



ΕΘΝΙΚΟ ΚΑΙ ΚΑΠΟΔΙΣΤΡΙΑΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

**ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ**

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

**Μελέτη της Απόδοσης των Αλγορίθμων Διαχείρισης Πόρων
σε δίκτυα 4^{ης} γενιάς σε περιβάλλον προσομοίωσης NS-3**

**Μαρινέλα – Ρόμπερτ – Μερτίρη
Joana – Hetem – Dulaj**

Επιβλέποντες: **Αθανασία Αλωνιστιώτη**, Επίκουρος Καθηγήτρια ΕΚΠΑ
Κωνσταντίνος Χατζηκοκολάκης, Υποψήφιος Διδάκτωρ ΕΚΠΑ
Γεώργιος Μπεϊνάς, Ερευνητής – Εξωτερικός Συνεργάτης

**ΑΘΗΝΑ
ΟΚΤΩΒΡΙΟΣ 2015**

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

Μελέτη της απόδοσης των αλγόριθμων διαχείρισης πόρων σε δίκτυα 4ης γενιάς σε περιβάλλον προσομοίωσης NS3

Μαρινέλα Ρ. Μερτίρη

A.M.: 1115201000131

Joana H. Dulaj

A.M.: 1115201000153

ΕΠΙΒΛΕΠΟΝΤΕΣ: **Αθανασία Αλωνιστιώτη**, Επίκουρος Καθηγήτρια ΕΚΠΑ
Κωνσταντίνος Χατζηκοκολάκης, Υποψήφιος Διδάκτωρ ΕΚΠΑ
Γεώργιος Μπεϊνάς, Ερευνητής-Εξωτερικός Συνεργάτης

ΠΕΡΙΛΗΨΗ

Στην παρούσα εργασία έγινε μελέτη των αλγορίθμων διαμοιρασμού πόρων, που έχουν υλοποιηθεί για LTE δίκτυα. Σκοπός είναι η σύγκριση των αλγορίθμων μέσα από ρεαλιστικά σενάρια προσομοίωσης. Για τα προγράμματα των προσομοιώσεων έχει χρησιμοποιηθεί ο δικτυακός προσομοιωτής του ns-3, ο οποίος παρέχει πληθώρα κλάσεων που μοντελοποιούν τα δομικά στοιχεία ενός LTE δικτύου.

Στα πλαίσια της παρούσας πτυχιακής εργασίας υλοποιήθηκαν δύο διαφορετικές τοπολογίες δικτύου, με διαφορετικά χαρακτηριστικά η καθεμία, όπως το πλήθος των χρηστών και οι διαθέσιμοι πόροι. Κάθε μεταβολή στην τοπολογία αποτελεί και ένα διαφορετικό σενάριο. Στη συνέχεια, για κάθε τοπολογία και για κάθε σενάριο, εφαρμόζονται τέσσερις αλγόριθμοι χρονοπρογραμματισμού και εξάγονται ορισμένα στατιστικά συμπεράσματα. Με αυτόν τον τρόπο γίνεται η σύγκριση των αλγορίθμων, ώστε να διαπιστωθεί ποιος είναι ο καταλληλότερος για κάθε περίπτωση. Οι αλγόριθμοι που μελετώνται πειραματικά είναι ο Round Robin, ο Proportional Fair και ο Token Bank Fair Queue. Ο τελευταίος υλοποιείται τόσο στο πεδίο του χρόνου όσο και στο πεδίο της συχνότητας.

Στην Ενότητα 2 γίνεται μελέτη των αλγορίθμων διαχείρισης πόρων και στην Ενότητα 3 παρουσιάζεται η επέκτασή τους στον προσομοιωτή ns-3. Τέλος, στην Ενότητα 4 αναλύονται τα σενάρια των πειραμάτων και τα αντίστοιχα αποτελέσματα, ενώ στην Ενότητα 5 συνοψίζονται τα συμπεράσματα που προκύπτουν από τα πειράματα.

ΘΕΜΑΤΙΚΗ ΠΕΡΙΟΧΗ: Διαμοιρασμός πόρων σε LTE δίκτυο

ΛΕΞΕΙΣ ΚΛΕΙΔΙΑ: LTE δίκτυο, eNB, Proportional Fair, Round Robin, Token Bank Fair Queue

ABSTRACT

In this project we study the resource allocation algorithms that have been developed for the LTE networks. Our purpose is to compare the algorithms, through realistic simulation scenarios. For the simulation programs, we have used the network simulation, ns-3, which provides various classes for modeling the basic elements of a LTE network.

For the purpose of this project, we created two different network topologies, with different parameters in each of them, such as the number of users and the available resources. Any modification in the topology is considered as a different scenario. Subsequently, we use four scheduling algorithms and we extract some statistical conclusions for each topology and each scenario in it. In this way, it is performed a comparison between the algorithms in order to be determined which is the most appropriate for each case. The algorithms that are studied experimentally are the Round Robin, the Proportional Fair and the Token Bank Fair Queue. We study the last one both in time and frequency domain.

In Section 2 we study the resource allocation algorithms and Section 3 presents their extension to ns-3 simulation. Finally, in Section 4 are discussed the scenarios for the experiments and the corresponding results, whereas Section 5 summarizes the conclusions of the experiments.

SUBJECT AREA: Resource allocation in LTE networks

KEYWORDS: LTE network, eNB, Proportional Fair, Round Robin, Token Bank Fair Queue

*Την πτυχιακή αυτή την αφιερώνουμε στις οικογένειές μας που μας στήριξαν με κάθε
τρόπο σε όλη την διάρκεια των σπουδών μας.*

ΕΥΧΑΡΙΣΤΙΕΣ

Για την εκπόνηση της παρούσας Πτυχιακής Εργασίας, θα θέλαμε να ευχαριστήσουμε τους επιβλέποντες, επικ. καθ. Αθανασία Αλωνιστιώτη, τον υποψήφιο διδάκτορα Κωνσταντίνο Χατζηκοκολάκη και τον ερευνητή Γεώργιο Μπεϊνά, για τον χρόνο τους, τη συνεργασία, την επιμονή και την πολύτιμη συμβολή τους στην ολοκλήρωση της.

ΠΕΡΙΕΧΟΜΕΝΑ

ΠΡΟΛΟΓΟΣ	13
1. ΕΙΣΑΓΩΓΗ.....	14
2. ΑΛΓΟΡΙΘΜΟΙ ΔΙΑΧΕΙΡΙΣΗΣ ΠΟΡΩΝ	15
2.1 Round Robin	20
2.2 Blind Equal Throughput.....	23
2.3 Token Bank Fair Queue	26
2.4 Proportional Fair	29
2.5 Priority Set Scheduler.....	32
2.6 Maximum Throughput	34
2.7 Throughput to Average	35
3. ΠΡΟΣΟΜΟΙΩΤΗΣ NS-3.....	36
3.1 Round Robin	37
3.2 Blind Equal Throughput.....	38
3.3 Token Bank Fair Queue	39
3.4 Proportional Fair	40
3.5 Priority Set Scheduler.....	40
3.6 Maximum Throughput	41
3.7 Throughput to Average	41
4. ΠΕΙΡΑΜΑΤΑ	42
4.1 1 ^ο Σενάριο Προσομοίωσης.....	42

4.1.1	Περιγραφή	42
4.1.1.1	Η τοπολογία της προσομοίωσης	42
4.1.1.2	Οι εφαρμογές των χρηστών	46
4.1.2	Παράμετροι Προσομοίωσης.....	47
4.1.3	Αποτελέσματα.....	48
4.1.3.1	Round Robin	48
4.1.3.2	Proportional Fair	52
4.1.3.3	Time Domain Token Bank Fair Queue	54
4.1.3.4	Frequency Domain Token Bank Fair Queue	56
4.2	2 ^ο Σενάριο Προσομοίωσης.....	59
4.2.1	Περιγραφή	59
4.2.1.1	Η τοπολογία της προσομοίωσης	59
4.2.1.2	Οι εφαρμογές των χρηστών	62
4.2.2	Παράμετροι προσομοίωσης.....	62
4.2.3	Αποτελέσματα.....	64
4.2.3.1	Round Robin	64
4.2.3.2	Proportional Fair	68
4.2.3.3	Time Domain Token Bank Fair Queue	70
4.2.3.4	Frequency Domain Token Bank Fair Queue	72
5.	ΣΥΜΠΕΡΑΣΜΑΤΑ	75
	ΠΙΝΑΚΑΣ ΟΡΟΛΟΓΙΑΣ	76
	ΣΥΝΤΜΗΣΕΙΣ – ΑΡΚΤΙΚΟΛΕΞΑ – ΑΚΡΩΝΥΜΙΑ	77
	ΑΝΑΦΟΡΕΣ	80

ΚΑΤΑΛΟΓΟΣ ΕΙΚΟΝΩΝ

Εικόνα 1: Αρχιτεκτονική LTE δικτύου [1].....	15
Εικόνα 2: Στοιβά πρωτοκόλλων επιπέδου χρήστη στο LTE [1]	16
Εικόνα 3: Πλέγμα πόρων για το downlink LTE [1]	17
Εικόνα 4: Διαμοιρασμός πόρων μεταξύ δύο χρηστών με τον TD RR αλγόριθμο [5].....	20
Εικόνα 5: Διαμοιρασμός πόρων μεταξύ δύο χρηστών με τον FD RR αλγόριθμο [5].....	21
Εικόνα 6: Χρονοπρογραμματισμός κατερχόμενης ζεύξης με τον TBFQ scheduler [2]...	27
Εικόνα 7: Σχηματική αναπαράσταση PF.....	32
Εικόνα 8: Δομικά στοιχεία που έχουν υλοποιηθεί στον ns-3 [18].....	36
Εικόνα 9: Διάγραμμα κλάσεων από τις οποίες κληρονομεί η <i>ns3::RrFfMacScheduler</i> [20]	37
Εικόνα 10: Συνάρτηση createBuilding()	43
Εικόνα 11: Τοπολογία 1 ^{ης} προσομοίωσης.....	44
Εικόνα 12: Συνολική διεκπεραιωτικότητα για κάθε σύνολο της κατερχόμενης ζεύξης για τον RR scheduler – Τοπολογία 1.....	49
Εικόνα 13: Συνολική διεκπεραιωτικότητα για κάθε σύνολο της ανερχόμενης ζεύξης για τον RR scheduler – Τοπολογία 1.....	49
Εικόνα 14: Μέση συνολική καθυστέρηση για κάθε σύνολο της κατερχόμενης ζεύξης για τον RR scheduler – Τοπολογία 1.....	50
Εικόνα 15: Μέση συνολική καθυστέρηση για κάθε σύνολο της ανερχόμενης ζεύξης για τον RR scheduler – Τοπολογία 1.....	50
Εικόνα 16: Συνολικός ρυθμός απωλειών δεδομένων για κάθε σύνολο της κατερχόμενης ζεύξης για τον RR scheduler – Τοπολογία 1.....	51
Εικόνα 17: Συνολικός ρυθμός απωλειών δεδομένων για κάθε σύνολο της ανερχόμενης ζεύξης για τον RR scheduler – Τοπολογία 1.....	51
Εικόνα 18: Συνολική διεκπεραιωτικότητα για κάθε σύνολο της κατερχόμενης ζεύξης για τον PF scheduler – Τοπολογία 1	53

Εικόνα 19: Μέση συνολική καθυστέρηση για κάθε σύνολο της κατερχόμενης ζεύξης για τον PF scheduler – Τοπολογία 1	53
Εικόνα 20: Συνολικός ρυθμός απωλειών δεδομένων για κάθε σύνολο της κατερχόμενης ζεύξης για τον PF scheduler – Τοπολογία 1	54
Εικόνα 21: Συνολική διεκπεραιωτικότητα για κάθε σύνολο της κατερχόμενης ζεύξης για τον Time Domain Token Bank Fair Queue scheduler – Τοπολογία 1.....	55
Εικόνα 22: Μέση συνολική καθυστέρηση για κάθε σύνολο της κατερχόμενης ζεύξης για τον Time Domain Token Bank Fair Queue scheduler – Τοπολογία.....	55
Εικόνα 23: Συνολικός ρυθμός απωλειών δεδομένων για κάθε σύνολο της κατερχόμενης ζεύξης για τον Time Domain Token Bank Fair Queue scheduler – Τοπολογία 1.....	56
Εικόνα 24: Συνολική διεκπεραιωτικότητα για κάθε σύνολο της κατερχόμενης ζεύξης για τον FD-TBFQ scheduler – Τοπολογία 1	57
Εικόνα 25: Μέση συνολική καθυστέρηση για κάθε σύνολο της κατερχόμενης ζεύξης για τον FD-TBFQ scheduler – Τοπολογία 1	57
Εικόνα 26: Συνολικός ρυθμός απωλειών δεδομένων για κάθε σύνολο της κατερχόμενης ζεύξης για τον FD-TBFQ scheduler – Τοπολογία 1	58
Εικόνα 27: Τοπολογία 2 ^{ης} προσομοίωσης.....	61
Εικόνα 28: Συνολική διεκπεραιωτικότητα για κάθε σύνολο της κατερχόμενης ζεύξης για τον RR scheduler – Τοπολογία 2.....	64
Εικόνα 29: Συνολική διεκπεραιωτικότητα για κάθε σύνολο της ανερχόμενης ζεύξης για τον RR scheduler – Τοπολογία 2.....	65
Εικόνα 30: Μέση συνολική καθυστέρηση για κάθε σύνολο της κατερχόμενης για τον RR scheduler – Τοπολογία 2.....	65
Εικόνα 31: Μέση συνολική καθυστέρηση για κάθε σύνολο της ανερχόμενης ζεύξης για τον RR scheduler – Τοπολογία 2.....	66
Εικόνα 32: Συνολικός ρυθμός απωλειών δεδομένων για κάθε σύνολο της κατερχόμενης ζεύξης για τον Round Robin scheduler – Τοπολογία 2.....	66
Εικόνα 33: Συνολικός ρυθμός απωλειών δεδομένων για κάθε σύνολο της ανερχόμενης ζεύξης για τον RR scheduler – Τοπολογία 2.....	67
Εικόνα 34: Συνολική διεκπεραιωτικότητα για κάθε σύνολο της κατερχόμενης ζεύξης για τον PF scheduler – Τοπολογία 2	68

Εικόνα 35: Μέση συνολική καθυστέρηση για κάθε σύνολο της κατερχόμενης ζεύξης για τον PF scheduler – Τοπολογία 2	69
Εικόνα 36: Συνολικός ρυθμός απωλειών δεδομένων για κάθε σύνολο της κατερχόμενης ζεύξης για τον PF scheduler – Τοπολογία 2	69
Εικόνα 37: Συνολική διεκπεραιωτικότητα για κάθε σύνολο της κατερχόμενης για τον TD-TBFQ scheduler – Τοπολογία 2	70
Εικόνα 38: Μέση συνολική καθυστέρηση για κάθε σύνολο της κατερχόμενης για τον TD-TBFQ scheduler – Τοπολογία 2	71
Εικόνα 39: Συνολικός ρυθμός απωλειών δεδομένων για κάθε σύνολο της κατερχόμενης ζεύξης για τον TD-TBFQ scheduler – Τοπολογία 2	71
Εικόνα 40: : Συνολική διεκπεραιωτικότητα για κάθε σύνολο της κατερχόμενης ζεύξης για τον FD-TBFQ scheduler – Τοπολογία 2	72
Εικόνα 41: Μέση συνολική καθυστέρηση για κάθε σύνολο της κατερχόμενης ζεύξης για τον FD-TBFQ scheduler – Τοπολογία 2	73
Εικόνα 42: Συνολικός ρυθμός απωλειών δεδομένων για κάθε σύνολο της κατερχόμενης ζεύξης για τον FD-TBFQ scheduler – Τοπολογία 2	73

ΚΑΤΑΛΟΓΟΣ ΠΙΝΑΚΩΝ

Πίνακας 1: Τιμές των παραμέτρων της 1 ^{ης} προσομοίωσης	47
Πίνακας 2: Τιμές των παραμέτρων της 2 ^{ης} προσομοίωσης	62

ΠΡΟΛΟΓΟΣ

Η παρούσα Πτυχιακή Εργασία εκπονήθηκε στα πλαίσια του Προπτυχιακού Προγράμματος Σπουδών του Τμήματος Πληροφορικής και Τηλεπικοινωνιών του Εθνικού και Καποδιστριακού Πανεπιστημίου Αθηνών το ακαδημαϊκό έτος 2014-2015.

Αντικείμενο της εργασίας είναι οι αλγόριθμοι διαμοιρασμού πόρων σε LTE δίκτυα και η σύγκριση των επιδόσεών τους μέσα από σενάρια προσομοίωσης. Για την μοντελοποίηση αυτών των σεναρίων χρησιμοποιήθηκε ο προσομοιωτής του ns-3.

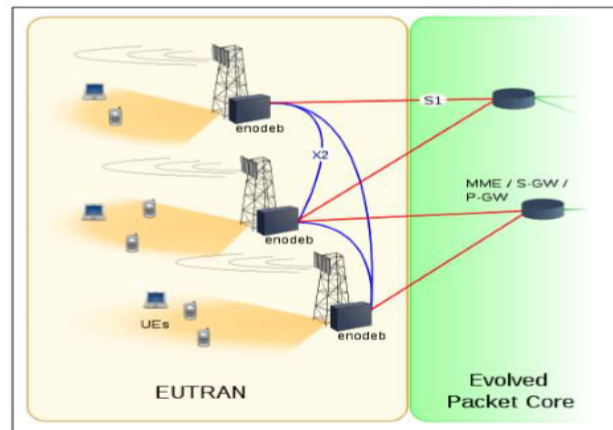
1. ΕΙΣΑΓΩΓΗ

Τα μελλοντικά ασύρματα και κινητά δίκτυα επικοινωνιών αναμένεται να προσφέρουν υψηλότερους ρυθμούς δεδομένων, να υποστηρίζουν ένα μεγάλο πλήθος συνδρομητών και να διασφαλίζουν την εκπλήρωση των απαιτήσεων για τις ποιότητες των υπηρεσιών (Quality of Service – QoS), δεδομένης της περιορισμένης διαθεσιμότητας του φάσματος συχνοτήτων και τις χρονικές μεταβολές των καναλιών. Το Long Term Evolution (LTE) δίκτυο έχει σχεδιαστεί από το 3GPP (3rd Generation Partnership Project) για να ανταπεξέλθει σε αυτές τις απαιτήσεις. Το έργο του LTE μπορεί να θεωρηθεί ως το ορόσημο που οδηγεί στην ανάπτυξη των δικτύων 4G (4th Generation – 4^{ης} Γενιάς). Το LTE χρησιμοποιεί διαφορετικές ζώνες συχνοτήτων ανάλογα με την περιοχή. Αυτές οι ζώνες είναι τα 900/1800/2600 MHz για την Ευρώπη, 1800/2600 MHz για την Ασία, 700 MHz για την Βόρεια Αμερική και 1800 MHz για την Αυστραλία. Κάποιες από τις απαιτήσεις που πρέπει να ικανοποιεί ένα LTE δίκτυο είναι ο υψηλός ρυθμός δεδομένων, ο μικρός χρόνος ανταπόκρισης (latency), η βελτιωμένη φασματική απόδοση, το κλιμακούμενο εύρος ζώνης (από 1,4 MHz μέχρι 20 MHz), η σχεδίαση ενός επίπεδου δικτύου που να βασίζεται εξ ολοκλήρου στην αρχιτεκτονική IP (Internet Protocol) και η βελτιστοποιημένη απόδοση για κινητές συσκευές. Έχουν αναπτυχθεί αρκετές τεχνολογίες, που έχουν ως αντικείμενο την κάλυψη αυτών των απαιτήσεων. Οι πιο σημαντικές από αυτές είναι η πολλαπλή πρόσβαση με ορθογώνια διαίρεση συχνότητας (Orthogonal Frequency-Division Multiple Access – OFDMA) για την κατερχόμενη ζεύξη (downlink) και η πολλαπλή πρόσβαση με διαίρεση συχνότητας σε ένα φέρον (Single-Carrier Frequency Division Multiple Access – SC-FDMA) για την ανερχόμενη (uplink), η χρήση τεχνολογιών πολλαπλών κεραιών και η τεχνική χωρικής πολυπλεξίας πολλαπλών εισόδων – πολλαπλών εξόδων (MIMO – Multiple-Input and Multiple-Output). Επίσης, σχετικά με τα σχήματα διαμόρφωσης, στο downlink χρησιμοποιείται το Quadrature Phase Shift Keying (QPSK), το Quadrature Amplitude Modulation με 16 σύμβολα (16 QAM) και το 64 QAM, ενώ στο uplink χρησιμοποιείται το QPSK και το 64 QAM.

Στην επόμενη ενότητα παρουσιάζονται οι έννοιες και τα δομικά στοιχεία του δικτύου, που χρειάζεται να γνωρίζουμε προκειμένου να καταλάβουμε τους αλγόριθμους διαμοιρασμού πόρων, και, στην συνέχεια, αναλύονται οι ίδιοι οι αλγόριθμοι. Έπειτα, στην Ενότητα 3, περιγράφεται η βιβλιοθήκη του ns-3 καθώς και ο τρόπος που έχουν υλοποιηθεί οι αλγόριθμοι χρονοπρογραμματισμού σε αυτήν. Στην Ενότητα 4 περιγράφονται αναλυτικά τα πειράματα που εκτελέσαμε, μαζί με κάποιες κλάσεις του ns-3 που χρησιμοποιήθηκαν, και παρουσιάζονται τα αποτελέσματά τους. Τέλος, στην Ενότητα 5 αναλύονται τα αποτελέσματα των πειραμάτων και προτείνονται θέματα για μελλοντική διερεύνηση.

2. ΑΛΓΟΡΙΘΜΟΙ ΔΙΑΧΕΙΡΙΣΗΣ ΠΟΡΩΝ

Το LTE είναι ένα εξελιγμένο σύστημα πακέτου (Evolved Packet System – EPS), που περιλαμβάνει το δίκτυο ραδιοφάσματος – Evolved UMTS (Universal Mobile Telecommunications System) Terrestrial Radio Access Network (EUTRAN) – και το δίκτυο κορμού – Evolved Packet Core (EPC). Στην Εικόνα 1 παρουσιάζεται η αρχιτεκτονική του δικτύου. [1]

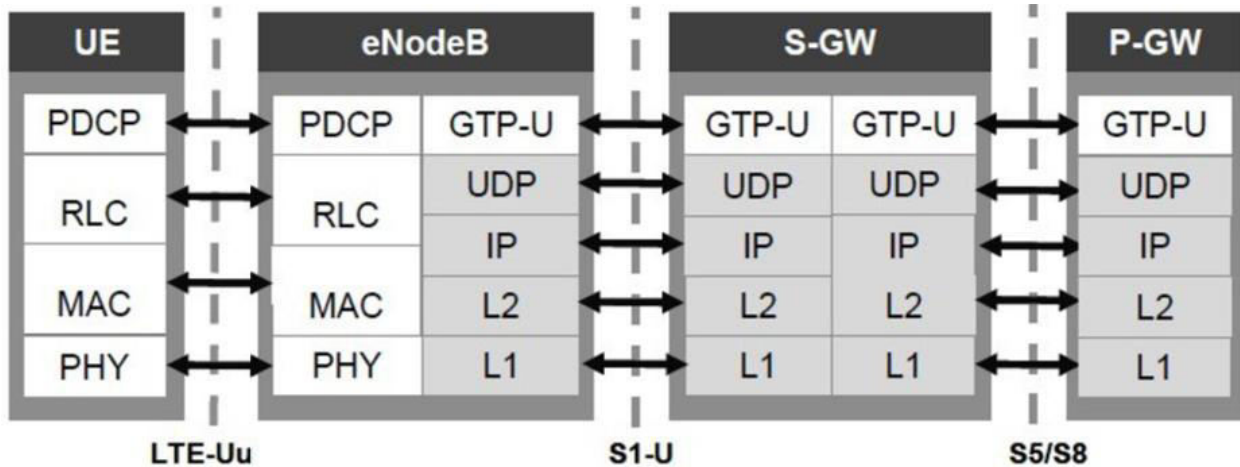


Εικόνα 1: Αρχιτεκτονική LTE δικτύου [1]

Τα επιμέρους τμήματα του δικτύου είναι τα:

- **eNodeB (evolved Node B – eNB):** Διαχειρίζεται όλες τις διεργασίες που σχετίζονται με την διεπαφή του ραδιοφάσματος και εκτελεί εργασίες όπως αυτές που εκτελούνται από τους NBs (Node B) και Radio Network Controller (RNC) μαζί στο UTRAN του UMTS.
- **MME (Mobility Management Entity):** Διαχειρίζεται την κινητικότητα, την ταυτότητα του UE και τις παραμέτρους ασφάλειας.
- **S-GW (Serving Getaway):** Είναι ο κόμβος που τερματίζει την διεπαφή στην πλευρά του EUTRAN.
- **P-GW (Packet data network Getaway):** Είναι ο κόμβος που τερματίζει την διεπαφή στην πλευρά του EPC.
- **Διεπαφές (Interfaces):** Αυτές είναι η S1, η οποία συνδέει τον eNB με το EPC, και η X2, η οποία συνδέει δύο eNBs και χρησιμοποιείται κυρίως για την λειτουργία της μεταβίβασης (handover).

Στην Εικόνα 2 παρουσιάζεται η στοίβα πρωτοκόλλων επιπέδου χρήστη στο LTE, μεταξύ των διεπαφών Uu (μεταξύ UE και eNB), S1 (μεταξύ eNB και S-GW) και S5/S8 (μεταξύ S-GW και P-GW).



Εικόνα 2: Στοιβά πρωτοκόλλων επιπέδου χρήστη στο LTE [1]

Τα επίπεδα πρωτοκόλλων, τα οποία βρίσκονται κάτω από το επίπεδο IP στο τερματικό του χρήστη, είναι τα ακόλουθα:

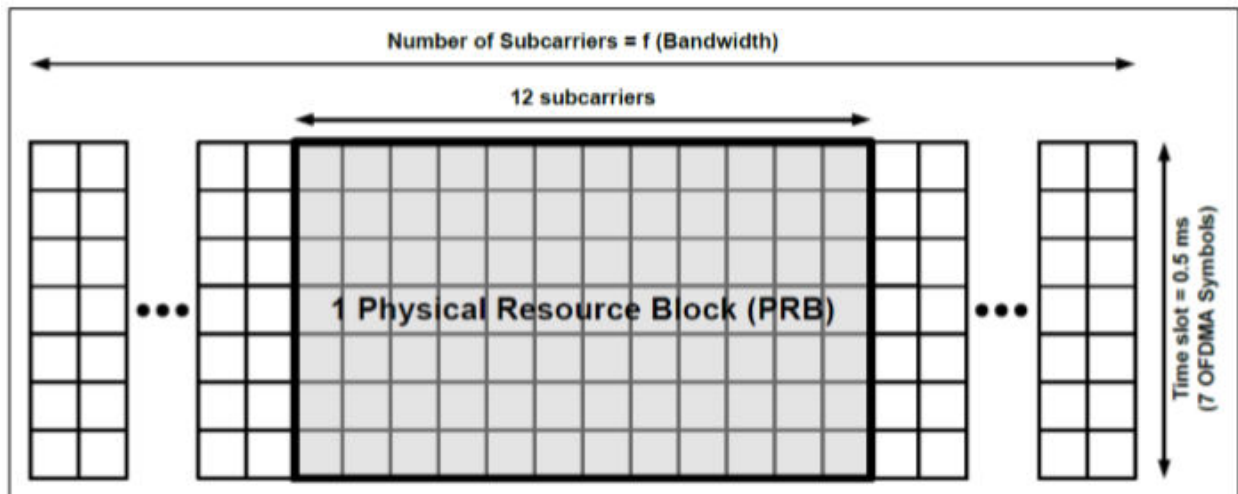
- **PDCP (Packet Data Convergence Protocol):** Η κύρια λειτουργία αυτού του πρωτοκόλλου είναι η συμπίεση της IP επικεφαλίδας.
- **RLC (Radio Link Control):** Το πρωτόκολλο αυτό είναι υπεύθυνο για την τμηματοποίηση και την συνένωση των PDCP-PDUs (Protocol Data Units) για ασύρματη μετάδοση. Επίσης, υλοποιεί διόρθωση σφαλμάτων με την μέθοδο της αυτόματης αίτησης για επανάληψη (Automatic Repeat reQuest – ARQ).
- **MAC (Medium Access Control):** Είναι υπεύθυνο για τον χρονοπρογραμματισμό των δεδομένων, σύμφωνα με τις προτεραιότητες, και την πολυπλεξία δεδομένων στο 1^ο επίπεδο μεταφοράς (Layer 1). Το επίπεδο MAC παρέχει επίσης διόρθωση σφαλμάτων με την μέθοδο του υβριδικού – ARQ (Hybrid ARQ).
- **PHY (Physical Layer):** Είναι το Layer 1 της διεπαφής του LTE-Uu, το οποίο ασχολείται με τις λειτουργίες του Direct – Sequence Code Division Multiple Access (DS-CDMA).
- **GTP-U (GPRS (General Packet Radio Service) Tunneling Protocol, User Plane):** Το πρωτόκολλο GTP-U βρίσκεται στους eNB, P-GW και S-GW και υλοποιεί την σήραγγα που χρησιμοποιείται για να στέλνει τα IP πακέτα του τερματικού χρήστη, που ανήκουν σε έναν φορέα (bearer) EPS, στο επίπεδο UDP/IP.
- **UDP (Unit Data Protocol) και IP (Internet Protocol):** Χρησιμοποιούνται ως τα καθιερωμένα και βασικά πρωτόκολλα μεταφοράς. Το UDP χρησιμοποιείται αντί του TCP (Transmission Control Protocol), επειδή τα υψηλότερα επίπεδα παρέχουν ήδη αξιόπιστη μεταφορά δεδομένων με την χρήση τεχνικών ανάκαμψης λόγω σφαλμάτων και αναμεταδόσεων. Τα IP πακέτα του EPC μπορούν να μεταδοθούν με διάφορες τεχνολογίες επιπέδου L2 και L1, όπως το Ethernet και το ATM (Asynchronous Transfer Mode).

Το OFDMA, που χρησιμοποιεί το LTE για μεταδόσεις στην κατερχόμενη ζεύξη, είναι η πιο κατάλληλη μέθοδος πολλαπλής πρόσβασης, γιατί δεν επηρεάζεται από την δια-συμβολική παρεμβολή (inter-symbol interference) και την επιλεκτική εξασθένιση συχνότητας. Στην μέθοδο αυτή, θεωρούμε ένα πλέγμα από πόρους στο πεδίο του χρόνου και της συχνότητας, το οποίο αναπαριστά τις συχνότητες των υπό-φερόντων (subcarriers) στον άξονα της συχνότητας και τα σύμβολα στον άξονα του χρόνου. Η βασική μονάδα πόρου του δικτύου αναπαρίσταται από ένα sub-carrier και ένα σύμβολο

του πλέγματος αυτού. Με αυτόν τον τρόπο, καθώς οι πόροι ανατίθενται τόσο στο πεδίο του χρόνου όσο και της συχνότητας, υπάρχει η δυνατότητα να εξυπηρετούνται ταυτόχρονα πολλοί χρήστες. Το OFDMA διαιρεί την ροή δεδομένων σε μικρότερες ροές, οι οποίες στέλνονται παράλληλα πάνω σε πολλά carriers. Αυτό έχει σαν αποτέλεσμα τα βήματα μετάδοσης να είναι αρκετά μεγάλα ώστε να αποφεύγονται τα προβλήματα της πολυδιόδευσης. Συνεπώς, λόγω της ορθογωνιότητας των πόρων που διατίθενται, δεν προκύπτουν παρεμβολές για τους χρήστες που βρίσκονται εντός της ίδιας κυψέλης. Ωστόσο, εξαιτίας της επαναχρησιμοποίησης του φάσματος σε γειτονικές κυψέλες, υπάρχει η πιθανότητα παρεμβολών ανάμεσα σε χρήστες που βρίσκονται στα όρια γειτονικών κυψελών και χρησιμοποιούν τους ίδιους ραδιοπόρους.

Το SC-FDMA περιέχει κάποια επιπλέον βήματα, σε σχέση με το OFDM, ενώ παράλληλα, τόσο για την αποστολή όσο και για την λήψη δεδομένων, χρησιμοποιούνται οι ίδιες φυσικές παράμετροι που χρησιμοποιούνται και στο downlink. Ο λόγος που στο uplink προτιμάται ο SC-FDMA από τον OFDMA είναι γιατί το κλάσμα της υψηλότερης τιμής ισχύος ως προς τη μέση τιμή ισχύος (Peak to Average Power Ratio – PAPR) είναι αρκετά υψηλό στο OFDMA. Αυτό σημαίνει ότι η συσκευή πρέπει να καταναλώνει μεγάλη ισχύ, για να μπορεί να υποστηρίξει την ισχύ κορυφής, ακόμα κι αν αυτή δεν εμφανίζεται συχνά στο σύστημα. Καθώς λοιπόν, το τερματικό του χρήστη, το οποίο πρέπει να λειτουργεί με όσο το δυνατόν μικρότερη ισχύ διατηρώντας μια καλή απόδοση, δεν μπορεί να υποστηρίξει συστήματα που επιβάλλουν μεγάλο PAPR, η χρήση της μεθόδου OFDMA είναι ακατάλληλη για το uplink. [2]

Η μικρότερη μονάδα πόρων που μπορεί να ανατεθεί στους χρήστες αποτελείται από δύο ομάδες πόρων φυσικού επιπέδου (Physical Resource Blocks – PRBs) και αναφέρεται ως τμήμα (chunk) ή ομάδα πόρων (Resource Block – RB). Στο downlink LTE ένα PRB αντιστοιχεί σε 12 subcarriers (συνολικά 180 kHz) και 7 OFDM σύμβολα (συνολικά 0,5 ms). Αυτό ισχύει για LTE συστήματα που δεν αποτελούνται από MBSFN (Multimedia Broadcast multicast Service Single Frequency Network) δίκτυα και στα οποία εφαρμόζεται η τεχνική του κλασικού κυκλικού προθέματος (Cyclic Prefix – CP). Το πλέγμα των πόρων του LTE (για το downlink) παρουσιάζεται στην Εικόνα 3. Όλα τα subcarriers είναι ορθογώνια μεταξύ τους και το κάθε ένα έχει εύρος ζώνης 15 kHz. Ωστόσο, το πλήθος των PRBs του συνολικού εύρους μετάδοσης στο LTE δεν είναι ακριβώς ίσο με την διαίρεση του συνολικού bandwidth με τα 180 kHz, γιατί μερικά από τα subcarriers κρατούνται για σηματοδότηση. Οι αποφάσεις χρονοπρογραμματισμού γίνονται σε κάθε TTI (Transmission Time Interval), το οποίο στο LTE ισούται με 1 ms. Τα δεδομένα μεταφέρονται στα UEs μέσα στα PRBs. Το eNB αναθέτει σε ένα συγκεκριμένο UE διαφορετικά RBs.



Εικόνα 3: Πλέγμα πόρων για το downlink LTE [1]

Καθώς κάποιος χρήστης μπορεί να καταναλώνει παραπάνω από μια υπηρεσίες, είναι χρήσιμο να περιγράψουμε την έννοια του LTE bearer. Το LTE bearer είναι ένα λογικό κανάλι, που υποστηρίζει τις απαιτήσεις των QoS του LTE. Τα bearers διαχωρίζονται σε κλάσεις και το κάθε ένα παίρνει από το δίκτυο μόνο έναν προσδιοριστή (identifier) QoS. Το bearer προσδιορίζει με μοναδικό τρόπο μια ροή πακέτων μεταξύ του UE και της πύλης εξόδου (gateway). Έτσι, οι διαφορετικές ροές κίνησης αντιστοιχούν σε διαφορετικά bearers, καθώς έχουν διαφορετικές απαιτήσεις. Τα EPS bearers μπορούν να διαιρεθούν σε δύο κατηγορίες, αυτά που εξασφαλίζουν σταθερό ρυθμό δεδομένων (Guaranteed Bit Rate – GBR) και αυτά που δεν τον εξασφαλίζουν (non-Guaranteed Bit Rate – non-GBR). Στο GBR το δίκτυο εξασφαλίζει το QoS αναθέτοντας ένα συγκεκριμένο πλήθος πόρων στο UE κατά την εγκαθίδρυση της σύνδεσης. Για παράδειγμα, η VoIP (Voice over IP) κίνηση μπορεί να ανατεθεί σε ένα GBR. Στα non-GBR bearers δεν ανατίθενται πάντα το ίδιο πλήθος από πόρους και δεν εξασφαλίζεται κάποιος συγκεκριμένος ρυθμός δεδομένων. Η κίνηση σε αυτά τα bearers προέρχεται κυρίως από υπηρεσίες βέλτιστης προσπάθειας.

Επειδή οι πόροι του ραδιοφάσματος είναι πεπερασμένοι, πρέπει να βρεθούν τρόποι ώστε να χρησιμοποιούνται όσο πιο αποδοτικά γίνεται, ώστε να είναι σε θέση το δίκτυο να ανταπεξέλθει στις ολοένα αυξανόμενες ανάγκες των χρηστών. Έτσι, ο χρονοπρογραμματιστής (scheduler) του MAC επιπέδου του eNodeB προσπαθεί να κάνει τον κατάλληλο καταμερισμό των πόρων μεταξύ των ενεργών χρηστών, λαμβάνοντας υπόψη συγκεκριμένες παραμέτρους όπως η απαιτούμενη ποιότητα υπηρεσίας (QoS), η καλύτερη δυνατή φασματική απόδοση, που να εξασφαλίζει την υψηλότερη απόδοση της κυψέλης κάτω από δεδομένες συνθήκες καναλιού, ο δίκαιος διαμοιρασμός των πόρων ανάμεσα στους χρήστες και τις εφαρμογές, οι καταστάσεις των καταχωρητών (buffers), ο περιορισμός των επιπτώσεων της παρεμβολής μέσω ειδικού χειρισμού των χρηστών που βρίσκονται στα όρια της κυψέλης και η εξισορρόπηση του φόρτου κίνησης μεταξύ των κυψελών. Όπως και σε παλιότερες ασύρματες τεχνολογίες, ο αλγόριθμος χρονοπρογραμματισμού που χρησιμοποιείται σε ένα δίκτυο δεν καθορίζεται από κάποιο πρότυπο, αλλά επαφίεται στο eNB η απόφαση για το ποιος αλγόριθμος θα χρησιμοποιηθεί.

Ο scheduler αλληλεπιδρά απ' ευθείας με την μονάδα Hybrid Automatic Repeat Request (HARQ), η οποία διαχειρίζεται τις αναμεταδόσεις, καθώς ένα καινούριο bearer μπορεί σε ένα TTI (Transmission Time Interval) είτε να κάνει μια νέα μετάδοση είτε να αναμεταδώσει τα πακέτα που είναι σε αναμονή. Ο χρονοπρογραμματισμός πακέτων μπορεί να γίνει τόσο στο πεδίο του χρόνου (Time Domain – TD) όσο και στο πεδίο της συχνότητας (Frequency Domain). Για τον χρονοπρογραμματισμό στο πεδίο της συχνότητας χρησιμοποιούνται πολλαπλά φέροντα (multi-carriers), έτσι ώστε ο scheduler να μπορεί να εξυπηρετεί χρήστες με υψηλή ποιότητα καναλιού και να αποφεύγει χρήστες που αντιμετωπίζουν μεγάλες εξασθενίσεις στα λαμβανόμενα σήματα. Αυτό είναι χρήσιμο για την μείωση των διακυμάνσεων του λόγου σήματος προς θόρυβο και παρεμβολή (Signal to Interference plus Noise Ratio – SINR) και την αύξηση της χωρητικότητας του συστήματος. Για την επίτευξη υψηλού κέρδους με έναν FD scheduler είναι σημαντικό το εύρος ζώνης (bandwidth) του καναλιού να είναι μικρότερο από το bandwidth του συστήματος, πράγμα το οποίο ισχύει σχεδόν για όλα τα ασύρματα συστήματα. Ο TD scheduler μπορεί να αυξήσει την διαφορικότητα με πολλαπλούς χρήστες (multiuser diversity), το οποίο εξαρτάται από την εξασθένηση συχνότητας και την ταχύτητα του χρήστη. Αν υπάρχουν μεγάλες εξασθενίσεις, ο TD scheduler μπορεί να επιλέξει την ακριβή στιγμή του χρονοπρογραμματισμού ώστε να αποφύγει αυτές τις εξασθενίσεις. Με αυτόν τον τρόπο, ο scheduler προσπαθεί να εξασφαλίσει το QoS της κίνησης.

Ο χρονοπρογραμματισμός που εξαρτάται από το κανάλι (Channel Dependent Scheduling – CDS) μπορεί, επίσης, να υλοποιηθεί τόσο στο πεδίο του χρόνου όσο και στο πεδίο της συχνότητας. Σε αυτήν την περίπτωση χρονοπρογραμματισμού, ο scheduler προσαρμόζεται στις διακυμάνσεις του καναλιού και επιτυγχάνεται, έτσι, η προσαρμογή επιπέδου ζεύξης. Ένας χρήστης με καλύτερη ποιότητα καναλιού θα πάρει περισσότερους πόρους, καθώς μπορεί να χρησιμοποιήσει πιο αποδοτικά αυτούς τους πόρους και να συμβάλει έτσι στην αύξηση της απόδοσης της κυψέλης.

Το Channel Dependent Scheduling απαιτεί να διατίθενται στο eNB πληθώρα πληροφοριών για τις συνθήκες καναλιού της ανερχόμενης και της κατερχόμενης ζεύξης. Για να υλοποιηθεί το CDS στο πεδίο της συχνότητας, οι πληροφορίες για την κατάσταση του καναλιού πρέπει να εστιάζουν στην συχνότητα. Για παράδειγμα, για τον καταμερισμό των πόρων της κατερχόμενης ζεύξης, ο scheduler λαμβάνει υπόψη του τις πληροφορίες που προκύπτουν από τι αναφορές του Channel Quality Indicator (CQI) που προέρχονται από τα User Equipments (UEs), για να εκτιμήσει την ποιότητα του καναλιού, καθώς και τις περιγραφές των QoS που χαρακτηρίζουν τους διάφορους EPS bearers των UEs. Το eNB μπορεί να ρυθμίσει αν η αναφορά του CQI θα αντιστοιχεί σε όλο το εύρος ζώνης του downlink ή σε ένα τμήμα αυτού, το οποίο ονομάζεται υπο-ζώνη (sub-band). Η αναφορά των CQI για διαφορετικά sub-bands απαιτεί περισσότερους πόρους για το uplink. Το eNB μπορεί να ρυθμίσει την αναφορά του CQI στους ακόλουθους τρόπους:

- Αναφορά ευρείας ζώνης (wideband): Το CQI που αναφέρεται αντιστοιχεί σε όλο το εύρος ζώνης της κατερχόμενης ζεύξης.
- Αναφορά συγκεκριμένης υπο-ζώνης, καθορισμένης από το eNB.
- Αναφορά υπο-ζώνης που έχει επιλεγεί από το UE.

Ο χρονοπρογραμματισμός που εξαρτάται από το κανάλι οδηγεί σε υψηλότερη απόδοση κυψέλης, ενώ παράλληλα πρέπει να καταναίμει τους πόρους με δίκαιο τρόπο ανάμεσα στους χρήστες. Υπάρχει, δηλαδή, μια σύγκρουση ανάμεσα στην δικαιοσύνη και στην απόδοση της κυψέλης (βλ. [3], [4]). Όπως θα δούμε υπάρχουν αρκετοί τρόποι χρονοπρογραμματισμού που προσπαθούν να ικανοποιήσουν και τις δύο συνθήκες.

Όταν ο χρονοπρογραμματισμός πραγματοποιείται ταυτόχρονα στο πεδίο του χρόνου και της συχνότητας, ανεξάρτητα από το αν λαμβάνει υπόψη του τις συνθήκες καναλιού, το πλήθος των εξυπηρετούμενων χρηστών σε κάθε TTI περιορίζεται από το πλήθος των καναλιών PDCCHs (Physical Downlink Control Channels) που μπορούν να διαμορφωθούν. Αυτό εξαρτάται, μεταξύ άλλων, από το bandwidth του συστήματος και το πλήθος των συμβόλων που χρησιμοποιούνται. Τα PDCCHs έχουν σχεδιαστεί για να παρέχουν πληροφορίες χρονοπρογραμματισμού τόσο για την ανερχόμενη όσο για την κατερχόμενη ζεύξη και, καθώς σε κάθε χρήστη ανατίθεται ένα PDCCH για ανάθεση πόρων, το μέγιστο πλήθος των εξυπηρετούμενων χρηστών σε κάθε TTI για την μία ζεύξη είναι το μισό του πλήθους των διαθέσιμων PDCCHs καναλιών.

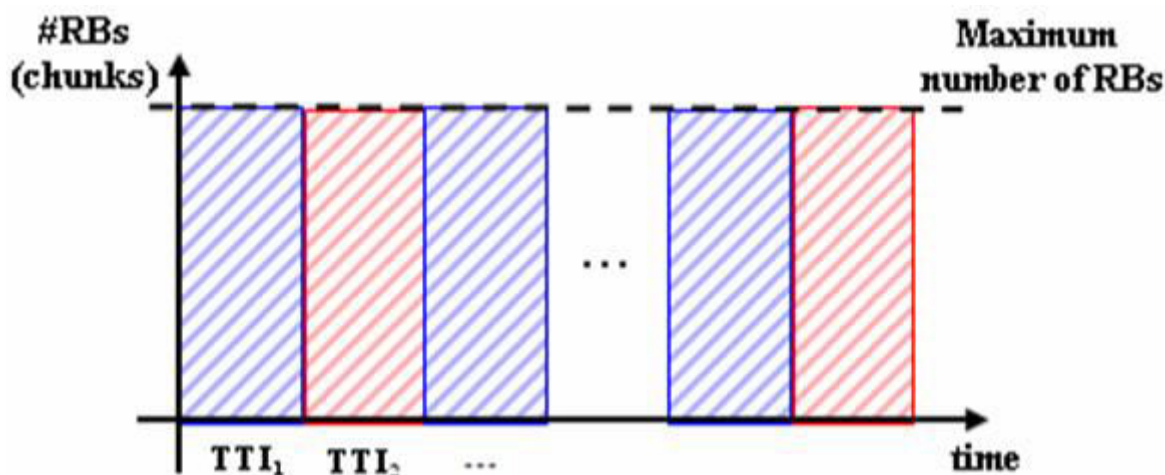
Η απόδοση ενός UE μπορεί να διαφοροποιείται σημαντικά ανάλογα με τον αλγόριθμο χρονοπρογραμματισμού που χρησιμοποιείται. Στην συνέχεια του κεφαλαίου παρουσιάζονται οι πιο γνωστοί αλγόριθμοι χρονοπρογραμματισμού, οι οποίοι είναι οι:

1. Round Robin (RR),
2. Blind Equal Throughput (BET),
3. Token Bank Fair Queue (TBFQ),
4. Proportional Fair (PF),
5. Priority Set (PSS),
6. Maximum Throughput (MT) και
7. Throughput To Average (TTA).

2.1 Round Robin

Ο αλγόριθμος αυτός επιτρέπει σε όλους τους χρήστες να χρησιμοποιούν ο ένας μετά τον άλλον τους πόρους του δικτύου. Δηλαδή, αν οι πόροι που μοιράζονται μεταξύ των χρηστών είναι χρονοθυρίδες, τότε μοιράζει ίσο πλήθος χρονοθυρίδων σε όλους τους χρήστες κυκλικά, χωρίς να δώσει σημασία στις προτεραιότητες των εφαρμογών ή στην κατάσταση του καναλιού. Έτσι προτιμάται όταν ο σταθμός βάσης δεν γνωρίζει ούτε για την κατάσταση του καναλιού, ούτε για τις ποιότητες υπηρεσίας των εφαρμογών.

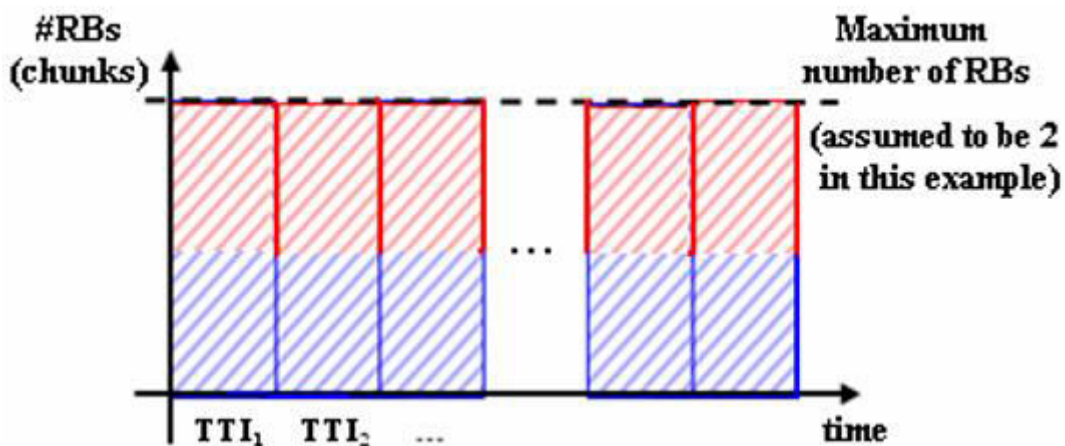
Υπάρχουν δύο εναλλακτικές μέθοδοι υλοποίησης του αλγορίθμου, μία για το πεδίο του χρόνου (Time Domain Round Robin – TD RR) και μία για το πεδίο της συχνότητας (Frequency Domain Round Robin – FD RR) (βλ. [5]). Η βασική διαφορά τους έγκειται στον τρόπο που μοιράζονται οι πόροι σε κάθε TTI. Στον TD RR, σε κάθε TTI ανατίθεται όλο το φάσμα συχνοτήτων του καναλιού στον χρήστη που έχει σειρά. Στο επόμενο TTI, οι πόροι αυτοί ανακτούνται από τον eNodeB και ανατίθενται στον επόμενο χρήστη. Δηλαδή, ο χρόνος που κάθε χρήστης κρατά απασχολημένους τους πόρους είναι ίσος με ένα TTI. Οι χρήστες τοποθετούνται σε μια ουρά αναμονής ώστε να εξυπηρετούνται όλοι με κυκλική σειρά. Έτσι, κάθε χρήστης που εξυπηρετείται σε ένα TTI, τοποθετείται κατόπιν στο τέλος αυτής της ουράς. Η Εικόνα 4 αναπαριστά την λειτουργία του αλγορίθμου TD RR για την περίπτωση που έχουμε δύο χρήστες. Τα χρώματα στο σχήμα αντιστοιχούν στους χρήστες. Όπως φαίνεται και από την εικόνα, στην περίπτωση των δύο χρηστών, ο κάθε ένας χρησιμοποιεί όλα τα τμήματα των πόρων (resource blocks), σε κάθε δύο TTI, ενώ ο συνολικός χρόνος δέσμευσης αυτών των πόρων είναι ο μισός του ολικού χρόνου όπου μένουν ανοιχτές οι συνδέσεις και διαβιβάζονται δεδομένα. Έτσι, ο κάθε ένας λαμβάνει το 50% των συνολικών πόρων, λαμβάνοντας υπόψη τόσο τους πόρους στο πεδίο συχνότητας όσο και τους πόρους στο πεδίο του χρόνου.



Εικόνα 4: Διαμοιρασμός πόρων μεταξύ δύο χρηστών με τον TD RR αλγόριθμο [5]

Στον FD RR οι πόροι σε ένα TTI μοιράζονται σε πολλούς χρήστες, οι οποίοι εξυπηρετούνται με κυκλική σειρά. Ο αλγόριθμος λειτουργεί ως εξής: Αρχικά, επιλέγονται κάποιοι χρήστες, έστω N , για τους οποίους θα γίνει διαμοιρασμός πόρων στο επόμενο TTI. Έπειτα, ο αλγόριθμος ελέγχει αν οι πόροι για τις PDCCH μεταδόσεις είναι αρκετοί για όλους τους επιλεγμένους χρήστες. Τέλος, αναθέτει τμήματα πόρων (resource blocks) σε αυτούς τους χρήστες. Όπως είναι φυσικό, κάθε χρήστης παίρνει διαφορετικά resource blocks και η ανάθεση των πόρων γίνεται σύμφωνα με την λογική του Round

Robin. Οι χρήστες οι οποίοι δεν επιλέχθηκαν για ανάθεση πόρων σε κάποιο TTI, θα επιλεγούν στα επόμενα TTIs. Ένας τρόπος υλοποίησης του αλγορίθμου αυτού είναι να ανατίθεται από ένα RB σε όλους τους επιλεγμένους χρήστες, προτού ξεκινήσει η κυκλική ανάθεση πόρων μέσα στο TTI. Αυτή η τεχνική αποδίδει όταν το πλήθος των χρηστών που περιμένουν να πάρουν πόρους είναι μικρότερο από το πλήθος των καναλιών PDCCHs για κάθε TTI, γιατί μόνο έτσι θα μπορεί ο καθένας να αξιοποιήσει το RB που παίρνει. Ωστόσο, όταν οι χρήστες είναι περισσότεροι από τα κανάλια, η παραπάνω τεχνική ανάθεσης ενός RB σε κάθε χρήστη δεν είναι αποδοτική. Αυτό συμβαίνει γιατί ανατίθενται πόροι σε χρήστες που δεν μπορούν να τους χρησιμοποιήσουν, αφού τα κανάλια PDCCHs δεν επαρκούν για όλους, πράγμα που έχει σαν αποτέλεσμα την σπατάλη πόρων. Στην Εικόνα 5 φαίνεται η λειτουργία του αλγορίθμου FD RR για δύο χρήστες και δύο resource blocks. Όπως και στην αντίστοιχη περίπτωση του TD RR, σε κάθε χρήστη ανατίθεται το 50% των συνολικών πόρων. Από την σύγκριση που γίνεται στο [5] μεταξύ TD RR και FD RR, εξάγεται το συμπέρασμα ότι η απόδοση του FD RR είναι μικρότερη εξαιτίας των περιορισμών που επιβάλλει το PDCCH. Ένας τρόπος αντιμετώπισης αυτού του προβλήματος είναι η ανάθεση περισσότερων RBs σε κάθε χρήστη, κατά την αρχική φάση, ώστε να αξιοποιηθεί όλο το εύρος ζώνης μετάδοσης. Έτσι η απόδοση του FD RR θα είναι σχεδόν ίδια με αυτήν του TD RR, ενώ παράλληλα ο FD RR θα είναι πιο κατάλληλος για υπηρεσίες όπου έχουμε μεταδόσεις μικρών πακέτων και απαιτήσεις για χαμηλή καθυστέρηση.



Εικόνα 5: Διαμοιρασμός πόρων μεταξύ δύο χρηστών με τον FD RR αλγόριθμο [5]

Στο [5] υλοποιείται μια προσομοίωση, προκειμένου να μελετηθεί η απόδοση του αλγορίθμου RR. Για την προσομοίωση θεωρείται ότι υπάρχει μία κυψέλη με ένα eNodeB, που χρησιμοποιεί συχνότητα φέροντος γύρω στα 2,6 GHz FDD (Frequency Division Duplex) και εύρος ζώνης 20 MHz. Επίσης, η διαμόρφωση της κεραίας είναι τύπου SISO (Single Input Single Output). Για λόγους απλοποίησης, έχει γίνει η υπόθεση ότι η ισχύς της κατερχόμενης ζεύξης είναι ίδια για όλα τα resource blocks και ότι όλα τα πακέτα λαμβάνονται σωστά. Επιπλέον, για όλη την διάρκεια της προσομοίωσης, οι χρήστες είναι στατικοί και έχουν το ίδιο SNR για όλα τα RBs που τους ανατίθενται. Στην προσομοίωση χρησιμοποιούνται διάφορες τιμές για SNR, οι οποίες προκύπτουν από τις απώλειες και την εξασθένηση των σημάτων και διαφοροποιούνται ανάλογα με την εφαρμοζόμενη μέθοδο διαμόρφωσης και κωδικοποίησης. Τέλος, υπάρχουν δύο κατηγορίες χρηστών. Στην μία δημιουργείται μια ροή υπηρεσίας τύπου CBR (Constant Bit Rate) με αναμενόμενη απόδοση γύρω στα 2 Mbps. Αυτό σημαίνει ότι αν η απόδοση είναι χαμηλότερη από την αναμενόμενη τιμή της, τότε οι χρήστες δεν μπορούν να εξυπηρετηθούν. Στην δεύτερη κατηγορία

δημιουργούνται υπηρεσίες βέλτιστης προσπάθειας, τύπου VBR (Variable Bit Rate), και η ελάχιστη αποδεκτή απόδοση είναι πάλι στα 2 Mbps. Ωστόσο, στην δεύτερη περίπτωση η απόδοση μπορεί να ξεπεράσει τα 2 Mbps, ενώ στην πρώτη θα είναι πάντα κοντά στην αναμενόμενη τιμή της.

Από τα πειράματα της παραπάνω προσομοίωσης παρατηρείται ότι καθώς αυξάνεται το SNR, αυξάνεται και η απόδοση της κυψέλης. Ωστόσο, στην περίπτωση του FD RR για την πρώτη κατηγορία χρηστών η καμπύλη της απόδοσης της κυψέλης συναρτήσει του SNR φτάνει κάποια στιγμή στον κόρο της. Αυτό σημαίνει ότι, μετά την τιμή κόρου του SNR, όσο κι αν αυξάνεται ο λόγος του σήματος προς θόρυβο, η συνολική απόδοση της κυψέλης μένει σχεδόν σταθερή. Οι χρήστες που τρέχουν υπηρεσίες CBR αντιμετωπίζουν αυτό το πρόβλημα τόσο εξαιτίας των περιορισμών που επιβάλλουν τα κανάλια PDCCHs, όπου επιτρέπουν την εξυπηρέτηση περιορισμένου αριθμού χρηστών, μικρότερου από το πλήθος των χρηστών της κυψέλης, σε κάθε TTI, όσο και εξαιτίας του τύπου της υπηρεσίας, όπου επιβάλει ρυθμό δεδομένων ακριβώς ίσο με 2 Mbps. Αντίθετα, για την κατηγορία των χρηστών που τρέχουν υπηρεσίες τύπου VBR, η καμπύλη της απόδοσης της κυψέλης σε συνάρτηση με το SNR δεν φτάνει στην κατάσταση κόρου, παρόλο που εξακολουθούν να ισχύουν οι περιορισμοί των καναλιών PDCCHs. Αυτό συμβαίνει γιατί οι χρήστες μπορούν να επιτύχουν ρυθμούς πάνω από 2 Mbps και, έτσι, η απόδοσή του συστήματος μπορεί να αυξηθεί και να ξεπεράσει αυτήν των χρηστών της πρώτης κατηγορίας, όπου ο ρυθμός δεδομένων τους περιορίζεται στα 2 Mbps. Επίσης, συγκρίνοντας τα TD RR και FD RR, παρατηρείται ότι η μεταβολή της απόδοσης της κυψέλης καθώς αυξάνονται οι χρήστες είναι ίδια όταν οι υπηρεσίες που τρέχουν στους χρήστες είναι του τύπου VBR. Συγκεκριμένα, η απόδοση της κυψέλης αυξάνεται μέχρις ότου το πλήθος των χρηστών φτάσει την τιμή κόρου. Από αυτό το σημείο και μετά, η απόδοση παραμένει σταθερή, ακόμα κι αν ο αριθμός των χρηστών συνεχίσει να αυξάνεται. Στην περίπτωση όμως όπου οι χρήστες τρέχουν CBR υπηρεσίες, οι τιμές της απόδοσης της κυψέλης για το ίδιο πλήθος χρηστών είναι διαφορετικές μεταξύ TD RR και FD RR. Αναλυτικότερα, η απόδοση των δύο αλγορίθμων είναι ίδια, μέχρι ο FD RR να φτάσει στο σημείο κορεσμού του, που είναι στους 20 χρήστες. Από αυτό το σημείο και έπειτα η απόδοση του TD RR είναι μεγαλύτερη, γιατί μπορούν να εξυπηρετηθούν περισσότεροι χρήστες απ' ότι στον FD RR.

Επίσης, από τον υπολογισμό της μέσης απόδοσης για κάθε χρήστη σε σχέση με το πλήθος των χρηστών παρατηρείται ότι, για υπηρεσίες CBR, η μέση απόδοση είναι σταθερή. Αντίθετα, για υπηρεσίες VBR, η μέση απόδοση του χρήστη μειώνεται, καθώς αυξάνεται το πλήθος των χρηστών. Αυτό συμβαίνει μέχρι το σημείο κορεσμού, όπου μετά από αυτό το σημείο η μέση απόδοση μένει σταθερή. Το σημείο κορεσμού υποδεικνύει το μέγιστο πλήθος χρηστών που μπορούν να εξυπηρετηθούν από την κυψέλη. Επιπλέον, όταν υπάρχουν πάνω από 20 χρήστες, όπου 20 είναι το σημείο κορεσμού του FD RR, η μέση απόδοση ανά χρήστη για τον FD RR είναι μεγαλύτερη από την αντίστοιχη απόδοση για τον TD RR, η οποία εξακολουθεί να μειώνεται. Αυτό συμβαίνει γιατί, για αυτήν την περίπτωση του FD RR, οι χρήστες που εξυπηρετούνται πραγματικά από το δίκτυο είναι λιγότεροι. Έτσι, οι πόροι διαμοιράζονται σε μικρότερο πλήθος χρηστών και ο καθένας τους λαμβάνει περισσότερους πόρους.

Το θετικό χαρακτηριστικό του αλγόριθμου Round Robin είναι ότι εξυπηρετούνται όλοι οι χρήστες και δεν υπάρχει κίνδυνος να μην εξυπηρετείται ένας χρήστης λόγω «λιμοκτονίας» (starvation), όπως συμβαίνει με άλλους αλγόριθμους. Επίσης, θεωρείται ως ο πιο δίκαιος αλγόριθμος διαμοιρασμού πόρων, καθώς σε όλες τις ανοιχτές συνδέσεις του δικτύου ανατίθενται ίσες ποσότητες πόρων. Ακόμη, ο Round Robin είναι ένας αλγόριθμος που μπορεί να υλοποιηθεί εύκολα και δεν χρειάζεται να

χρησιμοποιήσει επιπλέον πληροφορίες για την κατάσταση του συστήματος, πράγμα που πιθανότατα θα καθυστερούσε την εκτέλεσή του.

Ωστόσο, όπως αναφέραμε, ο αλγόριθμος λειτουργεί χωρίς να λαμβάνει υπόψη του κανένα κριτήριο προτεραιότητας ή απώλειας ζεύξεων. Για παράδειγμα, αναθέτει ίδιες ποσότητες πόρων σε όλες τις συνδέσεις, ακόμα και όταν κάποιες απαιτούν περισσότερους πόρους από άλλες. Μία περίπτωση όπου κάποια σύνδεση χρειάζεται περισσότερους πόρους από όσους της δίνονται είναι όταν η κατάσταση του καναλιού δεν είναι καλή, δηλαδή υπάρχουν μη ανεκτές απώλειες και γίνονται συχνές αναμεταδώσεις δεδομένων. Συνεπώς, η ποιότητα υπηρεσίας δεν θα είναι ίδια για όλες τις συνδέσεις και η συνολική απόδοση του δικτύου θα είναι μικρότερη, σε σχέση με άλλους αλγόριθμους που θα δούμε στη συνέχεια.

2.2 Blind Equal Throughput

Ο αλγόριθμος αυτός επιτυγχάνει την ίδια απόδοση για όλους τους χρήστες της κυψέλης, χωρίς να λάβει υπόψη την ποιότητα του καναλιού. Το μόνο που λαμβάνεται υπόψη είναι οι προτεραιότητες, οι οποίες ανατίθενται με βάση την μέση απόδοση που έχει κάθε χρήστης μέχρι στιγμής. Οι αποφάσεις χρονοπρογραμματισμού για τον αλγόριθμο του Blind Equal Throughput (BET) λαμβάνονται σύμφωνα με την ακόλουθη εξίσωση:

$$\hat{i}(t) = \arg \max_{j=1,\dots,N} \frac{1}{T_j(t)} \quad (1)$$

όπου το $T_j(t)$ είναι η προηγούμενη μέση απόδοση του UE j , την στιγμή t . Έτσι, ο χρονοπρογραμματιστής επιλέγει τον χρήστη με την μέγιστη προτεραιότητα, δηλαδή αυτόν με την ελάχιστη απόδοση. Δηλαδή, οι προτεραιότητες των χρηστών, P_j , δίνονται από το κλάσμα:

$$P_j = \frac{1}{T_j(t)} \quad (2)$$

Η απόδοση, $T_j(t)$, υπολογίζεται από τον τύπο :

$$T_j(t) = \begin{cases} \left(1 - \frac{1}{\tau}\right) T_j(t-1) + \frac{1}{\tau} R_j(t), & \text{αν ο χρήστης εξυπηρετήθηκε στο } t \\ \left(1 - \frac{1}{\tau}\right) T_j(t-1) & , \text{ αλλιώς} \end{cases} \quad (3)$$

όπου το $R_j(t)$ είναι ο ρυθμός δεδομένων που επιτυγχάνει ο χρήστης j την στιγμή t και ενημερώνεται σε κάθε TTΙ, ενώ το τ είναι το μήκος του χρονικού παραθύρου, μετρούμενο σε TTIs. Καθώς 1 TTΙ διαρκεί για 1 ms, οι συνηθισμένες τιμές του τ είναι μεταξύ 500 και 1000 TTIs. Έτσι, θα ισχύει ότι

$$0 < \frac{1}{\tau} < 1,$$

από το οποίο συνεπάγεται ότι θα ισχύει και η ανίσωση:

$$0 < 1 - \frac{1}{\tau} < 1.$$

Επίσης, το τ εξαρτάται από την ταχύτητα με την οποία κινείται ο χρήστης. Όταν ο χρήστης κινείται γρήγορα, το τ πρέπει να είναι μεγάλο, ενώ όταν κινείται αργά, το τ μπορεί να μειωθεί. Αυτό συμβαίνει γιατί όσο πιο αργά κινείται ένας χρήστης, τόσο πιο αργά θα μεταβάλλεται και ο στιγμιαίος ρυθμός δεδομένων του. (βλ. [1])

Σύμφωνα με την λογική του αλγόριθμου BET, κάθε φορά που γίνεται ανάθεση πόρων, ο χρήστης με την μικρότερη απόδοση αποκτά μεγαλύτερη προτεραιότητα έναντι των

άλλων χρηστών και οι διατιθέμενοι πόροι ανατίθενται σε αυτόν. Όταν κάποια στιγμή η απόδοσή αυτού του χρήστη γίνει ίση με ή μεγαλύτερη από την απόδοση κάποιου άλλου χρήστη της κυψέλης, τότε στον πρώτο χρήστη παύουν να του ανατίθενται πόροι, καθώς πλέον ο δεύτερος χρήστης αποκτά μεγαλύτερη προτεραιότητα. Με αυτόν τον τρόπο, οι χρήστες που έχουν πιο άσχημες συνθήκες καναλιού θα δεσμεύουν πιο συχνά πόρους. Έτσι, η απόδοση όλων των χρηστών είναι η ίδια, αλλά η συνολική απόδοση του συστήματος μειώνεται. (βλ. [2] και [6])

Υπάρχουν και εδώ δύο κατηγορίες του αλγόριθμου Blind Equal Throughput, μία για το πεδίο του χρόνου (Time Domain Blind Equal Throughput, TD BET) και μία για το πεδίο της συχνότητας (Frequency Domain Blind Equal Throughput, FD-BET). Στον αλγόριθμο TD-BET, ο χρονοπρογραμματιστής (scheduler) του eNB επιλέγει, σε κάθε TTI, τον χρήστη (UE – User Equipment) με την μεγαλύτερη προτεραιότητα και αναθέτει όλα τα RBs σε αυτόν τον χρήστη.

Από την άλλη, στον αλγόριθμο FD-BET, σε κάθε TTI ο χρονοπρογραμματιστής του eNB λειτουργεί ως εξής: Αρχικά, επιλέγει για ανάθεση πόρων το UE με την μεγαλύτερη προτεραιότητα, που θα είναι αυτό με την μικρότερη απόδοση. Στη συνέχεια, αναθέτει ένα RB σε αυτό το UE, υπολογίζει την αναμενόμενη απόδοση για αυτό το UE και την χρησιμοποιεί για να την συγκρίνει με την απόδοση που έχει υπολογιστεί ήδη για τα υπόλοιπα UEs. Ο scheduler συνεχίζει να αναθέτει RBs σε αυτό το UE, μέχρις ότου η αναμενόμενη απόδοσή του να μην είναι πια η μικρότερη μεταξύ των υπολογισμένων αποδόσεων όλων των χρηστών. Ύστερα ο scheduler θα ακολουθήσει την ίδια λογική για να αναθέσει RBs στο επόμενο UE που θα έχει την μικρότερη μέση απόδοση, μέχρι να ανατεθούν όλα τα RBs σε κάποιους χρήστες. Γενικά, η βασική αρχή του αλγορίθμου είναι ότι ο χρονοπρογραμματιστής κάνει ότι μπορεί για να επιτύχει ίση απόδοση μεταξύ όλων των UEs.

Στο [8] πραγματοποιείται μοντελοποίηση των αλγορίθμων TD-BET και FD-BET, στο ns3, και συγκρίνονται οι αποδόσεις τους. Τόσο για τον TD-BET όσο και για τον FD-BET, χρησιμοποιείται το CQI ευρείας ζώνης, καθώς ο αλγόριθμος δεν λαμβάνει υπόψη του την κατάσταση του καναλιού και συνεπώς τα αποτελέσματα θα είναι όμοια, ανεξαρτήτως CQI. Στα πειράματα που υλοποιούνται γίνεται η υπόθεση ότι ο RLC buffer του κάθε UE είναι πάντα κορεσμένος, ώστε να γίνεται ακόμα πιο εμφανής ο τρόπος που διαμοιράζονται οι πόροι στους δύο αυτούς αλγορίθμους. Επίσης, στο paper υλοποιούνται δύο σενάρια. Στο πρώτο σενάριο, όλα τα UE έχουν την ίδια απόσταση από το eNB, έτσι ώστε να έχουν όλοι το ίδιο CQI. Στο δεύτερο σενάριο, κάθε UE έχει διαφορετική απόσταση από το eNB, έτσι ώστε το καθένα να έχει διαφορετικό CQI.

Από τα πειράματα και για τα δύο σενάρια παρατηρείται, τόσο για τον TD-BET όσο και για τον FD-BET, ότι, παρόλο που η απόδοση όλων των χρηστών είναι ίδια, οι τιμές αναφοράς για τα δύο σενάρια είναι διαφορετικές. Στο πρώτο σενάριο, όλα τα UEs μπορούν να αποκτήσουν τον ίδιο ρυθμό δεδομένων, καθώς έχουν το ίδιο CQI. Συνεπώς, σε αυτήν την περίπτωση, ο FD-BET αναθέτει το κάθε RB στα UEs, τα οποία επιλέγονται με την σειρά, μέχρι που όλα τα RBs να έχουν ανατεθεί στους χρήστες. Από την άλλη, ο TD-BET λειτουργεί όπως ο Round Robin, ο οποίος αναθέτει όλα τα RBs σε ένα UE στο τρέχον TTI και στο επόμενο TTI αναθέτει όλα τα RBs σε κάποιο άλλο UE. Αν υποθέσουμε ότι το πλήθος των RBs είναι ακέραιο πολλαπλάσιο του N, όπου N είναι το πλήθος των UEs, τότε η αναμενόμενη απόδοση για τα FD-BET και TD-BET δίνεται από τις ακόλουθες εξισώσεις:

$$T_{fdbet}^I = \frac{S(MCS_{\frac{B}{N}})}{\tau} \quad (4)$$

$$T_{tdbet}^I = \frac{S(MCS, B)}{\tau * N} \quad (5)$$

όπου T_{fdbet}^I και T_{tdbet}^I είναι η απόδοση των αλγορίθμων FD-BET και TD-BET, αντίστοιχα, για το πρώτο σενάριο, το MCS (*Modulation And Code Scheme*) είναι το σχήμα διαμόρφωσης και κωδικοποίησης που χρησιμοποιείται για δεδομένο λόγο σήματος προς παρεμβολή και θόρυβο (SINR – Signal to Interference plus Noise Ratio), το B είναι το συνολικό πλήθος των RBs που μπορούν να χρησιμοποιηθούν, το $S(MCS, B)$ είναι το μέγεθος του μεταφερόμενου block και το τ είναι η διάρκεια του TTI. Όπως φαίνεται και από τις Εξισώσεις (4) και (5), ο FD-BET, σε κάθε TTI, αναθέτει $\frac{B}{N}$ RBs σε κάθε UE, ενώ ο TD-BET αναθέτει B RBs σε κάθε N TTIs. Από τα αποτελέσματα της προσομοίωσης προκύπτει ότι και για τους δύο αλγόριθμους BET, όλοι οι χρήστες έχουν την ίδια απόδοση, η οποία είναι σχεδόν ίδια με την αναμενόμενη απόδοση.

Στο δεύτερο σενάριο, ο FD-BET θα αναθέσει ένα διαφορετικό πλήθος από RBs σε κάθε UE. Από την άλλη, στον TD-BET, ο χρόνος μεταξύ δύο γεγονότων χρονοπρογραμματισμού για ένα UE δεν είναι πάντα ο ίδιος, αλλά μεταβάλλεται. Έτσι, δεν μπορούν να χρησιμοποιηθούν οι εξισώσεις του πρώτου σεναρίου, για τον υπολογισμό της αναμενόμενης απόδοσης. Έστω F_i το κλάσμα του χρόνου που ανατίθεται στο UE i για όλη την διάρκεια της προσομοίωσης και τα R_i^{fb} και T_i είναι ο συνολικός ρυθμός δεδομένων και η απόδοση, αντίστοιχα, που επιτυγχάνονται για το UE i . Τότε θα ισχύει ότι:

$$T_i = F_i * R_i^{fb} \quad (6)$$

Στον TD-BET, το άθροισμα όλων των F_i ισούται με μονάδα. Έτσι, μπορούμε να αναδιατάξουμε την Εξίσωση (6) στην ακόλουθη:

$$\sum_i \frac{T_i}{R_i^{fb}} = 1 \quad (7)$$

Μακροπρόθεσμα, όλα τα UE έχουν το ίδιο T_i κι έτσι μπορούμε να αντικαταστήσουμε το T_i με T . Συνεπώς, η αναμενόμενη απόδοση για τον TD-BET σε αυτό το σενάριο θα είναι η:

$$T_{tdbet}^{II} = \frac{1}{\sum_{i=1}^N \frac{1}{R_i^{fb}}} \quad (8)$$

Στις δοκιμές που πραγματοποιήθηκαν, κάθε UE αποκτά το ίδιο πλήθος από RBs, τόσο στον FD-BET όσο και στον TD-BET. Επιπλέον, ισχύει ότι το $\sum_i S(MCS, B_i)$ είναι περίπου ίσο με το $S(MCS, B)$, $B = \sum_i B_i$. Εδώ, το B_i είναι το πλήθος των RBs που ανατίθενται στο UE i σε ένα TTI, στον FD-BET. Εφόσον ένα UE μπορεί να αποκτήσει το ίδιο πλήθος από RBs και στις δύο περιπτώσεις, οι αποδόσεις του για τους FD-BET και TD-BET θα έχουν πολύ κοντινές τιμές. Επομένως, η Εξίσωση (8) μπορεί να χρησιμοποιηθεί επίσης για τον υπολογισμό του T_{fdbet}^{II} . Από τα αποτελέσματα του πειράματος παρατηρείται, όπως και στην περίπτωση του πρώτου σεναρίου, ότι η απόδοση όλων των χρηστών είναι σχεδόν ίδια με την αναμενόμενη τιμή.

Όπως και στον αλγόριθμο RR, ο BET δεν χρειάζεται να έχει γνώση της κατάστασης του καναλιού, πράγμα που καθιστά την υλοποίησή του αρκετά απλή. Επίσης, είναι ένας δίκαιος αλγόριθμος, υπό την έννοια ότι επιτυγχάνει ίδια απόδοση για όλους τους χρήστες, ακόμα κι αν κάποιοι έχουν χειρότερες συνθήκες καναλιού από άλλους.

2.3 Token Bank Fair Queue

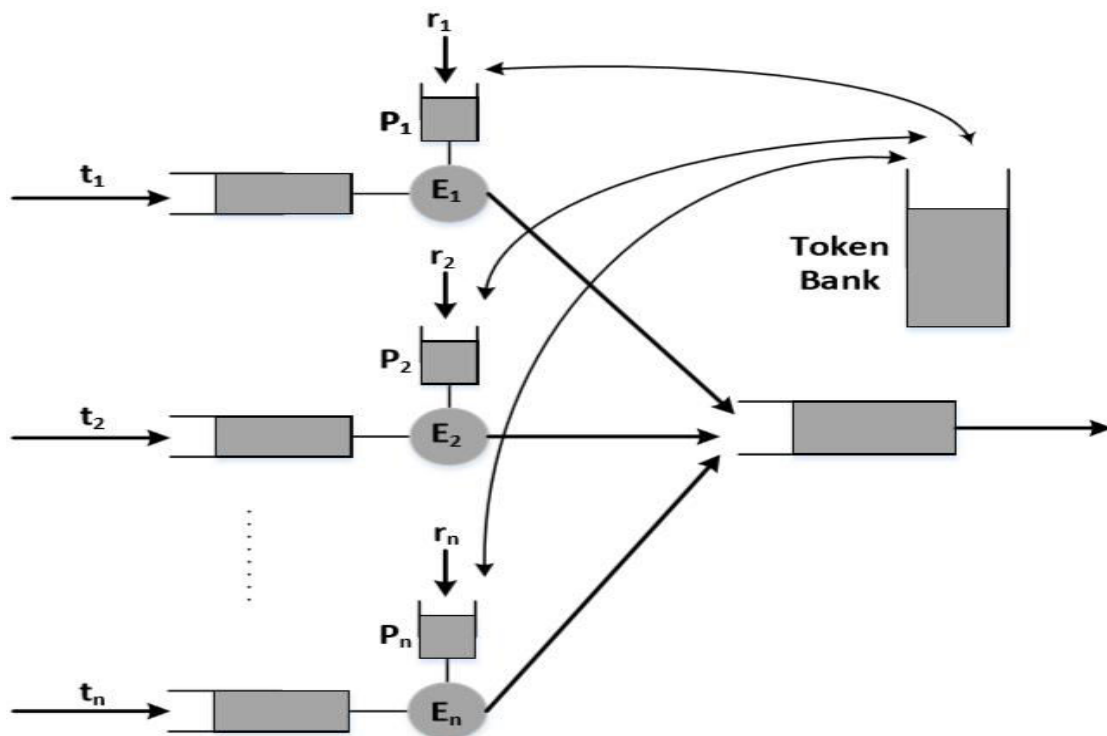
Ο μηχανισμός του Token Bank Fair Queue (TBFQ) scheduler αποσκοπεί στην εγγύηση ενός ελάχιστου ρυθμού δεδομένων, λαμβάνοντας υπόψη του τις απαιτήσεις σε ποιότητα υπηρεσίας (QoS). Πηγή έμπνευσης του TBFQ αλγόριθμου αποτελεί ο μηχανισμός του τρύπιου δοχείου (leaky bucket). Ο μηχανισμός του τρύπιου δοχείου είναι μια μέθοδος αστυνόμευσης που χρησιμοποιείται στα δίκτυα υπολογιστών. Στην μέθοδο αυτή χρησιμοποιείται ένα τρύπιο δοχείο το οποίο χωράει b κουπόνια (tokens). Πέρα από το δοχείο, χρησιμοποιεί και μια γεννήτρια παραγωγής κουπονιών, η οποία παράγει r κουπόνια το δευτερόλεπτο. Όταν το δοχείο είναι γεμάτο, δηλαδή όταν έχει ήδη b κουπόνια, τότε τα νέα κουπόνια που παράγονται απορρίπτονται, αλλιώς εισάγονται μέσα στο δοχείο. Ο μηχανισμός του τρύπιου δοχείου μπορεί να χρησιμοποιηθεί για την αστυνόμευση μιας ροής πακέτων ως εξής: Κάθε πακέτο πρέπει να πάρει ένα κουπόνι από το δοχείο, προτού μπορέσει να μεταδοθεί στο δίκτυο. Αν το δοχείο είναι άδειο, τότε πρέπει να περιμένει για ένα νέο κουπόνι. Καθώς το μέγιστο πλήθος κουπονιών που μπορούν να υπάρχουν στο δοχείο είναι b , τότε το μέγιστο πλήθος των πακέτων που μπορούν να υπάρχουν σε μια ροή είναι επίσης b . Ακόμη, επειδή ο ρυθμός παραγωγής κουπονιών είναι r , το μέγιστο πλήθος πακέτων που μπορούν να εισέλθουν στο δίκτυο σε διάρκεια t είναι $rt + b$. Έτσι, ο ρυθμός παραγωγής κουπονιών από την γεννήτρια περιορίζει την μακροπρόθεσμη μέση ταχύτητα με την οποία μπορούν να εισέρχονται πακέτα στο δίκτυο. [9]

Όπως αναφέραμε, η λειτουργία του TBFQ, η οποία παρουσιάζεται στην Εικόνα 6, μοιάζει αρκετά με τον μηχανισμό του τρύπιου δοχείου. Στον TBFQ η ροή κίνησης του UE i καθορίζεται από τις ακόλουθες 4 παραμέτρους: τον ρυθμό άφιξης πακέτων, t_i (bytes/sec), τον ρυθμό παραγωγής κουπονιών, r_i (bytes/sec), το μέγεθος της δεξαμενής κουπονιών, p_i (bytes) και τον μετρητή που καταγράφει το πλήθος των κουπονιών που έχουν ανταλλαχθεί μεταξύ της τράπεζας κουπονιών (token bank) και της ροής i , E_i . Ο μετρητής E_i μπορεί να είναι μικρότερος του μηδενός. Ο TBFQ χρησιμοποιεί μια κοινή τράπεζα κουπονιών, προκειμένου να εξισορροπήσει την κίνηση των ροών. Σε αυτήν, κάθε κουπόνι αντιπροσωπεύει ένα byte δεδομένων. Έτσι, κάθε k Bytes δεδομένων καταναλώνουν k κουπόνια. Αν ο ρυθμός παραγωγής κουπονιών, r_i , είναι μικρότερος από τον ρυθμό άφιξης πακέτων, t_i , τότε κάποια στιγμή η δεξαμενή κουπονιών του UE i θα αδειάσει. Τότε, εάν η ροή i έχει να στείλει και άλλα πακέτα, πρέπει να δανειστεί κουπόνια από την τράπεζα, ανάλογα με την προτεραιότητά της. Σε αυτήν την περίπτωση, το E_i μειώνεται τόσο όσο το πλήθος των κουπονιών που δανείστηκαν από την τράπεζα. Αντίθετα, αν $r_i > t_i$, τότε τα κουπόνια, που περισσεύουν και δεν μπορούν να εισαχθούν στην δεξαμενή κουπονιών του χρήστη, αποθηκεύονται στην τράπεζα και το E_i αυξάνεται κατά το πλήθος των προστιθέμενων κουπονιών. Έτσι, σε περιόδους όπου το t_i είναι μικρότερο του r_i , η δεξαμενή θα έχει πάντα αρκετά κουπόνια για να εξυπηρετήσει τα αφικνούμενα πακέτα και το E_i αυξάνεται. Από την άλλη, σε περιόδους όπου το t_i είναι μεγαλύτερο του r_i , τα κουπόνια που φεύγουν από την δεξαμενή, μέσα σε ένα διάστημα t , είναι πιο πολλά από αυτά που έρχονται. Σε αυτήν την περίπτωση, η σύνδεση μπορεί να δανειστεί κουπόνια από την τράπεζα. Η προτεραιότητα με την οποία μια σύνδεση μπορεί να δανειστεί κουπόνια από την τράπεζα εξαρτάται από την μετρική προτεραιότητας, P_i , η οποία υπολογίζεται από το κλάσμα:

$$P_i = \frac{E_i}{r_i} \quad (9)$$

Αναθέτοντας προτεραιότητες στις εισερχόμενες ροές με αυτόν τον τρόπο, μπορούμε να είμαστε πιο σίγουροι ότι οι ροές που ανήκουν σε UEs που υπόκεινται σε ποικίλες παρεμβολές και απώλειες δεδομένων θα έχουν υψηλότερο δείκτη προτεραιότητας,

καθώς θα βρίσκονται πιο συχνά στην κατάσταση με $t_i < r_i$ και έτσι θα συνεισφέρουν πιο συχνά στην τράπεζα.



Εικόνα 6: Χρονοπρογραμματισμός κατερχόμενη ζεύξης με τον TBFQ scheduler [2]

Όπως είναι προφανές, ο χρήστης που συνεισφέρει περισσότερο στην τράπεζα κουπονιών έχει μεγαλύτερη προτεραιότητα για να δανειστεί κουπόνια από αυτήν, όταν χρειάζεται. Από την άλλη, ο χρήστης που δανείζεται περισσότερα κουπόνια από την τράπεζα έχει μικρότερη προτεραιότητα να συνεχίσει να εξάγει κουπόνια. Έτσι, σε περίπτωση που πολλοί χρήστες έχουν τον ίδιο ρυθμό παραγωγής κουπονιών, τον ίδιο ρυθμό δεδομένων και το ίδιο μέγεθος για την δεξαμενή κουπονιών, ο χρήστης που υπόκειται στην υψηλότερη παρεμβολή θα έχει περισσότερες ευκαιρίες να δανειστεί κουπόνια από την τράπεζα. Επίσης, ο TBFQ μπορεί να επιβλέπει την κίνηση αλλάζοντας τον ρυθμό παραγωγής κουπονιών.

Αρχικά, ο αλγόριθμος του TBFQ σχεδιάστηκε για συστήματα ενός φέροντος (carrier) και έτσι δεν είναι κατάλληλος για OFDM συστήματα πολλαπλών φερόντων. Συνεπώς, απαιτούνται κάποιες επιπρόσθετες παράμετροι, προκειμένου να μπορεί να χρησιμοποιηθεί ο TBFQ σε OFDM συστήματα. Αυτές οι αλλαγές χρειάζονται για να μπορέσει ο scheduler να ανταποκριθεί στην υλοποίηση και στις απαιτήσεις των 4G δικτύων. Σε αυτά τα δίκτυα, η χρησιμοποίηση των πόρων και, κατά συνέπεια, η απόδοση του δικτύου μπορούν να βελτιωθούν αξιοποιώντας τα συστήματα πολλαπλής πρόσβασης που χρησιμοποιούνται. Επίσης, απαιτείται γρήγορος χρονοπρογραμματισμός, προκειμένου να μπορεί να χρησιμοποιηθεί η ανάδραση καναλιού, ενώ παράλληλα χρειάζεται να διατηρηθεί ένα ελάχιστο επίπεδο δικαιοσύνης και απόδοσης για όλους τους χρήστες.

Στο OFDM η βασική χρονοσυχνοτική μονάδα πόρων αναφέρεται ως ένα κομμάτι (chunk). Αυτό αποτελείται από μια τετράγωνη περιοχή χρόνου – συχνότητας, η οποία περιλαμβάνει ένα πλήθος από (μεταγενέστερα) OFDM σύμβολα και ένα πλήθος από γειτονικά subcarriers. Τα πακέτα από τις ροές κίνησης αντιστοιχίζονται αποκλειστικά με αυτά τα chunks, βασισμένα στις απαιτήσεις QoS, που προκύπτουν από το Radio Link Control (RLC) επίπεδο, καθώς και στην ανάδραση καναλιού, που λαμβάνεται από το φυσικό επίπεδο. Η ανάδραση καναλιού περιλαμβάνει τον λόγο SNR, ο οποίος υπολογίζεται από το UE στο τμήμα της κατερχόμενης ζεύξης του πλαισίου j .

Τελικά, κάποιες από τις επιπρόσθετες παραμέτρους των ροών που ελέγχει ο scheduler είναι το όριο οφειλών (dept limit), d_i , το οποίο αποτρέπει ένα UE από το να δανείζεται υπερβολικά πολλά κουπόνια από την τράπεζα κουπονιών, το μέγιστο πλήθος των κουπονιών που μπορεί το UE i να δανειστεί κάθε φορά από την τράπεζα, c_i , και το ελάχιστο πλήθος κουπονιών, C , που πρέπει να αποθηκεύσει στην τράπεζα το κάθε UE προκειμένου να μπορεί να δανειστεί περισσότερα κουπόνια όταν φτάσει το όριο dept limit. Δηλαδή, αν το E_i γίνει μικρότερο από το κατώφλι d_i , το UE i δεν μπορεί να δανειστεί επιπλέον κουπόνια από την τράπεζα. Αντίθετα, αν το E_i γίνει μεγαλύτερο ή ίσο από το όριο d_i , τότε το UE i πρέπει να αποθηκεύσει C κουπόνια στην τράπεζα, προκειμένου να μπορέσει να δανειστεί επιπλέον κουπόνια. [2] [10]

Υπάρχουν δύο αλγόριθμοι του TBFQ scheduler, ένας για το πεδίο της συχνότητας (FD-TBFQ) και ένας για το πεδίο του χρόνου (TD-TBFQ). Στον FD-TBFQ, ο scheduler επιλέγει πάντα το UE με την υψηλότερη προτεραιότητα και αναθέτει τα RBs με το υψηλότερο CQI υποζώνης μέχρι που να μην υπάρχουν άλλα πακέτα στον RLC buffer του UE ή μέχρι να έχουν ανατεθεί όλα τα RBs. Στον TD-TBFQ, αφού επιλεγεί το UE με την μέγιστη προτεραιότητα, ο scheduler αναθέτει όλα τα RBs σε αυτό το UE χρησιμοποιώντας το CQI ευρείας ζώνης.

Στο [8] υλοποιούνται οι αλγόριθμοι FD-TBFQ και TD-TBFQ και πραγματοποιούνται τα ίδια πειράματα που χρησιμοποιήθηκαν για τον BET. Προκειμένου να τεστάρουν τα χαρακτηριστικά ποιότητας υπηρεσίας, οι συγγραφείς του paper έχουν μοντελοποιήσει UDP ροές με διάφορους σταθερούς ρυθμούς δεδομένων. Επιπλέον, το μεσοδιάστημα ανάμεσα στις αποστολές των πακέτων έχει τεθεί σε 1 ms, για να μην αδειάζει ο RLC buffer. Καθώς τα CQI στενής και ευρείας ζώνης είναι σταθερά, ο FD-TBFQ και ο TD-TBFQ θα λειτουργήσουν με τον ίδιο τρόπο. Συνεπώς, παρακάτω θα αναφερθούμε μόνο στον FD-TBFQ scheduler.

Στο πρώτο σενάριο των πειραμάτων, όπου όλοι οι χρήστες έχουν την ίδια απόσταση από το eNB, επειδή κάθε ροή έχει τον ίδιο ρυθμό κίνησης, t_i , και τον ίδιο ρυθμό παραγωγής κουπονιών, r_i , ο FD-TBFQ scheduler θα εξασφαλίσει την ίδια απόδοση για όλα τα UEs. Επίσης, η ακριβής αναμενόμενη απόδοση του UE εξαρτάται από τον συνολικό ρυθμό της UDP ροής και δίνεται από την ακόλουθη εξίσωση

$$T_{fdtbfq}^i = \begin{cases} t_i, \text{αν } \sum_i t_i < T_t \\ \frac{T_t}{N}, \text{αν } \sum_i t_i \geq T_t \end{cases} \quad (10)$$

όπου

$$T_t = \frac{S(MCS, B)}{\tau}$$

Εδώ, το T_t είναι η μέγιστη συνολική απόδοση, η οποία ισούται με τον ρυθμό δεδομένων που επιτυγχάνεται όταν όλα τα RBs ανατίθενται σε ένα UE. Όταν ο συνολικός ρυθμός της κίνησης ($\sum_i t_i$) είναι μικρότερος από την μέγιστη συνολική απόδοση (T_t), ο FD-TBFQ μπορεί να ελέγχει την κίνηση μέσω του ρυθμού παραγωγής κουπονιών (r_i), έτσι ώστε η αναμενόμενη απόδοση του UE να ισούται με τον ρυθμό της κίνησής του. Αντιθέτως,

όταν ο συνολικός ρυθμός των ροών είναι μεγαλύτερος από την μέγιστη συνολική απόδοση T_t , το eNB δεν μπορεί να εξυπηρετήσει όλους τους χρήστες μέσα σε ένα TTI. Συνεπώς, σε κάθε TTI, ο FD-TBFQ scheduler θα αναθέτει όλα τα RBs σε ένα UE εξαιτίας του μεγάλου μεγέθους του RLC buffer. Όταν γίνει ο χρονοπρογραμματισμός αυτού του UE στο τρέχον TTI, ο μετρητής E_i μειώνεται για να μην επιλεγεί πάλι το ίδιο UE για ανάθεση πόρων στο επόμενο TTI. Επειδή το κάθε UE έχει τον ίδιο ρυθμό t_i , ο FD-TBFQ θα εξυπηρετήσει το κάθε UE με την σειρά και θα εξυπηρετεί μόνο ένα UE σε κάθε TTI. Συνεπώς, στην δεύτερη περίπτωση, η αναμενόμενη απόδοση για ένα UE ισούται με το κλάσμα της μέγιστης συνολικής απόδοσης διαιρούμενη με το πλήθος των UEs.

Από τα αποτελέσματα της προσομοίωσης για το πρώτο σενάριο φαίνεται ότι η απόδοση των χρηστών που βρίσκονται σε ίδια απόσταση από το eNB είναι ίδια. Επίσης, η απόδοση των χρηστών που βρίσκονται σχετικά κοντά στο eNB ισούται με τον ρυθμό της κίνησης. Αυτό συμβαίνει γιατί η απόδοση των UE περιορίζεται από τον ρυθμό παραγωγής κουπονιών. Αντίθετα, για τους χρήστες που βρίσκονται πιο μακριά, η απόδοσή τους είναι μικρότερη από τον ρυθμό της κίνησης. Αυτό συμβαίνει γιατί ο συνολικός ρυθμός της UDP κίνησης είναι μεγαλύτερος από την συνολική μέγιστη απόδοση.

Επιπλέον, στο δεύτερο σενάριο, όπου κάθε χρήστης έχει διαφορετική απόσταση από το eNB, οι αναμενόμενες αποδόσεις των UEs εξαρτώνται από την μέγιστη συνολική απόδοση, T_t . Ωστόσο, η μέγιστη συνολική απόδοση σε αυτήν την περίπτωση είναι διαφορετική από αυτήν του πρώτου σεναρίου. Όπως και στην περίπτωση το πρώτου σεναρίου, ο αλγόριθμος FD-TBFQ συμπεριφέρεται όπως ο αλγόριθμος BET, όταν ο συνολικός ρυθμός της κίνησης είναι μεγαλύτερος από την μέγιστη συνολική απόδοση. Συνεπώς, στο δεύτερο σενάριο, η απόδοση, T_{fdtbfq}^{II} , μπορεί να υπολογιστεί από την Εξίσωση (10), λαμβάνοντας όμως υπόψη ότι:

$$T_t = \frac{N}{\sum_{i=1}^N \frac{1}{R_i^{fb}}}$$

Για την προσομοίωση του δεύτερου σεναρίου οι χρήστες έχουν τοποθετηθεί σε διαφορετικές αποστάσεις από το eNB. Παρατηρείται ότι η απόδοση των χρηστών που έχουν τον ίδιο ρυθμό δεδομένων είναι σχεδόν ίδια για όλους τους χρήστες.

2.4 Proportional Fair

Ο PF είναι ένας ευρέως χρησιμοποιούμενος αλγόριθμος προγραμματισμού ανάθεσης πόρων και αποτελεί τον default αλγόριθμο που χρησιμοποιεί ο δικτυακός προσομοιωτής NS-3. Αρχικά υλοποιήθηκε στο πεδίο του χρόνου και αργότερα υιοθετήθηκε από το LTE για να αξιοποιήσει τις δυνατότητες του OFDMA και στο επίπεδο του χρόνου και στο επίπεδο της συχνότητας. Ο κύριος σκοπός της ύπαρξης συνδυαστικών συστημάτων που πραγματοποιούν προγραμματισμό ανάθεσης πόρων στο πεδίο του χρόνου (TDS) και στο πεδίο της συχνότητας (FDS) είναι για να επιτευχθεί μια ισορροπία ανάμεσα στην ολική απόδοση του συστήματος και δικαιοσύνη-ισότητα του ρυθμού δεδομένων ανάμεσα στους χρήστες μέσω της αξιοποίησης της πολυμορφίας πολλαπλών χρηστών. [11]

Η γνωστή και ως “Παράμετρος Δικαιοσύνης (FA)” στον PF αναφέρεται στην ποσότητα των πόρων που θα κατανεμηθούν σε ένα δοσμένο χρονικό παράθυρο. Η παρακάτω εξίσωση περιγράφει την παράμετρο αυτή στον PF scheduler. [11]

$$FA(\Delta T) = \frac{(\sum_{m=1}^M A_m(\Delta T))^2}{(M \times \sum_{m=1}^M A_m(\Delta T)^2)} \quad (11)$$

Όπου $0 < FA(\Delta T) \leq 1$ (Όταν $FA(\Delta T) = 1$, όλοι οι χρήστες λαμβάνουν την ίδια ποσότητα από πόρους), M είναι το πλήθος των χρηστών, m είναι ο δείκτης για τον εκάστοτε χρήστη κάθε φορά, και $A_m(\Delta T)$ είναι ο αριθμός των οντοτήτων (πόρων) που προγραμματίστηκαν να κατανεμηθούν στον χρήστη m το χρονικό διάστημα ΔT .

Ο PF scheduler υποστηρίζει έναν υψηλό βαθμό χρησιμοποίησης πόρων διατηρώντας παράλληλα μια καλή ισότητα μεταξύ των ροών του δικτύου. Ένας χρήστης είναι πιθανό να προγραμματιστεί να του ανατεθούν πόροι όταν η στιγμιαία ποιότητα του καναλιού του είναι υψηλή σχετικά με το μέσο όρο της ποιότητας του καναλιού του στην πάροδο του χρόνου. Έστω M_i είναι το σχήμα διαμόρφωσης και κωδικοποίησης (MCSI) που ανατίθεται σε ένα χρήστη ανάλογα με την αντίστοιχη ποιότητα καναλιού του (CQI). Χρησιμοποιώντας την χαρτογράφηση/ανάθεση TBS, $S(M_j, B)$ είναι το μέγεθος του μπλοκ των bit που μεταφέρονται όπως ορίζεται στο 3GPP πρότυπο TS36.213, όπου B είναι το πλήθος των RB που χρησιμοποιούνται. Ο αριθμός του μπλοκ πόρων (RB) προκύπτει από το εύρος ζώνης του καναλιού μετάδοσης. Έστω $T_j(t)$ είναι η προηγούμενη μέση απόδοση του χρήστη j . Συνεπώς ο ρυθμός που μπορεί να επιτευχθεί από το χρήστη j δίνεται από την ακόλουθη σχέση. [12]

$$R_j(k, t) = \frac{S(M_j, B)}{\tau} \quad (12)$$

όπου τ είναι το χρονικό διάστημα μετάδοσης.

Ο προγραμματισμός ανάθεσης πόρων στους χρήστες προκύπτει από την σχέση:

$$\hat{t}_k(t) = \underset{j=1, \dots, N}{\operatorname{argmax}} \frac{R_j(k, t)}{T_j(t)} \quad (13)$$

Η μέση απόδοση δίνεται από την σχέση:

$$T_j(t+1) = \begin{cases} (1-\lambda)T_j(t) + \lambda \hat{T}_j(t), & \text{if } j \text{ user is scheduled} \\ (1-\lambda)T_j(t), & \text{otherwise} \end{cases} \quad (14)$$

Το λ είναι μια σταθερά και το $T_j(t)$ είναι η πραγματική απόδοση που επιτυγχάνεται από τον χρήστη j στο sub frame t .

Λειτουργία Proportional Fair Scheduler

Αρχικά το eNodeB αποκτά το feedback της στιγμιαίας κατάστασης της ποιότητας του καναλιού (CQI) για κάθε UE k τη χρονική στιγμή t με βάση ένα ζητούμενο ρυθμό δεδομένων $R_{k,n}(t)$. Έπειτα παρακολουθεί τον μεταβαλλόμενη μέση τιμή της απόδοσης $T_{k,n}(t)$ του κάθε UE k σε κάθε φυσικό μπλοκ πόρων (PRB) n μέσα σε ένα χρονικό παράθυρο μήκους t_c . Η παράμετρος t_c ελέγχει την λανθάνουσα κατάσταση του συστήματος. Αν το t_c είναι μεγάλο ο scheduler προσεγγίζει τον MR αλγόριθμο, αν το t_c είναι μικρό ο scheduler προσεγγίζει τον RR αλγόριθμο. Ο μηχανισμός χρονοπρογραμματισμού πόρων δίνει προτεραιότητα στο UE k^* την t -στη χρονική στιγμή και η πόρους που ικανοποιούν τη μέγιστη σχετική κατάσταση της ποιότητας του καναλιού. Ο χρήστης k^* υπολογίζεται ως εξής [13]:

$$k^* = \underset{k=1, 2, \dots, K}{\operatorname{argmax}} \frac{[R_{k,n}(t)]^a}{[T_{k,n}(t)]^b} \quad (15)$$

Από την παραπάνω σχέση διακρίνουμε τις εξής τρεις περιπτώσεις

- Αν $a=b=1$, η σχέση (15) περιγράφει τον PF αλγόριθμο.
- Αν $a=1$, $b=0$, η σχέση (15) ακολουθεί τον MR αλγόριθμο.
- Αν $a=0$, $b=1$, η σχέση (15) ακολουθεί τον RR αλγόριθμο.

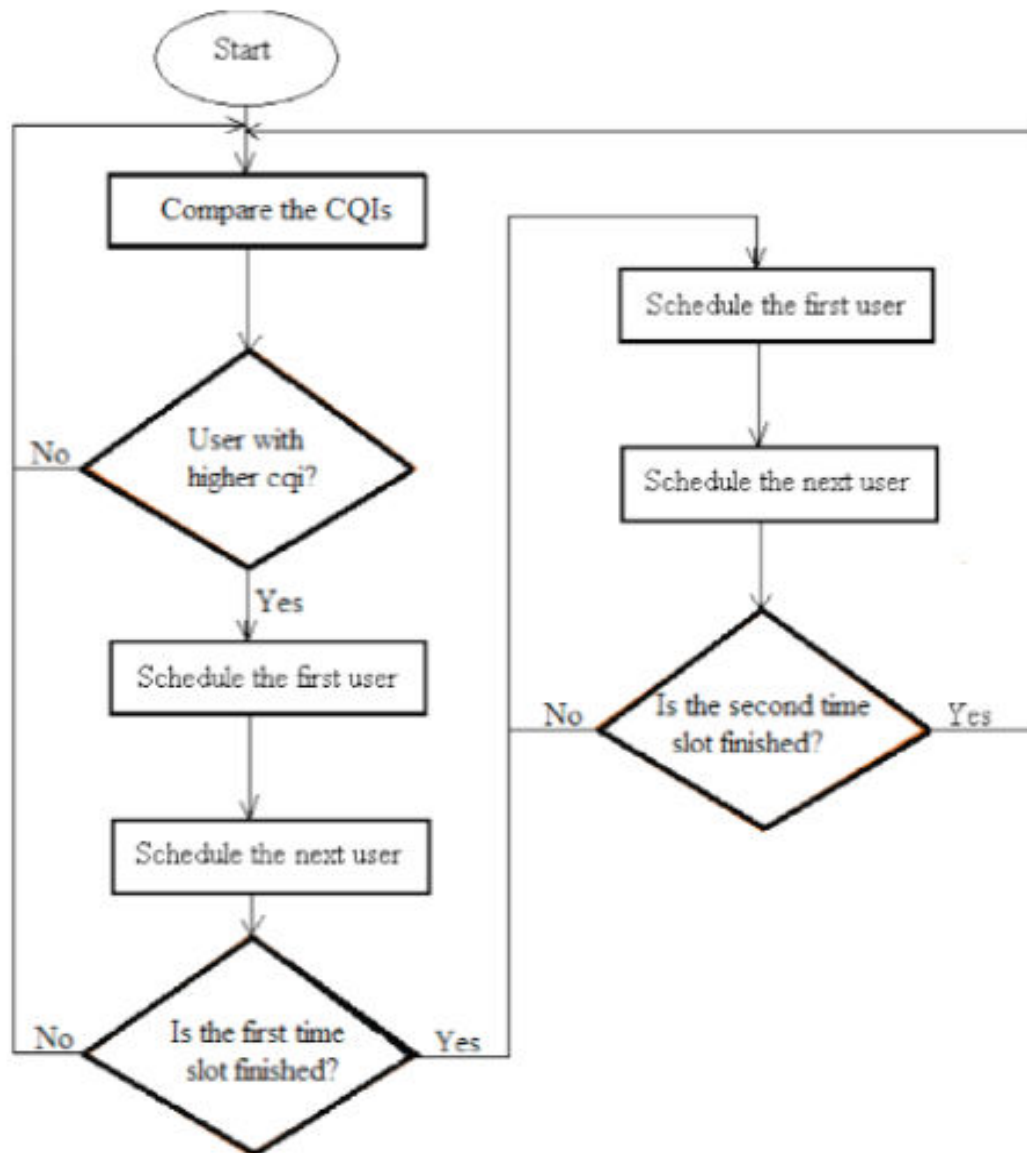
Μια άλλη προσέγγιση υπολογισμού της απόδοσης με βάση το [13] είναι η ακόλουθη.

$$T_{k,n}(t+1) = \begin{cases} \left(1 - \frac{1}{t_c}\right) T_{k,n}(t) + \frac{1}{t_c} R_{k,n}(t), & \text{όταν } k^* = k \\ \left(1 - \frac{1}{t_c}\right) T_{k,n}(t), & \text{όταν } k^* \neq k \end{cases} \quad (16)$$

Ο PF scheduler γενικά ενημερώνει την απόδοση $T_{k,n}(t)$ του k -στού χρήστη την t -στη χρονική στιγμή χρησιμοποιώντας την εκθετική μεταβαλλόμενη μέση τιμή του παραπάνω φίλτρου.

Μιας και παραπάνω γίνεται αναφορά στον MR αλγόριθμο αξίζει να γίνει μια αναφορά και στον τρόπο λειτουργίας αυτού του αλγορίθμου. Ο αλγόριθμος αυτός κάνει schedule τα UEs που έχουν υψηλότερο λόγο σήματος προς θόρυβο (SINR) από την απαιτούμενη τιμή του SINR κατώφλι (threshold), ενώ δεν προγραμματίζουν εκείνα τα UEs των οποίων η εμπειρία διαχωρίζει την εξασθένιση του καναλιού (channel fading). Ως αποτέλεσμα, ο MR scheduler παρέχει υψηλότερη χωρητικότητα και απόδοση από οποιονδήποτε άλλο αλγόριθμο προγραμματισμού ανάθεσης πόρων. Όμως, αγνοεί πλήρως την δικαιοσύνη ανάμεσα στα UEs μέσα σε μια κυψέλη. Είναι γνωστό, στα ασύρματα κινητά συστήματα, ότι τα UEs που τοποθετούνται σε διαφορετικές αποστάσεις έχουν διαφορετικές συνθήκες εξασθένισης σήματος. Συνεπώς, προγραμματίζοντας τα UEs που έχουν υψηλό SINR οδηγεί σε άδικο καταμερισμό πόρων ανάμεσα στα UEs της κυψέλης [14].

Παρακάτω ακολουθεί μια σχηματική απεικόνιση της λειτουργίας του PF scheduler όπως μας δίνεται στο [15].



Εικόνα 7: Σχηματική αναπαράσταση PF.

Τέλος, πειράματα (βλ. [4]) έχουν δείξει ότι όταν εφαρμόζεται ο αλγόριθμος του Proportional Fair στον scheduler του eNodeB, ο ρυθμός δεδομένων για τους χρήστες που βρίσκονται μακριά από την κεραία του σταθμού βάσης είναι χαμηλότερος από τον αντίστοιχο ρυθμό για την περίπτωση που εφαρμόζεται ο αλγόριθμος του Round Robin. Από την άλλη, η συνολική απόδοση της κυψέλης είναι υψηλότερη με τον Proportional Fair. Ωστόσο, η απόδοση του Round Robin θα βελτιωθεί αν χρησιμοποιηθούν κεραίες που δεν εκπέμπουν με τον ίδιο τρόπο παντού, αλλά εκπέμπουν με ποικίλους τρόπους στα διάφορα σημεία της κυψέλης, ανάλογα με τις απώλειες και τις ανάγκες του δικτύου.

2.5 Priority Set Scheduler

Ο PSS είναι ένας scheduler που ασχολείται με την ποιότητα της υπηρεσίας που προσφέρεται στον χρήστη (QoS-aware), λειτουργεί και στην περίοδο του χρόνου και της συχνότητας και μπορεί να εγγυηθεί τον απαιτούμενο ρυθμό δεδομένων [8]. Αυτός ο scheduler αποτελεί μια γενική λύση για QoS-aware ροές που απαιτούν εγγυημένο ρυθμό δεδομένων [6]. Για να εξασφαλίσει την δικαιοσύνη ανάμεσα στα UEs ο PSS

χρησιμοποιεί μια παράμετρο που λέγεται TBR, επιθυμητός ρυθμός δεδομένων bit, με την οποία ξεχωρίζει τις χαμηλές και τις υψηλές ροές προτεραιότητας. [2]

Στο επίπεδο του χρόνου ο PSS, δηλαδή ο TDPS(Time Domain Packet Scheduler), επινοήθηκε ώστε να επιλέγει τους χρήστες με την μεγαλύτερη προτεραιότητα. Συγκεκριμένα οι χρήστες με ροές μικρότερες από το TBR (δηλαδή αυτοί οι χρήστες που χρειάζονται επειγόντως να τους διατεθούν πόροι ώστε να ικανοποιήσουν τις απαιτήσεις του QoS τους) αποτελούν το υψηλής προτεραιότητας σύνολο. Οι υπόλοιποι χρήστες αποτελούν το σύνολο χαμηλής προτεραιότητας. Επομένως από τον TDPS δημιουργούνται δύο σύνολα χρηστών βασισμένα στο TBR.

- Το ένα είναι το σύνολο υψηλής προτεραιότητας που περιέχει UEs των οποίων η παλιά μέση διεκπεραιωτική ικανότητα είναι μικρότερη από το TBR. Ο TDPS σε αυτή την περίπτωση υπολογίζει τις μετρικές προτεραιότητας χρησιμοποιώντας τον αλγόριθμο BET.
- Το άλλο είναι το σύνολο χαμηλής προτεραιότητας που περιέχει UEs των οποίων η παλιά μέση διεκπεραιωτική ικανότητα είναι μεγαλύτερη (ή ίση) του TBR. Ο TDPS σε αυτή την περίπτωση υπολογίζει τις μετρικές προτεραιότητας χρησιμοποιώντας τον αλγόριθμο PF.

Από τη στιγμή που επιλεγεί από τον TDPS το UE με την μεγαλύτερη μετρική, ο FDPS διανέμει το RBG στο UE με την μεγαλύτερη μετρική χρησιμοποιώντας μια από τις ακόλουθες μετρικές που υπάρχουν στο LTE πακέτο του ns-3. [2]

- Proportional Fair Scheduled (PFsch)

$$\widehat{M}_{sch_k} = \max_{1 \leq j \leq N} \left(\frac{R_j(k, t)}{T_{sch_j}(t)} \right) \quad (17)$$

- Carrier over interference to average (Colta)

$$M_{col_k}(t) = \max_{1 \leq j \leq N} \left(\frac{Col(j, k)}{\sum_{k=0}^{N_{RBG}} Col(j, k)} \right) \quad (18)$$

Όπου $T_{sch_j}(t)$ είναι όμοιο με την προηγούμενη μέση τιμή της διεκπεραιωτικής ικανότητας του UE j, με τη διαφορά ότι ενημερώνεται η τιμή της μεταβλητής αυτής μόνο όταν το j-οστό UE εξυπηρετείται. Η μεταβλητή $Col(j, k)$ είναι μια εκτίμηση του λόγου σήματος προς θόρυβο (SINR) του RBG k του UE j. Και οι δύο μετρικές PFsch και Colta χρησιμεύουν ώστε να διαχωρίζεται ο scheduler στο πεδίο του χρόνου και στο πεδίο της συχνότητας αντίστοιχα.

Μια άλλη προσέγγιση είναι τα σύνολα να δημιουργηθούν εξαρτώμενα από το QCI του κάθε data bearer και να ταξινομηθούν ως GBR και non-GBR σύνολα. Έπειτα από αυτό το βήμα ο FDPS αναθέτει τους καλύτερους πόρους σε κάθε χρήστη του GBR συνόλου, ενημερώνοντας τη τιμή του bitrate που επιτεύχθει. Όταν όλοι οι χρήστες έχουν φτάσει τη τιμή του bitrate που επιθυμούν, αν υπάρχουν RB που είναι ακόμη διαθέσιμα, ο scheduler τα αναθέτει στους χρήστες του non-GBR συνόλου χρησιμοποιώντας τις μετρικές του PF.

Οι προτεραιότητες μπορούν να υπολογισθούν και με τον ακόλουθο τύπο για τον i-στο χρήστη όπως αναφέρεται και στο [16].

$$P_i = \frac{D_{HOL,i}}{\tau_i} \quad (19)$$

Αξίζει να σημειωθεί ότι αυξάνοντας την τιμή του P_i , η μετάδοση της κορυφής (του επικεφαλής) της γραμμής των πακέτων γίνεται πιο επείγουσα. Η κατανομή των πόρων εκτελείται έπειτα κατανέμοντας όλους τους πόρους που χρειάζονται ώστε να επιτύχει ο χρήστης με τη μεγαλύτερη προτεραιότητα τον απαιτούμενο ρυθμό δεδομένων bit. Αν αποδεδειχθούν κάποιοι πόροι μετά την κατανομή τους, οι ίδιες λειτουργίες επαναλαμβάνονται για τον χρήστη με τη δεύτερη μεγαλύτερη προτεραιότητα και έτσι συνεχίζει ο αλγόριθμος για τους υπόλοιπους χρήστες. Στο τέλος της διαδικασίας κάθε ελεύθερος πόρος κατανέμεται πάλι στους χρήστες που ακολουθούν με σειρά προτεραιότητας και αυτό συνεχίζεται όσο οι χρήστες έχουν δεδομένα να μεταδώσουν.

Από μια γενική άποψη, όλες αυτές οι προσεγγίσεις χρησιμοποιούν ταξινομημένες λίστες για να θέσουν ως προτεραιότητες τις πιο καθυστερημένες ροές και για να επιτευχθεί ο απαιτούμενος ρυθμός δεδομένων bit. Παρ'όλα αυτά, η πρώτη προσέγγιση στοχεύει στην υψηλή φασματική απόδοση δεδομένου ότι η διαδικασία λήψης αποφάσεων στο FDPS δεν ασχολείται με την ποιότητα της παρεχόμενης υπηρεσίας στον χρήστη (QoS-unaware). Οι άλλες δυο περιπτώσεις φαίνονται να είναι πιο ισχυρές υπό την άποψη των QoS προβλέψεων, λόγω της ισχυρής ιεράρχησης που εφαρμόζεται επίσης στην κατανομή των πόρων στο πεδίο της συχνότητας.

2.6 Maximum Throughput

Ο MT scheduler στοχεύει στη μεγιστοποίηση της ολικής απόδοσης ενός eNB. Διανέμει κάθε RBG στο UE που μπορεί να πετύχει το μέγιστο προσδοκώμενο ρυθμό δεδομένων στο τρέχον TTI. Δοσμένων N UEs, οι μετρικές προτεραιότητας, όπως παρουσιάζονται στο [8], που χρησιμοποιούνται στο πεδίο της συχνότητας (FD-MT) και στο πεδίο του χρόνου (TD-MT) είναι αντίστοιχα:

$$\hat{i}_k(t) = \operatorname{argmax}_{j=1,\dots,N} R_j(k, t) \quad (20)$$

$$\hat{i}(t) = \operatorname{argmax}_{j=1,\dots,N} R_j(t) \quad (21)$$

όπου $j=1,\dots,N$ είναι το εκάστοτε UE, $\hat{i}_k(t)$ είναι το επιλεγμένο UE από το FD-MT για μετάδοση στο RBG k τη χρονική στιγμή t , $\hat{i}(t)$ είναι το επιλεγμένο UE από το TD-MT για μετάδοση κατά το χρονικό διάστημα t πάνω σε όλα τα RBGs. Το $R_j(k, t)$ και το $R_j(t)$ είναι οι μετρικές προτεραιότητας που χρησιμοποιούνται για τον FD-MT και τον TD-MT αντίστοιχα. Συγκεκριμένα η μεταβλητή $R_j(k, t)$ είναι ο επιτεύξιμος ρυθμός δεδομένων για το RBG k και καθορίζεται με βάση την υποζώνη CQI του συγκεκριμένου RBG, ενώ $R_j(t)$ είναι ο επιτεύξιμος ρυθμός δεδομένων που καθορίζεται με βάση το φάσμα ευρείας ζώνης CQI. Χρησιμοποιώντας τις από πάνω μετρικές προτεραιότητας, ο MT πάντα διανέμει τους ράδιο-πόρους στα UE με την καλύτερη ποιότητα καναλιού. Συνεπώς, η αποτελεσματικότητα χρησιμοποίησης του δικτύου μπορεί να μεγιστοποιηθεί δεδομένου ότι οι πόροι του δικτύου πάντα αξιοποιούνται πλήρως. Ωστόσο, αυτή η «καιροσκοπική» συμπεριφορά δεν μπορεί να εξασφαλίσει δικαιοσύνη μεταξύ των UEs, η οποία αποτελεί έναν άλλο βασικό παράγοντα απόδοσης.

Ο MT είναι προφανώς ικανός να μεγιστοποιήσει την απόδοση της κυψέλης, αλλά από την άλλη πλευρά εκτελεί άδικο διαμερισμό πόρων μιας και οι χρήστες με κακές συνθήκες καναλιού (όπως οι χρήστες που βρίσκονται στις άκρες της κυψέλης) θα λάβουν ένα μικρό ποσοστό των διαθέσιμων πόρων ή σε ακραίες περιπτώσεις ίσως υποφέρουν από «λιμοκτονία πόρων».

Ένας πρακτικός scheduler πρέπει να είναι ενδιάμεσα μεταξύ του MT, που μεγιστοποιεί την απόδοση της κυψέλης, και του BET, που εγγυάται δίκαιη κατανομή απόδοσης μεταξύ των χρηστών, προκειμένου να ικανοποιήσει όσο το δυνατόν περισσότερο τις

γρήγορες αλλαγές των συνθηκών του καναλιού ενώ εξακολουθεί να ικανοποιεί ένα βαθμό δικαιοσύνης. [6]

2.7 Throughput to Average

Ο TTA αποτελεί έναν FD scheduler, δηλαδή ορίζεται μόνο στο πεδίο της συχνότητας. Σε αντίθεση με τον MT scheduler ο TTA αποτελεί μια εξισορρόπηση μεταξύ της αποδοτικότητας και της δικαιοσύνης μεταξύ των UEs σε μια κυψέλη, αποτελεί δηλαδή μια μέση προσέγγιση μεταξύ του MT και του PF scheduler [8]. Ο χρήστης $i_k(t)$ στον οποίον ανατίθεται το RBG k στο subframe t υπολογίζεται από την παρακάτω μετρική προτεραιότητας όπως αναφέρεται και στο [17] :

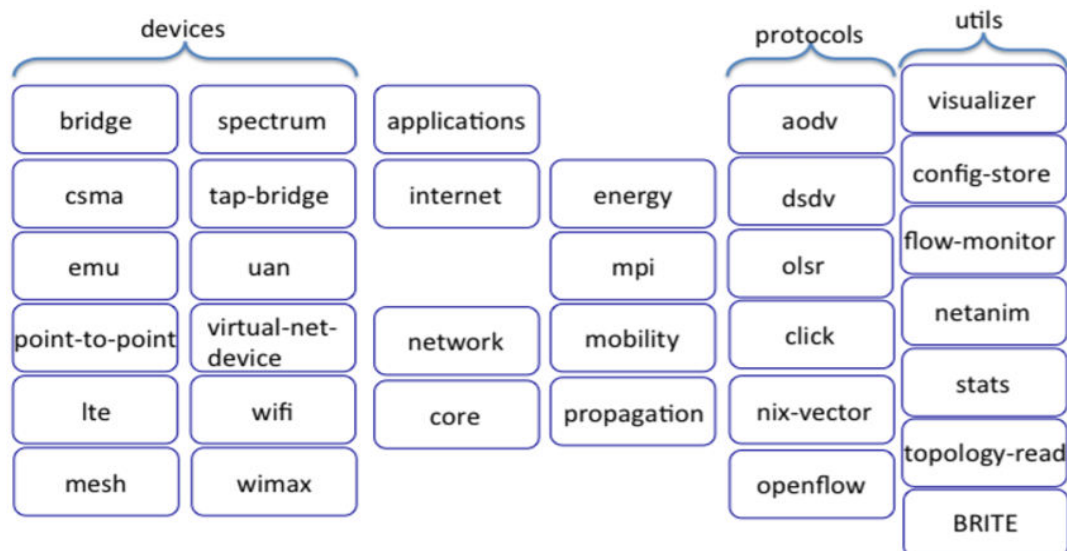
$$i_k(t) = \underset{j=1, \dots, N}{\operatorname{argmax}} \left(\frac{R_j(k,t)}{R_j(t)} \right) \quad (22)$$

Όπου $R_j(t)$ είναι ο επιτεύξιμος ρυθμός δεδομένων για τον χρήστη j στο subframe t . Η διαφορά μεταξύ των επιτεύξιμων ρυθμών $R_j(k,t)$ και $R_j(t)$ είναι στην επιλογή της τιμής του MCS. Για το $R_j(k,t)$ το MCS υπολογίζεται από την υποζώνη CQI ενώ για το $R_j(t)$ υπολογίζεται από την ευρεία ζώνη του CQI. Για την ακρίβεια, η «μέση τιμή» της επιτεύξιμης απόδοσης στο τρέχον TTI χρησιμοποιείται σαν ένας παράγοντας κανονικοποίησης της επιτεύξιμης απόδοσης για ένα συγκεκριμένο RBG. Άρα, ο TTA scheduler εγγυάται ότι τα καλύτερα RBs ανατίθενται σε κάθε χρήστη. Σαν συνέπεια ο TTA πρέπει να εξασφαλίσει ένα ισχυρό επίπεδο δικαιοσύνης σε ένα χρονικό παράθυρο ενός μοναδικού TTI. Πράγματι, όσο υψηλότερη είναι η ολική προσδοκώμενη απόδοση ενός χρήστη τόσο χαμηλότερη θα είναι η μετρική του σε ένα μοναδικό RB. Συνεπώς αυτός ο scheduler εκμεταλλεύεται την ευαισθησία του καναλιού για την διασφάλιση ενός ελάχιστου επιπέδου εξυπηρέτησης κάθε χρήστη.

3. ΠΡΟΣΟΜΟΙΩΤΗΣ NS-3

Ο ns-3 είναι ένας προσομοιωτής, ο οποίος έχει υλοποιηθεί σε C++, ενώ περιέχει επίσης συνδεδεμένα αρχεία για Python. Τα προγράμματα της προσομοίωσης είναι είτε εκτελέσιμα C++ αρχεία είτε προγράμματα σε Python και η βιβλιοθήκη απαρτίζεται από περίπου 300.000 γραμμές C++ κώδικα, όπως υποδεικνύεται στο [18]. Ο ns-3 είναι κομμάτι του LENA project και ο κώδικάς του είναι ανοιχτός, σύμφωνα με το GNU GPLv2 (GNU General Public License version 2). Υποστηρίζεται κυρίως από συστήματα Linux, OS X και FreeBSD και δεν είναι προς τα πίσω συμβατή με την βιβλιοθήκη του ns-2.

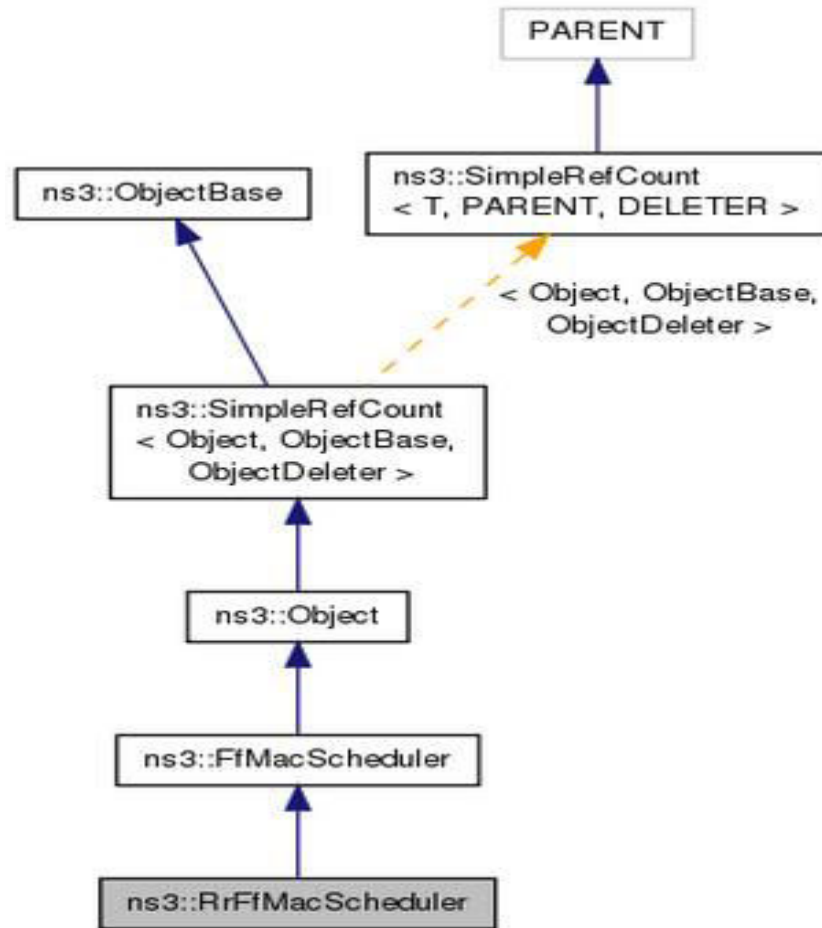
Στην Εικόνα 8 παρουσιάζονται τα βασικά δομικά στοιχεία που έχουν υλοποιηθεί στον ns-3. Στην παρούσα εργασία θα ασχοληθούμε κυρίως με το στοιχείο (module) του LTE, που φαίνεται στην πρώτη στήλη της Εικόνας 8. Το module αυτό παρέχει μια βασική υλοποίηση των LTE συσκευών, συμπεριλαμβανομένου των μοντέλων διάδοσης καθώς και των επιπέδων PHY και MAC. Τα στοιχεία του που έχουν υλοποιηθεί μέχρι στιγμής, μερικά από τα οποία παρουσιάζονται στο [19], κάνουν δυνατή την προσομοίωση αρκετών πλευρών των συστημάτων LTE, όπως είναι για παράδειγμα ο χρονοπρογραμματισμός στο downlink για τα επίπεδα RRM (Radio Resource Management) και MAC. Επίσης, παρέχουν ένα πολύ καλό υπόβαθρο για περεταίρω επέκταση, καθώς και για την ανάπτυξη ενός ολοκληρωμένου εργαλείου στο μέλλον. Συγκεκριμένα, τα πιο σημαντικά χαρακτηριστικά που παρέχονται από το υλοποιημένο LTE στοιχείο είναι μια βασική υλοποίηση τόσο του τερματικού (UE), όσο και του σταθμού βάσης (eNB), οι οντότητες του Radio Resource Control (RRC) επιπέδου για τα UE και eNB, ένα σύγχρονο AMC σύστημα για το downlink, η διαχείριση των bearer μαζί με τις QoS παραμέτρους τους, οι ουρές του MAC και τα στιγμιότυπα του RLC, η διαχείριση του CQI, υποστήριξη για τον χρονοπρογραμματισμό πακέτων για την ανερχόμενη και την κατερχόμενη ζεύξη, ένα μοντέλο για το επίπεδο του PHY και ένα μοντέλο για το κανάλι με το μοντέλο απωλειών του εξωτερικού E-UTRAN. Τα χαρακτηριστικά αυτά αναλύονται στο [19].



Εικόνα 8: Δομικά στοιχεία που έχουν υλοποιηθεί στον ns-3 [17]

3.1 Round Robin

Ο αλγόριθμος του Round Robin υλοποιείται από την κλάση *RrFfMacScheduler* του *ns3*, η οποία κληρονομεί τα χαρακτηριστικά και τις μεθόδους της από την αφηρημένη κλάση *FfMacScheduler*. Το διάγραμμα των κλάσεων από τις οποίες κληρονομεί η *RrFfMacScheduler* φαίνεται στην Εικόνα 9. [20]



Εικόνα 9: Διάγραμμα κλάσεων από τις οποίες κληρονομεί η *ns3::RrFfMacScheduler* [20]

Ο scheduler μοιράζει ίσο πλήθος χρονοθυρίδων σε όλους τους χρήστες και με κυκλική σειρά, χωρίς να δώσει σημασία στις προτεραιότητες των εφαρμογών. Οι διαθέσιμοι πόροι μοιράζονται μόνο ανάμεσα σε ενεργείς ροές, όπως είναι για παράδειγμα τα λογικά κανάλια των οποίων η ουρά του RLC επιπέδου δεν είναι άδεια (βλ. [21]). Αν το πλήθος των RBGs είναι μεγαλύτερο από το πλήθος των ενεργών ροών, τότε όλες οι ροές μπορούν να εξυπηρετηθούν στο ίδιο χρονικό υποπλαίσιο (subframe). Αλλιώς, αν το πλήθος των RBGs είναι μικρότερο από το πλήθος των ενεργών ροών, δεν μπορούν να εξυπηρετηθούν όλες οι ροές σε ένα subframe. Συνεπώς, στο επόμενο subframe η ανάθεση θα ξεκινήσει από την τελευταία ροή που δεν εξυπηρετήθηκε. Το MCS που θα χρησιμοποιηθεί για τον κάθε χρήστη επιλέγεται σύμφωνα με τα λαμβανόμενα CQIs ευρείας ζώνης.

Σχετικά με τον HARQ, ο RR υλοποιεί τον μη προσαρμοστικό τρόπο, ο οποίος υποδηλώνει ότι στην εξυπηρέτηση των αναμεταδόσεων ο RR χρησιμοποιεί τις ίδιες παραμέτρους που χρησιμοποίησε για το αρχικό πακέτο. Αυτό σημαίνει ότι διατηρεί τα ίδια RBGs και MCS. Τα UEs, στα οποία ανατίθενται πόροι για αναμετάδοση τύπου

HARQ, δεν λαμβάνονται υπόψη για την μετάδοση νέων δεδομένων, ακόμα κι αν έχουν την ευκαιρία να μεταδώσουν στο ίδιο TTI. Τέλος, το HARQ μπορεί να απενεργοποιηθεί, μέσα από το σύστημα παραμέτρων του *ns3*, για να είναι συμβατό με παλιότερες δοκιμές και με παλιότερο κώδικα. Αυτό μπορεί να γίνει με την εντολή:

`Config::SetDefault("ns3::RrFfMacScheduler::HarqEnabled", BooleanValue(false));`

Τα ορίσματα, που μπορούμε να μεταβάλλουμε μέσω της συνάρτησης *Config::SetDefault*, για την κλάση *RrFfMacScheduler* είναι τα παρακάτω [20]:

- **CqiTimerThreshold:** Ορίζει το πλήθος των TTIs για το οποίο ένα CQI μπορεί να είναι έγκυρο. Η προεπιλεγμένη τιμή αυτής της μεταβλητής είναι 1000, δηλαδή 1 δευτερόλεπτο (second) και ο τύπος της μεταβλητής είναι η κλάση *ns3::UintegerValue*.
- **HarqEnabled:** Ορίζει αν θα είναι ενεργοποιημένος ή όχι ο μηχανισμός του HARQ. Η κλάση από την οποία ορίζεται είναι η *BooleanValue* και η προεπιλεγμένη τιμή της είναι *true*, δηλαδή ο μηχανισμός HARQ είναι ενεργοποιημένος, εφόσον ο χρήστης δεν μεταβάλει την μεταβλητή *HarqEnabled*.
- **UIGrantMcs:** Ορίζει το MCS που χρησιμοποιείται για το uplink. Πρέπει να είναι ακέραιος μεταξύ 0 και 15, ενώ η προεπιλεγμένη τιμή είναι τι 0. Στο *ns3* είναι αντικείμενο της κλάσης *ns3::UintegerValue*.
- **UICqiFilter:** Το χαρακτηριστικό αυτό ορίζεται από την κλάση *FfMacScheduler* και ορίζει το φίλτρο που θα εφαρμοστεί στα CQI του uplink που λαμβάνονται. Είναι αντικείμενο της κλάσης *ns3::EnumValue* και η αρχική τιμή της είναι η *ALL_UL_CQI*. Οι τιμές που μπορεί να πάρει είναι οι ακόλουθες [21]:
 - **SRS_UL_CQI:** με το οποίο μόνο το CQI που βασίζεται στο Sounding Reference Signal (SRS) αποθηκεύεται στις παραμέτρους του scheduler.
 - **PUSCH_UL_CQI:** με το οποίο μόνο το CQI που βασίζεται στο κανάλι Physical Uplink Shared Chanel (PUSCH) αποθηκεύεται στις παραμέτρους του scheduler.
 - **ALL_UL_CQI:** με το οποίο όλα τα CQI αποθηκεύονται στην ίδια παράμετρο.

3.2 Blind Equal Throughput

Ο αλγόριθμος του Blind Equal Throughput στον *ns3* υλοποιείται από την κλάση *TdBetFfMacScheduler* [22] για το πεδίο του χρόνου και τον *FdBetFfMacScheduler* [23] για το πεδίο της συχνότητας. Η ιεραρχία των κλάσεων από τις οποίες κληρονομούν αυτές οι δύο κλάσεις είναι όμοια με αυτήν της *RrFfMacScheduler* (βλ. Εικόνα 9), καθώς όλες προκύπτουν από την κλάση *FfMacScheduler*.

Ο scheduler του BET αποσκοπεί στο να παρέχει ίση απόδοση σε όλους τους χρηστές που είναι συνδεδεμένοι με ένα eNB. Η μετρική που χρησιμοποιεί ο BET δίνεται από την Εξίσωση (1). Στον TD BET, ο scheduler επιλέγει το UE με την μεγαλύτερη μετρική απόδοσης και αναθέτει όλα τα RBGs σε αυτό το UE. Στον FD BET, από την άλλη, σε κάθε TTI ο scheduler επιλέγει το UE με την μεγαλύτερη προτεραιότητα, δηλαδή αυτό με την μικρότερη μέση απόδοση. Μετά ο scheduler αναθέτει ένα RBG σε αυτό το UE, υπολογίζει την αναμενόμενη απόδοση για αυτό το UE και την χρησιμοποιεί για να την συγκρίνει με την απόδοση, $T_j(t)$, που έχει υπολογιστεί ήδη για τα υπόλοιπα UEs. Ο scheduler συνεχίζει να αναθέτει RBG σε αυτό το UE, μέχρι που η αναμενόμενη απόδοσή του να μην είναι πια η μικρότερη μεταξύ των περασμένων αποδόσεων, $T_j(t)$, των υπόλοιπων χρηστών. Έστερα ο scheduler θα ακολουθήσει την ίδια λογική για να αναθέσει RBG στο επόμενο UE που θα έχει την μικρότερη μέση απόδοση, μέχρι να

ανατεθούν όλα τα RBGs στα UEs. Η αρχή του αλγορίθμου είναι ότι ο scheduler κάνει ότι μπορεί για να επιτύχει ίση απόδοση μεταξύ όλων των UEs. [21]

Οι κλάσεις *TdBetFfMacScheduler* και *FdBeFftMacScheduler* δέχονται τα ίδια ορίσματα. Αυτά είναι τα **CqiTimerThreshold**, **HarqEnabled**, **UIGrantMcs** και **UICqiFilter**, τα οποία ορίζονται και αρχικοποιούνται όπως και στον Round Robin.

3.3 Token Bank Fair Queue

Στον *ns3* ο αλγόριθμος του Token Bank Fair Queue υλοποιείται από τις κλάσεις *TdTbfqFfMacScheduler* [24] για τον TD TBFQ και *FdTbfqFfMacScheduler* [25] για τον FD TBFQ. Ο αλγόριθμος TBFQ λαμβάνει υπόψη του τις απαιτήσεις σε ποιότητα υπηρεσίας και ακολουθεί τη λογική του μηχανισμού του τρύπιου δοχείου. Στον αλγόριθμο αυτόν η ροή της κίνησης για έναν χρήστη i χαρακτηρίζεται από τον ρυθμό άφιξης δεδομένων, t_i (byte/sec), τον ρυθμό παραγωγής κουπονιών, r_i (byte/sec), το μέγεθος της δεξαμενής κουπονιών, p_i (byte), και έναν μετρητή που κρατάει τον αριθμό των κουπονιών που δανείζονται από ή δίνονται στην τράπεζα κουπονιών, E_i . Ο μετρητής E_i μπορεί να είναι μικρότερος του μηδενός. Κάθε K Bytes δεδομένων καταναλώνουν k κουπόνια. Επίσης, ο TBFQ διατηρεί μια διαμοιραζόμενη τράπεζα κουπονιών, B , ώστε να εξισορροπεί την κίνηση μεταξύ διαφορετικών ροών. Αν ο ρυθμός παραγωγής κουπονιών, r_i , είναι μεγαλύτερος από τον ρυθμό άφιξης δεδομένων, t_i , τότε τα κουπόνια που υπερχειλίζουν από την δεξαμενή προστίθενται στην τράπεζα κουπονιών και ο E_i αυξάνεται τόσο όσα είναι τα κουπόνια που μπαίνουν στην τράπεζα. Αλλιώς, η ροή i χρειάζεται να εξάγει κουπόνια από την τράπεζα, βασιζόμενη στην μετρική προτεραιότητας $frac{E_i}{r_i}$, και ο E_i αυτή τη φορά μειώνεται. Όπως είναι προφανές, ο χρήστης που συνεισφέρει περισσότερο στην τράπεζα κουπονιών έχει μεγαλύτερη προτεραιότητα για να δανειστεί κουπόνια. Αντιθέτως, ο χρήστης που δανείζεται περισσότερα κουπόνια από την τράπεζα έχει μικρότερη προτεραιότητα να συνεχίσει να εξάγει κουπόνια. Έτσι, σε περίπτωση που πολλοί χρήστες έχουν τον ίδιο ρυθμό παραγωγής κουπονιών, τον ίδιο ρυθμό δεδομένων και το ίδιο μέγεθος για την δεξαμενή κουπονιών, ο χρήστης που υπόκειται στην υψηλότερη παρεμβολή θα έχει περισσότερες ευκαιρίες να δανειστεί κουπόνια από την τράπεζα. Επίσης, ο TBFQ μπορεί να αστυνομεύει την κίνηση θέτοντας τον ρυθμό παραγωγής κουπονιών έτσι ώστε να περιορίσει την απόδοση. Επιπλέον, για κάθε ροή ο TBFQ αποθηκεύει τις ακόλουθες παραμέτρους: (βλ. [21])

- Το όριο οφειλών (dept limit), d_i : Αν το E_i γίνει μικρότερο από αυτό το κατώφλι, ο χρήστης i δεν μπορεί να δανειστεί περαιτέρω κουπόνια από την τράπεζα. Αυτό υπάρχει για να εμποδίζει πιθανώς κακόβουλους χρήστες να δανείζονται υπερβολικά κουπόνια.
- Το όριο πίστωσης (credit limit), c_i : Κάθε φορά που το UE i δανείζεται από την τράπεζα, μπορεί να δανειστεί το πολύ c_i κουπόνια.
- Το κατώφλι πίστωσης (credit threshold), C : Αν το E_i φτάσει το dept limit, το UE i πρέπει να αποθηκεύσει C κουπόνια στην τράπεζα, προκειμένου να μπορέσει να δανειστεί επιπλέον κουπόνια από την τράπεζα.

Στο FD-TBFQ, ο scheduler επιλέγει πάντα το UE με την υψηλότερη προτεραιότητα και αναθέτει τα RBGs με το υψηλότερο subband CQI μέχρι που να μην υπάρχουν άλλα πακέτα στον RLC buffer του UE ή μέχρι να έχουν ανατεθεί όλα τα RBGs. Στον TD-TBFQ, αφού επιλεγεί το UE με την μέγιστη προτεραιότητα, αναθέτει όλα τα RBGs σε αυτό το UE χρησιμοποιώντας το wideband CQI.

Οι παράμετροι που δέχονται οι κλάσεις του TBFQ στο *ns3* είναι οι ίδιοι και για τις δύο κλάσεις, *TdTbfqFfMacScheduler* και *FdTbfqFfMacScheduler*, όπως φαίνεται από τα [24]

και [25]. Συγκεκριμένα, αυτά είναι τα **CqiTimerThreshold**, **HarqEnabled**, **UIGrantMcs** και **UICqiFilter**, τα οποία ορίζονται και αρχικοποιούνται όπως και στον Round Robin. Σε αυτά προστίθενται και τα ακόλουθα ορίσματα:

- **DeptLimit**: Προσδιορίζει το dept limit της ροής. Είναι του τύπου *ns3::IntegerValue* και η προεπιλεγμένη τιμή της είναι το - 625.000 (bytes).
- **CreditLimit**: Προσδιορίζει το credit limit της ροής. Είναι του τύπου *ns3::UIntegerValue* και η προεπιλεγμένη τιμή της είναι 625.000 (bytes).
- **TokenPoolSize**: Προσδιορίζει τη μέγιστη τιμή της δεξαμενής κουπονιών της ροής. Είναι του τύπου *ns3::UIntegerValue* και η προεπιλεγμένη τιμή της είναι 1 (byte).
- **CreditableThreshold**: Προσδιορίζει το credit threshold της ροής. Είναι του τύπου *ns3::UIntegerValue* και η προεπιλεγμένη τιμή της είναι 0 (bytes).

3.4 Proportional Fair

Η κλάση PffMacScheduler υλοποιεί τον Proportional Fair scheduler. Η κλάση αυτή κληρονομεί τα χαρακτηριστικά και τις μεθόδους της από την αφηρημένη κλάση FfMacScheduler. Η σχέση της κλάσης PffMacScheduler με την FfMacScheduler είναι όμοια με την σχέση των RrFfMacScheduler και FfMacScheduler, που παρουσιάζεται στην Εικόνα 9. Το πλήρες σχεδιάγραμμα υπάρχει στο [26].

Ο PF όπως αναφέραμε και στο Κεφάλαιο 2 αναθέτει πόρους στον χρήστη του οποίου η στιγμιαία ποιότητα καναλιού είναι η μεγαλύτερη συγκριτικά με τη μέση τιμή των συνθηκών του καναλιού του καθ' όλη τη διάρκεια.

Όσον αφορά το HARQ, ο PF υλοποιεί μια μη προσαρμοστική έκδοση, πράγμα που συνεπάγεται ότι, κατά την κατανομή (allocation) στην αναμετάδοση, ο scheduler επιχειρεί να χρησιμοποιήσει την ίδια διαμόρφωση κατανομής πόρων του αρχικού μπλοκ, το οποίο σημαίνει ότι διατηρούνται τα ίδια RBGs και το ίδιο MCS. Τα UEs που διατίθενται για HARQ αναμεταδόσεις δεν χρησιμοποιούνται για την μετάδοση νέων δεδομένων, σε περίπτωση που υπάρχει δυνατότητα μετάδοσης τους στο ίδιο TTI. Τέλος, το HARQ μπορεί να απενεργοποιηθεί με την ακόλουθη εντολή:

```
Config::SetDefault("ns3::PffMacScheduler::HarqEnabled", BooleanValue(false);
```

Για την κλάση *PffMacScheduler* μπορούμε να μεταβάλλουμε τα ίδια ορίσματα όπως και για την κλάση *RrFfMacScheduler* μέσω της συνάρτησης *Config::SetDefault*.

3.5 Priority Set Scheduler

Η κλάση PssFfMacScheduler υλοποιεί τον Priority Set scheduler. Η κλάση αυτή κληρονομεί τα χαρακτηριστικά και τις μεθόδους της από την αφηρημένη κλάση FfMacScheduler, όπως φαίνεται και στο [27].

Ο PSS είναι ένας QoS aware scheduler που συνδυάζει τις λειτουργίες χρονοπρογραμματισμού πακέτων του TD και του FD σε έναν scheduler. Ελέγχει την δικαιοσύνη μεταξύ των UEs χρησιμοποιώντας το TBR όπως αναφέραμε και στο Κεφάλαιο 2. Με το TBR χωρίζει τα Ues σε δύο σύνολα στο πεδίο του χρόνου. Επιλέγει τα Ues με τις μεγαλύτερες μετρικές από τα σύνολα αυτά και τα προωθεί στον FD scheduler. Στον PSS ο FD scheduler αναθέτει το RBG k στο UE n που μεγιστοποιεί την επιλεγμένη μετρική. Και η μετρική του PF scheduler και η μετρική του Colta διαχωρίζουν τον FD scheduler από τον TD στον PSS. Επιπλέον ο PSS FD scheduler παρέχει μια μετρική βάρους $W[n]$ για έλεγχο δικαιοσύνης σε περίπτωση μικρού αριθμού των UEs. [21]

$$W[n] = \max\left(1, \frac{TBR}{T_j(t)}\right) \quad (23)$$

Όπου $T_j(t)$ είναι η προηγούμενη απόδοση του χρήστη j . Επομένως, στο RBG k , ο FD scheduler επιλέγει το UE j που μεγιστοποιεί τη μετρική του πεδίου συχνότητας (M_{sch} , M_{Col}) από το $W[n]$. Αυτή η στρατηγική θα διασφαλίσει ότι η απόδοση του UE με την χαμηλότερη ποιότητα θα τείνει προς το TBR.

Τέλος, το HARQ μπορεί να απενεργοποιηθεί με την ακόλουθη εντολή:

`Config::SetDefault("ns3::PssFfMacScheduler::HarqEnabled", BooleanValue(false));`

Και για την κλάση *PssFfMacScheduler* μπορούμε να μεταβάλλουμε τα ίδια ορίσματα που αναφέρουμε για την κλάση *RrFfMacScheduler* μέσω της συνάρτησης *Config::SetDefault*.

3.6 Maximum Throughput

Ο αλγόριθμος του Maximum Throughput στον *ns3* υλοποιείται από την κλάση *TdMtFfMacScheduler* [28] για το πεδίο του χρόνου και τον *FdMtFfMacScheduler* [29] για το πεδίο της συχνότητας. Η ιεραρχία των κλάσεων από τις οποίες κληρονομούν αυτές οι δύο κλάσεις είναι όμοια με αυτήν της *RrFfMacScheduler* (βλ. Εικόνα 9), καθώς όλες προκύπτουν από την κλάση *FfMacScheduler*.

Όπως αναφέραμε και στο Κεφάλαιο 2, ο MT scheduler στοχεύει στην μεγιστοποίηση της ολικής απόδοσης του eNB. Αναθέτει κάθε RB στο χρήστη που μπορεί να επιτύχει τον μέγιστο επιτεύξιμο ρυθμό στο τρέχον TTI. Στον FDMT, σε κάθε TTI, ο MAC scheduler αναθέτει RBGs στο UE που έχει τον υψηλότερο επιτεύξιμο ρυθμό υπολογισμένο από την υποζώνη CQI. Στον TDMT, σε κάθε TTI, ο MAC scheduler επιλέγει το UE που έχει τον υψηλότερο επιτεύξιμο ρυθμό υπολογισμένο από το ευρείας ζώνης CQI και του αναθέτει όλα τα RBGs. Ο επιτεύξιμος ρυθμός δεδομένων υπολογίζεται όπως και στον PF. Στην περίπτωση που πολλά UEs έχουν τον ίδιο επιτεύξιμο ρυθμό, επιλέγεται το πρώτο UE που δημιουργήθηκε. Παρόλο που ο MT μπορεί να μεγιστοποιήσει την απόδοση της κυψέλης δεν μπορεί να παρέχει δικαιοσύνη στα UEs με κακές συνθήκες καναλιού.[21]

3.7 Throughput to Average

Η κλάση *TtaFfMacScheduler* [30] υλοποιεί τον Throughput To Average scheduler. Η κλάση αυτή κληρονομεί τα χαρακτηριστικά και τις μεθόδους της από την αφηρημένη κλάση *FfMacScheduler*. Η ιεραρχία των κλάσεων από τις οποίες κληρονομεί αυτή η κλάση είναι όμοια με αυτήν της *RrFfMacScheduler* (βλ. Εικόνα 9), καθώς όλες προκύπτουν από την κλάση *FfMacScheduler*.

Ο TTA είναι μια μέση προσέγγιση μεταξύ του MT και του PF. Ο TTA μπορεί να υλοποιηθεί μόνο στο πεδίο της συχνότητας επειδή ο επιτεύξιμος ρυθμός ενός συγκεκριμένου RBG σχετίζεται μόνο με το FD scheduling. [21]

4. ΠΕΙΡΑΜΑΤΑ

Στο κεφάλαιο αυτό παρουσιάζονται τα πειράματα που εκτελέσαμε κατά την διάρκεια της εργασίας και τα αποτελέσματά τους, σε μορφή διαγραμμάτων. Για την ανάλυση των αποτελεσμάτων χρησιμοποιούνται τα PDCP αρχεία του ns-3 και ένα script που εξαγεί συμπεράσματα από αυτά. Τα PDCP αρχεία περιέχουν την έξοδο του προγράμματος, που σχετίζεται με το PDCP επίπεδο, . Για κάθε πείραμα, δημιουργούνται δύο αρχεία, ένα για το uplink και ένα για το downlink. Για την εξαγωγή στατιστικών πληροφοριών από αυτά τα αρχεία χρησιμοποιείται το script statistics.py. Αυτό δέχεται σαν όρισμα το PDCP αρχείο για το οποίο θέλουμε να πάρουμε τα στατιστικά χαρακτηριστικά και, για κάθε μέτρηση του PDCP αρχείου, υπολογίζει και εκτυπώνει την απόδοση σε KBps, την μέση καθυστέρηση του δικτύου, τον ρυθμό επιτυχίας και τον ρυθμό απώλειας δεδομένων. Στην συνέχεια, θα αναλύσουμε τα δύο σενάρια προσομοίωσης που έχουν υλοποιηθεί. Για το κάθε ένα θα περιγράψουμε την τοπολογία του μαζί με κάποιες χρήσιμες κλάσεις του ns-3, έναν συνοπτικό πίνακα με τις παραμέτρους που αλλάξαμε στις εκτελέσεις των πειραμάτων και τα αποτελέσματά του.

4.1 1^ο Σενάριο Προσομοίωσης

4.1.1 Περιγραφή

Η πρώτη προσομοίωση που υλοποιήθηκε αφορά την μελέτη ενός εμπορικού κέντρου (Shopping Mall), με 4 κεραίες και πολλούς κινούμενους χρήστες. Δόθηκε, επίσης, ιδιαίτερη έμφαση στους διάφορους αλγόριθμους ανάθεσης πόρων και στην αντίστοιχη απόδοση του δικτύου και των χρηστών, για κάθε περίπτωση. Για την προσομοίωση φτιάξαμε, σε C++ και με τη χρήση της βιβλιοθήκης ns-3, το πρόγραμμα *building_sim.cc*, μέσω του οποίου αλλάζουμε διάφορες παραμέτρους. Οι παράμετρες αυτές έχουν να κάνουν με τον αριθμό των χρηστών (*numberOfUes*), τον αριθμό των resource blocks (*rb*) και τον τύπο του scheduler (*schedulerType*) που επιθυμούμε. Οι προεπιλεγμένες τιμές, που ανατίθενται σε αυτές τις μεταβλητές σε περίπτωση που δεν ορίζονται από τον χρήστη, είναι *numberOfUes*=10, *rb*=6, *schedulerType*="rr". Επίσης, η προσομοίωσή μας τρέχει για 100sec, ενώ τα πειράματά μας τα εκτελέσαμε για 10, 15, 20 και 25 χρήστες και για 6 και 15 resource blocks. Οι schedulers που χρησιμοποιήθηκαν είναι μόνο οι Proportional Fair, Round Robin, Frequency Domain Token Bank Fair Queue και Time Domain Token Bank Fair Queue. Οι παραπάνω τύποι scheduler αναγνωρίζονται από το πρόγραμμά μας, μέσω της μεταβλητής *schedulerType* ως "pf", "rr", "fdtbfq", "tdtbfq", αντίστοιχα. Στη συνέχεια αυτής της ενότητας περιγράφεται το πείραμα και κάποιες κλάσεις του ns-3, που θα μας χρησιμεύουν.

4.1.1.1 Η τοπολογία της προσομοίωσης

Στην προσομοίωση έχουμε ένα εμπορικό κέντρο με διαστάσεις $50 \times 50 \text{ m}^2$ και ύψος 10 m. Το κέντρο είναι στην ουσία ένα αστικό κτίριο, που περιβάλλεται από εξωτερικά παράθυρα. Αποτελείται από έναν όροφο που έχει 4 δωμάτια ίσου μεγέθους. Δηλαδή, το κτίριο μοιράζεται σε 4 ίσα τετραγωνικά τμήματα, διαστάσεων 25×25 , και το κάθε ένα από αυτά αποτελεί ένα δωμάτιο. Για την δημιουργία του κτιρίου έχει υλοποιηθεί η συνάρτηση *createBuilding()*, η οποία παρουσιάζεται στην Εικόνα 10. Το *build* που δέχεται σαν όρισμα η συνάρτηση, είναι ένας δείκτης σε αντικείμενο της κλάσης *Building*. Έτσι, εφόσον το *build* είναι δείκτης, μεταβάλλεται από την συνάρτηση και μπορεί να χρησιμοποιηθεί από το κυρίως πρόγραμμα.

```

void createBuilding(Ptr<Building> build)
{
    double x_min = 0.0;
    double x_max = 50.0;
    double y_min = 0.0;
    double y_max = 50.0;
    double z_min = 0.0;
    double z_max = 10.0;

    //build = CreateObject<Building>();
    build->SetBoundaries(Box(x_min, x_max, y_min, y_max, z_min, z_max));
    build->SetBuildingType(Building::Residential);
    build->SetExtWallsType(Building::ConcreteWithWindows);
    build->SetNFloors(1);
    build->SetNRoomsX(2);
    build->SetNRoomsY(2);
}

```

Εικόνα 10: Συνάρτηση createBuilding()

Όπως φαίνεται και στην Εικόνα 10, η δημιουργία του τρισδιάστατου κτιρίου πραγματοποιείται μέσω της κλάσης *ns3::Building*. Οι παράμετροι της κλάσης που τροποποιούνται από την συνάρτηση είναι ακόλουθες:

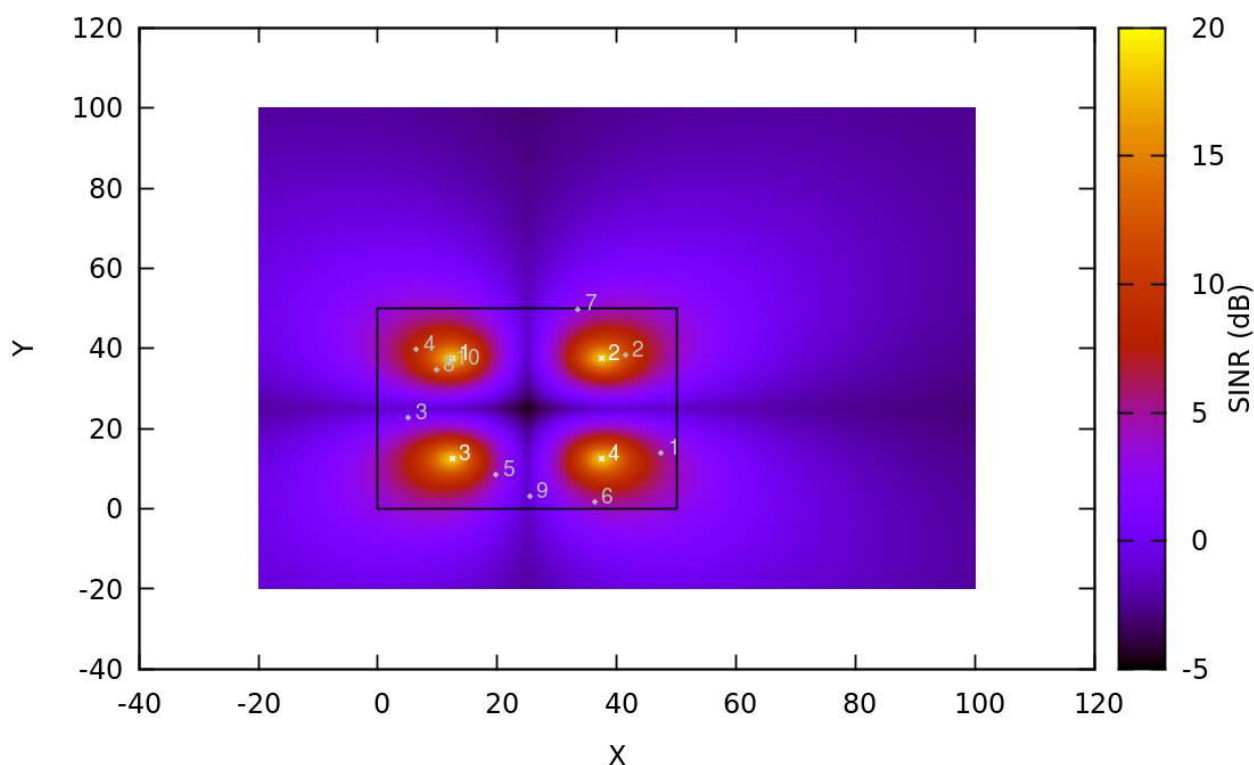
- **NRoomsX**: Αντιστοιχεί στο πλήθος των δωματίων στον άξονα X. Είναι μεταβλητή του τύπου *ns3::UIntegerValue* και η τιμή που τις αποδίδεται είναι 2 δωμάτια.
- **NRoomsY**: Αντιστοιχεί στο πλήθος των δωματίων στον άξονα Y. Είναι, επίσης, μεταβλητή του τύπου *ns3::UIntegerValue* και ορίζεται σε 2 δωμάτια.
- **NFloors**: Το πλήθος των ορόφων του κτιρίου. Ορίζεται από την κλάση *ns3::UIntegerValue* με τιμή 1 όροφος.
- **Boundaries**: Τα όρια του κτιρίου, τα οποία ορίζονται από την κλάση *BoxValue*. Η τιμή τους είναι η (0,50,0,50,0,10), πράγμα που ορίζει ένα κτίριο διαστάσεων 50×50×10. Συγκεκριμένα, τα (x_min, x_max) ορίζουν τα σημεία όπου αρχίζει και τελειώνει το κτίριο στον άξονα x. Με τον ίδιο τρόπο χρησιμοποιούνται τα (y_min, y_max) και (z_min, z_max) για τους άξονες y και z αντίστοιχα. Έτσι, προσδιορίζονται η ακριβής θέση και οι διαστάσεις του κτιρίου στο τρισδιάστατο σύστημα συντεταγμένων.
- **Type**: Ο τύπος του κτιρίου. Ορίζεται μέσω της κλάσης *ns3::EnumValue* και η προεπιλεγμένη τιμή του είναι το “Residential”. Κάποιες άλλες τιμές που μπορεί να πάρει η παράμετρος αυτή είναι οι “Office” και “Commercial”.
- **ExternalWallsType**: Ο τύπος του υλικού από το οποίο είναι φτιαγμένοι οι εξωτερικοί τοίχοι. Ορίζεται μέσω της κλάσης *ns3::EnumValue* και η αρχική της τιμή είναι “ConcreteWithWindows”, που σημαίνει ότι οι τοίχοι είναι από μπετόν και καλύπτονται από παράθυρα. Άλλες δυνατές τιμές είναι οι “Wood” για ξύλινους τοίχους, “ConcreteWithoutWindows” για τοίχους από μπετόν, χωρίς παράθυρα, και “StoneBlocks” για τοίχους από ογκόλιθους.

Για περισσότερες λεπτομέρειες σχετικά με την κλάση *ns3::Building* δείτε τις αναφορές [31] και [32].

Το κάθε δωμάτιο του κτιρίου λειτουργεί ως μία ξεχωριστή φεμτοκυψέλη (femtocell), με έναν σταθμό βάσης στο κέντρο της. Οι κεραίες των σταθμών βάσεων (eNBs) εκπέμπουν ισοτροπικά. Επιπλέον, για τα eNBs προσδιορίζονται το εύρος ζώνης, το οποίο θα είναι το ίδιο για το uplink και το downlink και μετράται σε Resource Blocks, καθώς και ο αλγόριθμος χρονοπρογραμματισμού που θα υλοποιεί ο scheduler. Αυτές οι

παράμετροι δίνονται κατά την κλήση του προγράμματος, ώστε να μπορέσουμε να εκτελέσουμε πειράματα με διαφορετικούς συνδυασμούς αυτών των παραμέτρων. Ακόμη, η ισχύς εκπομπής της κάθε κεραίας είναι 20 dBm, καθώς, σύμφωνα με το [33] τόσο είναι η ισχύς εκπομπής της κεραίας μιας κυψέλης femto. Όσον αφορά τους χρήστες, αυτοί τοποθετούνται αρχικά σε τυχαίες θέσεις μέσα στο κτίριο και στη συνέχεια βαδίζουν με τυχαίο τρόπο μέσα σ' αυτό. Η διεύθυνση της κίνησης του κάθε χρήστη αλλάζει κάθε 2 sec, ενώ η ταχύτητά του είναι σταθερή στο 1 m/s. Επίσης, το κάθε UE συνδέεται αυτόματα με το eNB από το οποίο λαμβάνει την μεγαλύτερη ισχύ.

Ένα ενδεικτικό διάγραμμα του κτιρίου, με τις 4 φεμτοκυψέλες και 10 χρήστες, φαίνεται στην Εικόνα 11. Σε αυτήν την εικόνα παρουσιάζονται και τα επίπεδα του SINR (σε dB) για κάθε σημείο του κτιρίου, καθώς και έξω από αυτό, έτσι όπως προκύπτουν από την ύπαρξη και τα χαρακτηριστικά των τεσσάρων κεραιών.



Εικόνα 11: Τοπολογία 1^ης προσομοίωσης

Το διάγραμμα που παρουσιάζεται στην Εικόνα 11 έχει προκύψει από την χρήση του Radio Environment Map. Με την χρήση του *RadioEnvironmentMapHelper* είναι δυνατό να αποθηκευτεί σε ένα αρχείο ένας χάρτης για το περιβάλλον του ραδιοφάσματος (Radio Environment Map – REM). Αυτός μπορεί να είναι, για παράδειγμα, ένα διδιάστατο πλέγμα από τιμές που αντιπροσωπεύουν τον λόγο σήμα προς θόρυβο για το downlink, δίνοντας, σε κάθε σημείο, έμφαση στο eNB που έχει το ισχυρότερο σήμα. Οι παράμετροι που μπορούν να οριστούν από την κλάση *RadioEnvironmentMapHelper* και έχουν μεταβληθεί για της ανάγκες της προσομοίωσής μας είναι οι ακόλουθες (βλ. και [34]):

- **ChannelPath:** Το μονοπάτι προς το κανάλι για το οποίο θα δημιουργηθεί το REM. Είναι αντικείμενο της κλάσης *StringValue* και η αρχική της τιμή είναι η *"/ChannelList/0"*.
- **OutputFile:** Το όνομα του αρχείου στο οποίο θα αποθηκευτεί το REM. Ορίζεται από την κλάση *StringValue* και η αρχική του τιμή είναι το *"rem.out"*.
- **XMin:** Η ελάχιστη x συντεταγμένη του χάρτη. Ορίζεται μέσω της κλάσης *ns3::DoubleValue* και η τιμή που της αποδίδεται είναι το -20.

- **YMin:** Η ελάχιστη y συντεταγμένη του χάρτη. Ορίζεται μέσω της κλάσης *ns3::DoubleValue* και η τιμή της είναι το -20.
- **XMax:** Η μέγιστη x συντεταγμένη του χάρτη. Ορίζεται από την κλάση *ns3::DoubleValue* και η τιμή της είναι το 100.
- **YMax:** Η μέγιστη y συντεταγμένη του χάρτη. Ορίζεται από την κλάση *ns3::DoubleValue* και η τιμή της είναι το 100.
- **XRes:** Η ανάλυση κατά τον άξονα του x , δηλαδή το πλήθος των σημείων για οποία ορίζονται τιμές στον x άξονα. Ορίζεται μέσω της κλάσης *ns3::UIntegerValue* και η επιλεγμένη τιμή της είναι το 800.
- **Yres:** Η ανάλυση κατά τον άξονα του y . Ορίζεται, επίσης, από την κλάση *ns3::UIntegerValue* και η επιλεγμένη τιμή του είναι το 600.
- **Z:** Η τιμή του επιπέδου z για το οποίο θα δημιουργηθεί ο χάρτης. Ορίζεται από την κλάση *ns3::DoubleValue* και η τιμή τίθεται στο 1.

Δηλαδή, οι διαστάσεις του REM είναι από -20 έως 100 για τους άξονες x και y , ενώ η ανάλυση που χρησιμοποιείται είναι 800 για τον άξονα x και 600 για τον y . Επίσης, το ύψος στο οποίο υπολογίζεται το REM, δηλαδή το Z , είναι 1. Ακόμη, το REM παράγεται για το PDSCCH (Physical Downlink Shared Channel) κανάλι.

Το REM αποθηκεύεται σε ένα αρχείο ASCII και περιέχει 4 στήλες. Η 1η, η 2η και η 3η στήλη αντιστοιχούν στις συντεταγμένες x , y και z , αντίστοιχα, ενώ η 4η αντιστοιχεί στις τιμές του SINR σε γραμμικές μονάδες. Για την σχεδίαση του REM χρησιμοποιείται το script *my_plot_script*, το οποίο παρουσιάζεται στο [35]. Το SINR, που υπολογίζεται μέσω αυτού του script, αναπαρίσταται σε dB και όχι σε γραμμικές τιμές. Επίσης, κατά την παραγωγή του REM θα χρησιμοποιηθούν οι συναρτήσεις *PrintGnuplottableBuildingListToFile()*, *PrintGnuplottableUeListToFile()* και *PrintGnuplottableEnbListToFile()*, οι οποίες είναι ίδιες με τις αντίστοιχες συναρτήσεις που έχουν υλοποιηθεί για την προσομοίωση του *lena-dual-stripe* (βλ. [36]). Έτσι, παράγονται αρχεία συμβατά με το gnuplot, που περιέχουν πληροφορίες για τις θέσεις των UEs και eNBs καθώς και για τα κτίρια που υπάρχουν. Τα αρχεία αυτά είναι τα *ues.txt*, *enbs.txt* και *buildings.txt*. Συνεπώς, για την σχηματική αναπαράσταση όλων των κόμβων και των κτιρίων της προσομοίωσης, εκτελείται η ακόλουθη εντολή:

```
gnuplot -p enbs.txt ues.txt buildings.txt my_plot_script
```

Τέλος, το δίκτυο έχει υλοποιηθεί σε τοπολογία EPC, η χρήση του οποίου επιτρέπει την χρήση της δικτύωσης IPv4 μαζί με LTE συσκευές. Δηλαδή, δίνεται η δυνατότητα χρησιμοποίησης συνηθισμένων εφαρμογών του ns-3 και υποδοχών (sockets) του IPv4 μαζί με ένα LTE δίκτυο, καθώς και η σύνδεση ενός δικτύου LTE μαζί με ένα οποιοδήποτε άλλο IPv4 δίκτυο που μπορεί να υπάρχει στην προσομοίωση. Για την προσομοίωση ενός EPC δικτύου χρησιμοποιείται η κλάση *LteHelper*, μαζί με την κλάση *EpcHelper*, οι οποίες αναλαμβάνουν την δημιουργία των EPC οντοτήτων και της τοπολογίας του δικτύου. Για περισσότερες λεπτομέρειες σχετικά με το EPC στον ns-3 μπορείτε να ανατρέξετε στο [35].

Με την χρήση του EPC, το κάθε eNB συνδέεται με ένα S-GW/P-GW, ενώ το P-GW με την σειρά του συνδέεται, μέσω internet, με έναν απομακρυσμένο κόμβο (remote host). Δηλαδή, το P-GW και το remote host είναι δικτυακές συσκευές με IPv4 διευθύνσεις. Οι συσκευές αυτές στέλνουν δεδομένα με ρυθμό 100 Gb/s και με MTU (MAC-level Maximum Transmission Unit – Μέγιστη Μονάδα Μετάδοσης σε επίπεδο MAC) 1500 bytes. Το κανάλι με το οποίο συνδέονται τα P-GW και remote host έχει καθυστέρηση μετάδοσης ίση με 0,01 sec. Επίσης, και τα UEs λειτουργούν σαν κόμβοι του δικτύου internet και έχουν διευθύνσεις IP.

4.1.1.2 Οι εφαρμογές των χρηστών

Για την δημιουργία των εφαρμογών (applications) χρησιμοποιείται η κλάση *UdpClientHelper* (βλ. [37]). Για να είναι η προσομοίωση πιο ρεαλιστική, έχουν οριστεί 10 διαφορετικά applications, τα οποία ανατίθενται στους χρήστες με κυκλική σειρά. Συγκεκριμένα, κάθε application χαρακτηρίζεται από έναν μοναδικό αριθμό από το 0 μέχρι το 9. Έπειτα, λαμβάνονται οι χρήστες, με την σειρά που εμφανίζονται στην προσομοίωση, και σε κάθε έναν ανατίθεται από ένα application. Δηλαδή, το πρώτο UE θα τρέχει το application 0, το δεύτερο το 1, κι ούτω καθεξής. Αν έχει ανατεθεί και το application 9 σε κάποιο UE και υπάρχουν και άλλα UEs για τα οποία δεν έχει οριστεί κάποια εφαρμογή, τότε η ανάθεση τους συνεχίζεται κανονικά ξεκινώντας πάλι από το application 0.

Για όλες τις εφαρμογές έχει οριστεί ότι το μέγεθος των μεταδιδόμενων πακέτων θα είναι ίσο με 1024 bytes. Οι εφαρμογές της προσομοίωσης είναι οι ακόλουθες:

- Application 0: Πρόκειται για το κατέβασμα (download) ενός αρχείου από τον απομακρυσμένο κόμβο (remote host). Ο ρυθμός μετάδοσης δεδομένων είναι 11722 Kbps (Kbit per second), δηλαδή περίπου 1465 KBps (Kbyte per second), και το μέγεθος του αρχείου είναι 3MBytes. Οι τιμές αυτές έχουν προκύψει από το [38]. Η εφαρμογή αρχίζει την στιγμή 0,01 sec και τερματίζει όταν έχουν μεταδοθεί όλα τα πακέτα του αρχείου.
- Application 1: Πρόκειται για μετάδοση video προς τον UE. Ο ρυθμός μετάδοσης δεδομένων είναι 657 Kbps, όπως προκύπτει από το [39], και έχει επιλεγεί να στέλνονται συνολικά το πολύ 2000 πακέτα. Η εφαρμογή αρχίζει την στιγμή 0,01 sec και τερματίζει όταν έχουν μεταδοθεί όλα τα πακέτα του αρχείου.
- Application 2: Πρόκειται για το ανέβασμα (upload) ενός αρχείου από τον χρήστη. Ο ρυθμός μετάδοσης δεδομένων είναι 1788 Kbps και το μέγεθος του αρχείου είναι 1 MByte. Οι τιμές αυτές έχουν προκύψει από το [38]. Η εφαρμογή αρχίζει την στιγμή 0,01 sec και τερματίζει όταν έχουν μεταδοθεί όλα τα δεδομένα του αρχείου.
- Application 3: Πρόκειται για κατέβασμα αρχείου, μεγέθους 10 MBytes. Ο ρυθμός μετάδοσης δεδομένων είναι 13463 Kbps, όπως προκύπτει από το [38]. Η εφαρμογή αρχίζει την στιγμή 0,01 sec και τερματίζει όταν έχουν μεταδοθεί όλα τα πακέτα του αρχείου.
- Application 4: Πρόκειται για το upload ενός αρχείου, μεγέθους 10 MBytes. Ο ρυθμός μετάδοσης δεδομένων είναι 1920 Kbps. Η τιμή αυτή έχει προκύψει από το [38]. Η εφαρμογή αρχίζει την στιγμή 0,01 sec και τερματίζει όταν έχουν μεταδοθεί όλα τα δεδομένα του αρχείου.
- Application 5: Πρόκειται για την μεταφορά ενός αρχείου, με μέγεθος ίσο με 10 MBytes. Ο ρυθμός μετάδοσης δεδομένων είναι 240 Kbps, όπως προκύπτει από το [40]. Η εφαρμογή αρχίζει την στιγμή 0,01 sec και τερματίζει όταν έχουν μεταδοθεί όλα τα πακέτα του αρχείου.
- Application 6: Πρόκειται για κατέβασμα μουσικής σε smartphone. Αν το μέγεθος του αρχείου ήχου είναι περίπου 0,23 MB, τότε ο ρυθμός μετάδοσης δεδομένων θα είναι περίπου 60 MB/hour, δηλαδή 16,66 KBps, όπως προκύπτει από το [40]. Η εφαρμογή αρχίζει την στιγμή 0,01 sec και τερματίζει όταν έχουν μεταδοθεί όλα τα πακέτα.
- Application 7: Πρόκειται για συνεχής ροή (streaming) video σε smartphone, το οποίο μπορεί να συνδεθεί και να αξιοποιήσει το 4G δίκτυο. Ο ρυθμός μετάδοσης δεδομένων είναι 350 MB/hour, δηλαδή 97,22 KBps, όπως προκύπτει από το [40]. Η εφαρμογή αρχίζει την στιγμή 0,01 sec και τερματίζει στα 10 sec.

- Application 8: Πρόκειται για βιντεοκλήση με tablet. Ο ρυθμός μετάδοσης δεδομένων είναι 41,67 KBps, όπως προκύπτει από το [40]. Η εφαρμογή αρχίζει την στιγμή 0,01 sec και τερματίζει με το πέρας της προσομοίωσης.
- Application 9: Πρόκειται για εφαρμογή VoIP με την χρήση tablet. Ο ρυθμός μετάδοσης δεδομένων είναι 8,33 KBps. Η τιμή αυτή έχει προκύψει από το [40]. Η εφαρμογή αρχίζει την στιγμή 0,01 sec και τερματίζει όταν τερματίσει και η προσομοίωση.

Πέρα από τις εφαρμογές, για κάθε UE ορίζεται και από ένα λογικό κανάλι, bearer, πάνω στο οποίο πραγματοποιείται η κίνηση των δεδομένων. Οι λειτουργίες των EPS bearers υλοποιούνται από την κλάση *ns3::EpsBearer* (βλ. [41]). Όπως έχουμε αναφέρει, τα bearers σχετίζονται με τις απαιτούμενες ποιότητες υπηρεσιών. Γι' αυτό και το κάθε ένα χαρακτηρίζεται και από έναν δείκτη (indicator) QoS. Στον ns-3 οι δείκτες αυτοί είναι οι GBR_CONV_VOICE, GBR_CONV_VIDEO, GBR_GAMING, GBR_NON_CONV_VIDEO, NGBR_IMS, NGBR_VIDEO_TCP_OPERATOR, NGBR_VOICE_VIDEO_GAMING, NGBR_VIDEO_TCP_PREMIUM και NGBR_VIDEO_TCP_DEFAULT. Προκειμένου να ικανοποιείται κάθε εφαρμογή, έχουμε επιλέξει τα bearers της προσομοίωσής μας να ανήκουν όλα στην κατηγορία *NGBR_VIDEO_TCP_DEFAULT*.

4.1.2 Παράμετροι Προσομοίωσης

Στον Πίνακα 1 παρουσιάζονται συνοπτικά οι τιμές των διάφορων παραμέτρων της προσομοίωσης.

Πίνακας 1: Τιμές των παραμέτρων της 1^{ης} προσομοίωσης

Πλήθος κτιρίων	1
Διαστάσεις κτιρίου (μήκος×πλάτος×ύψος)	50 × 50 × 10
Τύπος κτιρίου	Αστικό
Πλήθος ορόφων στο κτίριο	1
Πλήθος δωματίων στο κτίριο	4
Εύρος ζώνης	6, 15 RBs
Αλγόριθμοι για τον scheduler	Round Robin Proportional Fair Time Domain Token Bank Fair Queue Frequency Domain Token Bank Fair Queue
Πλήθος από eNBs	4
Κεραίες σταθμού βάσης	Ισοτροπικές
Θέσεις των κεραιών	Στα κέντρα των δωματίων

Ισχύς εκπομπής κεραιών	20 dBm
Πλήθος χρηστών	10, 15, 20 και 25
Τύπος κινητικότητας χρηστών	Περπατάνε με τυχαία κατεύθυνση μέσα στο κτίριο
Ταχύτητα κίνησης χρηστών	1 m/s
Όρια κίνησης χρηστών	Εντός του κτιρίου
Διάρκεια προσομοίωσης	100 sec

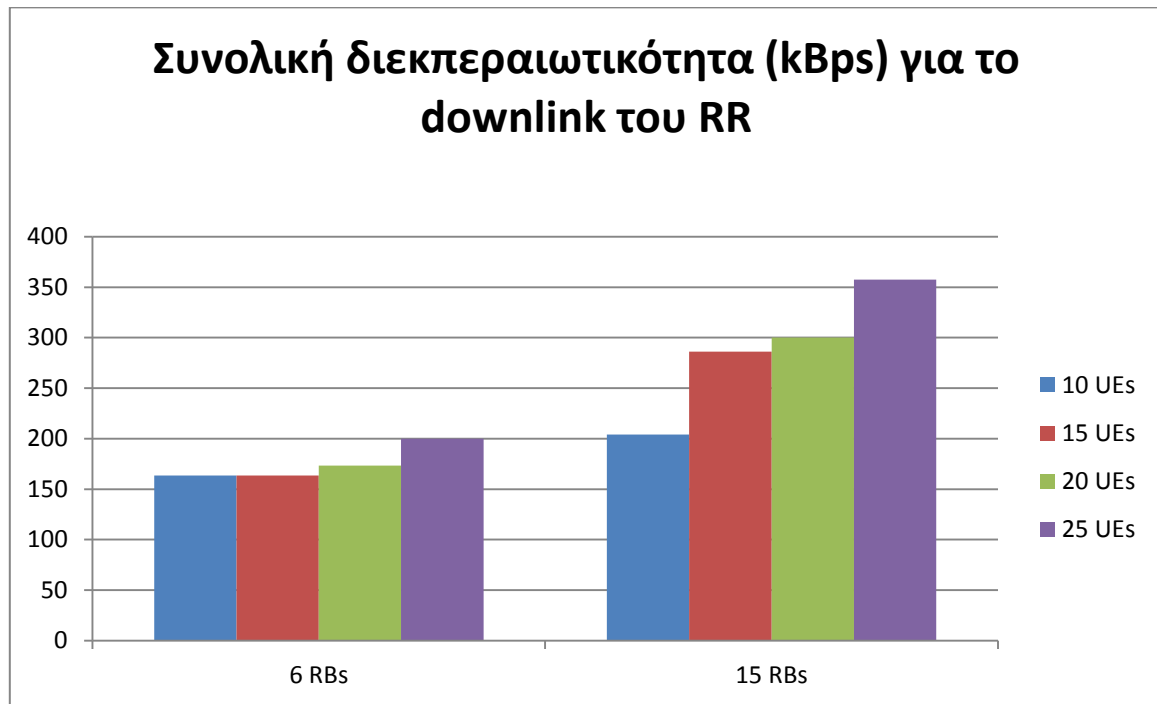
4.1.3 Αποτελέσματα

Τα πειράματά μας διαχωρίζονται σε 8 διαφορετικά σύνολα. Συγκεκριμένα, έχουμε δύο σύνολα που αφορούν το πλήθος των διαθέσιμων πόρων. Στο πρώτο τα eNBs έχουν εύρος 6 Resource Blocks και στο δεύτερο 15. Το κάθε ένα από αυτά τα δύο σύνολα χωρίζεται σε τέσσερα ακόμη, που σχετίζονται με το πλήθος των χρηστών, οι οποίοι είναι 10, 15, 20 και 25. Έτσι, κάθε διαφορετικός συνδυασμός από RBs και UEs αποτελεί ένα σύνολο. Έτσι σε ένα διάγραμμα θα εμφανίζονται και τα 8 αυτά σύνολα.

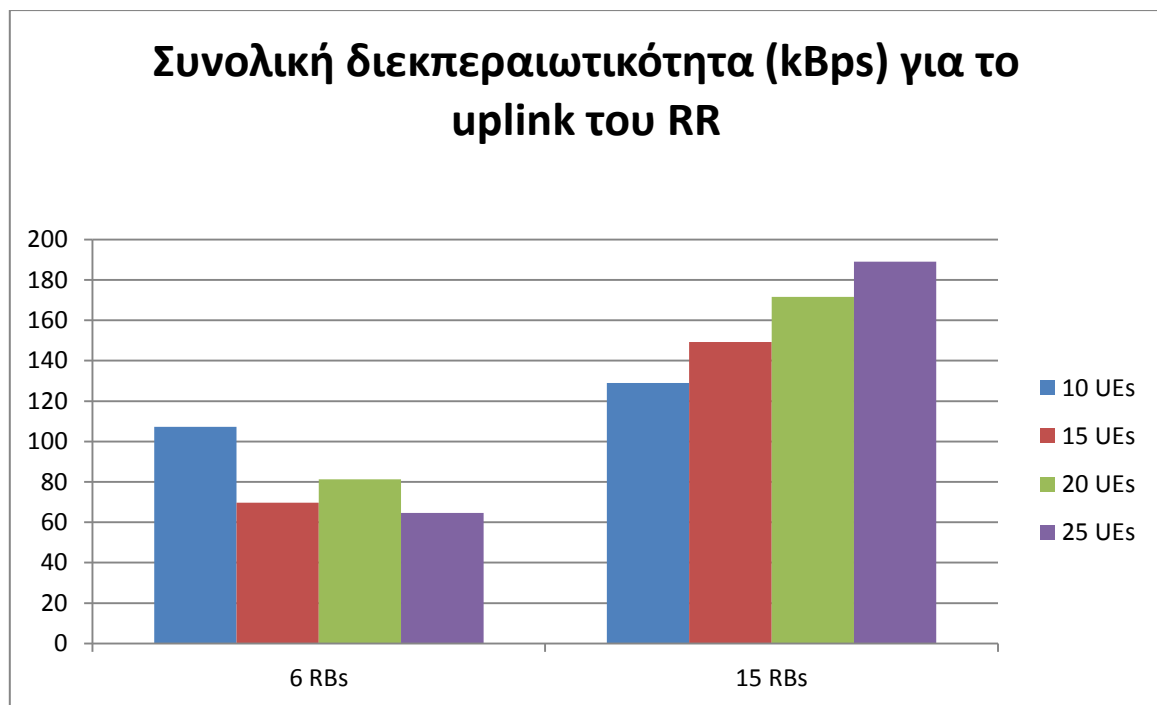
Στις επόμενες ενότητες παρουσιάζονται τα διαγράμματα για τους RR, PF, TD-TBFQ και FD-TBFQ schedulers. Για τον RR, τα διαγράμματα αυτά παρουσιάζονται τόσο για την κατερχόμενη όσο και για την ανερχόμενη ζεύξη. Ωστόσο, καθώς ο ns-3 έχει ορισμένο στο uplink μόνο τον RR scheduler, για τους υπόλοιπους αλγόριθμους θα δείξουμε τα αποτελέσματα μόνο της κατερχόμενης ζεύξης. Για κάθε scheduler, και για κάθε ζεύξη, υπολογίζονται 3 μετρικές και έτσι δημιουργούνται τρία διαγράμματα: ένα για την ολική απόδοση, ένα για την μέση καθυστέρηση και ένα για τις συνολικές απώλειες δεδομένων. Δηλαδή, στον οριζόντιο άξονα βρίσκονται τα 8 σύνολα που προαναφέρθηκαν, ενώ στον κατακόρυφο βρίσκεται μια από τις 3 μετρικές.

4.1.3.1 Round Robin

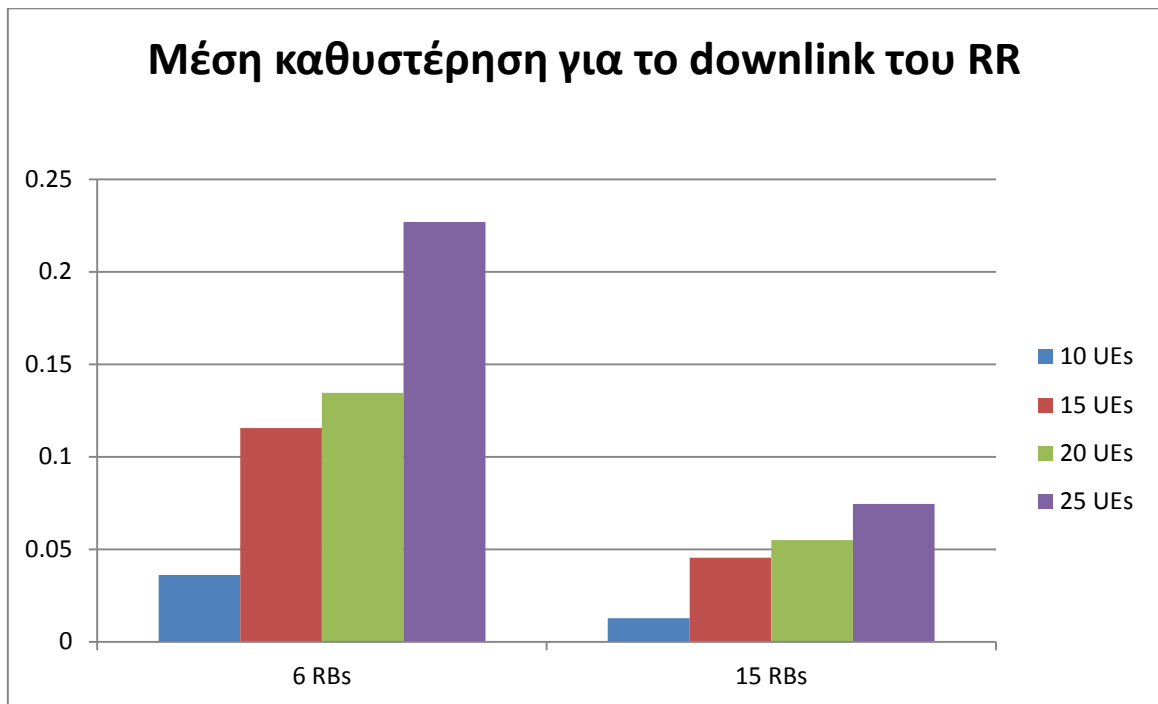
Στις Εικόνες 12 – 17 παρουσιάζονται τα αποτελέσματα για τον αλγόριθμο του Round Robin.



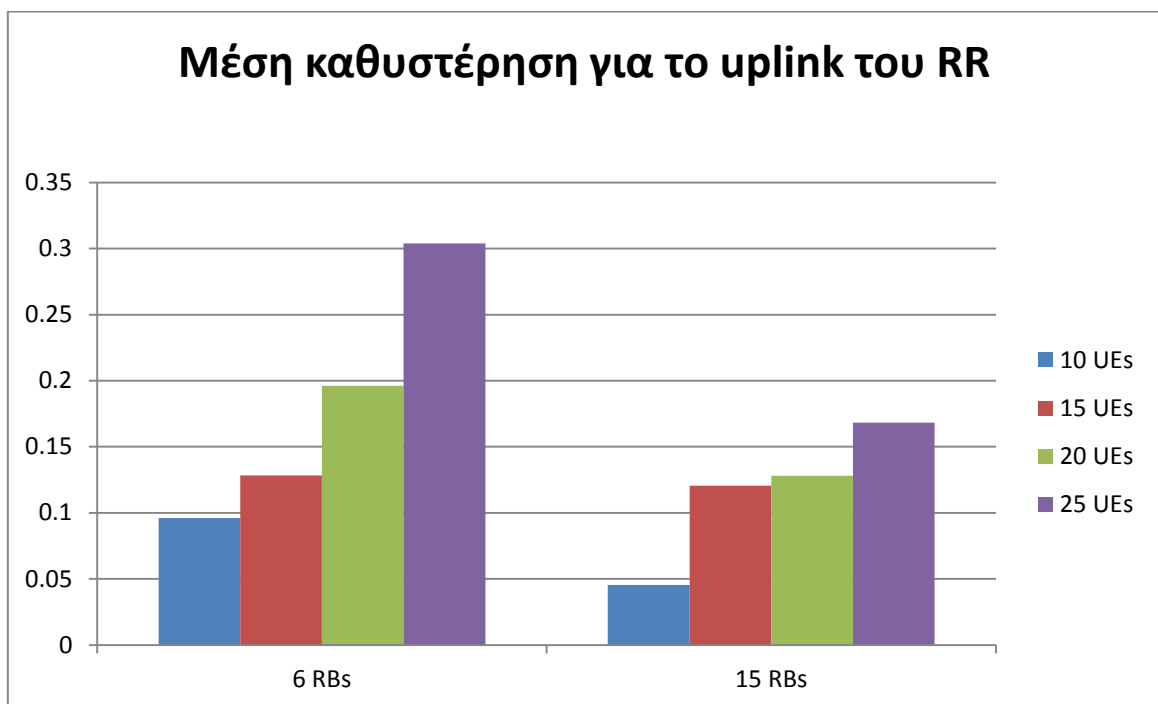
Εικόνα 12: Συνολική διεκπεραιωτικότητα για κάθε σύνολο της κατερχόμενης ζεύξης για τον RR scheduler – Τοπολογία 1



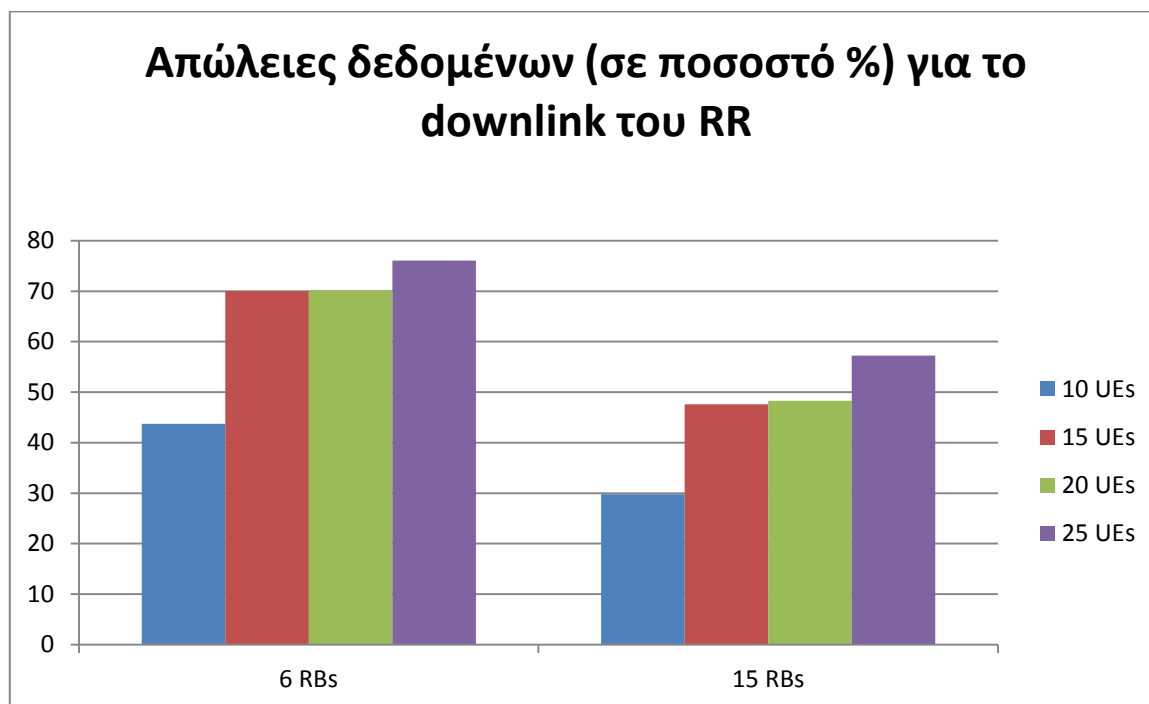
Εικόνα 13: Συνολική διεκπεραιωτικότητα για κάθε σύνολο της ανερχόμενης ζεύξης για τον RR scheduler – Τοπολογία 1



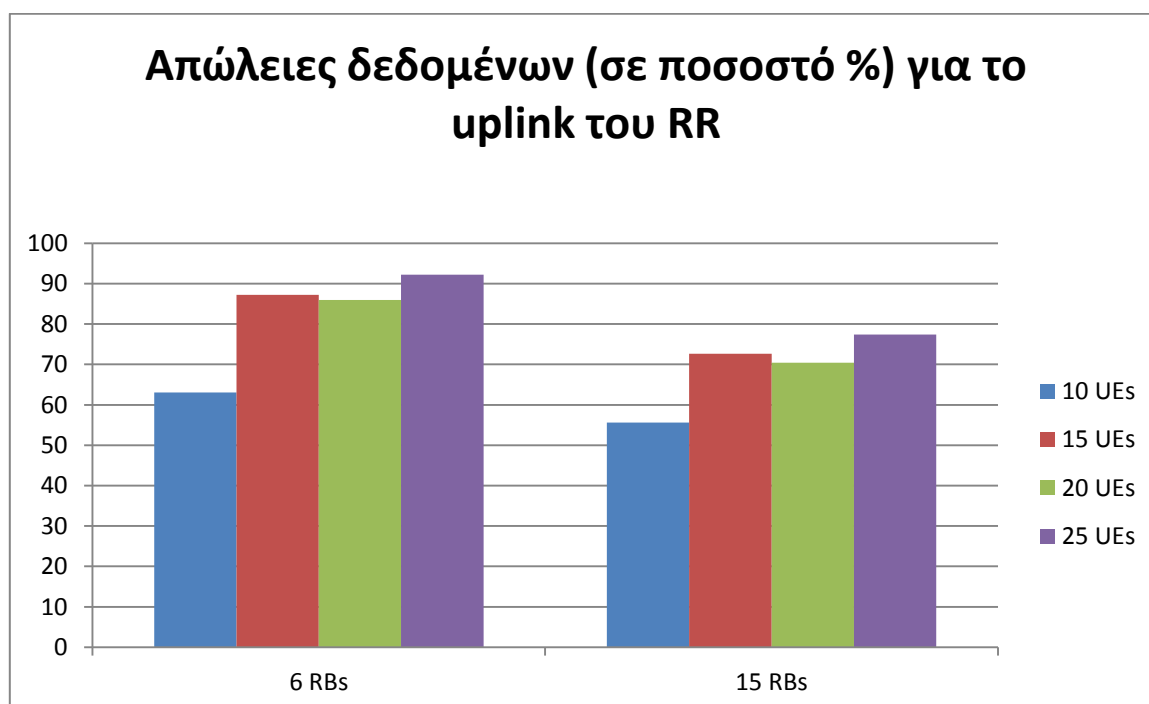
Εικόνα 14: Μέση συνολική καθυστέρηση για κάθε σύνολο της κατερχόμενης ζεύξης για τον RR scheduler – Τοπολογία 1



Εικόνα 15: Μέση συνολική καθυστέρηση για κάθε σύνολο της ανερχόμενης ζεύξης για τον RR scheduler – Τοπολογία 1



Εικόνα 16: Συνολικός ρυθμός απωλειών δεδομένων για κάθε σύνολο της κατερχόμενης ζεύξης για τον RR scheduler – Τοπολογία 1



Εικόνα 17: Συνολικός ρυθμός απωλειών δεδομένων για κάθε σύνολο της ανερχόμενης ζεύξης για τον RR scheduler – Τοπολογία 1

Όσον αφορά την συνολική απόδοση του συστήματος, για το downlink (Εικόνα 12) παρατηρούμε ότι, και για τα δύο σύνολα των 6 και 15 RBs, η απόδοση αυξάνεται καθώς αυξάνεται το πλήθος των χρηστών. Αυτό συμβαίνει γιατί υπάρχουν περισσότερες ενεργές εφαρμογές στο δίκτυο. Έτσι, η κίνηση στο δίκτυο θα είναι μεγαλύτερη και γι’

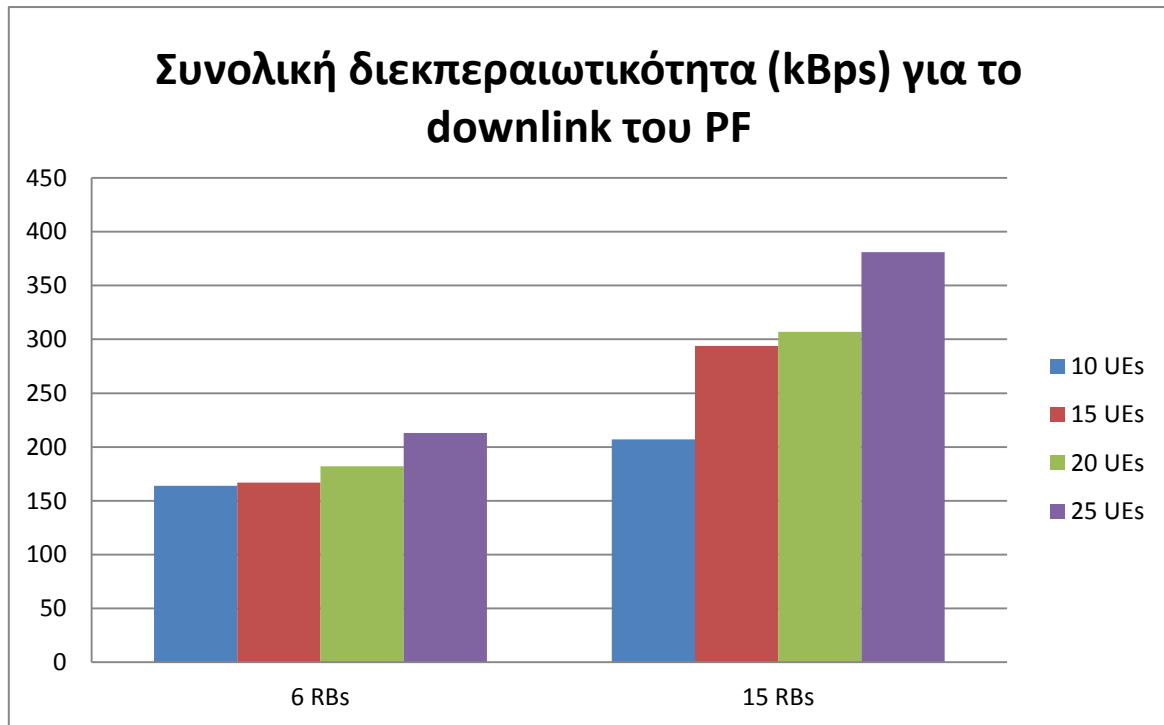
αυτό αυξάνεται και το throughput, το οποίο χαρακτηρίζει το πλήθος των δεδομένων που διαβιβάζονται επιτυχώς στην κατεύθυνση του downlink. Επίσης, η απόδοση στην περίπτωση των 15 RBs είναι πάντα μεγαλύτερη από αυτήν της περίπτωσης των 6 RBs. Αυτό είναι λογικό, καθώς οι πόροι του δικτύου είναι περισσότεροι. Δηλαδή, σε κάθε χρήστη ανατίθενται κάθε φορά περισσότερα RBs και έχει έτσι την δυνατότητα να στείλει επιτυχώς περισσότερα δεδομένα. Από την άλλη, η απόδοση του uplink (Εικόνα 13) είναι μικρότερη από την απόδοση του downlink, για κάθε αντίστοιχο συνδυασμό από Resource Blocks και πλήθος χρηστών. Επίσης, αν και στην περίπτωση των 15 RBs η αύξηση της απόδοσης για το uplink είναι όμοια με την αντίστοιχη αύξηση για το downlink, αυτό δεν ισχύει για την περίπτωση των 6 RBs. Σε αυτό το σύνολο πειραμάτων, η απόδοση του δικτύου δεν αυξάνεται με την αύξηση των χρηστών. Αντιθέτως, η απόδοση είναι μεγαλύτερη όταν στο κτίριο υπάρχουν μόνο 10 χρήστες. Αυτό θεωρούμε ότι συμβαίνει καθώς οι πόροι δεν επαρκούν για την εξυπηρέτηση των χρηστών και έτσι όταν προστίθενται παραπάνω χρήστες θα εμφανίζονται μεγαλύτερες απώλειες και η συνολική απόδοση θα μειώνεται. Αυτό επιβεβαιώνεται και από το διάγραμμα της Εικόνας 17, όπου το ποσοστό των απωλειών ξεπερνά το 80% για τους 15 και 20 χρήστες και το 90% για τους 25.

Σχετικά με την μέση καθυστέρηση (Εικόνα 14 και Εικόνα 15) παρατηρείται ότι, σε κάθε περίπτωση, όταν αυξάνεται το πλήθος των χρηστών, μεγαλώνει και η μέση καθυστέρηση του δικτύου, καθώς υπάρχει μεγαλύτερη συμφόρηση σε αυτό. Επίσης, όταν το πλήθος των UEs είναι σταθερό και αλλάζει μόνο το πλήθος των RBs, παρατηρούμε ότι η μέση καθυστέρηση στα 6 RBs είναι μεγαλύτερη σε σχέση με την καθυστέρηση στα 15 RBs. Αυτό εξηγείται από το γεγονός ότι όταν υπάρχουν περισσότεροι πόροι που διαμοιράζονται ανάμεσα στους χρήστες, τότε το κάθε UE θα έχει μεγαλύτερο διαθέσιμο εύρος ζώνης και έτσι θα μπορεί να στέλνει και να λαμβάνει δεδομένα με μεγαλύτερο ρυθμό. Τέλος, όπως η απόδοση του δικτύου ήταν καλύτερη για το downlink, σε σχέση με το uplink, το ίδιο ισχύει και για την συνολική μέση καθυστέρηση. Δηλαδή, για κάθε συνδυασμό από RBs και UEs, η μέση καθυστέρηση στο uplink είναι μεγαλύτερη από την αντίστοιχη καθυστέρηση του downlink.

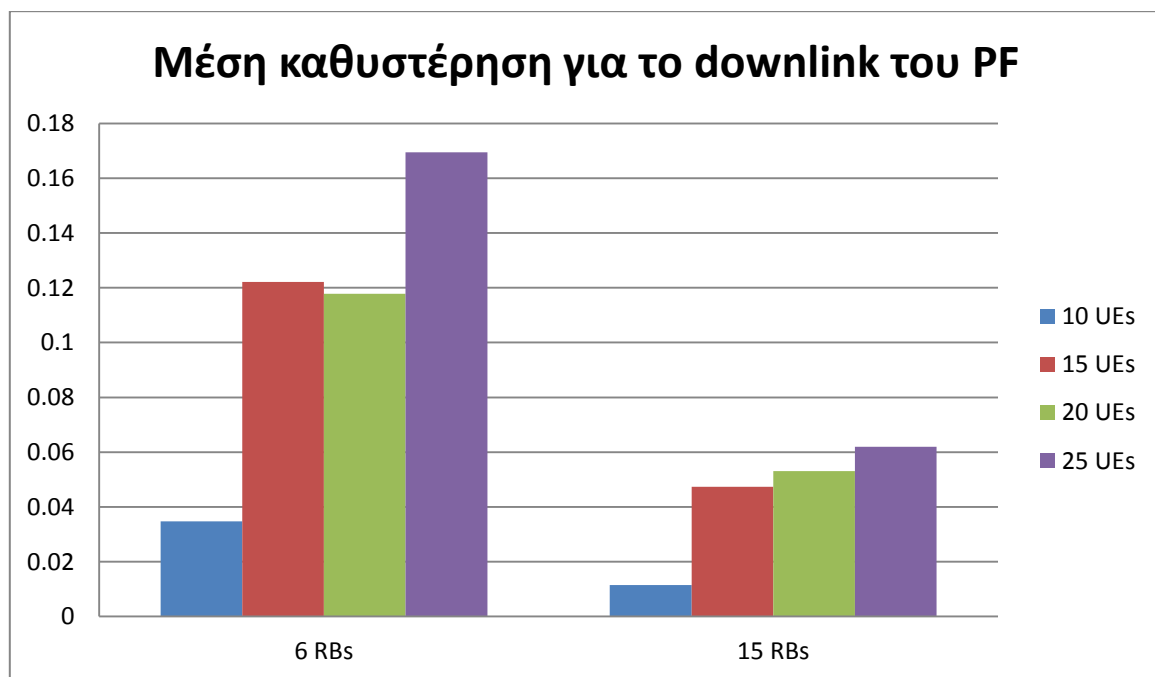
Στην Εικόνα 16 και στην Εικόνα 17 φαίνεται το ποσοστό των απωλειών δεδομένων για την κατερχόμενη και την ανερχόμενη ζεύξη, αντίστοιχα. Το πρώτο πράγμα που παρατηρούμε και στις δύο περιπτώσεις είναι ότι το ποσοστό αυτό είναι πολύ μεγάλο, καθώς τις περισσότερες φορές ξεπερνά το 50%, ενώ στο uplink φτάνει μέχρι και το 90%. Η μόνη περίπτωση που το ποσοστό αυτό είναι κάτω από 50% είναι στο downlink, όταν υπάρχουν 10 ενεργοί χρήστες. Σε αυτήν την περίπτωση έχουμε περίπου 44% απώλειες όταν το bandwidth είναι ίσο με 6 RBs και 30% όταν είναι ίσο με 15 RBs. Όπως αναφέρθηκε και προηγουμένως, η κατάσταση του δικτύου είναι χειρότερη στην ανερχόμενη ζεύξη, συγκριτικά με την κατερχόμενη. Έτσι, οι απώλειες δεδομένων, για ίδιο πλήθος χρηστών και πόρων, είναι μεγαλύτερες στο uplink. Επίσης, τόσο στην κατερχόμενη όσο και στην ανερχόμενη ζεύξη, παρατηρείται ότι, για ίδιο πλήθος από RBs, με την αύξηση των χρηστών αυξάνεται και το ποσοστό των απολεσθέντων δεδομένων. Ωστόσο, στην περίπτωση των 10 και 15 UEs, το ποσοστό αυτό είναι σχεδόν ίδιο, με την διαφορά ότι στο downlink είναι ελαφρώς μεγαλύτερο για τους 15 χρήστες, ενώ στο uplink είναι λίγο μεγαλύτερο στους 10 χρήστες. Τέλος, όπως είναι λογικό, όταν έχουμε ίδιο πλήθος από UEs αλλά αυξάνουμε το πλήθος των διαθέσιμων πόρων, το ποσοστό των απωλειών μειώνεται. Αυτό ισχύει και για τις δύο κατευθύνσεις.

4.1.3.2 Proportional Fair

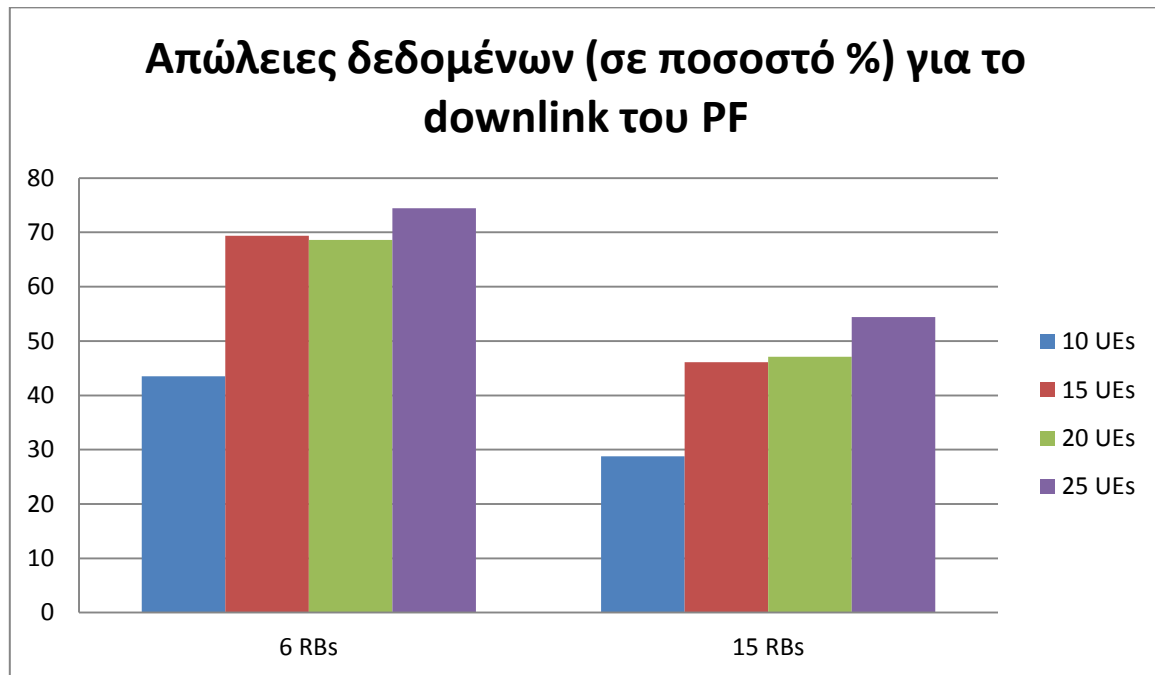
Στις εικόνες που ακολουθούν παρουσιάζονται τα διαγράμματα των αποτελεσμάτων του Proportional Fair scheduler στην κατερχόμενη ζεύξη.



Εικόνα 18: Συνολική διεκπεραιωτικότητα για κάθε σύνολο της κατερχόμενης ζεύξης για τον PF scheduler – Τοπολογία 1



Εικόνα 19: Μέση συνολική καθυστέρηση για κάθε σύνολο της κατερχόμενης ζεύξης για τον PF scheduler – Τοπολογία 1



Εικόνα 20: Συνολικός ρυθμός απωλειών δεδομένων για κάθε σύνολο της κατερχόμενης ζεύξης για τον PF scheduler – Τοπολογία 1

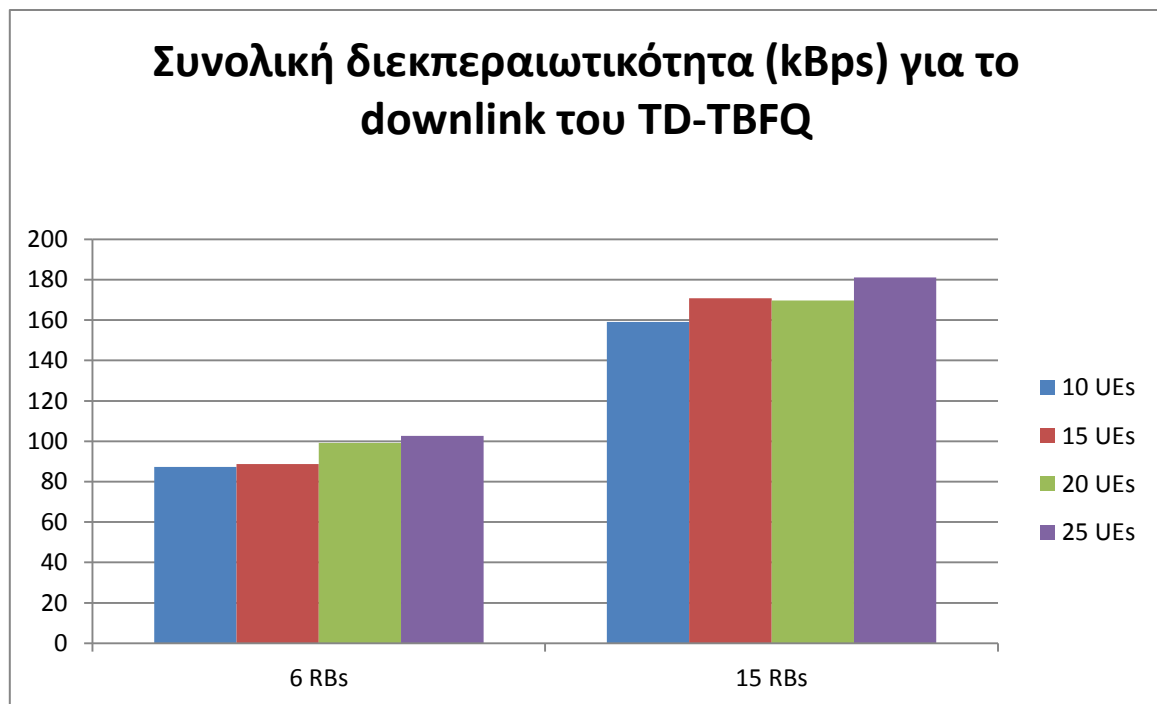
Παρατηρούμε, στην Εικόνα 18 ότι η ολική απόδοση στο downlink αυξάνεται όσο αυξάνονται οι χρήστες και όσο αυξάνονται οι διαθέσιμοι πόροι. Η διεκπεραιωτικότητα αυξάνεται όσο αυξάνονται οι χρήστες διότι υπάρχουν περισσότερες ενεργές εφαρμογές στο δίκτυο με αποτέλεσμα να αυξηθεί η επιτυχής ροή αποστολής και λήψης δεδομένων. Για 15 RBs οι τιμές της διεκπεραιωτικότητας παρουσιάζουν αισθητή διαφορά από ότι για 6 RBs, γεγονός που είναι αναμενόμενο μιας και υπάρχουν περισσότεροι διαθέσιμοι πόροι για κάθε χρήστη και συνεπώς έχει την δυνατότητα να στείλει επιτυχώς περισσότερα πακέτα.

Η μέση συνολική καθυστέρηση στην κατερχόμενη ζεύξη, όπως φαίνεται και στην Εικόνα 19, είναι μεγαλύτερη όταν έχουμε 6 RBs από όταν έχουμε 15 RBs διότι υπάρχουν περισσότεροι διαθέσιμοι πόροι για τον κάθε χρήστη οπότε αυξάνεται ο ρυθμός αποστολής και λήψης για το κάθε UE. Επιπλέον η μέση καθυστέρηση αυξάνεται καθώς αυξάνονται οι χρήστες στο δίκτυο μιας και δημιουργείται μεγάλος φόρτος εξυπηρέτησης.

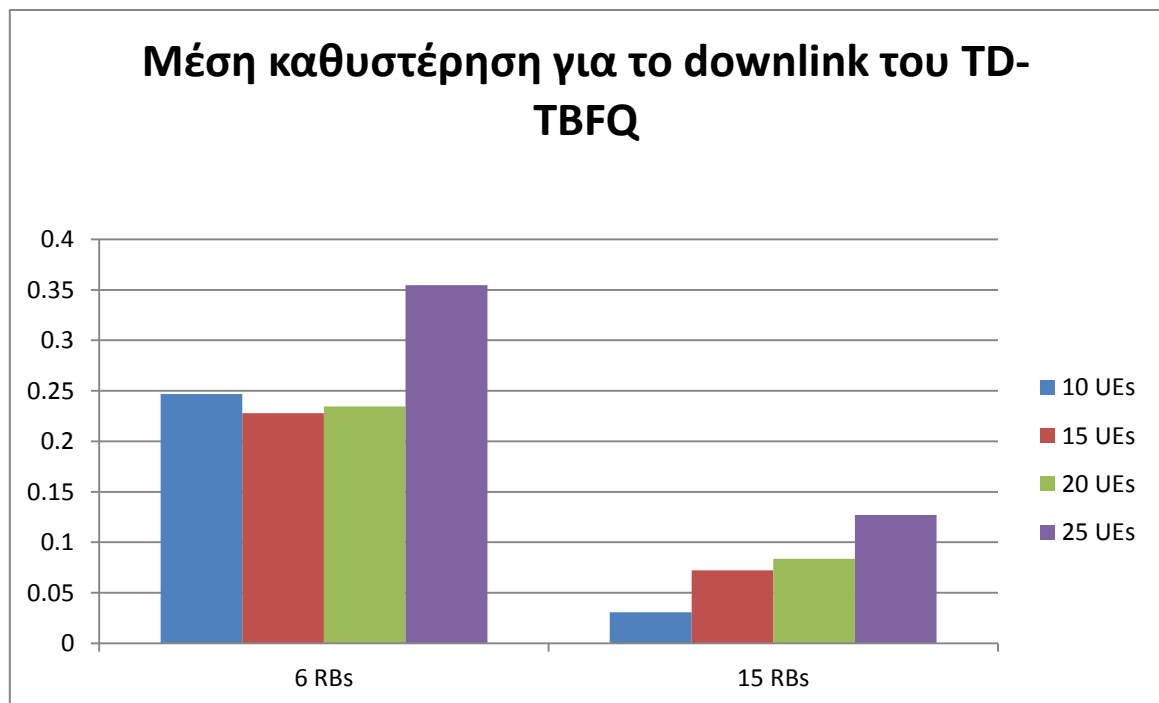
Σχετικά με τις απώλειες δεδομένων, Εικόνα 20, είναι μεγαλύτερες όταν τα RBs είναι 6 και ξεπερνούν το 70% στην κατερχόμενη ζεύξη. Από την άλλη, όταν τα RBs είναι 15 και οι χρήστες 25, οι απώλειες ξεπερνούν το 50%. Άρα το ποσοστό απωλειών είναι καλύτερο όσο αυξάνονται τα RBs για τον ίδιο αριθμό χρηστών. Επιπλέον οι απώλειες αυξάνονται καθώς αυξάνονται οι χρήστες για τον ίδιο αριθμό RBs.

4.1.3.3 Time Domain Token Bank Fair Queue

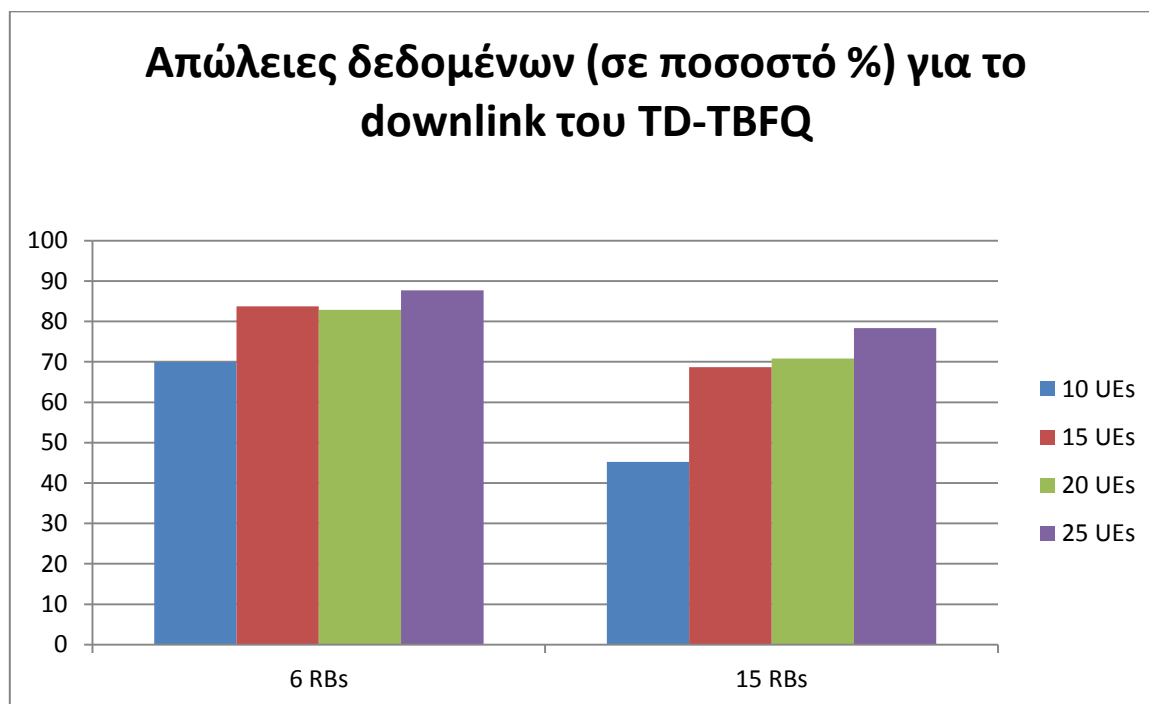
Στις επόμενες εικόνες παρουσιάζονται τα διαγράμματα των αποτελεσμάτων του TD-TBFQ scheduler, για την κατερχόμενη και την ανερχόμενη ζεύξη.



Εικόνα 21: Συνολική διεκπεραιωτικότητα για κάθε σύνολο της κατερχόμενης ζεύξης για τον Time Domain Token Bank Fair Queue scheduler – Τοπολογία 1



Εικόνα 22: Μέση συνολική καθυστέρηση για κάθε σύνολο της κατερχόμενης ζεύξης για τον Time Domain Token Bank Fair Queue scheduler – Τοπολογία



Εικόνα 23: Συνολικός ρυθμός απωλειών δεδομένων για κάθε σύνολο της κατερχόμενης ζεύξης για τον Time Domain Token Bank Fair Queue scheduler – Τοπολογία 1

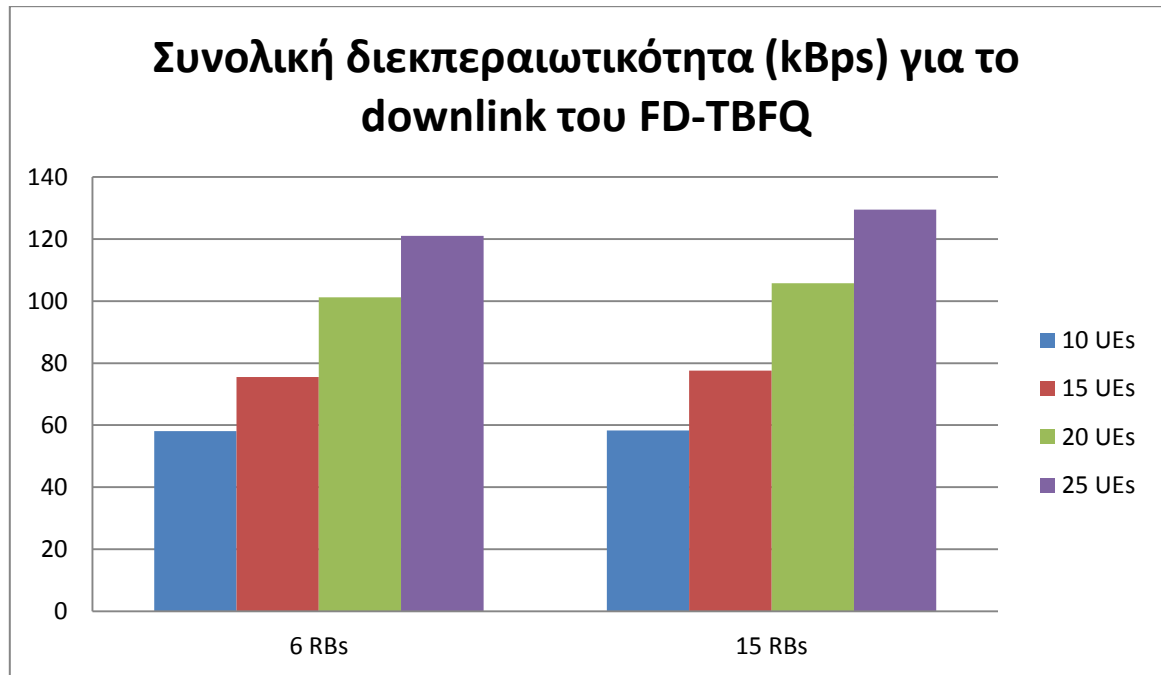
Με την εφαρμογή του αλγορίθμου του TBFQ για το πεδίου του χρόνου παρατηρείται ότι η συνολική διεκπεραιωτικότητα του συστήματος για το downlink (Εικόνα 21) αυξάνεται καθώς αυξάνεται το πλήθος των UEs. Επίσης, όπως και στους προηγούμενους αλγόριθμους, η διεκπεραιωτικότητα στην περίπτωση των 15 RBs είναι πάντα μεγαλύτερη από αυτήν της περίπτωσης των 6 RBs, καθώς το κάθε UE έχει την δυνατότητα δεσμεύσει συνολικά περισσότερους πόρους και να στείλει επιτυχώς περισσότερα δεδομένα.

Σχετικά με την μέση καθυστέρηση (Εικόνα 22) θα περιμέναμε ότι αυτή θα αυξάνεται καθώς προστίθενται παραπάνω χρήστες στο δίκτυο. Ωστόσο, αν και αυτό ισχύει για την περίπτωση των 15 RBs, δεν ισχύει στην περίπτωση των 6 RBs. Εδώ, η μέση καθυστέρηση στους 10 χρήστες είναι κοντά στο 0,25. Στην συνέχεια, στους 15 χρήστες μειώνεται, ενώ στους 20 παρουσιάζει μια πολύ μικρή αύξηση. Ωστόσο, η αύξηση, που παρουσιάζεται στους 20 χρήστες, είναι αμελητέα σε σχέση με την μείωση που υπέστη η καθυστέρηση στους 15 χρήστες. Έτσι, η τιμή της εξακολουθεί να είναι μικρότερη από αυτής με τους 10 χρήστες. Τέλος, στους 25 χρήστες, η καθυστέρηση αυξάνεται και φθάνει το 0,35 sec. Επιπλέον, όπως και με τους παραπάνω αλγορίθμους, παρατηρείται ότι για ίδιο πλήθος χρηστών, αλλά διαφορετικό πλήθος από RBs, η μέση καθυστέρηση είναι μεγαλύτερη όταν χρησιμοποιούνται λιγότερα RBs.

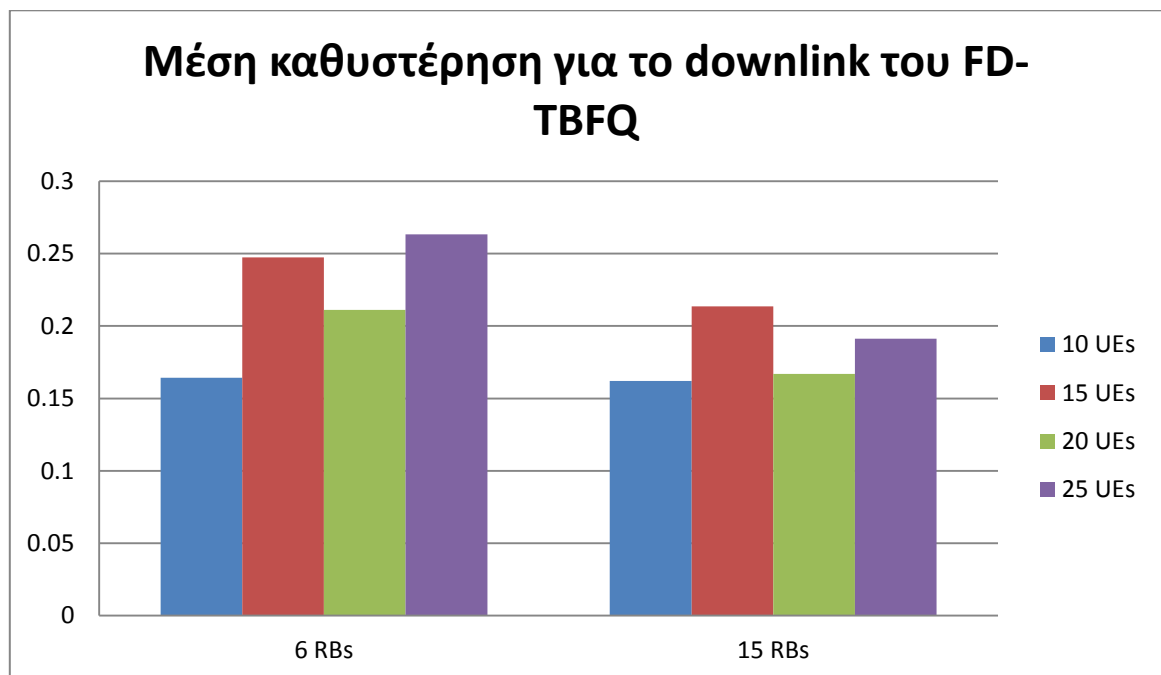
Τα ποσοστά των απωλειών δεδομένων για την κατερχόμενη ζεύξη, παρουσιάζονται στην Εικόνα 23, αντίστοιχα. Δυστυχώς, το ποσοστό αυτό είναι πολύ μεγάλο, καθώς, σχεδόν σε όλες τις περιπτώσεις ξεπερνά το 50%. Επίσης, και για τον TD-TBFQ ισχύει η γενική παρατήρηση πως, το ποσοστό των απωλειών αυξάνεται με την αύξηση των χρηστών, ενώ μειώνεται με την αύξηση των διαθέσιμων πόρων.

4.1.3.4 Frequency Domain Token Bank Fair Queue

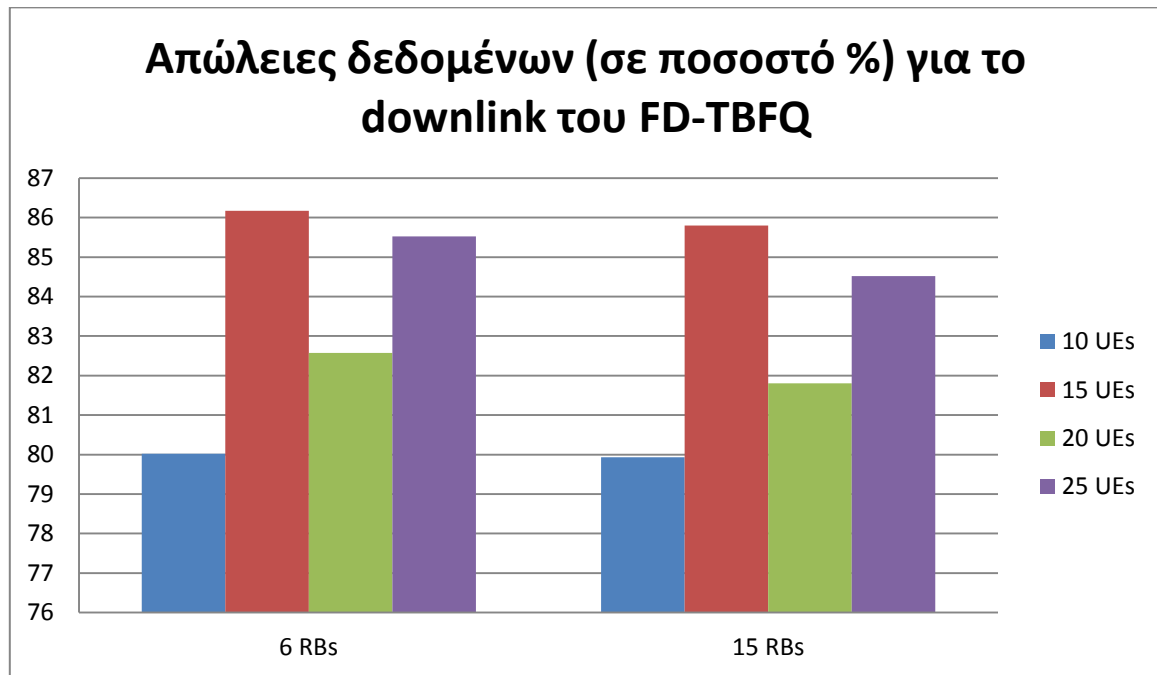
Στις εικόνες που ακολουθούν παρουσιάζονται τα διαγράμματα των αποτελεσμάτων του Frequency Domain Token Bank Fair Queue scheduler στην κατερχόμενη ζεύξη.



Εικόνα 24: Συνολική διεκπεραιωτικότητα για κάθε σύνολο της κατερχόμενης ζεύξης για τον FD-TBFQ scheduler – Τοπολογία 1



Εικόνα 25: Μέση συνολική καθυστέρηση για κάθε σύνολο της κατερχόμενης ζεύξης για τον FD-TBFQ scheduler – Τοπολογία 1



Εικόνα 26: Συνολικός ρυθμός απωλειών δεδομένων για κάθε σύνολο της κατερχόμενης ζεύξης για τον FD-TBFQ scheduler – Τοπολογία 1

Παρατηρούμε, στην Εικόνα 24, ότι η ολική διεκπεραιωτικότητα στο downlink αυξάνεται όσο αυξάνονται οι χρήστες και όσο αυξάνονται οι διαθέσιμοι πόροι. Όπως και σε όλους τους υπόλοιπους αλγορίθμους, οι τιμές της διεκπεραιωτικότητας εμφανίζουν διαφορά ανάμεσα στα 6 και 15 RBs, καθώς όταν υπάρχουν περισσότεροι διαθέσιμοι πόροι, ο κάθε χρήστης θα έχει την δυνατότητα να στέλνει επιτυχώς περισσότερα πακέτα. Συγκριτικά με τους άλλους αλγορίθμους ο PF παρουσιάζει την καλύτερη απόδοση στο downlink.

Η μέση συνολική καθυστέρηση στην κατερχόμενη ζεύξη, όπως φαίνεται και στην Εικόνα 25, είναι μεγαλύτερη όταν έχουμε 6 RBs από όταν έχουμε 15 RBs. Σε αυτόν τον αλγόριθμο, όταν υπάρχουν 6 RBs, η μέση καθυστέρηση στο downlink είναι μεγαλύτερη όταν έχουμε 15 χρήστες απ' ότι όταν έχουμε 20 χρήστες, πράγμα που έρχεται σε αντίθεση με αυτό που θα περιμέναμε. Το ίδιο ισχύει και για την περίπτωση των 15 RBs.

Συγκριτικά με τους άλλους αλγορίθμους ο PF παρουσιάζει την μικρότερη μέση συνολική καθυστέρηση.

Σχετικά με τις απώλειες δεδομένων (Εικόνα 26), αυτές είναι μεγαλύτερες όταν τα RBs είναι 6. Συγκεκριμένα, για τα 6 RBs, οι μεγαλύτερες απώλειες για την κατερχόμενη ζεύξη προκύπτουν όταν οι χρήστες είναι 15, όπου και ξεπερνούν το 86%. Επίσης, η μεταβολή των απωλειών ως συνάρτηση του πλήθους των χρηστών παρουσιάζει όμοια συμπεριφορά με την αντίστοιχη μεταβολή της καθυστέρησης, που είδαμε στην Εικόνα 25. Επιπλέον, παρατηρείται ότι το ποσοστό απωλειών είναι καλύτερο όσο αυξάνονται τα RBs για τον ίδιο αριθμό χρηστών.

Συγκριτικά με τους άλλους αλγορίθμους ο PF παρουσιάζει το μικρότερο ποσοστό απωλειών.

4.2 2^ο Σενάριο Προσομοίωσης

4.2.1 Περιγραφή

Η δεύτερη κατηγορία πειραμάτων βασίζεται στην έτοιμη προσομοίωση της ns-3 βιβλιοθήκης, *lena-dual-stripe*, της οποίας ο κώδικας παρουσιάζεται στο [36]. Για την προσομοίωσή μας έχει διατηρηθεί η τοπολογία του *lena-dual-stripe*, αλλά έχουν αλλάξει οι εφαρμογές των χρηστών, ενώ το πλήθος τους και ο αλγόριθμος διαμοιρασμού πόρων μπορούν να μεταβάλλονται από τις παραμέτρους του προγράμματος. Το κύριο χαρακτηριστικό της προσομοίωσης είναι ότι υπάρχουν δύο διαφορετικές κατηγορίες κυψελών: οι μακροκυψέλες (macrocells) και οι φεμτοκυψέλες. Έτσι, και οι χρήστες διαχωρίζονται σε δύο κατηγορίες, ανάλογα με το είδος της κυψέλης στην οποία συνδέονται. Το πρόγραμμα που φτιάξαμε για τον σκοπό αυτό είναι το *building-sim-lena.cc* και μπορεί να δεχτεί μέχρι δύο παραμέτρους. Η πρώτη παράμετρος ορίζει το πλήθος των χρηστών της κάθε κατηγορίας (*nHomeUes* και *nMacroUes*). Δηλαδή, το πλήθος των UEs στις δύο παραπάνω κατηγορίες που αναφέραμε θα είναι το ίδιο. Η δεύτερη παράμετρος ορίζει τον τύπο του scheduler (*schedulerType*) που επιθυμούμε. Οι προεπιλεγμένες τιμές, που ανατίθενται σε αυτές τις μεταβλητές σε περίπτωση που δεν ορίζονται από τον χρήστη, είναι *nHomeUes* = *nMacroUes* = 10 και *schedulerType* = "rr". Η προσομοίωσή μας τρέχει για 100 sec, ενώ τα πειράματά μας τα εκτελέσαμε για 10, 15, 20 και 25 χρήστες, σε κάθε κατηγορία. Οι schedulers που χρησιμοποιήθηκαν είναι οι ίδιοι με της προηγούμενης προσομοίωσης, δηλαδή οι Proportional Fair, Round Robin, Frequency Domain Token Bank Fair Queue και Time Domain Token Bank Fair Queue. Οι παραπάνω τύποι scheduler αναγνωρίζονται από το πρόγραμμά μας, μέσω της μεταβλητής *schedulerType* ως "pf", "rr", "fdtbfq", "tdtbfq", αντίστοιχα. Τέλος, το πλήθος των Resource Blocks είναι σταθερά 15 RBs για τις μακροκυψέλες και 6 RBs για τις φεμτοκυψέλες. Στη συνέχεια αυτής της ενότητας περιγράφεται το πείραμα και κάποιες κλάσεις του ns-3, που θα μας χρησιμεύσουν.

4.2.1.1 Η τοπολογία της προσομοίωσης

Στο δίκτυο συμπεριλαμβάνονται 2 κτίρια. Τα κτίρια αυτά περιλαμβάνουν 2 σειρές δωματίων, όπου η κάθε σειρά έχει 10 δωμάτια. Δηλαδή, το κάθε κτίριο αποτελείται από 20 δωμάτια και έναν όροφο. Επίσης, το κάθε κτίριο τοποθετείται κάπου τυχαία, εντός της περιοχής του δικτύου, και έχει διαστάσεις 100 × 20 m² και ύψος 3 m.

Η δημιουργία των κτιρίων πραγματοποιείται από την συνάρτηση *FemtocellBlockAllocator::Create()*, την οποία μπορείτε να δείτε στο [36]. Για τον σκοπό αυτόν χρησιμοποιείται η κλάση *ns3::GridBuildingAllocator*, η οποία επιτρέπει την δημιουργία δισδιάστατων κτιρίων. Οι παράμετροι της κλάσης που τροποποιούνται από την συνάρτηση είναι ακόλουθες:

- **GridWidth:** Ορίζει το πλήθος των αντικειμένων που θα υπάρχουν σε μια οριζόντια γραμμή. Η μεταβλητή *GridWidth* είναι του τύπου *ns3::UIntegerValue* και η τιμή που τις αποδίδεται είναι 1 αντικείμενο. Στην περίπτωσή μας ένα αντικείμενο είναι ένα κτίριο και, συνεπώς, δεν θα υπάρχουν δύο κτίρια στην ίδια ευθεία.
- **LengthX:** Αντιστοιχεί στο μήκος του τοίχους του κτιρίου στον x άξονα. Είναι μεταβλητή του τύπου *ns3::DoubleValue* και ορίζεται στην τιμή των 100 m.
- **LengthY:** Αντιστοιχεί στο μήκος του τοίχους του κτιρίου στον y άξονα. Είναι, επίσης, μεταβλητή του τύπου *ns3::DoubleValue* και ορίζεται στην τιμή των 20 m.

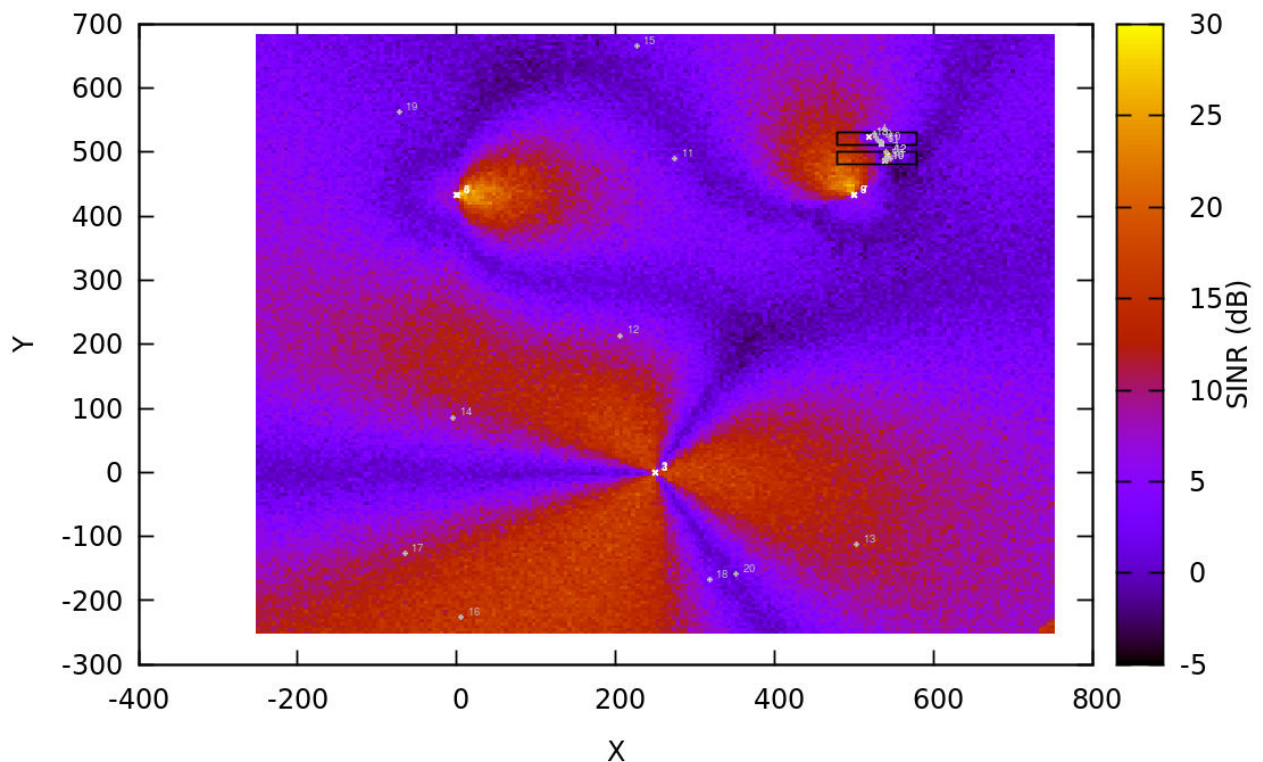
- **DeltaX:** Η οριζόντια απόσταση (στον άξονα x) μεταξύ των κτιρίων, η οποία ορίζεται από την κλάση *ns3::DoubleValue*. Η τιμή της είναι 10 m.
- **DeltaY:** Η κατακόρυφη απόσταση (στον άξονα y) μεταξύ των κτιρίων, η οποία ορίζεται από την κλάση *ns3::DoubleValue*. Η τιμή της είναι 10 m.
- **Height:** Το ύψος του κτιρίου, δηλαδή το ύψος στο οποίο βρίσκεται η κορυφή του. Ορίζεται μέσω της κλάσης *ns3::DoubleValue* και η τιμή της είναι 3 m.
- **NRoomsX:** Το πλήθος των δωματίων στον x άξονα, το οποίο ορίζεται σε 10 δωμάτια.
- **NRoomsY:** Το πλήθος των δωματίων στον y άξονα, το οποίο ορίζεται σε 2 δωμάτια.
- **NFloors:** Το πλήθος των ορόφων, το οποίο ορίζεται σε έναν όροφο.
- **MinX:** Αντιστοιχεί στην x συντεταγμένη, από την οποία ξεκινάει η δημιουργία του κτιρίου. Ορίζεται από την κλάση *ns3::DoubleValue* και στο πείραμα μας η τιμή της είναι το 478,548.
- **MinY:** Αντιστοιχεί στην y συντεταγμένη, από την οποία ξεκινάει η δημιουργία του κτιρίου. Ορίζεται από την κλάση *ns3::DoubleValue* και στο πείραμα μας η τιμή της είναι το 480,745.

Για περισσότερες λεπτομέρειες σχετικά με την κλάση *ns3::GridBuildingAllocator* δείτε το [42].

Στο δίκτυο υπάρχουν συνολικά 4 femtocells και 3 macrocells. Οι σταθμοί βάσης των μακροκυψελών βρίσκονται έξω από τα κτίρια και αποτελούνται από 3 κεραίες, οι οποίες εκπέμπουν παραβολικά προς διαφορετικές κατευθύνσεις. Το άνοιγμα του λοβού εκπομπής αυτών των κεραιών είναι 70° και η εκπεμπόμενη ισχύς τους είναι 46 dBm. Επίσης, η EARFCN συχνότητα για το downlink είναι στα 100 Hz και στο uplink είναι στα 18100 Hz, ενώ το εύρος ζώνης και για τις δύο κατευθύνσεις είναι ίσο με 15 RBs. Από την άλλη, οι σταθμοί βάσης των φεμτοκυψελών τοποθετούνται τυχαία μέσα στα κτίρια και εκπέμπουν ισοτροπικά, με ισχύς εκπομπής ίση με 20 dBm. Οι EARFCN συχνότητες για αυτά τα eNBs είναι, επίσης 100 Hz για την κατερχόμενη και 18100 Hz για την ανερχόμενη ζεύξη, ενώ και για τις δύο το εύρος ζώνης ισούται με 6 RBs.

Όσον αφορά τους χρήστες, οι μισοί από αυτούς τοποθετούνται, τυχαία, μέσα σε δωμάτια όπου υπάρχει κάποιος σταθμός βάσης μιας φεμτοκυψέλης. Οι υπόλοιποι, βρίσκονται, επίσης τυχαία, μέσα στην εμβέλεια του δικτύου. Το δίκτυο εκτείνεται από το -250 μέχρι το 750 στον οριζόντιο άξονα και από το -250 στο 683 στον κατακόρυφο. Επιπλέον, τα UEs θα βρίσκονται πάντα σε ύψος 1,5 m.

Ένα ενδεικτικό διάγραμμα της τοπολογίας του δικτύου, με 20 χρήστες (10 για τις φεμτοκυψέλες και 10 για τις μακροκυψέλες), φαίνεται στην Εικόνα 27. Σε αυτήν την εικόνα παρουσιάζονται και τα επίπεδα του SINR (σε dB) για κάθε σημείο του δικτύου.

Εικόνα 27: Τοπολογία 2^{ης} προσομοίωσης

Όπως και με την 1^η προσομοίωση, το διάγραμμα της Εικόνας 27 έχει προκύψει από την χρήση του Radio Environment Map. Το REM παράγεται με τον ίδιο τρόπο που παράγεται και στην 1^η προσομοίωση, σύμφωνα με τα [35] και [36]. Ωστόσο, οι παράμετροι, που ορίστηκαν για την παραγωγή του REM, δεν είναι ίδιες με προηγούμενως. Οι διαφορές είναι ότι, πλέον, οι διαστάσεις του REM είναι ίδιες με τις διαστάσεις όλου του δικτύου, ενώ το ύψος στο οποίο γίνονται οι μετρήσεις είναι στο 1,5 m, καθώς αυτό είναι και το ύψος όπου βρίσκονται τα UEs. Επίσης, η ανάλυση για τους άξονες x και y έχει ελαττωθεί στα 200 σημεία. Τέλος, για αυτήν την περίπτωση REM έχει μεταβληθεί επιπλέον η μεταβλητή *UseDataChannel*. Αν η μεταβλητή αυτή είναι true, τότε το REM θα δημιουργηθεί για το PDSCH κανάλι. Αλλιώς, θα δημιουργηθεί για το PDCCH κανάλι. Είναι του τύπου *BooleanValue*. Στην περίπτωσή μας η τιμή της έχει οριστεί ως true και έτσι τα αποτελέσματα θα αφορούν το PDSCH κανάλι.

Ως μοντέλο απωλειών του δικτύου ορίζεται το *HybridBuildingsPropagationLossModel*, το οποίο αποτελεί έναν συνδυασμό από διαφορετικά μοντέλα, ικανά να υπολογίσουν την εξασθένιση για τις συχνότητες μεταξύ των 200 και 2600 MHz σε διαφορετικά περιβάλλοντα, καθώς και μέσα και έξω από τα κτίρια, όπως αναφέρεται και στο [43]. Τα επιπλέον χαρακτηριστικά που ορίζονται για το μοντέλο απωλειών είναι τα ακόλουθα:

- **ShadowSigmaExtWalls:** Ορίζει την τυπική απόκλιση της κανονικής κατανομής που χρησιμοποιείται για τον υπολογισμό της σκίασης λόγω των εξωτερικών τοίχων. Είναι μεταβλητή του τύπου *ns3::DoubleValue* και της ανατίθεται η τιμή 0.
- **ShadowSigmaOutdoor:** Ορίζει την τυπική απόκλιση της κανονικής κατανομής που χρησιμοποιείται για τον υπολογισμό της σκίασης λόγω των εξωτερικών κόμβων. Είναι μεταβλητή του τύπου *ns3::DoubleValue* και της ανατίθεται η τιμή 1.
- **ShadowSigmaIndoor:** Ορίζει την τυπική απόκλιση της κανονικής κατανομής που χρησιμοποιείται για τον υπολογισμό της σκίασης λόγω των εσωτερικών κόμβων. Είναι μεταβλητή του τύπου *ns3::DoubleValue* και της ανατίθεται η τιμή 1,5.

- **Los2NlosThr:** Ορίζει το κατώφλι κάτω από το οποίο μεταβαίνουμε από κατάσταση ορατότητας (Line of Sight – LoS) σε κατάσταση μη ορατότητας (Non Line of Sight – NLoS). Είναι μεταβλητή του τύπου *ns3::DoubleValue* και της ανατίθεται η τιμή 10^6 m. Με αυτόν τον τρόπο, σε οποιοδήποτε σημείο του δικτύου θα ισχύει πάντα το LoS μοντέλο.

Επίσης, ως ο τύπος του φασματικού καναλιού που θα χρησιμοποιηθεί για τα UEs ορίζεται το *MultiModelSpectrumChannel*, το οποίο μπορεί να διαχειριστεί διαφορετικά φασματικά μοντέλα, σύμφωνα με το [44].

Τέλος, το δίκτυο έχει υλοποιηθεί σε τοπολογία EPC, όπως και η 1^η προσωμοίωση.

4.2.1.2 Οι εφαρμογές των χρηστών

Για την δεύτερη προσομοίωση έχουν χρησιμοποιηθεί οι ίδιες εφαρμογές που υλοποιήθηκαν για την πρώτη και αναλύονται στην Ενότητα 4.1.1.2. Ωστόσο, δύο από αυτές έχουν μεταβληθεί. Αυτές είναι οι ακόλουθες:

- Application 1: Για την μετάδοση video δεν έχει οριστεί κάποιος περιορισμός στο συνολικό πλήθος των πακέτων που θα σταλούν. Αντιθέτως, η εφαρμογή αρχίζει την στιγμή 0,01 sec και η μετάδοση συνεχίζεται μέχρι και το τέλος της προσομοίωσης.
- Application 7: Η ροή του video δεν σταματάει μετά τα 10 sec, αλλά συνεχίζει μέχρι το τέλος της προσομοίωσης.

Επίσης, και σε αυτήν την προσομοίωση, έχει οριστεί ένα bearer για κάθε UE. Τα bearers ανήκουν όλα στην κατηγορία *NGBR_VIDEO_TCP_DEFAULT*.

4.2.2 Παράμετροι προσομοίωσης

Στον Πίνακα 2 παρουσιάζονται συνοπτικά οι τιμές των διάφορων παραμέτρων της προσομοίωσης.

Πίνακας 2: Τιμές των παραμέτρων της 2^{ης} προσομοίωσης

Πλήθος κτιρίων	2	
Διαστάσεις κτιρίων (μήκος×πλάτος×ύψος)	100 × 20 × 3	
Απόσταση μεταξύ των κτιρίων	10 m στον άξονα y	
Πλήθος ορόφων στο κάθε κτίριο	1	
Πλήθος δωματίων στο κάθε κτίριο	20	
Φεμτοκυψέλες	Πλήθος από κυψέλες (eNBs)	4
	Εύρος ζώνης	6 RBs
	Κεραίες σταθμού βάσης	Ισοτροπικές

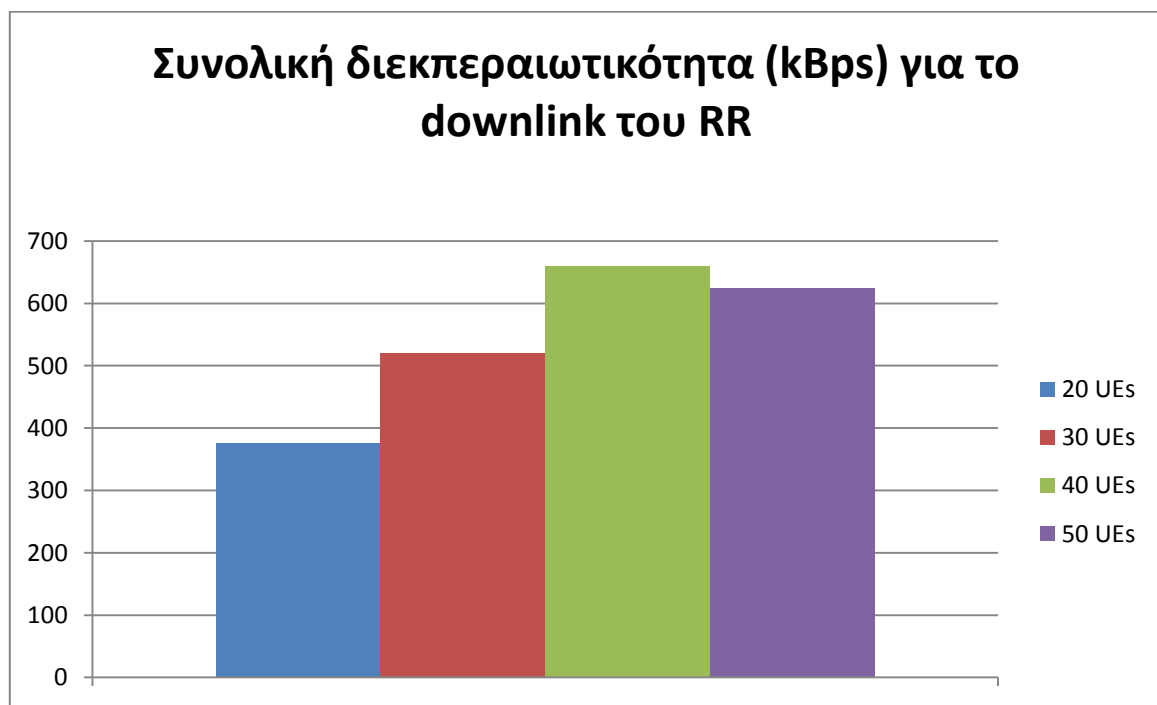
	Θέσεις των κεραιών	Τυχαία μέσα στα κτίρια
	Ισχύς εκπομπής κεραιών	20 dBm
	EARFCN για downlink	100 Hz
	EARFCN για uplink	18.100 Hz
Μακροκυψέλες	Πλήθος από κυψέλες (eNBs)	3
	Εύρος ζώνης	15 RBs
	Κεραίες σταθμού βάσης	Κεραίες με 3 τομείς που εκπέμπουν παραβολικά, με γωνία ανοίγματος 70°
	Θέσεις των κεραιών	Έξω από τα κτίρια
	Ισχύς εκπομπής κεραιών	46 dBm
	EARFCN για downlink	100 Hz
	EARFCN για uplink	18.100 Hz
Αλγόριθμοι για τον scheduler	Round Robin Proportional Fair Time Domain Token Bank Fair Queue Frequency Domain Token Bank Fair Queue	
Πλήθος χρηστών	20, 30, 40 και 50	
Κατανομή χρηστών	Οι μισοί συνδέονται με φεμτοκυψέλες και οι άλλοι μισοί με μακροκυψέλες.	
Ύψος των UEs	1,5 m	
Εμβέλεια δικτύου στον x άξονα	-250 – 750	
Εμβέλεια δικτύου στον y άξονα	-250 – 683	
Διάρκεια προσομοίωσης	100 sec	

4.2.3 Αποτελέσματα

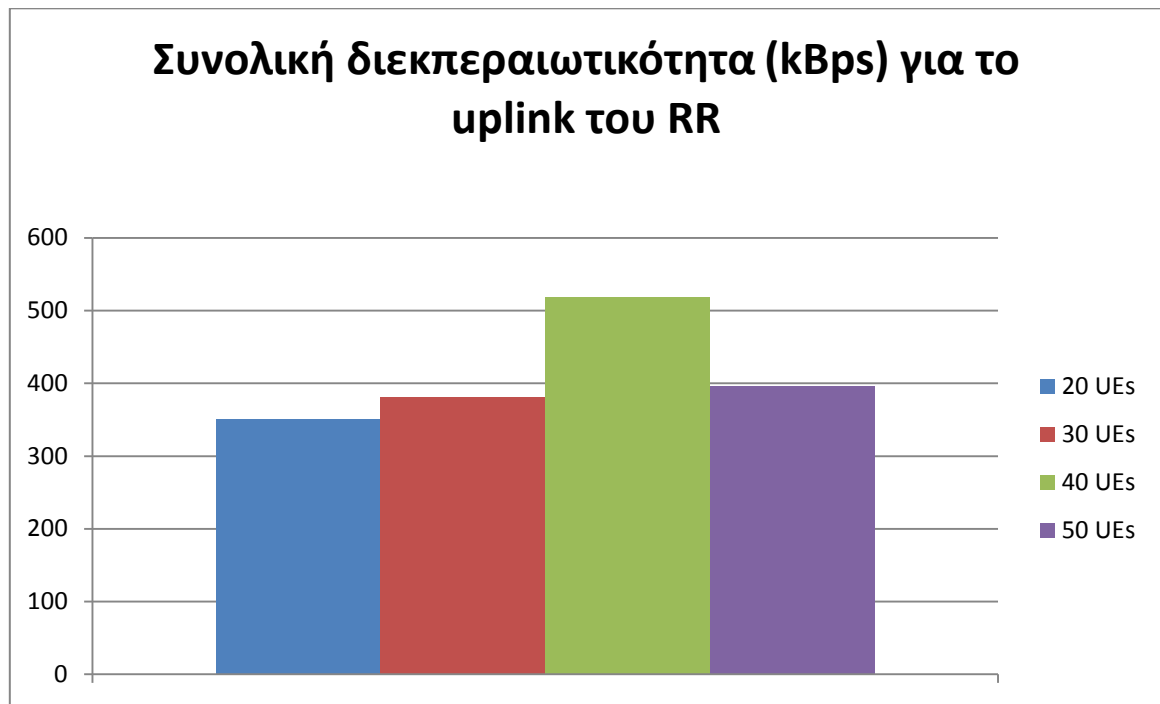
Τα πειράματά μας διαχωρίζονται σε 4 διαφορετικά σύνολα, τα οποία σχετίζονται με το πλήθος των χρηστών, οι οποίοι συνολικά είναι 20, 30, 40 και 50. Για τον υπολογισμό της απόδοσης, των καθυστερήσεων και των απωλειών του δικτύου λαμβάνονται υπόψη όλοι οι χρήστες, χωρίς να γίνει κάποιος διαχωρισμός ανάμεσα σε αυτούς που συνδέονται σε φεμτοκυψέλες και εκείνους που συνδέονται σε μακροκυψέλες. Στις επόμενες ενότητες παρουσιάζονται τα διαγράμματα για τους PF, TD-TBFQ και FD-TBFQ schedulers, για την κατερχόμενη ζεύξη, και του RR για την κατερχόμενη και την ανερχόμενη ζεύξη.

4.2.3.1 Round Robin

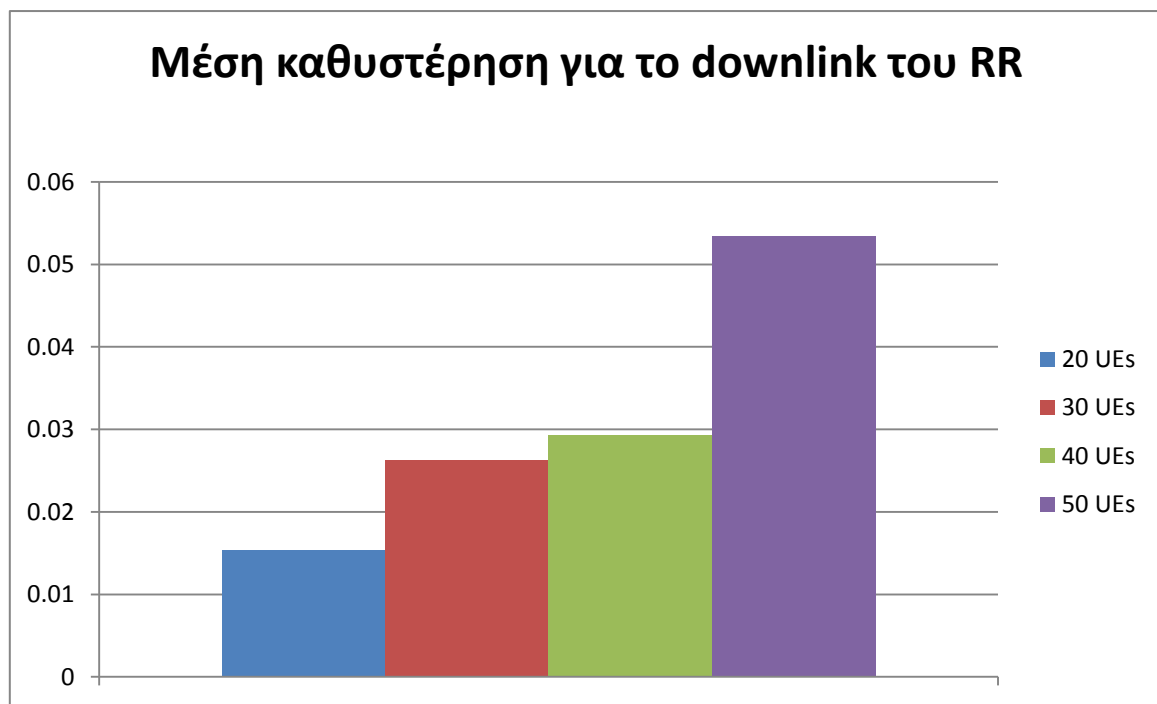
Στις Εικόνες 28 – 33 παρουσιάζονται τα αποτελέσματα για τον αλγόριθμο του Round Robin.



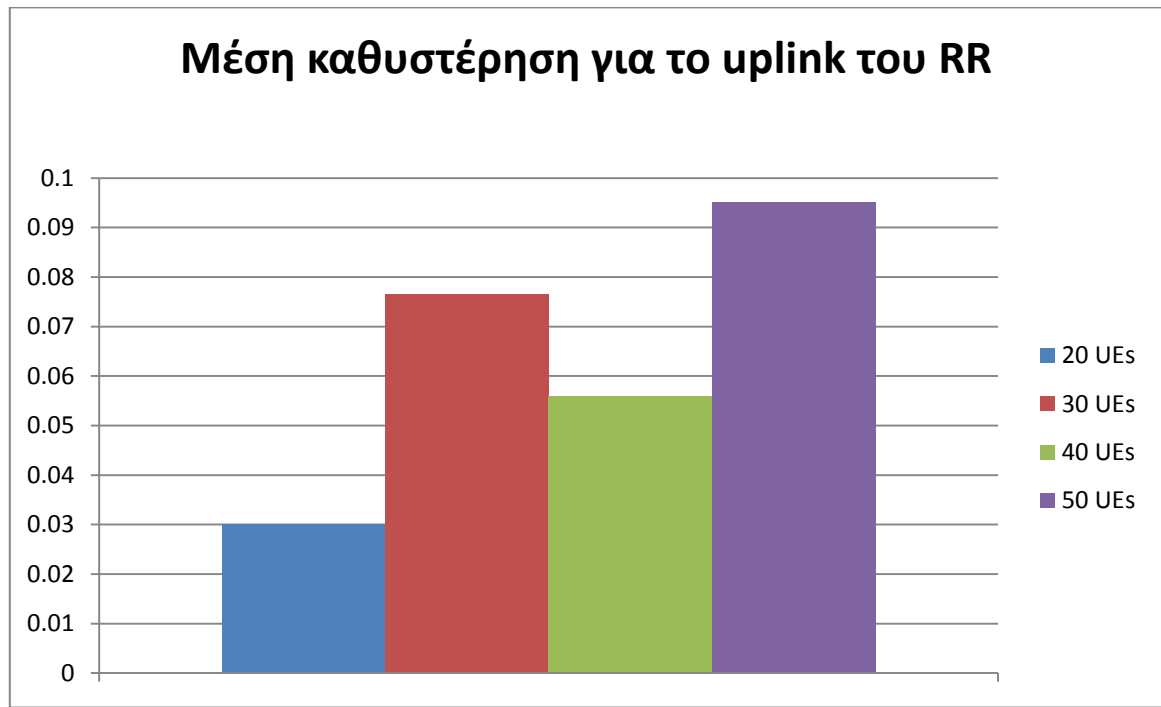
Εικόνα 28: Συνολική διεκπεραιωτικότητα για κάθε σύνολο της κατερχόμενης ζεύξης για τον RR scheduler – Τοπολογία 2



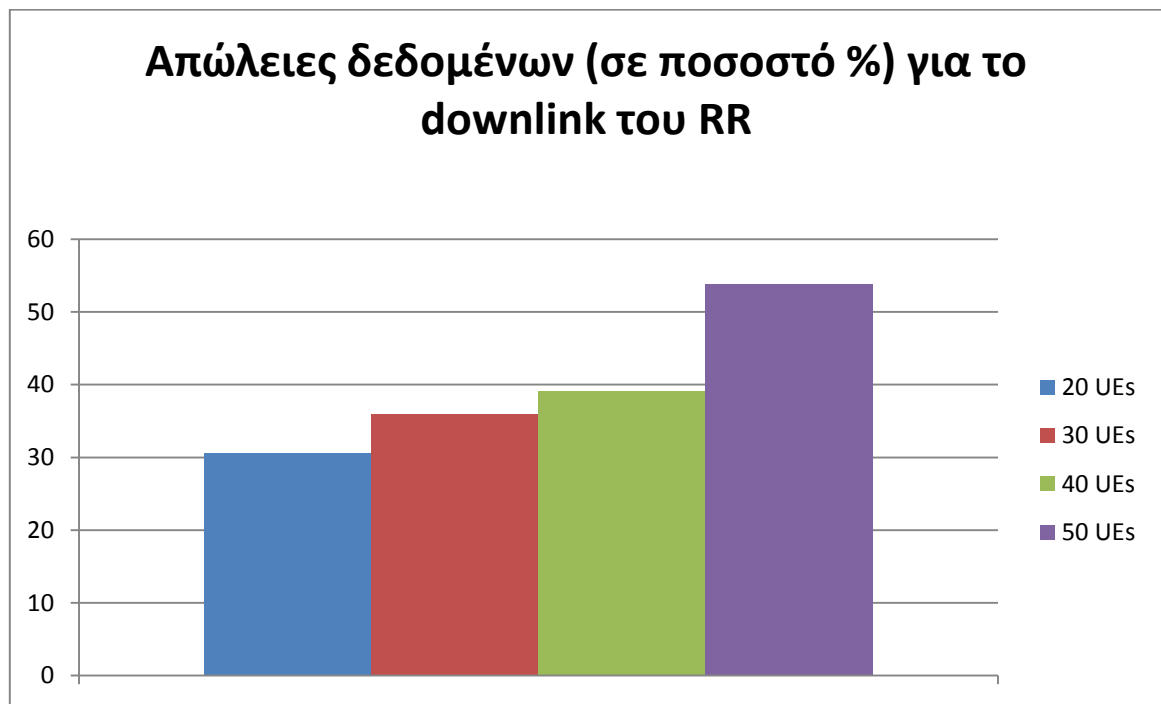
Εικόνα 29: Συνολική διεκπεραιωτικότητα για κάθε σύνολο της ανερχόμενης ζεύξης για τον RR scheduler – Τοπολογία 2



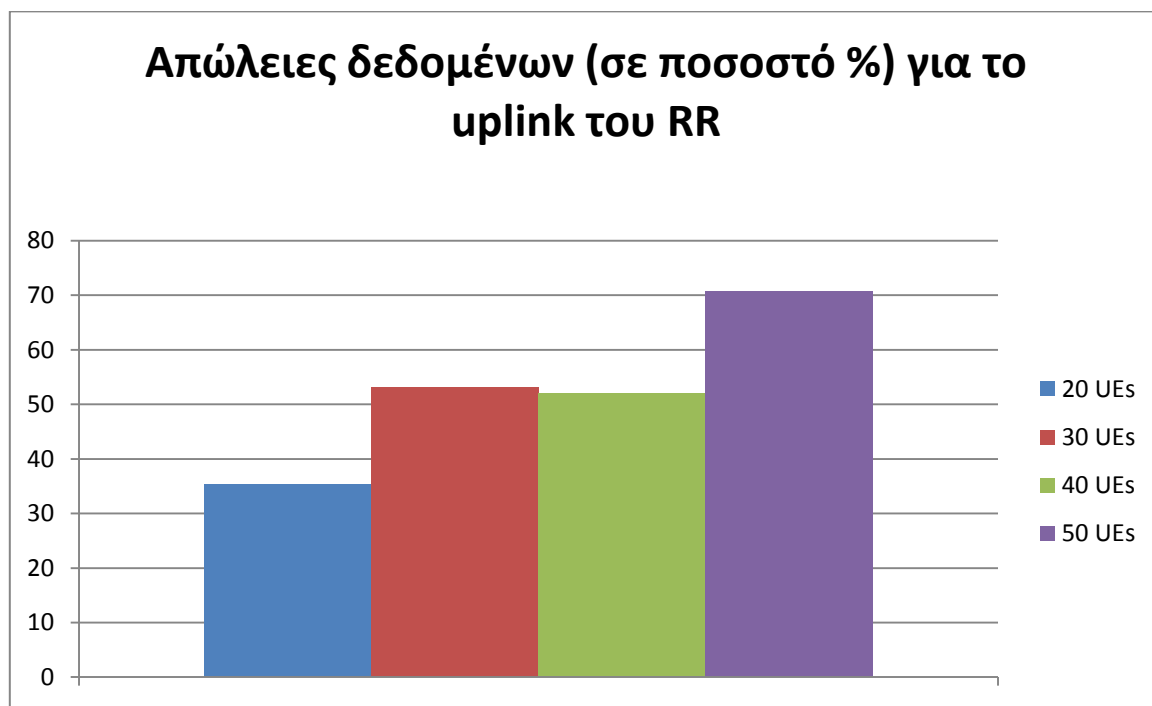
Εικόνα 30: Μέση συνολική καθυστέρηση για κάθε σύνολο της κατερχόμενης για τον RR scheduler – Τοπολογία 2



Εικόνα 31: Μέση συνολική καθυστέρηση για κάθε σύνολο της ανερχόμενης ζεύξης για τον RR scheduler – Τοπολογία 2



Εικόνα 32: Συνολικός ρυθμός απωλειών δεδομένων για κάθε σύνολο της κατερχόμενης ζεύξης για τον Round Robin scheduler – Τοπολογία 2



Εικόνα 33: Συνολικός ρυθμός απωλειών δεδομένων για κάθε σύνολο της ανερχόμενης ζεύξης για τον RR scheduler – Τοπολογία 2

Όσον αφορά την συνολική διεκπεραιωτικότητα (throughput) του συστήματος, για την κατερχόμενη ζεύξη (Εικόνα 28) παρατηρούμε ότι με την αύξηση των χρηστών, η απόδοση αυξάνεται, καθώς θα υπάρχουν περισσότερες ενεργές εφαρμογές στο δίκτυο. Ωστόσο, το throughput στα 50 UEs είναι μικρότερο απ' ό,τι στα 40, πιθανότατα γιατί υπάρχει μεγάλη συμφόρηση στο δίκτυο και έτσι θα υφίστανται και μεγαλύτερες απώλειες. Αυτό επιβεβαιώνεται και από το διάγραμμα της Εικόνας 32, όπου βλέπουμε πως στους 50 χρήστες το ποσοστό των απωλειών για το downlink αυξάνεται απότομα. Από την άλλη, το throughput στην ανερχόμενη ζεύξη (Εικόνα 29) είναι μικρότερο από το αντίστοιχο throughput της κατερχόμενης, για ίδιο πλήθος χρηστών. Επίσης, ο τρόπος που αυξάνεται το throughput, ως συνάρτηση του πλήθους των χρηστών, στο uplink είναι όμοιος με τον τρόπο που αυξάνεται στο downlink. Δηλαδή, στους 50 χρήστες, το throughput του δικτύου, αντί να αυξηθεί, μειώνεται.

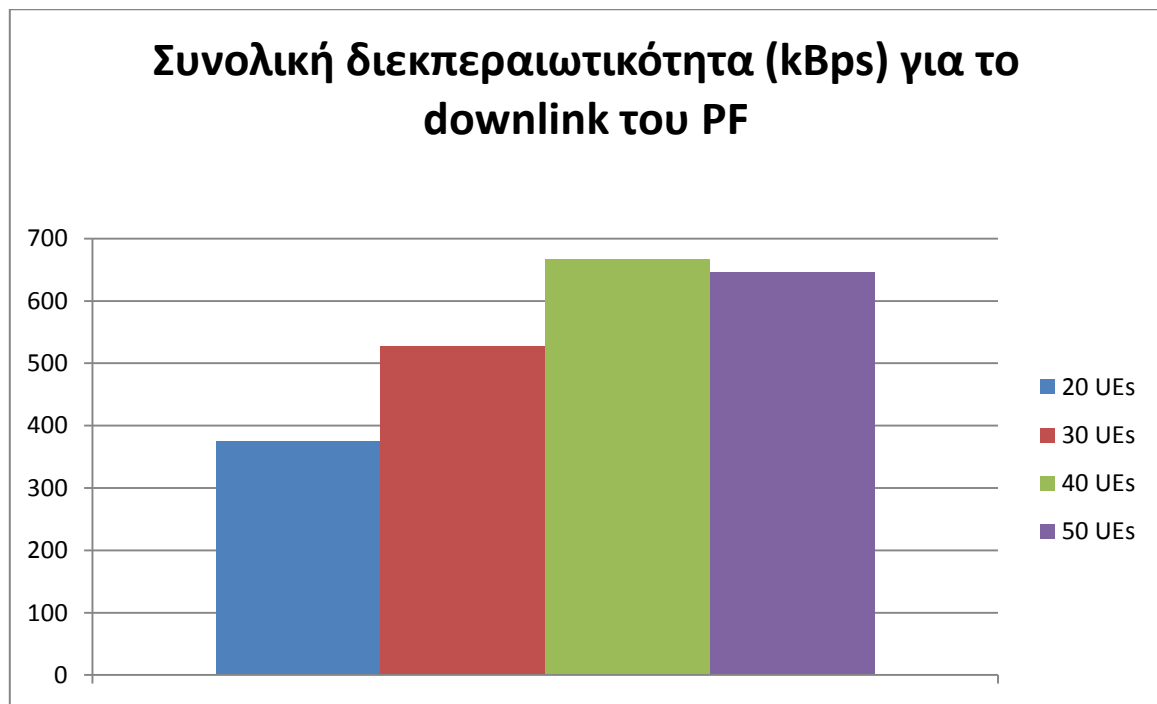
Σχετικά με την μέση καθυστέρηση στην κατερχόμενη ζεύξη, η οποία φαίνεται στην Εικόνα 30, παρατηρείται ότι, καθώς αυξάνεται το πλήθος των χρηστών, μεγαλώνει και η μέση καθυστέρηση του δικτύου, γιατί υπάρχει μεγαλύτερη συμφόρηση σε αυτό. Ωστόσο, στην ανερχόμενη ζεύξη (Εικόνα 31), η μέση καθυστέρηση δεν παρουσιάζει την ομαλή αύξηση που παρατηρείται στην κατερχόμενη. Συγκεκριμένα, η καθυστέρηση στους 40 χρήστες είναι αισθητά μικρότερη σε σχέση με τους 30, αν και παραμένει μικρότερη από την περίπτωση των 50 χρηστών και μεγαλύτερη από αυτής των 20 χρηστών. Τέλος, όπως η απόδοση του δικτύου ήταν καλύτερη για το downlink, σε σχέση με το uplink, έτσι και η μέση καθυστέρηση, για το ίδιο πλήθος χρηστών, είναι μικρότερη στο downlink απ' ό,τι στο uplink.

Στις Εικόνες 32 και 33 φαίνεται το ποσοστό των απωλειών δεδομένων για την κατερχόμενη και την ανερχόμενη ζεύξη, αντίστοιχα. Γενικά, παρατηρείται πως όσο αυξάνονται οι χρήστες, αυξάνονται και οι απώλειες. Στο downlink, το ποσοστό των απωλειών ξεκινάει από περίπου 30%, για τα 20 UEs, και φτάνει σχεδόν μέχρι το 54%, στα 50 UEs. Αντίστοιχα, στο uplink το ποσοστό αυτό ξεκινάει από το 35% και φτάνει μέχρι το 70%. Οι διαφορές στις απώλειες μεταξύ uplink και downlink επιβεβαιώνουν την

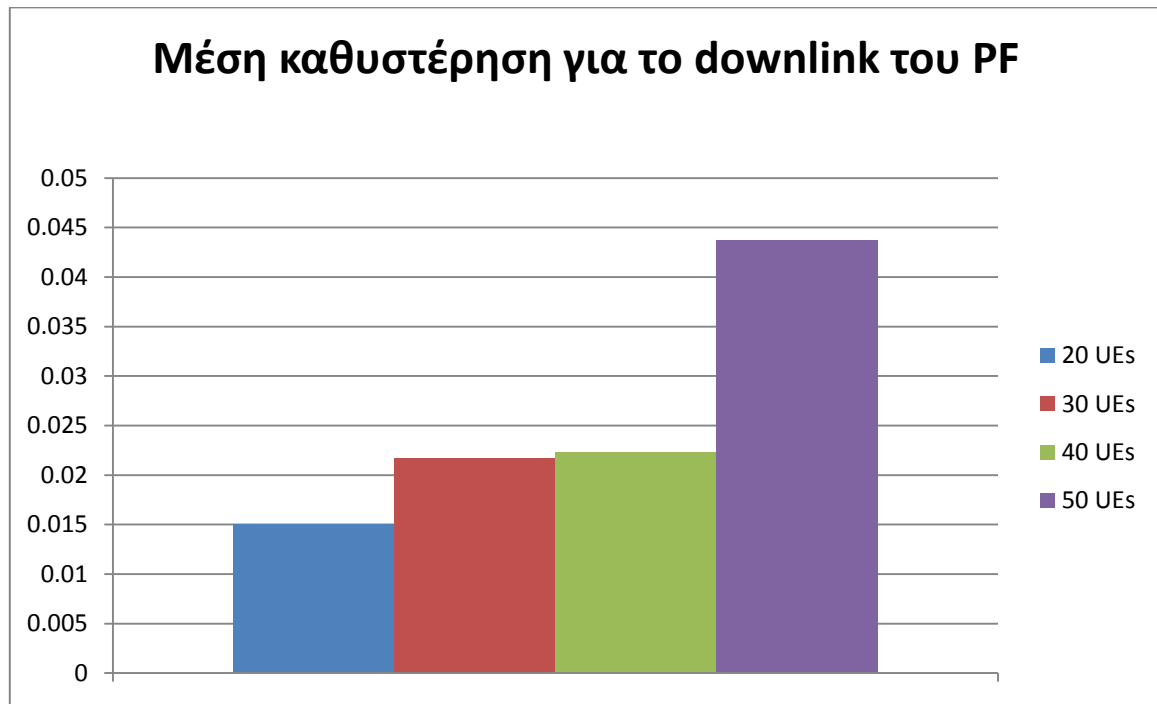
παρατήρηση που κάναμε προηγουμένως, ότι δηλαδή η κατάσταση του δικτύου είναι χειρότερη στην ανερχόμενη ζεύξη, συγκριτικά με την κατερχόμενη. Τέλος, το μόνο αναπάντεχο που παρατηρείται, όσον αφορά το ποσοστό των απωλειών, βρίσκεται στην περίπτωση των 40 χρηστών, για την κατεύθυνση του uplink. Σε αυτήν την περίπτωση, όπως φαίνεται στην Εικόνα 33, το ποσοστό αυτό είναι μόλις 1% μικρότερο από την περίπτωση των 30 χρηστών και όχι μεγαλύτερο, όπως θα περιμέναμε.

4.2.3.2 Proportional Fair

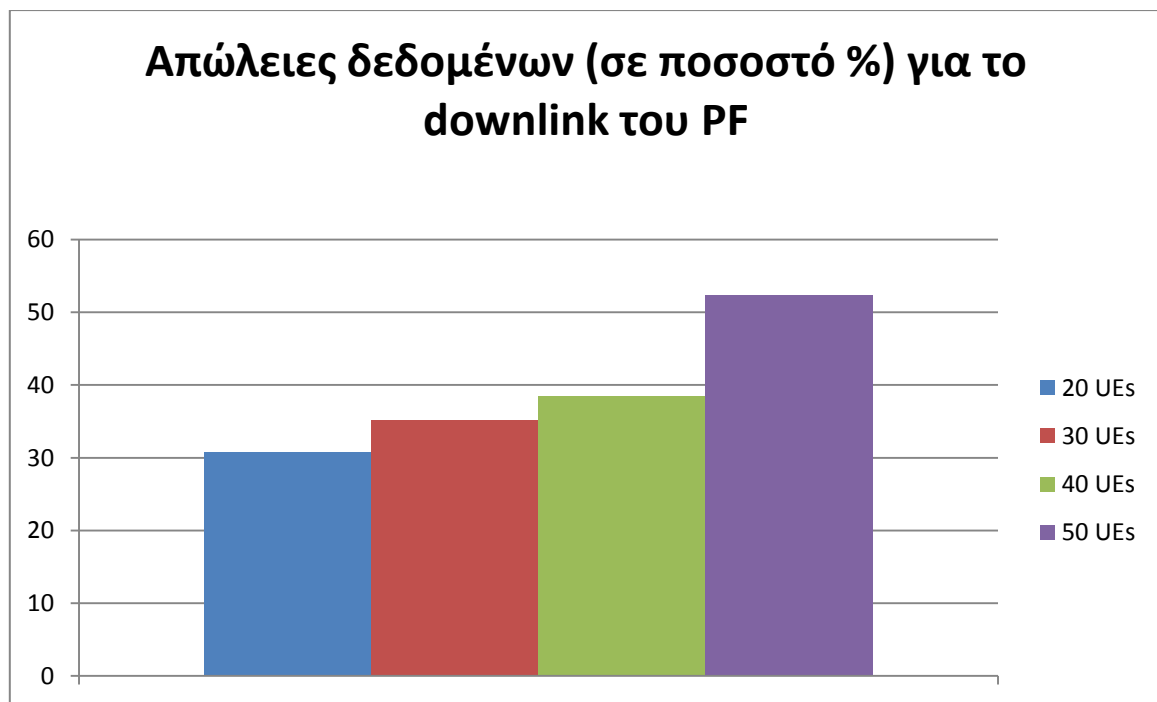
Στις ακόλουθες εικόνες παρουσιάζονται τα αποτελέσματα του Proportional Fair scheduler.



Εικόνα 34: Συνολική διεκπεραιωτικότητα για κάθε σύνολο της κατερχόμενης ζεύξης για τον PF scheduler – Τοπολογία 2



Εικόνα 35: Μέση συνολική καθυστέρηση για κάθε σύνολο της κατερχόμενης ζεύξης για τον PF scheduler – Τοπολογία 2



Εικόνα 36: Συνολικός ρυθμός απωλειών δεδομένων για κάθε σύνολο της κατερχόμενης ζεύξης για τον PF scheduler – Τοπολογία 2

Σχετικά με την συνολική διεκπεραιωτικότητα του συστήματος στο downlink, όπως βλέπουμε και στην Εικόνα 34, παρατηρούμε ότι η διεκπεραιωτικότητα όταν έχουμε 50 χρήστες στο δίκτυο είναι μικρότερη από όταν έχουμε 40 χρήστες, μιας και υπάρχει μεγάλη κίνηση στο δίκτυο οπότε αυξάνονται οι απώλειες και μειώνεται η διεκπεραιωτικότητα (φαίνεται και από το διάγραμμα απωλειών στο downlink, Εικόνα

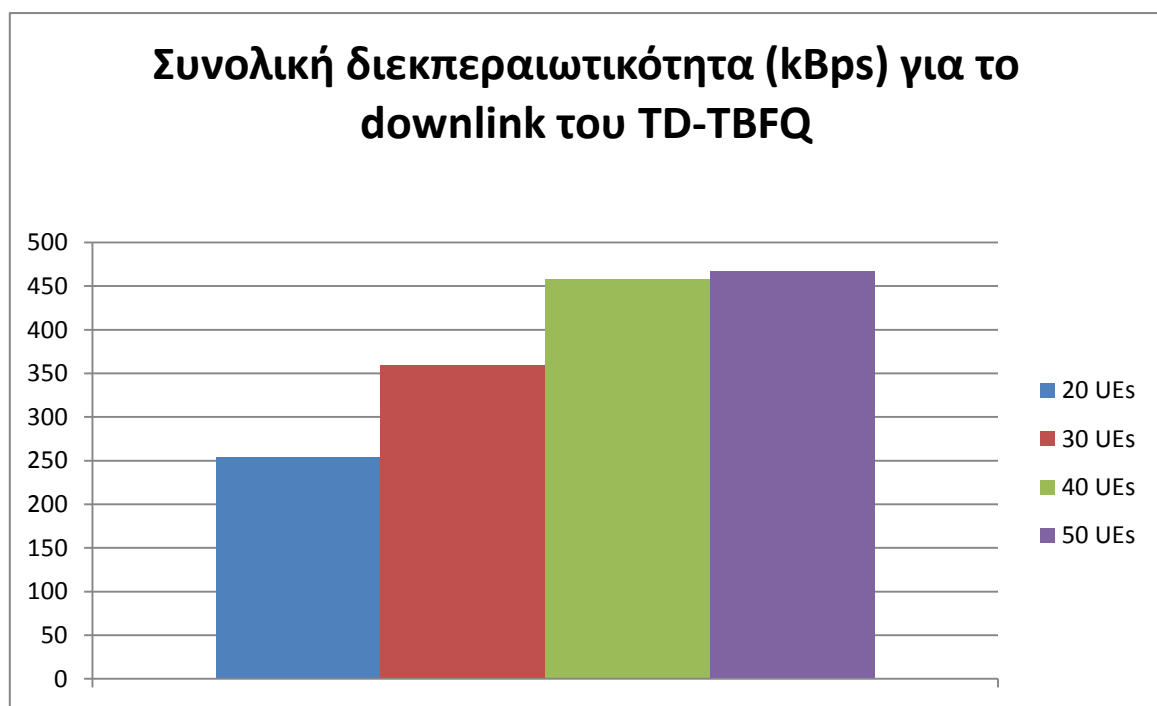
36). Επιπλέον παρατηρούμε ότι η διεκπεραιωτικότητα αυξάνεται καθώς αυξάνονται οι χρήστες, αφού αυξάνονται οι ενεργές εφαρμογές που θα υπάρχουν στο δίκτυο.

Όπως φαίνεται και στην Εικόνα 35, η μέση καθυστέρηση του συστήματος στο downlink, αυξάνεται όσο αυξάνονται οι χρήστες και γίνεται μέγιστη όταν στο δίκτυο υπάρχουν 50 χρήστες, διότι τότε παρουσιάζεται μεγάλη συμφόρηση.

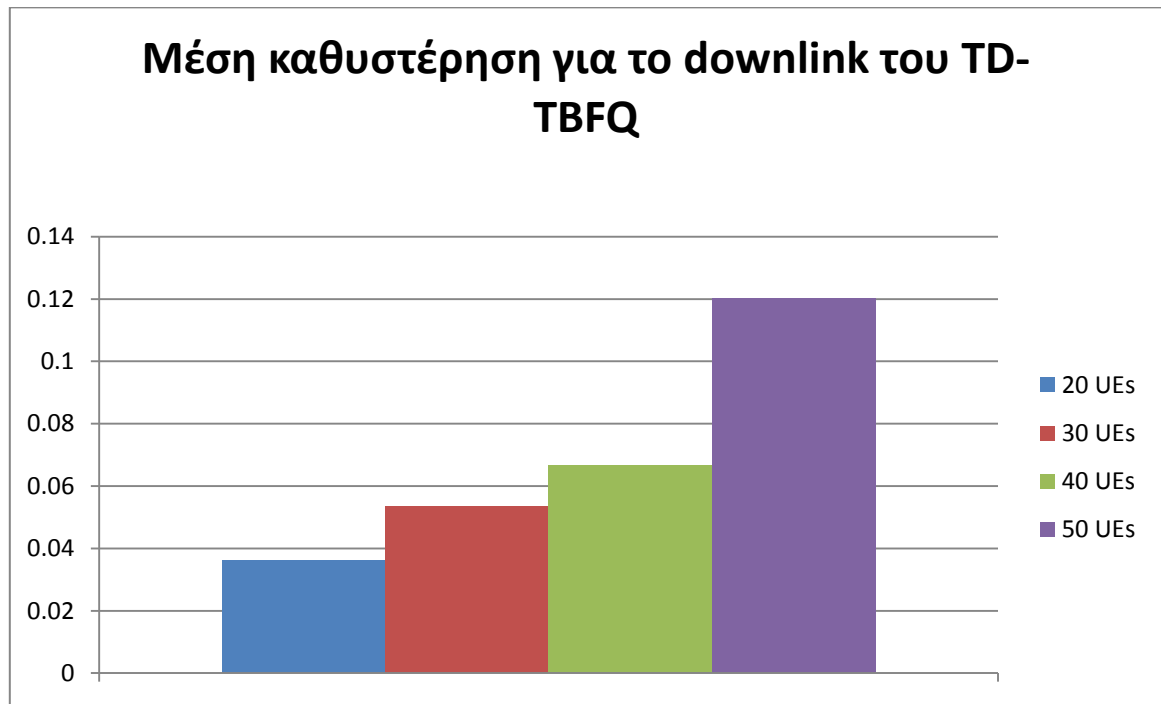
Τέλος, στην Εικόνα 36 παρουσιάζονται οι απώλειες του δικτύου στην κατερχόμενη ζεύξη. Παρατηρούμε ότι στο downlink οι απώλειες αυξάνονται όσο αυξάνεται και ο αριθμός των χρηστών.

4.2.3.3 Time Domain Token Bank Fair Queue

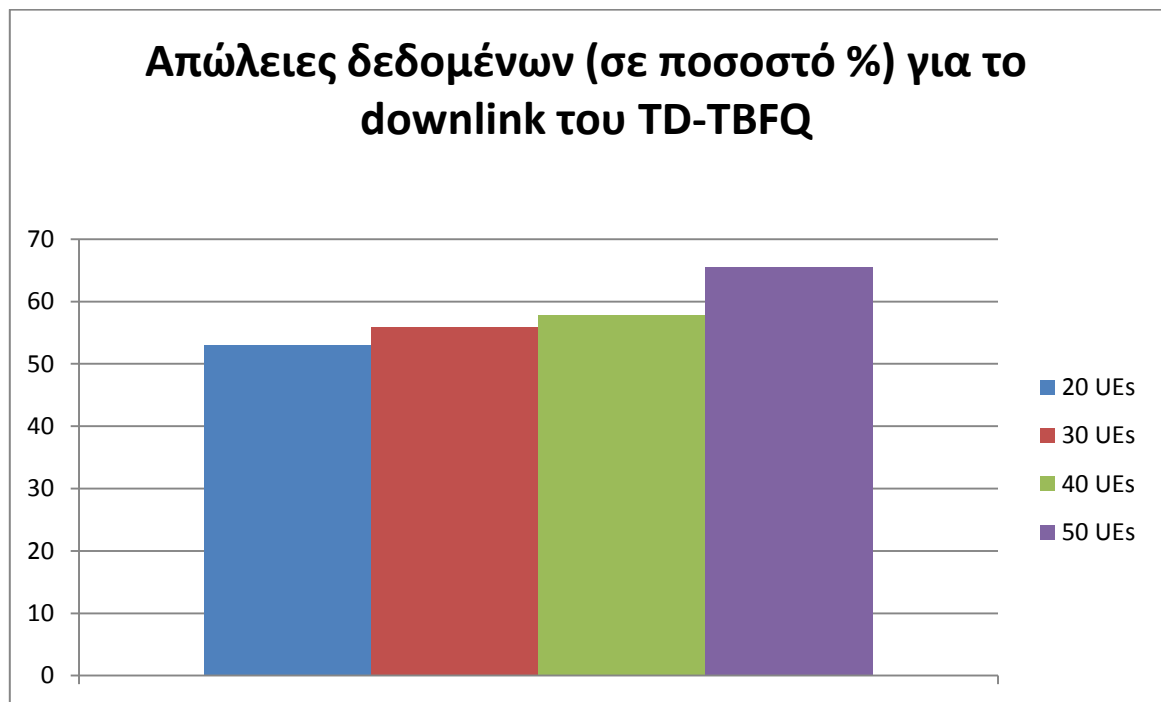
Στις επόμενες εικόνες παρουσιάζονται τα αποτελέσματα του TD-TBFQ αλγορίθμου.



Εικόνα 37: Συνολική διεκπεραιωτικότητα για κάθε σύνολο της κατερχόμενης για τον TD-TBFQ scheduler – Τοπολογία 2



Εικόνα 38: Μέση συνολική καθυστέρηση για κάθε σύνολο της κατερχόμενης για τον TD-TBFQ scheduler – Τοπολογία 2



Εικόνα 39: Συνολικός ρυθμός απωλειών δεδομένων για κάθε σύνολο της κατερχόμενης ζεύξης για τον TD-TBFQ scheduler – Τοπολογία 2

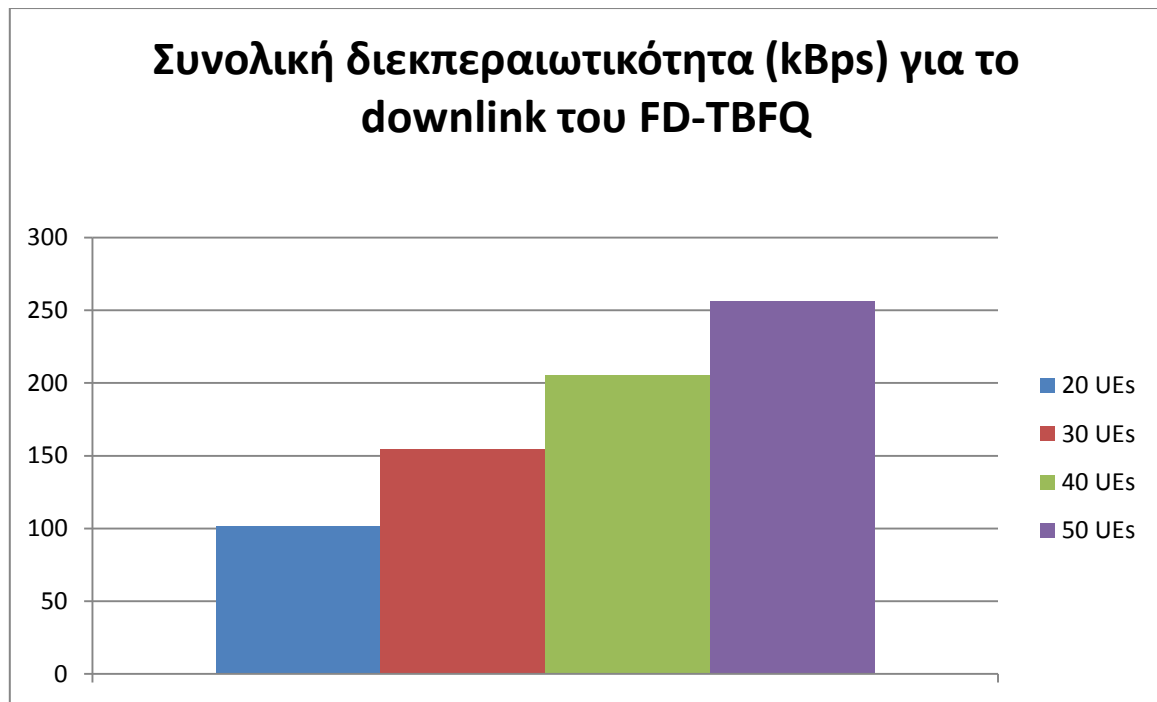
Με την εφαρμογή του αλγορίθμου του TD-TBFQ παρατηρείται ότι η συνολική διεκπεραιωτικότητα του συστήματος για την κατερχόμενη ζεύξη (Εικόνα 37) αυξάνεται καθώς αυξάνεται το πλήθος των UEs και φτάνει μέχρι τα 450 kbps.

Σχετικά με την μέση καθυστέρηση (Εικόνα 38) παρατηρούμε ότι αυτή αυξάνεται καθώς προστίθενται παραπάνω χρήστες στο δίκτυο. Στους 50 χρήστες, η καθυστέρηση φτάνει το 0,12 sec.

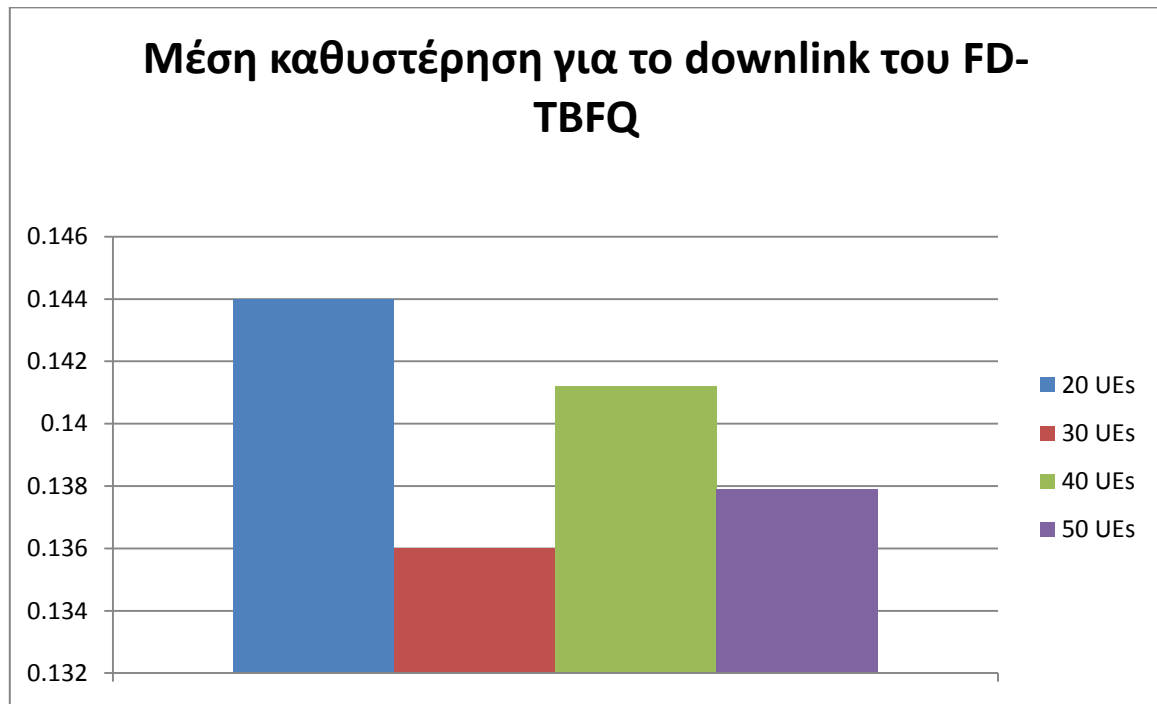
Τα ποσοστά των απωλειών δεδομένων για την κατερχόμενη ζεύξη παρουσιάζονται στην Εικόνα 39. Όπως και με τους προηγούμενους αλγορίθμους, το ποσοστό αυτό αυξάνεται με την αύξηση των χρηστών.

4.2.3.4 Frequency Domain Token Bank Fair Queue

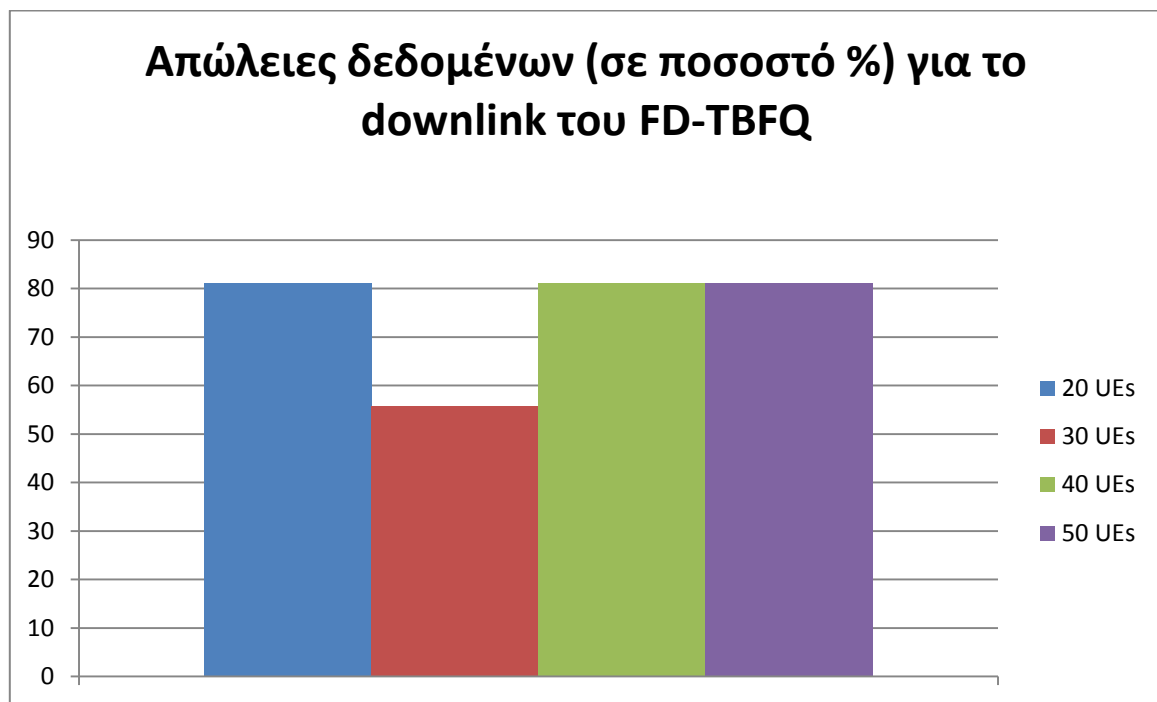
Στα ακόλουθα διαγράμματα παρουσιάζονται τα αποτελέσματα του FD-TBFQ.



Εικόνα 40: : Συνολική διεκπεραιωτικότητα για κάθε σύνολο της κατερχόμενης ζεύξης για τον FD-TBFQ scheduler – Τοπολογία 2



Εικόνα 41: Μέση συνολική καθυστέρηση για κάθε σύνολο της κατερχόμενης ζεύξης για τον FD-TBFQ scheduler – Τοπολογία 2



Εικόνα 42: Συνολικός ρυθμός απωλειών δεδομένων για κάθε σύνολο της κατερχόμενης ζεύξης για τον FD-TBFQ scheduler – Τοπολογία 2

Σχετικά με τη διεκπεραιωτικότητα του συστήματος στο downlink για τον FD-TBFQ (Εικόνα 40), παρατηρούμε ότι αυτή αυξάνεται καθώς αυξάνονται οι χρήστες.

Συγκριτικά με τους άλλους αλγόριθμους, στην τοπολογία 2, ο PF παρουσιάζει την καλύτερη απόδοση στο downlink.

Όσον αφορά την μέση καθυστέρηση του δικτύου στο downlink (Εικόνα 41), αυτή παρουσιάζει περίεργη συμπεριφορά καθώς εμφανίζει μέγιστη τιμή όταν έχουμε 20 χρήστες και ελάχιστη όταν έχουμε 50 χρήστες.

Συγκριτικά με τους άλλους αλγορίθμους, στην τοπολογία 2, ο PF παρουσιάζει την μικρότερη καθυστέρηση στο downlink.

Στην Εικόνα 42 παρουσιάζονται τα ποσοστά απωλειών της κατερχόμενης ζεύξης. Τα ποσοστά απωλειών, στο downlink, είναι περίπου ίδια όταν έχουμε 20,40 και 50 χρήστες, ενώ στους 30 χρήστες εμφανίζεται το μικρότερο ποσοστό απωλειών δεδομένων.

Συγκριτικά με τους άλλους αλγορίθμους, στην τοπολογία 2, ο PF παρουσιάζει καλύτερα ποσοστά απωλειών δεδομένων στο downlink.

5. Συμπεράσματα

Στην παρούσα πτυχιακή μελετήθηκαν οι αλγόριθμοι διαμοιρασμού πόρων και εκτελέστηκαν πειράματα για ένα σύνολο από αυτούς. Τα πειράματα εκτελέστηκαν για δυο διαφορετικές τοπολογίες.

Στην πρώτη τοπολογία, στην περίπτωση ενός εμπορικού κέντρου, αλλάζουμε το πλήθος των χρηστών και των αριθμό των διαθέσιμων πόρων. Παρατηρήσαμε ότι την καλύτερη απόδοση την είχε ο PF scheduler.

Στην δεύτερη τοπολογία, όπου στο δίκτυο υπάρχουν μικροκυψέλες και μακροκυψέλες, αλλάζουμε μόνο το πλήθος των χρηστών ενώ ο αριθμός των διαθέσιμων πόρων παραμένει σταθερός για τις μικροκυψέλες και τις μακροκυψέλες. Παρατηρήσαμε και πάλι ότι ο PF παρουσιάζει καλύτερη απόδοση.

Ο PF έχει τις καλύτερες επιδόσεις διότι σκοπός του είναι η μεγιστοποίηση της συνολικής απόδοσης του δικτύου διατηρώντας δικαιοσύνη απόδοσης πόρων ανάμεσα στους χρήστες του δικτύου, καθώς δεν επιτρέπει κάποιος χρήστης να λιμοκτονήσει από μη ανάθεση πόρων. Έτσι ο PF είναι πιο δίκαιος από τον RR και πιο απλός από τον TBFQ. Συνεπώς τα αποτελέσματά μας ήταν αναμενόμενα. Ωστόσο, ο αλγόριθμος του TBFQ εγγυάται μια ελάχιστη απόδοση για τους χρήστες, η οποία καθορίζεται από τον τύπο του bearer του. Έτσι, θα ήταν προτιμότερο να ορίσουμε διαφορετικό τύπο bearer ανάλογα με την εφαρμογή που εκτελείται. Άρα, για τις εφαρμογές που έχουν υψηλότερο ρυθμό δεδομένων θα ορίζεται μια ελάχιστη απόδοση μεγαλύτερη από την αντίστοιχη απόδοση που ορίζεται για εφαρμογές με μικρότερο ρυθμό. Αυτό θα έχει ως αποτέλεσμα ο TBFQ να λειτουργεί πιο αποδοτικά και να παρουσιάζει καλύτερες επιδόσεις.

Ως ανοιχτό πεδίο αφήνεται η πειραματική δοκιμή των υπόλοιπων αλγορίθμων, ώστε να μπορούμε να έχουμε πληρέστερη εικόνα της απόδοσής των αλγορίθμων scheduling που υπάρχουν στον προσομοιωτή NS-3.

ΠΙΝΑΚΑΣ ΟΡΟΛΟΓΙΑΣ

Ξενόγλωσσος όρος	Ελληνικός Όρος
Quality of service	Ποιότητα υπηρεσίας
Latency	Χρόνος ανταπόκρισης
Downlink	Κατερχόμενη ζεύξη
Uplink	Ανερχόμενη ζεύξη
Interface	Διεπαφή
Handover	Μεταβίβαση
Bearer	Φορέας
Inter-symbol interference	Δια-συμβολική παρεμβολή
Carrier	Φέρον
Chunk	Κομμάτι, τμήμα
Cyclic Prefix	Κυκλικό Πρόθεμα
Identifier	Προσδιοριστής
Getaway	Πύλη εξόδου
Scheduler	Χρονοπρογραμματιστής
Multipath	Πολλαπλές διαδρομές
Buffer	Καταχωρητής
Bandwidth	Εύρος ζώνης
Multiuser Diversity	Διαφορικότητα με πολλαπλούς χρήστες
Channel Dependent Scheduling	Χρονοπρογραμματισμός εξαρτώμενος από το κανάλι
Subband	Υποζώνη
Wideband	Ευρεία ζώνη
Orthogonal Frequency-Division Multiple Access	Πολλαπλή πρόσβαση με ορθογώνια διαίρεση συχνότητας
Radio resource allocation	Ανάθεση Ραδιοπόρων
Resource Block	Τμήμα με πόρους
Starvation	Λιμοκτονία
Token Bank	Τράπεζα Κουπονιών
Dept limit	Όριο οφειλών
Threshold	Κατώφλι
Second	Δευτερόλεπτο
Default	Προεπιλεγμένος
Subframe	Υποπλαίσιο
Credit limit	Όριο πίστωσης
Credit threshold	Κατώφλι πίστωσης
Allocation	Κατανομή
Socket	Υποδοχέας
Point-to-point	Σημείο προς σημείο
Indicator	Δείκτης
Application	Εφαρμογή
Throughput	Απόδοση
Femtocell	Φεμτοκυψέλη
Remote host	Απομακρυσμένος κεντρικός υπολογιστής
Macrocell	Μακροκυψέλη
Module	Στοιχείο

ΣΥΝΤΜΗΣΕΙΣ – ΑΡΚΤΙΚΟΛΕΞΑ – ΑΚΡΩΝΥΜΙΑ

QoS	Quality of Service
LTE	Long Term Evolution
3GPP	3 rd Generation Partnership Project
4G	4 th Generation
IP	Internet Protocol
OFDMA	Orthogonal Frequency – Division Multiple Access
SC-FDMA	Single Carrier Frequency Division Multiple Access
MIMO	Multiple Input Multiple Output
QPSK	Quadrature Phase Shift Keying
QAM	Quadrature Amplitude Modulation
EPS	Evolved Packet System
UMTS	Universal Mobile Telecommunications System
EUTRAN	Evolved UMTS Terrestrial Radio Access Network
EPC	Evolved Packet Core
eNB	Evolved Node B
RNC	Radio Network Controller
MME	Mobility Management Entity
S-GW	Serving Gateway
P-GW	Packet data network Gateway
PDCCP	Packet Data Convergence Protocol
RLC	Radio Link Control
PDU	Protocol Data Unit
ARQ	Automatic Repeat reQuest
HARQ	Hybrid ARQ
MAC	Medium Access Control
PHY	Physical Layer
DS-CDMA	Direct – Sequence Code Division Multiple Access
GPRS	General Packet Radio Service
GTP-U	GPRS Tunneling Protocol – User interface
UDP	Unit Data Protocol
TCP	Transmission Control Protocol
ATM	Asynchronous Transfer Mode

PAPR	Peak to Average Power Ratio
PRB	Physical Resource Block
RB	Resource Block
MBSFN	Multimedia Broadcast multicast Service Single Frequency Network
CP	Cyclic Prefix
TTI	Transmission Time Interval
GBR	Guaranteed Bit Rate
VoIP	Voice over IP
TD	Time Domain
FD	Frequency Domain
SINR	Signal to Interference And Noise Ratio
CDS	Channel Dependent Scheduling
CQI	Channel Quality Indication
UE	User Equipment
PDCCH	Physical Downlink Control Channel
RR	Round Robin
PF	Proportional Fair
BET	Blind Equal Throughput
TTA	Throughput To Average
MT	Maximum Throughput
PSS	Priority Set Scheduler
TBFQ	Token Bank Fair Queue
MCS	Modulation And Code Scheme
RRA	Radio Resource Allocation
SNR	Signal to Noise Ratio
FDD	Frequency Division Duplex
SISO	Single Input Single Output
CBR	Constant Bit Rate
VBR	Variable Bit Rate
DCI	Downlink Control Information
MCSI	Modulation and Coding Scheme Index
TBS	Transport Block Size
TDS	Time Domain Scheduling
FDS	Frequency Domain Scheduling

MR	Maximum Rate
TBR	Target Bit Rate
TDPS	Time Domain Packet Scheduler
FDPS	Frequency Domain Packet Scheduler
RBG	Resource Block Group
QCI	QoS Class Identifier
SRS	Sounding Reference Signal
PUSCH	Physical Uplink Shared Channel
GPLv2	General Public License version 2
GSoC	Google Summer of Code
RRM	Radio Resource Management
RRC	Radio Resource Control
REM	Radio Environment Map
EARFCN	E-UTRA Absolute Radio Frequency Channel Number
PDSCH	Physical Downlink Shared Channel
KB	Kilo Byte
GB	Giga Byte
Gb	Giga bit
MTU	MAC-level Maximum Transmission Unit
Kbps	Kilo bit per second
KBps	Kilo Byte per second
LoS	Line of Sight
NLoS	Non Line of Sight

ΑΝΑΦΟΡΕΣ

- [1] Yangyang Dong, "LTE MAC Scheduler Investigation", master mini project, Dept. Communication Networks, University of Bremen, Sept. 2011.
- [2] Narges Shojaedin, "TCP-Aware Scheduling in LTE Networks, thesis, Dept. Computer Science, University of Calgary, Alberta, 2014.
- [3] I. Gutierrez and F. Bader, *Scheduling and resource allocation*, Bentham Science Publishers Ltd, 2011, p. 8.
- [4] M.T. Kawser, H.M.A.B. Farid, A.R. Hasin, A.M.J. Sadik, and I.K. Razu, "Performance Comparison between Round Robin and Proportional Fair Scheduling Methods for LTE", *International Journal of Information and Engineering*, vol. 2, no. 5, Sept. 2012.
- [5] O. Iosif, and I. Banica, "On the Analysis of Packet Scheduling in Downlink", *The Fourth International Conference On Communication Theory, Reliability, and Quality Of Service (CTRQ 2011)*
- [6] F. Capozzi, G. Piro, L.A. Grieco, G. Boggia, and P. Camarda, Downlink Packet Scheduling in LTE Cellular Networks: Key Design Issues and a Survey, Dept. Electrical and Electronics, Bari polytechnic Inst., pp.17-23.
- [7] E. Dahlman, S. Parkvall, and J. Skold, 4G LTE / LTE-Advanced for Mobile Broadband, *Elsevier Ltd*, 2011, pp.100.
- [8] D. Zhou, N. Baldo, and M. Miozzo, *Implementation and Validation of LTE Downlink Schedulers for ns-3*, WNS3, In Proc. Int. ICST Conf. On Simulation Tools and Techniques, March 2013.
- [9] J.F. Kurose and K.W. Ross, *Δικτύωση Υπολογιστών – Προσέγγιση από Πάνω προς τα Κάτω*, Μ. Γκιούρδας, 4^η Έκδοση, Αθήνα, 2011, p. 649 – 651.
- [10] G. Indumathi, and K. Murugesan, ed., *Across-Layer Resource Scheduling with QoS Guarantees using Adaptive Token Bank Fair Queuing Algorithm in Wireless Networks*, Journal of Engineering Science and Technology, vol. 6, no 3, 2011
- [11] Carlos Fon Cuesta, Resource Allocation and Time-Frequency Scheduling in LTE, Seminar Mobile Communication, TUWEIN, Dec. 2007
- [12] Aditya Tiwari, Shashikant Suman, Pranav Viswanathan, LTE Protocol, "Verification of MAC Scheduling algorithms in NetSim™", 2014, p. 4.
- [13] Yaser Barayan and Ivica Kostanic, "Performance Evaluation of Proportional Fairness Scheduling in LTE", San Francisco, October 2013.
- [14] Yueming Cai, Jiang Yu, Youyun Xu, Mulin Cai, "A comparison of packet scheduling algorithms for OFDMA systems", *The second International Conference On Signal Processing and Communication Systems, ICSPCS*, vol., no., December 2008, p. 1-5.
- [15] T. Dikamba, "Downlink Scheduling in 3GPP Long Term Evolution (LTE)", MSc Thesis TUDelft, Delft University of Technology, March 2011.
- [16] D. Skoutas, A. Rouskas, "Scheduling with qos provisioning in mobile broadband wireless systems", *Wireless Conference (EW)*, April 2010, p. 422-428.
- [17] V. Conan, F. Rebecchi, R. Bruno, A. Passarella, E. Biondi, C. Boldrini, G. Mainetto, M. Dias de Amorim, E. Zeydan, A. S. Tan, E. Pierattelli and D. Azzarelli, "Design and evaluation of enabling techniques for mobile data traffic offloading, *WP3 – Offloading foundations and enablers*, vol. 2, Apr. 2015.
- [18] Konstantinos Katsaros, *Introduction to ns-3*, Center for Communication System Research, University of Surrey, 2013.
- [19] G. Piro, N. Baldo, and M. Miozzo, "An LTE module for ns-3 network simulator, Bari polytechnic Inst., GSoC, 2010.
- [20] Ns3 page, "ns3::RrFfMacScheduler Class Reference", https://www.nsnam.org/doxygen/classns3_1_1_rr_ff_mac_scheduler.html [Προσπελάστηκε 19/8/15]
- [21] "Design Documentation", <https://www.nsnam.org/docs/release/3.21/models/html/lte-design.html> [Προσπελάστηκε 19/8/15]
- [22] Ns3 page, "ns3::TdBetFfMacScheduler Class Reference", https://www.nsnam.org/doxygen/classns3_1_1_td_bet_ff_mac_scheduler.html [Προσπελάστηκε 20/8/15]
- [23] Ns3 page, "ns3::FdBetFfMacScheduler Class Reference", https://www.nsnam.org/doxygen/classns3_1_1_fd_bet_ff_mac_scheduler.html [Προσπελάστηκε 21/8/15]
- [24] Ns3 page, "ns3::TdTbfqFfMacScheduler Class Reference", https://www.nsnam.org/doxygen/classns3_1_1_td_tbfq_ff_mac_scheduler.html [Προσπελάστηκε 21/8/15]
- [25] Ns3 page, "ns3::FdTbfqFfMacScheduler Class Reference", https://www.nsnam.org/doxygen/classns3_1_1_fd_tbfq_ff_mac_scheduler.html [Προσπελάστηκε 21/8/15]

- [26] Ns3 page, "ns3::PffMacScheduler Class Reference",
https://www.nsnam.org/doxygen/classns3_1_1_pf_ff_mac_scheduler.html [Προσπελάστηκε
Αύγουστος 2015]
- [27] Ns3 page, "ns3::PssFfMacScheduler Class Reference",
https://www.nsnam.org/doxygen/classns3_1_1_pss_ff_mac_scheduler.html#details [Προσπελάστηκε
Αύγουστος 2015]
- [28] Ns3 page, "ns3::TdMtFfMacScheduler Class Reference",
https://www.nsnam.org/doxygen/classns3_1_1_td_mt_ff_mac_scheduler.html#details
[Προσπελάστηκε Αύγουστος 2015]
- [29] Ns3 page, "ns3::FdMtFfMacScheduler Class Reference",
https://www.nsnam.org/doxygen/classns3_1_1_fd_mt_ff_mac_scheduler.html#details
[Προσπελάστηκε Αύγουστος 2015]
- [30] Ns3 page, "ns3::TtaFfMacScheduler Class Reference",
https://www.nsnam.org/doxygen/classns3_1_1_tta_ff_mac_scheduler.html#details [Προσπελάστηκε
Αύγουστος 2015]
- [31] Ns3 page, "ns3::Building Class Reference",
https://www.nsnam.org/doxygen/classns3_1_1_building.html [Προσπελάστηκε 1/9/2015]
- [32] Ns3 page, "building.cc", https://www.nsnam.org/doxygen/building_8cc_source.html#l00039
[Προσπελάστηκε 1/9/2015]
- [33] J. Hamalainen, "Femtocells: Technology and Developments", Wireless Information Theory Summer School, Centre for Wireless Communications, Aalto University, 2011, p. 42.
- [34] Ns3 page, "ns3::RadioEnvironmentMapHelper Class Reference",
https://www.nsnam.org/doxygen/classns3_1_1_radio_environment_map_helper.html
[Προσπελάστηκε 1/9/2015]
- [35] Ns3 page, "User Documentation", <https://www.nsnam.org/docs/release/3.21/models/html/lte-user.html>
[Προσπελάστηκε 1/9/2015]
- [36] Ns3 page, "lena-dual-stripe.cc", https://www.nsnam.org/doxygen/lena-dual-stripe_8cc_source.html
[Προσπελάστηκε 1/9/2015]
- [37] Ns3 page, "ns3::UdpClientHelper Class Reference",
https://www.nsnam.org/doxygen/classns3_1_1_udp_client_helper.html [Προσπελάστηκε 1/9/2015]
- [38] B. Theiss, "The Great Mobile Network Test", *Focus*, connect, 2014, pp. 30.
- [39] Infographic, "How much bandwidth does streaming use?" Jan. 2013,
<http://www.theglobeandmail.com/technology/tech-news/how-much-bandwidth-does-streaming-use/article7365916> [Προσπελάστηκε 1/9/2015]
- [40] L. Korowajczuk, "How to Dimension User Traffic in 4G Networks" May 2014,
<http://www.slideshare.net/alhafhussain1023/how-to-dimension-user-traffic-in-lte> [Προσπελάστηκε
1/9/2015]
- [41] Ns3 page, "ns3::EpsBearer Struct Reference",
https://www.nsnam.org/doxygen/structns3_1_1_eps_bearer.html [Προσπελάστηκε 1/9/2015]
- [42] Ns3 page, "ns3::GridBuildingAllocator Class Reference",
https://www.nsnam.org/docs/release/3.18/doxygen/classns3_1_1_grid_building_allocator.html
[Προσπελάστηκε 1/10/2015]
- [43] Ns3 page, "ns3::HybridBuildingsPropagationLossModel Class Reference",
https://www.nsnam.org/docs/release/3.18/doxygen/classns3_1_1_hybrid_buildings_propagation_loss_model.html [Προσπελάστηκε 2/10/2015]
- [44] Ns3 page, "ns3::MultiModelSpectrumChannel Class Reference",
https://www.nsnam.org/doxygen/classns3_1_1_multi_model_spectrum_channel.html
[Προσπελάστηκε 2/10/2015]