



Εθνικό Μετσόβιο Πολυτεχνείο

Σχολή Ηλεκτρολόγων Μηχανικών και

Μηχανικών Υπολογιστών

Τομέας Επικοινωνιών, Ηλεκτρονικής και

Συστημάτων Πληροφορικής

Ανάπτυξη Συστήματος Δυναμικής Δειγματοληψίας Ποιότητας Δικτύων Κινητής Τηλεφωνίας

Διπλωματική Εργασία

Κωτσιανδρής Μάριος

Επιβλέπων: Μ. Θεολόγου

Καθηγητής

ΑΘΗΝΑ, ΣΕΠΤΕΜΒΡΙΟΣ 2011



Εθνικό Μετσόβιο Πολυτεχνείο

Σχολή Ηλεκτρολόγων Μηχανικών και

Μηχανικών Υπολογιστών

Τομέας Επικοινωνιών, Ηλεκτρονικής και

Συστημάτων Πληροφορικής

Ανάπτυξη Συστήματος Δυναμικής Δειγματοληψίας Ποιότητας Δικτύων Κινητής Τηλεφωνίας

Διπλωματική Εργασία

Κωτσιανδρής Μάριος

Επιβλέπων: Μιχαήλ Θεολόγου

Καθηγητής

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή:

.....
Μιχαήλ Θεολόγου
Καθηγητής Ε.Μ.Π

.....
Ευστάθιος Συκάς
Καθηγητής Ε.Μ.Π

.....
Στασινόπουλος Γεωργιος
Καθηγητής Ε.Μ.Π

.....
Κωτσιανδρής Μάριος

Διπλωματούχος Ηλεκτρολόγος Μηχανικός και Μηχανικός Υπολογιστών Ε.Μ.Π

Copyright © Μάριος Π. Κωτσιανδρής, 2011

Με επιφύλαξη παντός δικαιώματος. All rights reserved.

Απαγορεύεται η αντιγραφή, αποθήκευση και διανομή της παρούσας εργασίας, εξ' ολοκλήρου ή τμήματος αυτής, για εμπορικό σκοπό. Επιτρέπεται η ανατύπωση, αποθήκευση και διανομή για σκοπό μη κερδοσκοπικό, εκπαιδευτικής ή ερευνητικής φύσης, υπό την προϋπόθεση να αναφέρεται η πηγή προέλευσης και να διατηρείται το παρόν μήνυμα. Ερωτήματα που αφορούν τη χρήση της εργασίας για κερδοσκοπικό σκοπό πρέπει να απευθύνονται προς τον συγγραφέα.

Οι απόψεις και τα συμπεράσματα που περιέχονται σε αυτό το έγγραφο εκφράζουν τον συγγραφέα και δεν πρέπει να ερμηνευθεί ότι αντιπροσωπεύουν τις επίσημες θέσεις του Εθνικού Μετσόβιου Πολυτεχνείου.

Αφιερώνεται στην οικογένειά μου

Κωτσιανδρής Μάριος

Aut viam inveniam aut faciam.

Πίνακας περιεχομένων

Πίνακας περιεχομένων	8
Περίληψη.....	12
Abstract	13
1. Γενικά για την Ποιότητα Υπηρεσιών.....	14
1.1. Εισαγωγή.....	14
1.2. Βασικά χαρακτηριστικά της κίνησης.....	15
1.3. Ποιότητα Υπηρεσιών Ασύρματων και Κινητών Δικτύων	16
2. Ποιότητα Υπηρεσιών σε Κυψελωτά Δίκτυα.....	18
2.1. Εισαγωγή.....	18
2.2. Ορισμός	18
2.2.1. Γιατί χρειαζόμαστε την ποιότητα υπηρεσιών.....	19
2.2.2. Προκλήσεις Ποιότητας Υπηρεσίας.....	19
2.2.3. Τι έχει γίνει μέχρι τώρα στον τομέα της Ποιότητας Υπηρεσίας	20
2.3. Ποιότητα Υπηρεσιών σε Δίκτυο Τρίτης Γενιάς – Ανάλυση Μηχανισμών και Χαρακτηριστικών.....	21
2.3.2. Απ’ άκρη σ’ άκρη υπηρεσίες και υπηρεσία μεταφοράς UMTS.....	21
2.3.3. Υπηρεσία Μεταφοράς Ράδιο-Πρόσβασης και Υπηρεσία Μεταφοράς Πυρήνα Δικτύου	22
2.3.4. Radio Bearer Service και RAN Access Bearer Service	22
2.3.5. Η ραχοκοκαλιά των δικτυακών υπηρεσιών.....	22
2.4. Κλάσεις Ποιότητας Υπηρεσίας δικτύου UMTS και Χαρακτηριστικά τους.....	22
2.4.1. Κλάση Συνομιλίας.....	23
2.4.2. Κλάση Streaming	24
2.4.3. Διαδραστική Κλάση	24
2.4.4. Κλάση Υποβάθρου.....	24
2.4.5. Ιδιότητες Ποιότητας Υπηρεσίας.....	24
2.5. Διαφορετικά Μοντέλα Ποιότητας Υπηρεσίας σε Κυψελωτά Δίκτυα.....	25
2.5.1. Ανεκτικό σε Σφάλματα μοντέλο Δυναμικού Καταμερισμού	26
2.5.2. Μοντέλο Έλεγχου Εισόδου Κλήσεων.....	30
2.5.4. Μοντέλο Δυναμικού Καταμερισμού με την χρήση Επαναδιαπραγμάτευσης.....	33
2.5.5. Ατομική Ποιότητα Υπηρεσίας για Υπηρεσίες Ομιλίας.....	34
2.6. Συνδυασμός Κυψελωτών και Ασύρματων δικτύων για επίτευξη καλύτερης Ποιότητας Υπηρεσίας.....	36
2.6.1. Δομή Μοντέλου Ποιότητας Υπηρεσίας Αυστηρής Διασύνδεσης.....	37

2.6.2.	Χαλαρή Διασύνδεση Κυψελωτού με Ασύρματο Τοπικό Δίκτυο	38
2.7.	Ανάλυση δικτύων 3.5G και 4G.	40
2.7.1.	High Speed Packet Access	41
2.7.2.	3GPP - Long Term Evolution	42
2.7.3.	LTE Advanced.....	48
3.	Μηχανισμοί και Παράμετροι Ποιότητας Υπηρεσίας	49
3.1.	Εισαγωγή.....	49
3.2.	Ταξινόμηση.....	50
3.2.1.	Επίπεδο Ζεύξης Δεδομένων.....	51
3.8.1.	Ομιλία	61
3.8.2.	Βίντεο Τηλεφωνία.....	62
3.8.3.	Βίντεο Συνδιάσκεψη.....	63
3.8.4.	Υπηρεσία Μηνυμάτων Πολυμέσων.....	64
3.8.5.	Εφαρμογές Ροής Πολυμέσων	64
3.8.6.	Λήψη Δεδομένων	65
3.8.7.	Πρόσβαση στο διαδίκτυο και πλοήγηση.....	66
3.8.8.	Εταιρική Πρόσβαση	67
3.9.	Μέθοδοι Μέτρησης Ποιότητας Υπηρεσίας Ομιλίας.....	68
3.9.1.	Ληφθείσα Ποιότητα Υπηρεσίας για Ομιλία πάνω από πρωτόκολλο Διαδικτύου.	68
3.9.2.	Μέθοδοι Εξέτασης.....	68
3.9.3.	Προδιαγραφή Δοκιμής σε Επίπεδο Εφαρμογών.....	69
4.	Σύμφωνα Επιπέδων Εξυπηρέτησης	75
4.1.	Εισαγωγή.....	75
4.2.	Ορισμός και Γενικές Πληροφορίες.....	75
4.2.1.	Ιστορία και Εξέλιξη.....	76
4.2.2.	Νέες Κατευθύνσεις και εφαρμογές.....	77
4.2.3.	Κοινά Μεγέθη Μέτρησης.....	78
4.4.	Οντότητες στα Συμφωνητικά Επιπέδου Εξυπηρέτησης.....	79
4.4.1.	Συστατικά Συμφωνητικών Επιπέδου Υπηρεσιών.....	80
4.4.2.	Αρχιτεκτονική Υπηρεσιών Ιστού	84
4.5.	Προδιαγραφές Επιπέδων Εξυπηρέτησης.....	84
4.5.1.	Υπηρεσίες και Παράμετροι.....	85
4.6.	Σύμφωνα Επιπέδου Εξυπηρέτησης σε Ασύρματα Δίκτυα	87
4.7.	Σύμφωνα Επιπέδου Εξυπηρέτησης και Ποιότητα Υπηρεσίας.....	89

5. Εργαλεία Μέτρησης Ποιότητας Υπηρεσίας	92
5.2. QVoice Symphony	92
5.3. NEMO Analyze	95
5.4. Network Analysis NITRO (NAN)	96
5.5. TEMS Investigation	96
5.6. XCAL	98
5.7. LTE Base Station Emulator - LTE BSE (AWT 700A)	99
5.7.1. Χαρακτηριστικά	99
5.7.2. Σύνοψη Δοκιμής	100
5.8. LTE Air Interface Protocol Analyzer – LTE Air Sniffer	100
5.8.1. Χαρακτηριστικά και Πλεονεκτήματα	101
5.9. RF Parametric Tester – Versatile Subscriber Station Tester (VST)	101
5.9.1. Η Χρήση του VST	101
5.9.2. Βασικά Χαρακτηριστικά	102
5.10. LTE Test Tool	102
5.10.1. Βασικά Χαρακτηριστικά	102
5.11. Handheld XCAL	103
5.11.1. Βασικά Χαρακτηριστικά	103
5.11.2. Βασικές Πληροφορίες	103
5.12. XCAP	104
5.12.1. Βασικά Χαρακτηριστικά	104
5.13. XCAL - MO	105
5.13.1. Βασικά Χαρακτηριστικά	105
5.13.2. Βελτιστοποίηση Δικτύου LTE απ' άκρο σ' άκρο.	105
5.14. Measurement Lab	105
5.15. 6QM	109
5.16. QoSMeT	110
5.17. Network Access Neutrality Observatory	111
6. Android – Περιγραφή Εφαρμογής	113
6.1. Εισαγωγή	113
6.2. Η Αρχιτεκτονική του Android	114
6.2.1. Εφαρμογές	114
6.2.2. Πλαίσιο Εφαρμογών	114
6.2.3. Βιβλιοθήκες	115
6.2.4. Χρόνος Εκτέλεσης του Android	115

6.2.5.	Πυρήνας Linux	116
6.3.	Ιστορική Αναδρομή των Εκδόσεων.....	116
6.4.	Δομή Εφαρμογής.....	117
6.4.1.	Processes και Threads	117
6.4.2.	Applications and Tasks	117
6.5.	Σύστημα Δυναμικής Δειγματοληψίας WhereAmI	118
6.5.1.	Η εφαρμογή WhereAmI	118
6.5.2.	Υλοποίηση του Συστήματος.....	129
5.6.	Διάρθρωση Συστήματος	151
6.	Παράρτημα - Πηγαίος Κώδικας Εφαρμογής.....	156
7.	Βιβλιογραφία - Αναφορές	225

Περίληψη

Σε μια εποχή όπου σημειώνεται ραγδαία εξέλιξη και ανάπτυξη στον τομέα των τηλεπικοινωνιών και ειδικότερα των δικτύων κινητής τηλεφωνίας και των υπηρεσιών που αυτά υποστηρίζουν, οι χρήστες απαιτούν μια συνεχώς αυξανόμενη αλλά και εγγυημένη ποιότητα υπηρεσιών και εξυπηρέτησης. Η μετάβαση από υπηρεσίες απλής φωνητικής επικοινωνίας, σε πολύπλοκες υπηρεσίες ανταλλαγής δεδομένων οδήγησαν τους παρόχους να στρέψουν το ενδιαφέρον τους όχι μόνο στην εξέλιξη, αναβάθμιση και ανάπτυξη των δικτύων τους αλλά και στην συνεχή παρακολούθηση του.

Για το λόγο αυτό, συστήματα παρακολούθησης της συμπεριφοράς αλλά και της κίνησης των δικτύων κινητής τηλεφωνίας έχουν κάνει αισθητή την παρουσία τους τα τελευταία χρόνια κερδίζοντας ολοένα και περισσότερο το ενδιαφέρον της βιομηχανίας κινητής τηλεφωνίας. Ως κύριος στόχος των συστημάτων αυτών είναι η καταγραφή της ποιότητας υπηρεσιών που λαμβάνει ο χρήστης – πελάτης της εταιρείας. Τα αποτελέσματα χρησιμοποιούνται από τους διαχειριστές των δικτύων αλλά και τους παρόχους υπηρεσιών για τη βελτίωση της ποιότητας εμπειρίας των πελατών τους. Έτσι η ανάγκη για ανάπτυξη συστημάτων παρακολούθησης δικτύων κινητής τηλεφωνίας γίνεται όλο και μεγαλύτερη.

Σημαντικός παράγοντας για την ανάπτυξη τέτοιων συστημάτων ήταν η εμφάνιση έξυπνων συσκευών κινητής τηλεφωνίας, γνωστών και ως smart phones και ειδικότερα καινοτόμων λειτουργικών συστημάτων όπως το Android. Το Android είναι ένα λογισμικό ανοιχτού κώδικα, το οποίο υποστηρίζεται πλήρως από την εταιρεία Google. Στο λογισμικό αυτό μπορεί οποιοσδήποτε χρήστης να αναπτύξει τη δική του εφαρμογή σύμφωνα με τις ανάγκες αλλά και τις επιθυμίες του. Βασικό χαρακτηριστικό στοιχείο του Android, που το κάνει ιδανικό για ανάπτυξη συστημάτων καταγραφής κινητών δικτύων είναι η υποστήριξη υπηρεσιών εντοπισμού θέσης. Με τον τρόπο αυτό γίνεται παράλληλα αντιστοίχιση της ποιότητας υπηρεσιών ενός δικτύου με την ακριβή τοποθεσία του κινητού. Έτσι δημιουργείται ένας χάρτης πληροφοριών για την ποιότητα των υπηρεσιών που αντιλαμβάνονται οι χρήστες σε όλη την έκταση του δικτύου.

Στα πλαίσια της συγκεκριμένης διπλωματικής αναπτύχθηκε ένα σύστημα καταγραφής της ποιότητας του δικτύου μιας εταιρείας κινητής τηλεφωνίας. Το σύστημα αυτό επιτρέπει στο χρήστη να εκτελέσει δυναμική δειγματοληψία του δικτύου κινητής τηλεφωνίας τόσο ως προς το χρόνο όσο και ως προς τον χώρο. Επίσης δίνεται η δυνατότητα για δειγματοληψία με βάση την ταχύτητα του κινητού, ενώ παράλληλα το σύστημα επιτρέπει την αποθήκευση των καταγραφών σε βάση δεδομένων καθώς και την κρυπτογράφηση τους για ασφαλή μεταφορά σε απομακρυσμένο server.

Λέξεις Κλειδιά

Ποιότητα Υπηρεσίας, Ποιότητα Δικτύου, Μέτρηση Ισχύος Σήματος, Android, Δυναμική Δειγματοληψία Δικτύου, Σύμφωνα Επιπέδων Εξυπηρέτησης

Abstract

In an era where there is rapid growth and development in telecommunications and especially mobile telephony networks and services they support, users require an ever increasing and guaranteed quality of products and services. The transition from simple voice services, to complex data exchange services led providers to focus their interest not only in evolving, upgrading and developing their networks but also to continuous monitoring.

In the frame of network maintenance, systems that monitor the behavior and traffic of mobile networks have made their presence felt in recent years winning more and more interest in mobile phone industry. The primary goal of these systems is set to recording quality of services received by the user - customer of the mobile phone company. The results are used by network operators and service providers to improve the customer's quality of experience. Thus the need to develop monitoring systems of mobile networks is increasing.

An important factor for the development of such systems was the emergence of intelligent mobile devices known as smart phones and especially innovative operating systems such as Android. The Android is open source software, which is fully supported by the company Google. In this software any user is able to develop his own application in accordance with his needs and desires. A key feature of Android, which makes it ideal for building systems that capture and monitor mobile networks, is the support location based services. Using Location Detection Services in parallel with network monitoring a very important corresponding is created between network quality and location of the mobile terminal. This creates a map full of information on the quality of service perceived by users across the network.

The scope of this diploma thesis is to develop a system able to record the network quality of a mobile phone company. This system allows the user to perform dynamic sampling of mobile network parameters both in time and in space. Also the system allows the user to perform sampling based on the speed of the mobile, while in the same time it allows the storage of records in a database and encrypts them for safe transfer to a remote server.

Key Words

Quality of service, Network Quality, RSSI measurement, Android, Dynamic Network Sampling, Service Level Agreements

1. Γενικά για την Ποιότητα Υπηρεσιών

1.1. Εισαγωγή

Η ερμηνεία του όρου ποιότητα υπηρεσιών λαμβάνει διαφορετικές εκδοχές, αναλόγως το αντικείμενο με το οποίο σχετίζεται ή περιγράφει [5]. Όσο να φο ρά την δικτύωση υπολογιστών, αλλά και γενικά τα τηλεπικοινωνιακά συστήματα, με τον όρο ποιότητα υπηρεσίας ή αλλιώς Quality of Service (QoS), όπως θα την αναφέρουμε συχνά στο κείμενο, περιγράφουμε τους μηχανισμούς ελέγχου για την εξοικονόμηση πόρων, αλλά και την ταυτόχρονη επιτυχημένη παροχή υπηρεσιών. Με τον όρο αυτό, κυρίως αναφερόμαστε στον σχεδιασμό, αλλά και στην υλοποίηση ενός πλάνου που θέτει ως στόχο την εγγυημένη παροχή υπηρεσιών ή συγκεκριμένων προδιαγραφών λειτουργίας, για μια υπηρεσία ή εφαρμογή. Επίσης, αναφερόμαστε και στους μηχανισμούς με τους οποίους πραγματοποιείται το πλάνο αυτό, όπως είναι οι μηχανισμοί για τον καθορισμό προτεραιότητας ανάμεσα σε διαφορετικές ροές δεδομένων, διαφορετικούς χρήστες, αλλά και διαφορετικές εφαρμογές.

Αν θέλαμε να αναφέρουμε παραδείγματα παραγόντων που επιδρούν καταλυτικά στον καθορισμό, αλλά και στην εκτίμηση της QoS, θα έπρεπε να αναφέρουμε κυρίως κάποια από τα χαρακτηριστικά μεγέθη τα οποία αναλόγως με τις τιμές τους καθορίζουν αν θα έχουμε μια ασφαλή και ορθή μετάδοση δεδομένων και ως συνέπεια μια ολοκληρωμένη παροχή υπηρεσίας ή όχι. Έτσι ένας απαιτούμενος ρυθμός μετάδοσης δεδομένων, η δικτυακή καθυστέρηση στην οποία υπόκεινται τα πακέτα, η απόρριψη πακέτων δεδομένων, αλλά και άλλα μεγέθη που αναλύουμε στην συνέχεια είναι αυτά τα οποία καθορίζουν το επίπεδο υπηρεσιών που παρέχονται σε έναν πελάτη.

Είναι θεμελιώδες να αναφέρουμε τη μεγάλη διαφορά ανάμεσα σε δίκτυα και υπηρεσίες βέλτιστης προσπάθειας (best effort services or networks) τα οποία δεν είναι εν γένει δυνατό να υποστηρίξουν παροχή ποιότητας υπηρεσίας στην έκταση τους και σε δίκτυα που σχεδιάζονται και κατασκευάζονται με σκοπό να εξυπηρετήσουν και να υποστηρίξουν QoS στους χρήστες τους. Η βασική διαφορά έγκειται στην ύπαρξη μηχανισμών παρακολούθησης, καταγραφής και ελέγχου της δικτυακής κίνησης. Οι μηχανισμοί λειτουργούν με τέτοιο τρόπο, ώστε να μειώνονται στο ελάχιστο οι πιθανότητες για συμφόρηση του δικτύου και κατ' επέκταση για σύγκρουση μεταδιδόμενων πακέτων, αλλά και εξυπηρέτησης και εγκατάστασης νέων συνδέσεων.

Όσον αφορά τον τηλεπικοινωνιακό κλάδο, η υπεύθυνη αρχή για τον καθορισμό αλλά και την συνολική αποτίμηση της ποιότητας υπηρεσιών είναι η Διεθνής Ένωση Τηλεπικοινωνιών (International Telecommunication Union – ITU). Η εκτίμηση της ποιότητας υπηρεσίας μιας σύνδεσης γίνεται κοιτώντας τη σύνδεση σφαιρικά και αποτιμώντας τις απαιτήσεις που έχουν όλα τα συμβαλλόμενα μέλη. Σε πολλές περιπτώσεις, χρησιμοποιείται και ως μετρικό μέγεθος για την ποιότητα των παροχών που θα λάβει ένας πελάτης, και αυτό γιατί η ποιότητα υπηρεσιών καθορίζεται από τη λαμβανόμενη ποιότητα του πελάτη και όχι από την προσφερόμενη ποιότητα του πάροχου. Για παράδειγμα, υπάρχει ένα κάτω όριο εγγυημένης ποιότητας υπηρεσίας που θα πρέπει να λάβει ένας πελάτης, αλλά και ένα άνω όριο το οποίο θα εγγυάται την εξοικονόμηση πόρων, αλλά και την σωστή διαχείριση του δικτύου.

Είναι βασικό να αναφέρουμε, πως η έννοια της ποιότητας υπηρεσίας από την πλευρά των χρηστών και την πλευρά των σχεδιαστών δικτύων αποκλίνει. Από την πλευρά του παρόχου υπηρεσιών διαδικτύου Internet Service Provider (ISP), η QoS προσεγγίζεται σαν την παρουσία και την εξέλιξη μηχανισμών σε επίπεδο συστήματος. Παρ' όλα αυτά δεν είναι και

τόσο βέβαιο πως οι βελτιώσεις αυτές θα είναι ευδιάκριτες και από την πλευρά του χρήστη. Βέβαια από την άλλη πλευρά, η αντίληψη για την ποιότητα υπηρεσίας, που είναι ικανοποιητική ή ανεκτή από έναν χρήστη, διαφέρει από χρήστη σε χρήστη. Για το λόγο αυτό, έχουμε πολλά και διαφορετικά επίπεδα ποιότητας υπηρεσίας, τα οποία και παρέχονται στους χρήστες συνήθως με διαφορετικό οικονομικό αντίκρισμα.

1.2. Βασικά χαρακτηριστικά της κίνησης

Όσον αφορά τα δίκτυα μεταγωγής πακέτου, υπάρχουν πολλοί παράγοντες [1] που διαμορφώνουν την ποιότητα υπηρεσίας για μια σύνδεση [5]. Ταυτόχρονα αποτελούν και τα βασικά χαρακτηριστικά τα οποία βιώνει και η μεταφορά ενός πακέτου, από τον αποστολέα στον παραλήπτη. Αναφορικά θα μπορούσαμε να αναφέρουμε πως τα βασικότερα είναι τα εξής ακόλουθα [9].

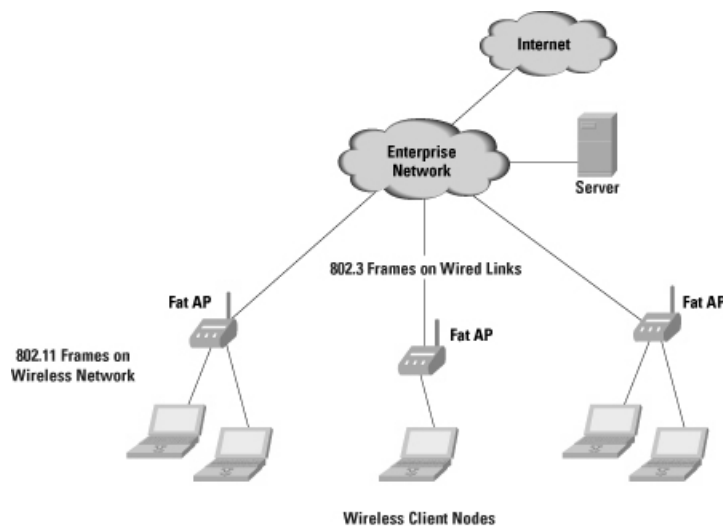
- **Χαμηλή Διέλευση – Low Throughput:** Με τον όρο αυτό αναφερόμαστε πρακτικά στην ευκολία ή στη δυσκολία για την μετάδοση πακέτων μέσα στο δίκτυο. Λόγω της μεγάλης συμμετοχής χρηστών σε ένα δίκτυο και αντίστοιχα και του μεταδιδόμενου φόρτου δεδομένων, είναι συχνό φαινόμενο να αντιστοιχεί πολύ μικρός ρυθμός μετάδοσης bit σε κάθε χρήστη ή σε κάθε υπηρεσία. Το φαινόμενο αυτό με τη σειρά του οδηγεί σε μια ικανοποιητική λειτουργία μιας εφαρμογής και αντίστοιχα σε μια κακής ποιότητας υπηρεσίας σύνδεση.
- **Απώλεια πακέτων – Packet Loss:** Οφείλεται κυρίως στην ανικανότητα ενδιάμεσων δρομολογητών να παραδώσουν ορισμένα πακέτα, αλλά και στην ύπαρξη σφαλμάτων στα δεδομένα πακέτων. Ο υπολογισμός γίνεται θεωρώντας ως χαμένο κάθε πακέτο το οποίο δεν ανήκει σε μια χρονική αντιστοιχία άφιξης σε σχέση με το πακέτο ελέγχου.
- **Διάρκεια Διακοπής Σύνδεσης:** Υπολογίζεται από την διάρκεια των συνεχόμενων χαμένων πακέτων. Έτσι η ανάλυση της διάρκειας διακοπής είναι άμεσα εξαρτημένη από τον ρυθμό μετάδοσης της καταγεγραμμένης ροής.
- **Απόλυτη Διασπορά στην Καθυστέρηση Πακέτων – Absolute Jitter:** Είναι η διαφορά στην καθυστέρηση ανάμεσα σε διαδοχικά πακέτα. Η αιτία για την εμφάνιση της διασποράς οφείλεται κυρίως στην ποικιλομορφία των εμποδίων, που συναντά ένα πακέτο κατά τη διάρκεια της μεταφοράς του από τον αποστολέα στον παραλήπτη. Χαρακτηριστικό παράδειγμα είναι η διαφορετική θέση σε ουρές ενδιάμεσων δρομολογητών που λαμβάνει κάθε πακέτο, αλλά και το διαφορετικό φορτίο το οποίο θα συναντήσει στο δίκτυο.
- **Καθυστέρηση – Delay:** Ίσως ένα από τα σημαντικότερα μεγέθη που παρουσιάζεται σε μία σύνδεση. Συμβαίνει συνήθως σε περιπτώσεις υψηλού φόρτου δικτύου και μπορεί να δημιουργηθεί υπό προϋποθέσεις ακόμη και σε περιπτώσεις υψηλής διέλευσης. Λόγω του χαρακτήρα των υπηρεσιών που καλείται ένα δίκτυο να υποστηρίξει, είναι βασική η μείωση όσο το δυνατό της καθυστέρησης παράδοσης πακέτων. Πολλές υπηρεσίες, όπως η Voice over Internet Protocol - VoIP που θα αναφέρουμε και στη συνέχεια, είναι εξαιρετικά ευαίσθητες στην καθυστέρηση που θα υποστούν τα μεταδιδόμενα πακέτα.

1.3. Ποιότητα Υπηρεσιών Ασύρματων και Κινητών Δικτύων

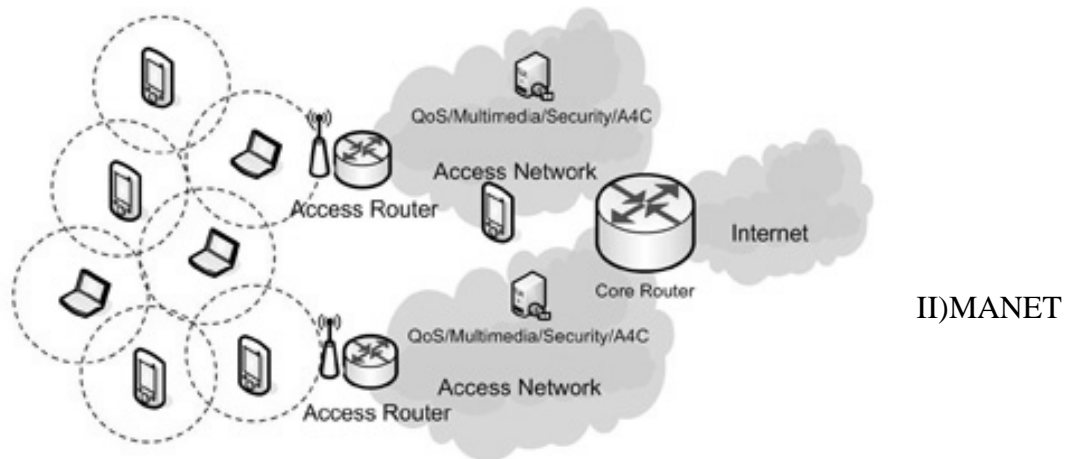
Από την εμφάνισή τους το 1970, τα κινητά ασύρματα δίκτυα έχουν γίνει όλο και περισσότερο δημοφιλή στην βιομηχανία των υπολογιστών. Αυτό έχει γίνει ιδιαίτερα εμφανές την περασμένη δεκαετία, στην οποία τα ασύρματα δίκτυα προσαρμόστηκαν στην επιτρεπτή κινητικότητα των κόμβων.

Υπάρχουν δύο είδη ασύρματων δικτύων. Το πρώτο είναι γνωστό σαν δομημένο δίκτυο, ένα δίκτυο με σταθερές και συνδεδεμένες, με καλώδιο, πύλες. Οι γέφυρες (bridges) αυτών των δικτύων είναι γνωστές σαν σταθμοί βάσης (base stations). Η κινητή μονάδα αυτών δικτύων συνδέεται και επικοινωνεί με το κοντινότερο σταθμό βάσης, που βρίσκεται μέσα στην ακτίνα επικοινωνίας του. Καθώς το κινητό μεταφέρεται από το εύρος ενός σταθμού, στο εύρος ενός άλλου μεταφέρεται και η σύνδεση από τον παλιό σταθμό στον καινούριο, με αποτέλεσμα το κινητό να συνεχίζει να επικοινωνεί, όμοια με πριν, μέσα στο δίκτυο. Τυπική εφαρμογή αυτών των δικτύων παρατηρούμε στα ασύρματα τοπικά δίκτυα (WLANs).

Το δεύτερο είδος το ασύρματο κινητού δικτύου είναι το μη-δομημένο ασύρματο (Infrastructure less) δίκτυο, γνωστό και ως ασύρματο κινητό δίκτυο ad-hoc (wireless mobile ad-hoc network, MANET). Τα Ad-hoc δίκτυα δεν έχουν κανένα σταθερό δρομολογητή, όλοι οι κόμβοι είναι ικανοί να κινηθούν και να συνδεθούν δυναμικά με έναν αυθαίρετο τρόπο. Οι κόμβοι αυτών των δικτύων λειτουργούν σαν δρομολογητές, οι οποίοι ανακαλύπτουν και διατηρούν τη δρομολόγηση τους στους άλλους κόμβους του δικτύου. Παραδείγματα εφαρμογών ενός MANET αποτελούν οι επείγουσες επιχειρήσεις διάσωσης, οι στρατιωτικές επεμβάσεις, καθώς και οι συνεδριάσεις ή συζητήσεις, στις οποίες τα άτομα επιθυμούν να μοιράζονται γρήγορα πληροφορίες.



I) WLAN



Εικόνα 1. Διαφορά ανάμεσα σε Δομημένα Δίκτυα (πχ. WLAN) και MANET.
 Πηγή: www.cisco.com και www.personal.psu.edu

Στο κεφάλαιο που ακολουθεί, θα αναλύσουμε τα κυψελωτά συστήματα και δίκτυα και ειδικότερα θα δούμε τους μηχανισμούς, που βοηθούν στην αύξηση της ποιότητας υπηρεσιών που παρέχονται. Τα κυψελωτά δίκτυα βρίσκουν άμεση εφαρμογή στις κινητές επικοινωνίες, όπως τις γνωρίζουμε σήμερα. Έτσι τα συστήματα αυτά αίρουν το βάρος της ευθύνης, τόσο για ορθή ποιότητα υπηρεσιών ομιλίας όσο και για σωστή και γρήγορη περιήγηση στο Internet.

2. Ποιότητα Υπηρεσιών σε Κυβελωτά Δίκτυα

2.1. Εισαγωγή

Η αύξηση της χρήσης των κινητών τηλεφώνων αλλά αντίστοιχα και των χρηστών κινητής τηλεφωνίας έφερε στο προσκήνιο την ανάγκη για ανάπτυξη υπηρεσιών και εφαρμογών, οι οποίες θα ξεπερνούσαν τα όρια των υπηρεσιών ομιλίας και θα εμπλούτιζαν τα δίκτυα με υπηρεσίες βίντεο, περιήγησης στο διαδίκτυο αλλά και υπηρεσίες αποστολής και λήψης δεδομένων γενικότερα. Για το λόγο αυτό τα δίκτυα κινητής τηλεφωνίας ξεκίνησαν να υποδέχονται νέες τεχνολογίες για την υποστήριξη του συνεχώς αυξανόμενου φορτίου αλλά και απαιτήσεων. Η μετάβαση ανάμεσα σε διαφορετικού τύπου τεχνολογίες χαρακτηρίστηκε ως γενιά. Έτσι ενώ ξεκινήσαμε από δίκτυα που χρησιμοποιούσαν τεχνολογίες πρώτης γενιάς, κυρίως αναλογικά συστήματα (1G), μεταφερθήκαμε σταδιακά σε δίκτυα τεχνολογιών δεύτερης (2G), τρίτης (3G) και εν τέλει τέταρτης γενιάς (4G).

Η ανάπτυξη και εξέλιξη των δικτύων κινητής τηλεφωνίας έδωσε πρόσφορο έδαφος, όπως είπαμε, για τη δημιουργία υπηρεσιών με μεγαλύτερο φόρτο για το δίκτυο αλλά και με μεγαλύτερες απαιτήσεις από αυτό. Σαν αποτέλεσμα οι χρήστες των υπηρεσιών αυτών απαιτούσαν ολοένα και μεγαλύτερους ρυθμούς μετάδοσης αλλά και πιο ολοκληρωμένη εξυπηρέτηση από τις εταιρείες. Οι συνθήκες αυτές εισήγαγαν το μέγεθος της ποιότητας υπηρεσίας ή αλλιώς της QoS στα κυβελωτά αλλά και γενικά στα ασύρματα δίκτυα.

Στο κεφάλαιο αυτό θα γίνει ανάλυση του δικτύου UMTS. Η ανάλυση θα περιλαμβάνει τόσο την αρχιτεκτονική του δικτύου όσο και της περιγραφή των μηχανισμών που χρησιμοποιούνται για την μετάδοση πληροφοριών και την υποστήριξη υπηρεσιών. Το δίκτυο UMTS ανήκει στα δίκτυα τρίτης γενιάς. Επιλέξαμε την ανάλυση του λόγω της καθολικότητας του, αφού είναι πλέον ένα από πιο εξαπλωμένα δίκτυα, τόσο στην Ελλάδα όσο και στο εξωτερικό.

2.2. Ορισμός

Η QoS σε κυβελωτά δίκτυα ορίζεται σαν την ικανότητα των παρόχων μιας κυβελωτής υπηρεσίας να δίνουν μια ικανοποιητική υπηρεσία, η οποία να περιλαμβάνει ποιότητα φωνής, δύναμη σήματος, μικρό ποσοστό φραγής για τις κλήσεις, καθώς και μικρή πιθανότητα απόρριψης, υψηλές ταχύτητες μετάδοσης δεδομένων για πολυμέσα και εφαρμογές δεδομένων. Για υπηρεσίες βασισμένες στο διαδίκτυο, η QoS όπως έχουμε προαναφέρει εξαρτάται από τους παρακάτω παράγοντες:

- **Διέλευση (Throughput):** Ο ρυθμός με τον οποίο τα πακέτα κινούνται στο δίκτυο. Ο μεγαλύτερος ρυθμός είναι φυσικά πάντα προτιμότερος.
- **Καθυστέρηση (Delay):** Είναι ο χρόνος τον οποίο ένα πακέτο χρειάζεται, για να πάει από τον αποστολέα στον παραλήπτη. Προτιμάται πάντα μια ελάχιστη καθυστέρηση.
- **Ρυθμός Απώλειας Πακέτων (Packet Loss Rate):** Είναι ο ρυθμός με τον οποίο χάνονται πακέτα. Το μέγεθος αυτό πρέπει να είναι όσο το δυνατό μικρότερο.
- **Αξιοπιστία (Reliability):** Η ικανότητα για την επίτευξη μιας σύνδεσης.

Όπως γίνεται κατανοητό η ικανοποιητική υποστήριξη μιας υπηρεσίας είναι απαιτητική και πολύπλευρη αφού θέτει ως δεδομένο της εκπλήρωση των παραπάνω παραγόντων. Η

πρόκληση έγκειται στην αλληλένδετη επίδραση που έχει ο ένας παράγοντας πάνω στον άλλο, αφού οι σχέσεις τους μπορούν να χαρακτηριστούν ως σχέσεις αντιστρόφως ανάλογων ποσών.

2.2.1. Γιατί χρειαζόμαστε την ποιότητα υπηρεσιών

Αν φανταστούμε μια κατάσταση, στην οποία μπορούμε μετά βίας να ακούσουμε το συνομιλητή στην άλλη άκρη της γραμμής του τηλεφώνου ή ακόμη και αν σκεφτούμε μια κατάσταση διακοπής κλήσης κατά τη διάρκεια της κίνησης μας, τότε σίγουρα θα καταλάβουμε γιατί είναι απαραίτητη η QoS. Αυτές οι καταστάσεις είναι προφανώς ανεπιθύμητες και σίγουρα το οικονομικό αντίκρισμα, το οποίο πληρώνουν οι χρήστες, θα έπρεπε να είναι ανάλογο με την ποιότητα υπηρεσίας την οποία και απολαμβάνουν. Η επικοινωνία παίζει ένα απίστευτα σημαντικό ρόλο στις μέρες μας και η υποστήριξη της, καθώς και η εξέλιξή της, απαιτούν να δοθεί μεγάλη προτεραιότητα στην ποιότητα υπηρεσιών που αυτή προσφέρει.

Είναι σημαντικό να διαφοροποιήσουμε (ταξινομήσουμε) την κίνηση [17], η οποία βασίζεται σε ένα επίπεδο προτεραιότητας. Κάποιες κλάσεις κίνησης μπορούν να λάβουν υψηλότερη προτεραιότητα από κάποιες άλλες. Για παράδειγμα στις υπηρεσίες φωνής (τηλεφωνικές συνομιλίες) θα πρέπει να δίνεται μεγαλύτερη προτεραιότητα σε σχέση με την κίνηση πακέτων, αφού η «φωνή» θεωρείται μέχρι και σήμερα σαν η πιο σημαντική υπηρεσία. Θα πρέπει να σημειωθεί ότι οι πελάτες, που πληρώνουν περισσότερα για να πάρουν μια υπηρεσία καλύτερης ποιότητας, θα πρέπει να λάβουν ιδιαίτερη μεταχείριση ανάλογη των χρημάτων που καταβάλουν, χωρίς βέβαια αυτό να επηρεάζει και να υποβαθμίζει τους υπόλοιπους πελάτες.

2.2.2. Προκλήσεις Ποιότητας Υπηρεσίας

Στα ασύρματα κινητά δίκτυα, η QoS αναφέρεται στην μέτρηση ενός συστήματος με καλή ποιότητα μετάδοσης, διαθεσιμότητα της υπηρεσίας, αλλά και ελάχιστη καθυστέρηση. Στα δίκτυα 4G αναμένεται να έχουμε αξιοπιστία τουλάχιστον 99.999%, η οποία ονομάζεται και ως αξιοπιστία πέντε εννέα. Οι μεγαλύτερες προκλήσεις, όταν αναφερόμαστε στην ποιότητα υπηρεσιών σε ασύρματα κυβελωτά δίκτυα, είναι ο ποικίλος ρυθμός των χαρακτηριστικών του κάθε καναλιού, ο καταμερισμός του εύρους ζώνης, τα επίπεδα ανοχής σε σφάλματα και η υποστήριξη ανάμεσα σε ετερογενή ασύρματα δίκτυα.

Είναι πολύ βασικό το γεγονός ότι κάθε επίπεδο (Φυσικό, MAC, IP, TCP και Στρώμα Εφαρμογής), έχει τους δικούς του μηχανισμούς για την παροχή μιας QoS. Είναι επίσης σημαντικό, να εξασφαλισθεί η ποιότητα υπηρεσίας σε κάθε στρώμα, έτσι ώστε το δίκτυο να είναι πιο ευέλικτο και ανεκτικό σε θέματα ποιότητας υπηρεσίας. Κάποιες από τις άλλες προκλήσεις είναι η αποδοτική χρήση του φάσματος, λόγω της μείωσης της προσφοράς του.

Ο καταμερισμός του εύρους ζώνης διαδραματίζει έναν τεράστιο ρόλο, αναφορικά με αυτό το θέμα [17]. Πρέπει να γίνει κατανοητό ότι το εύρος ζώνης θα πρέπει να μοιράζεται με αποδοτικό τρόπο και επιπλέον το πλεονάζον εύρος ζώνης δεν θα πρέπει να ξοδεύεται ή να παραμένει αχρησιμοποίητο. Κάποια μοντέλα, όπως αυτό της επαναδιαπραγματεύσης, εξαλείφουν αυτό το θέμα, με τον διαμοιρασμό του εναπομείναντος εύρους ζώνης σε κλάσεις μικρότερης προτεραιότητας.

Τα πράγματα γίνονται ακόμη πιο περίπλοκα, όταν υπηρεσίες φωνής και δεδομένων πρέπει να στηριχθούν πάνω στο ίδιο δίκτυο. Οι υπηρεσίες φωνής είναι εξαιρετικά ευαίσθητες σε καθυστέρηση και απαιτούν υπηρεσίες πραγματικού χρόνου. Από την άλλη πλευρά, οι υπηρεσίες δεδομένων μπορεί να είναι λιγότερο ευαίσθητες στο επίπεδο της καθυστέρησης, αλλά είναι πολύ ευαίσθητες στην απώλεια δεδομένων και, επιπλέον, αναμένουν πακέτα χωρίς λάθη στα δεδομένα τα οποία μεταφέρουν. Έτσι, και οι δύο αυτοί οι παράγοντες πρέπει να ληφθούν υπ' όψιν για την παροχή QoS, τόσο για υπηρεσίες φωνής όσο και για υπηρεσίες δεδομένων.

2.2.3. Τι έχει γίνει μέχρι τώρα στον τομέα της Ποιότητας Υπηρεσίας

Στα δίκτυα πρώτης γενιάς 1G και στα δίκτυα δεύτερης γενιάς 2G, όπως είναι τα δίκτυα GSM, CDMA υπήρχε μόνο μία πλευρά της QoS και αυτή στηριζόταν στην παροχή καλύτερης ποιότητας υπηρεσιών τηλεφώνου (voice services), καθώς δεν είχαμε μεταφορά δεδομένων. Η προσφορά ποιότητας ομιλίας ήταν η μεγαλύτερη ανησυχία. Τώρα στα δίκτυα τρίτης γενιάς 3G, η ποιότητα υπηρεσιών πρέπει να παρέχεται, τόσο καλά στη φωνή όσο και στα δεδομένα. Σίγουρα δίνεται ακόμη μεγαλύτερη προτεραιότητα στις υπηρεσίες ομιλίας, αφού θεωρούνται πρωταρχικές υπηρεσίες. Είναι όλες ευαίσθητες στην καθυστέρηση και όλες απαιτούν υπηρεσίες πραγματικού χρόνου. Οι υπηρεσίες δεδομένων αποτελούνται από κείμενο και πολυμέσα. Αυτές οι υπηρεσίες είναι, όπως προαναφέραμε, λιγότερο ευαίσθητες στην καθυστέρηση, αλλά απαιτούν αυξημένα επίπεδα διέλευσης και πολύ μικρότερο αν όχι μηδενικό ρυθμό απωλειών.

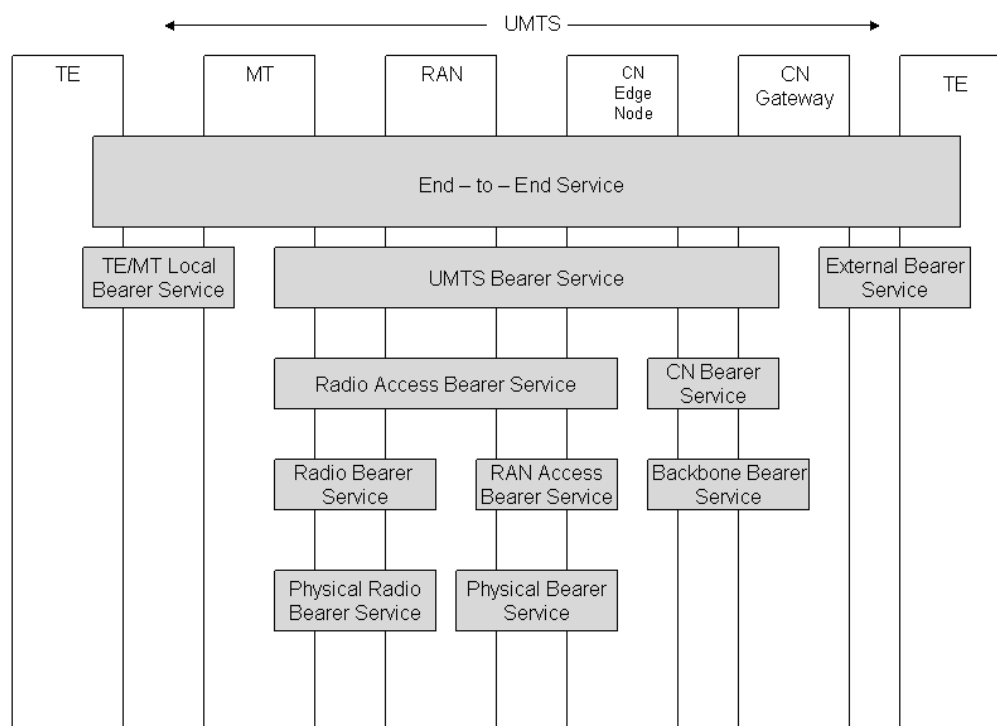
Για την μελέτη της επίδρασης των χαρακτηριστικών ενός κυψελωτού δικτύου σε μια υπηρεσία, αλλά και στην γενικότερη ποιότητα εξυπηρέτησης υπηρεσιών που παρέχονται, επιλέξαμε να παραθέσουμε το παράδειγμα ενός συστήματος UMTS, το οποίο είναι από τα πιο χαρακτηριστικά παραδείγματα συστημάτων που εφαρμόζονται σε δίκτυα τρίτης γενιάς. Η δυνατότητα που έχει στην υποστήριξη υπηρεσιών, τόσο ομιλίας όσο και δεδομένων, το καθιστούν άριστο παράδειγμα μελέτης και ανάλυσης.

Το UMTS ή Universal Mobile Telecommunications System είναι ίσως ένα από τα πιο χαρακτηριστικά συστήματα, που χρησιμοποιούν τεχνολογία τρίτης γενιάς. Το γεγονός ότι στηρίζεται στο WCDMA (Wideband Coded Division Multiple Access) μοντέλο, του δίνει τη δυνατότητα να υποστηρίζει μεγάλες ταχύτητες αποστολής, αλλά και λήψης δεδομένων. Χρησιμοποιεί πλαίσια χρόνου, που έχουν μήκος 10ms, ενώ το κάθε χρονοπλαίσιο διαθέτει 15 χρονοσχισμές. Έτσι, το UMTS έχει τη δυνατότητα υποστήριξης υψηλών ρυθμών μετάδοσης, που ανάλογα με την υλοποίηση, αλλά και τις συνθήκες, μπορούν να φτάσουν τα 384kbps αλλά ακόμη και τα 2Mbps.

2.3. Ποιότητα Υπηρεσιών σε Δίκτυο Τρίτης Γενιάς – Ανάλυση Μηχανισμών και Χαρακτηριστικών

2.3.1. Αρχιτεκτονική Ποιότητας Υπηρεσίας σε UMTS δίκτυο

Μια υπηρεσία απ' άκρο σ' άκρο υποθέτει ότι η επικοινωνία λαμβάνει χώρα από έναν τερματικό εξοπλισμό σε έναν άλλο [4]. Στον χρήστη μια δικτυακής υπηρεσίας παρέχεται μια ποιότητα υπηρεσιών, και είναι ο χρήστης αυτός που θα αποφασίσει αν είναι ικανοποιημένος με αυτή η QoS ή όχι. Μια υπηρεσία μεταφοράς, με καθαρά ορισμένα χαρακτηριστικά και διαδικασίες, πρέπει να τεθεί σε λειτουργία από την πηγή έως τον προορισμό, για να μπορεί να ανταποκριθεί με τις απαιτήσεις της ποιότητας υπηρεσίας. Η αρχιτεκτονική των υπηρεσιών μεταφοράς για τα δίκτυα UMTS φαίνονται στην παρακάτω εικόνα. Οι υπηρεσίες μεταφοράς προσφέρουν QoS, η οποία βασίζεται στα υποδεεστέρα επίπεδα.



Εικόνα 2. Αρχιτεκτονική UMTS. Πηγή: www.cse.wustl.edu

2.3.2. Απ' άκρη σ' άκρη υπηρεσίες και υπηρεσία μεταφοράς UMTS

Όπως φαίνεται και στην παραπάνω εικόνα (Εικόνα 2), το επίπεδο υπηρεσιών end - to - end είναι το υψηλότερο επίπεδο της αρχιτεκτονικής QoS. Κάνει δυνατή την επικοινωνία από ένα τερματικό στο άλλο. Παρατηρούμε πως το τερματικό είναι συνδεδεμένο στο δίκτυο UMTS, με μέσα κινητού τερματισμού [4]. Οι υπηρεσίες απ' άκρο σ' άκρο, οι οποίες και χρησιμοποιούνται από τον τερματικό εξοπλισμό, μπορούν να υλοποιηθούν από τα υποδεεστέρα επίπεδα, που ονομάζονται TE/MT Τοπική Υπηρεσία Μεταφοράς (TE/MT Local Bearer Service), UMTS Υπηρεσία Μεταφοράς (UMTS Bearer Service) και Εξωτερική Υπηρεσία Μεταφοράς (External Bearer Service) [17]. Ο χειριστής του UMTS παρέχει υπηρεσίες, που δίνονται από τις υπηρεσίες μεταφοράς του UMTS. Έτσι μπορούμε να καταλάβουμε ότι οι υπηρεσίες μεταφοράς του UMTS είναι αυτές που ουσιαστικά δίνουν την ποιότητα υπηρεσίας στο συγκεκριμένο δίκτυο

2.3.3. Υπηρεσία Μεταφοράς Ράδιο-Πρόσβασης και Υπηρεσία Μεταφοράς Πυρήνα Δικτύου

Η υπηρεσία μεταφοράς ενός δικτύου UMTS αποτελείται από δύο μέρη, τα οποία είναι η Υπηρεσία Μεταφοράς Ράδιο-Πρόσβασης (Radio Access Bearer Service) και η Υπηρεσία Μεταφοράς Πυρήνα Δικτύου (Core Network Bearer Service). Οι δύο αυτές υπηρεσίες είναι υπεύθυνες για την σωστή λειτουργία της υπηρεσίας μεταφοράς πάνω στην τοπολογία του δικτύου, λαμβάνοντας υπ' όψιν χαρακτηριστικά, όπως είναι η φορητότητα και το προφίλ του χρήστη.

Η Radio Access Bearer Service είναι υπεύθυνη για την παροχή της μεταφοράς της σηματοδοσίας και συνάντησης, ανάμεσα στον MT και στον τελευταίο κόμβο του CN, με ποιότητα υπηρεσίας ισοδύναμη με αυτή που έχει συμφωνηθεί με την UMTS υπηρεσία μεταφοράς ή με μια προκαθορισμένη QoS για σηματοδοσία.

Η Core Network Bearer Service ενός πυρήνα UMTS δικτύου συνδέει τον ακραίο κόμβο του UMTS CN, μέσω της πύλης εξόδου (gateway), με το εξωτερικό δίκτυο. Η υπηρεσία αυτή ελέγχει και χειρίζεται την ραχοκοκαλιά της αποδοτικότητας του δικτύου, με σκοπό να παρέχει την συμφωνηθείσα UMTS υπηρεσία μεταφοράς.

2.3.4. Radio Bearer Service και RAN Access Bearer Service

Η RABS υλοποιείται από την Radio Bearer Service και μια RAN Access - Bearer Service. Η Radio Bearer Service περιλαμβάνει όλες τις υπηρεσίες της διεπαφής ασύρματης μεταφοράς. Για την υποστήριξη άνισης προστασίας από σφάλματα, η RAN και το MT έχουν την ικανότητα να κατακερματίσουν και να επανασυνδέσουν την ροή δεδομένων του χρήστη σε διαφορετικές υπό-ροές, οι οποίες ζητούνται από την Radio Access Bearer Service. Ο κατακερματισμός και η επανένωση δίνεται από τη μορφή του περιεχομένου της Μονάδας Δεδομένων Υπηρεσίας (Service Data Unit (SDU)), η οποία σηματοδοτείται κατά την εγκατάσταση της Radio Access Bearer.

2.3.5. Η ραχοκοκαλιά των δικτυακών υπηρεσιών

Ο πυρήνας των δικτυακών υπηρεσιών μεταφοράς χρησιμοποιεί μια χαρακτηριστική υπηρεσία του πυρήνα του δικτύου (backbone network service). Η Backbone Network Service περιλαμβάνει τα στρώματα 1 και 2 και τις λειτουργίες τους και επιλέγεται ανάλογα με την προτίμηση του διαχειριστή να εκπληρώσει τις υποχρεώσεις του για την QoS σε ένα CN Bearer Service.

2.4. Κλάσεις Ποιότητας Υπηρεσίας δικτύου UMTS και Χαρακτηριστικά τους

Υπάρχουν 4 διαφορετικές κλάσεις QoS σε ένα δίκτυο UMTS:

- Κλάση συνομιλίας (Conversational Class)
- Κλάση Streaming(Streaming Class)
- Διαδραστική κλάση(Interactive Class)
- Κλάση υποβάθρου (Background Class)

Οι βασικές διαφορές ανάμεσα σε αυτές τις κλάσεις QoS συγκεντρώνονται στην ευαισθησία που έχει η κάθε κλάση, στην καθυστέρηση λήψης πακέτων δεδομένων. Η κλάση συνομιλίας είναι προορισμένη για κίνηση, η οποία είναι πολύ ευαίσθητη στην καθυστέρηση, ενώ η κλάση υποβάθρου είναι γενικά πιο αναισθητη στην καθυστέρηση.

Η κλάση συνομιλίας, καθώς και η κλάση streaming, χρησιμοποιούνται για να στηρίζουν ροές δεδομένων πραγματικού χρόνου, οι οποίες είναι πολύ ευαίσθητες στην καθυστέρηση. Παραδείγματα της τάξης συνομιλίας είναι υπηρεσίες πραγματικού χρόνου, όπως η τηλεφωνική κλήση με χρήση βίντεο, καθώς και το data streaming.

Η διαδραστική κλάση και η κλάση υποβάθρου προορίζονται κυρίως για εφαρμογές, όπως είναι οι εφαρμογές ιστού (www), ηλεκτρονική αλληλογραφία (email), FTP, News, Telnet. Από την στιγμή που αυτές οι κλάσεις είναι λιγότερο ευαίσθητες στη καθυστέρηση, συγκρινόμενες πάντα με τις κλάσεις συνομιλίας και streaming, και οι δύο αυτές κλάσεις παρέχουν καλύτερο ρυθμό εμφάνισης σφάλματος, με μέσα κωδικοποίησης του καναλιού και επανεκπομπής [4]. Αυτό σημαίνει ότι η επαναμετάδοση ενός πακέτου γίνεται όταν έχουμε σφάλμα σε πακέτο, απώλεια πακέτου ή ακόμη και λανθασμένη σειρά παραλαβής πακέτων. Ο λόγος είναι ότι αυτές οι κλάσεις είναι σχετικά αναισθητες στην καθυστέρηση, αλλά απαιτούν υψηλά επίπεδα διέλευσης και μικρό ποσοστό λαθών.

Η κεντρική διαφορά, ανάμεσα στην διαδραστική και στην κλάση υποβάθρου, είναι πως η διαδραστική κλάση χρησιμοποιείται κυρίως για εφαρμογές αλληλεπίδρασης, όπως είναι το Interactive Email ή το Interactive Web Browsing, ενώ η κλάση υποβάθρου προορίζεται για την παρασκηνακή κίνηση, για παράδειγμα τα downloads που συμβαίνουν σε δεύτερο σκηνικό ή την ηλεκτρονική αλληλογραφία. Ο αλγόριθμος χρονοπρογραμματισμού δίνει περισσότερη προτεραιότητα στην διαδραστική κλάση απ' ό τι στην κλάση υποβάθρου. Για τον λόγο αυτό, η κλάση υποβάθρου χρησιμοποιεί τους πόρους μετακίνησης, όταν οι διαδραστικές εφαρμογές δεν τους χρειάζονται.

2.4.1. Κλάση Συνομιλίας

Οι εφαρμογές οι οποίες χρησιμοποιούν αυτή την κλάση περιλαμβάνουν ομιλία τηλεφωνίας, voice over IP (φωνή που μεταδίδεται ενθλακωμένη σε πακέτα IP) και συνομιλίες βίντεο. Οι συζητήσεις πραγματικού χρόνου πραγματοποιούνται πάντα μεταξύ ομάδων ανθρώπων και έτσι αυτό είναι το μοναδικό μοντέλο, το οποίο καθορίζει τα χαρακτηριστικά του αυστηρά από το πώς γίνεται η λήψη της πληροφορίας από τους ανθρώπους.

Ένα μοντέλο συζητήσεων πραγματικού χρόνου χαρακτηρίζεται από το χρόνο μεταφοράς, ο οποίος θα πρέπει να είναι μικρός, λόγω της φύσης που δίνει η συζήτηση στο μοντέλο αυτό και την ίδια στιγμή η ποικιλότητα του χρόνου ανάμεσα στις οντότητες της ροής πληροφορίας, θα πρέπει να διατηρείται, όπως και στις συνομιλίες πραγματικού χρόνου. Η μέγιστη καθυστέρηση μεταφοράς ορίζεται από το πόσο μεγάλη καθυστέρηση μπορούν να ανεχθούν οι άνθρωποι για ήχο και βίντεο.

Για το λόγο αυτό, τα όρια για αποδεκτή καθυστέρηση μεταφοράς είναι πολύ άκαμπτα, και εάν η καθυστέρηση μεταφοράς δεν είναι αρκετά μικρή, τότε επηρεάζεται η ποιότητα της υπηρεσίας. Η καθυστέρηση μεταφοράς θα πρέπει να είναι μικρότερη και πιο άκαμπτη από την καθυστέρηση round trip που έχει αυτή η κλάση. Ως round trip χρόνο ορίζουμε το χρονικό διάστημα για την μεταφορά ενός πακέτου, από το ένα άκρο στο άλλο και πίσω.

2.4.2. Κλάση Streaming

Οι εφαρμογές για την κλάση αυτή περιλαμβάνουν την παρακολούθηση, τόσο ως ήχο όσο και ως εικόνα ενός βίντεο πραγματικού χρόνου. Το μοντέλο αυτό χαρακτηρίζεται από τις χρονικές σχέσεις ανάμεσα στις οντότητες πληροφορίας, οι οποίες θα πρέπει να διασφαλίζονται σε μια συγκεκριμένη ροή, χωρίς βέβαια αυτό να εγείρει υποχρεώσεις για μικρή καθυστέρηση. Έτσι, καταλαβαίνουμε ότι είναι ουσιώδες να έχουμε συγχρονισμό ανάμεσα στα πακέτα ήχου και εικόνας, για να επιτύχουμε ικανοποιητική ποιότητα, χωρίς αυτό όμως να μας υποχρεώνει να μειώσουμε κατά πολύ την καθυστέρηση παράδοσης των πακέτων αυτών.

2.4.3. Διαδραστική Κλάση

Οι εφαρμογές για την κλάση αυτή περιλαμβάνουν την περιήγηση στον Ιστό, την ανάκτηση βάσεων δεδομένων, την είσοδο σε server κτλ. Η διαδραστική κίνηση είναι ένα τηλεπικοινωνιακό μοντέλο, το οποίο χαρακτηρίζεται από το πρότυπο της αίτησης και της απάντησης στον τελικό χρήστη. Η καθυστέρηση ταξιδιού (Round Trip delay) είναι το πιο σημαντικό μέγεθος για την κλάση αυτή. Σημαντικό επίσης μέγεθος είναι και ο ρυθμός λαθών, ο οποίος θα πρέπει να είναι ο ελάχιστος δυνατός για την μεταφορά δεδομένων.

2.4.4. Κλάση Υποβάθρου

Είναι μια κλάση υπηρεσίας, στην οποία οι εφαρμογές τρέχουν στο παρασκήνιο. Διάφορα παραδείγματα περιλαμβάνουν ένα πρόγραμμα ηλεκτρονικής αλληλογραφίας, το οποίο τρέχει στο παρασκήνιο. Μπαίνει σε κατάσταση αναμονής για την περισσότερη ώρα και “ξυπνά”, όταν ένα email καταφθάνει. Άλλα παραδείγματα αφορούν τα SMS ή το κατέβασμα από βάσεις δεδομένων. Η κίνηση παρασκηνίου χαρακτηρίζεται από το γεγονός ότι ο προορισμός δεν αναμένει δεδομένα σε ένα συγκεκριμένο χρόνο. Έτσι, αυτή η κλάση είναι λιγότερο ευαίσθητη στην καθυστέρηση και άλλο ένα χαρακτηριστικό είναι ότι τα περιεχόμενά της μπορούν να παραδοθούν με χαμηλές ταχύτητες.

2.4.5. Ιδιότητες Ποιότητας Υπηρεσίας

Οι ιδιότητες της υπηρεσίας μεταφοράς UMTS περιγράφουν την υπηρεσία που παρέχεται από ένα δίκτυο UMTS στον χρήστη της εν λόγω υπηρεσίας. Μια ομάδα από ιδιότητες ποιότητας υπηρεσίας πρακτικά ορίζουν την υπηρεσία. Στη συνέχεια αναλύουμε τα βασικότερα χαρακτηριστικά μεγέθη της υπηρεσίας [17].

- **Μέγιστο Bit Rate (kbps)** ορίζεται σαν το μέγιστο πλήθος bits που μπορεί να παραδοθεί από ένα UMTS, αλλά και σε ένα UMTS, σε/από ένα Σημείο Πρόσβασης της Υπηρεσίας (Service Access Point) μέσα σε μια περίοδο χρόνου, προς τη διάρκεια του χρόνου αυτού. Το μέγιστο Bit Rate (ρυθμός bit) είναι το άνω όριο ταχύτητας που μπορεί ένας χρήστης ή μια υπηρεσία να δεχτεί ή να παρέχει αντίστοιχα. Ο λόγος της ύπαρξης της ιδιότητας αυτής είναι:

- (1) να περιορίσει τον παραδιδόμενο ρυθμό bit σε εφαρμογές ή εξωτερικά δίκτυα και
- (2) να ορίσει το διαφορετικό bit rate για εφαρμογές που το χρειάζονται.

- **Εγγυημένο Bit Rate (kbps)** είναι ο εξασφαλισμένος αριθμός bits που παραδίδονται σε ένα SAP από ένα UMTS δίκτυο, σε μία περίοδο χρόνου (δεδομένου πάντα ότι υπάρχουν δεδομένα να αποσταλούν) διαιρεμένος με την διάρκεια της περιόδου αυτής. Ο σκοπός της ιδιότητας αυτής είναι να περιγράψει την ταχύτητα, που θα πρέπει να εγγυηθεί η υπηρεσία μεταφοράς UMTS στον χρήστη ή στην εφαρμογή.
- **Σειρά Παράδοσης (y/n)** αναφέρει για το αν η υπηρεσία μεταφοράς του UMTS πρέπει να παρέχει προκαθορισμένης σειράς παράδοση ή όχι. Ο λόγος της ιδιότητας αυτής είναι για να ξέρουμε πότε χρειάζεται μια τέτοιους είδους παράδοση και πότε όχι.
- **Μέγιστο μέγεθος SDU(Single Dwelling Unit):** είναι το μέγιστο μέγεθος SDU για οποίο το δίκτυο θα ικανοποιήσει την συμφωνημένη QoS. Η ιδιότητα αυτή συνήθως χρησιμοποιείται για τον έλεγχο της πρόσβασης και την περιπολία του δικτύου για την καλύτερη δυνατή μεταφορά.
- **SDU format πληροφορίας (bits):** ορίζει σε μια λίστα όλα τα πιθανά μεγέθη SDU.
- **SDU error ratio (λόγος λαθών SDU):** περιλαμβάνει το τμήμα των SDU, τα οποία έχουν χαθεί ή έχουν ανιχνευτεί σαν εσφαλμένα πακέτα. Ο λόγος εσφαλμένων SDU ορίζεται μόνο και μόνο για συμμορφώσει την κίνηση. Η ιδιότητα αυτή χρησιμοποιείται για να ταξινομήσει τα πρωτόκολλα, τους αλγορίθμους και το μοντέλο ανίχνευσης λαθών.
- **Residual bit error ratio (εναπομείναν ρυθμός δεδομένων):** περιλαμβάνει το μη ανιχνευμένο λόγο εσφαλμένων bit στα παραδοθέντα SDU. Εάν δεν έχει ζητηθεί ανίχνευση λαθών, το residual error ratio περιλαμβάνει την αναλογία λαθών στα ήδη παραδοθέντα SDUs. Η ιδιότητα αυτή χρησιμοποιείται για την ταξινόμηση και αναγνώριση των radio interface πρωτοκόλλων, αλγορίθμων και κωδικοποίησης ανίχνευσης σφαλμάτων.
- **Delivery of erroneous SDU(y/n):** Περιλαμβάνει αν τα εσφαλμένα SDUs που ανιχνεύονται θα πρέπει να παραδίδονται ή να απορρίπτονται. Χρησιμοποιείται για την απόφαση εάν η ανίχνευση λαθών είναι απαιτούμενη ή όχι και επίσης εάν τα πλαίσια τα οποία ανιχνεύονται σαν εσφαλμένα πρέπει να απορριφθούν ή όχι.
- **Καθυστέρηση μεταφοράς (ms):** ορίζεται σαν τη μέγιστη καθυστέρηση για το 95% της κατανομής καθυστέρησης για όλα τα παραδοθέντα SDUs, κατά τη διάρκεια ζωής μιας υπηρεσίας μεταφοράς, όπου καθυστέρηση για ένα SDU ορίζεται ως ο χρόνος από την αίτηση για μεταφορά και SDU σε ένα SAP μέχρι την παράδοση του σε ένα άλλο SAP.
- **Καθορισμός προτεραιότητας κίνησης:** ορίζει την σχετική σημασία της διαχείρισης όλων των SDU που ανήκουν στον μεταφορέα UMTS συγκρινόμενο με τα SDU άλλων μεταφορέων.

2.5. Διαφορετικά Μοντέλα Ποιότητας Υπηρεσίας σε Κυψελωτά Δίκτυα

Υπάρχουν πολλά μοντέλα QoS, τα οποία έχουν εφαρμοστεί σε κυψελωτά συστήματα και κάθε μοντέλο έχει τα θετικά του και τα αρνητικά του. Στο κεφάλαιο αυτό θα προσπαθήσουμε να δούμε κάποια από τα βασικά και αποτελεσματικά μοντέλα QoS, που είναι αυτά που χρησιμοποιούνται για να παρέχουν υπηρεσίες φωνής και δεδομένων.

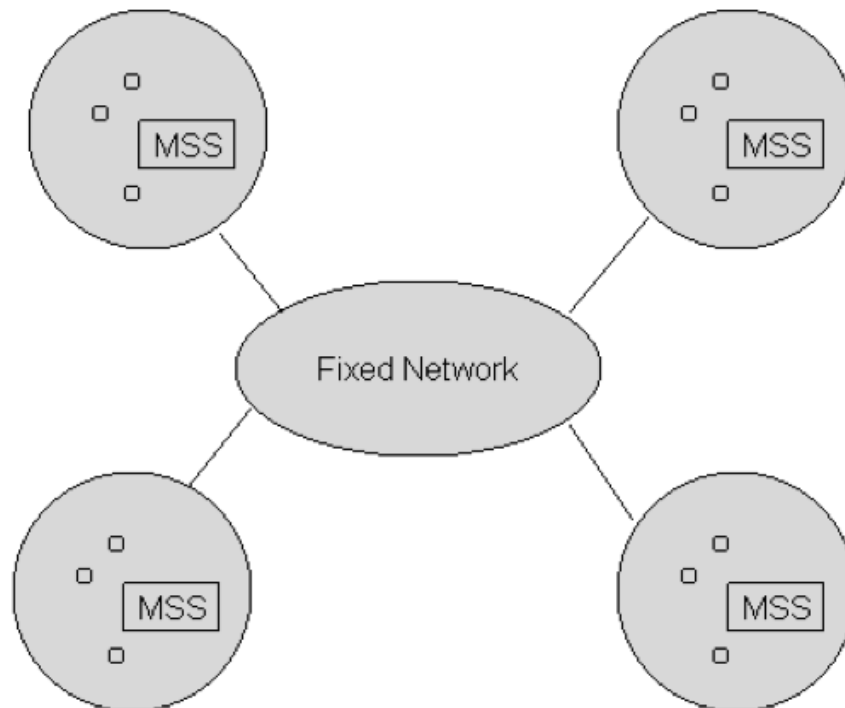
Το Ανεκτικό σε Σφάλματα μοντέλο Δυναμικού Καταμερισμού (**Fault Tolerant Dynamic Allocation** scheme) [20] αναφέρεται στις μεθόδους επαναχρησιμοποίησης των καναλιών, με αποδοτικό τρόπο ανάμεσα σε δύο κυψέλες, οι οποίες και χωρίζονται από μια ελάχιστη απόσταση έτσι ώστε να μην παρεμβάλουν η μία στην άλλη. Τα κανάλια καταμερίζονται δυναμικά, το οποίο έρχεται σε αντίθεση με το στατικό καταμερισμό, όπου τα κανάλια είναι μοιρασμένα και δεσμευμένα εξ αρχής.

Το επόμενο μοντέλο είναι το μοντέλο Έλεγχου Εισόδου Κλήσης (**Call Admission Control - CAC**) [23], το οποίο απασχολεί το φραγμό των κλήσεων σε πρώτο επίπεδο, ο οποίος είναι βασισμένος στο διαθέσιμο εύρος ζώνης και στο χειρισμό κλήσεων. Ο αλγόριθμος είναι βασισμένος σε δύο μοντέλα, τα οποία χρησιμοποιούνταν νωρίτερα και είχαν τα ονόματα Μοντέλο πρωθύστερης - κλήσης (pre request scheme) και μοντέλο προστασίας καναλιού (guard channel scheme). Ο αλγόριθμος CAC χρησιμοποιεί και τα δύο μοντέλα και δίνει επίδοση, σε όρους ρυθμών επιτυχούς ολοκλήρωσης της κλήσης (Successful Call Completion Rates - SCCR), ενώ ταυτόχρονα παρέχει εξασφαλισμένη QoS για εγκεκριμένους χρήστες. Στις τεχνικές πρόβλεψης της φορητότητας, οι κλήσεις που χάνονται μειώνονται, κυρίως γιατί οι πιθανότητες αποκλεισμού αλλά και οι πιθανότητες απόρριψης μειώνονται σημαντικά. Στο συγκεκριμένο μοντέλο πρόβλεψης φορητότητας, οι πληροφορίες τοπολογίας του δρόμου συλλέγονται και αποθηκεύονται σε μια βάση δεδομένων και υπολογίζεται το μονοπάτι ή η τροχιά του κινητού φορέα. Καμία υπόθεση δε γίνεται για το σχήμα της κυψέλης.

Το μοντέλο επαναδιαπραγματεύσεως είναι ένα μοντέλο, όπου ο καταμερισμός του εύρους ζώνης αλλάζει δυναμικά, βασισμένος πάνω στην διαθεσιμότητα. Εάν μια υπηρεσία χαμηλής προτεραιότητας χρησιμοποιεί εύρος ζώνης μικρότερο από αυτό το οποίο έχει ζητήσει, τότε το περισσευούμενο εύρος ζώνης που είναι διαθέσιμο από την ολοκλήρωση μιας υπηρεσίας υψηλής προτεραιότητας, θα δοθεί στην υπηρεσία χαμηλής προτεραιότητας, το οποίο συνεπάγεται και την αύξηση της QoS για την υπηρεσία μικρής προτεραιότητας. Το μοντέλο αυτό προϋποθέτει, ότι οι υπηρεσίες υψηλότερης προτεραιότητας παίρνουν το απαιτούμενο εύρος ζώνης και δεν επηρεάζονται με κανένα τρόπο.

2.5.1. Ανεκτικό σε Σφάλματα μοντέλο Δυναμικού Καταμερισμού

Στο μοντέλο αυτό, τα κανάλια μοιράζονται δυναμικά, με βάση την υπάρχουσα ζήτηση και έτσι αυξάνεται η χρησιμοποίηση των καναλιών, όπως και της QoS. Το μοντέλο καταμερισμού καναλιών μπορεί να έχει τόσο συγκεντρωτικό χαρακτήρα όσο και κατανεμημένο [20]. Στη συγκεντρωτική προσέγγιση, υπάρχει ένας κεντρικός χειριστής ο οποίος είναι υπεύθυνος για το διαμοιρασμό των καναλιών, ενώ οι απαιτήσεις/αιτήσεις στέλνονται στον χειριστή. Αντιθέτως στην κατανεμημένη προσέγγιση υπάρχει ένας Σταθμός Κινητής Υπηρεσίας (Mobile Service Station - MSS) σε κάθε κυψέλη, ο οποίος είναι υπεύθυνος για το διαμοιρασμό των καναλιών για τη συγκεκριμένη κυψέλη. Η κατανεμημένη προσέγγιση είναι πιο κλιμακωτή και αξιόπιστη, ενώ, λόγω της κλιμακούμενης δομής της, μπορεί να βρει εφαρμογή σε αλγόριθμο.



Εικόνα 3. Μοντέλο Ασύρματων Δικτύων Επικοινωνιών
 Πηγή: www.cse.wustl.edu

Η παραπάνω εικόνα (Εικόνα 3) δείχνει το μοντέλο της ασύρματης επικοινωνίας. Υπάρχουν πολλές κυψέλες που αποτελούνται από πολλούς Κινητούς Φορείς (Mobile Hosts - MH) και από Σταθμούς Κινητών Υπηρεσιών (Mobile Service Station - MSS). Οι κυψέλες είναι συνδεδεμένες σε ένα σταθερό δίκτυο. Κάθε MH, όποτε θέλει να επικοινωνήσει με κάποιον άλλο MH, πρέπει να απαιτήσει ένα κανάλι από τον MSS για επικοινωνία.

Υπάρχουν δύο τύποι καναλιών, που ονομάζονται κανάλια επικοινωνίας και κανάλια έλεγχου. Ένα κανάλι επικοινωνίας χρησιμοποιείται για να υποστηρίξει την επικοινωνία, ανάμεσα σε έναν MH και σε ένα MSS, ενώ ένα κανάλι ελέγχου χρησιμοποιείται για την αποστολή μηνυμάτων ελέγχου, όπως αυτά που στέλνει ο MH στον MSS για να ζητήσει ένα κανάλι επικοινωνίας. Ο MSS διαμοιράζει τα κανάλια στον MH, βασισμένος στο αν υπάρχουν ή όχι ομοκαναλικές (ομοδιαυλικές) παρεμβολές. Εάν δεν υπάρχουν τέτοιου είδους παρεμβολές, τότε μοιράζει το κανάλι στον MH για επικοινωνία.

Υπάρχουν δύο διαφορετικές προσεγγίσεις για την ζήτηση καναλιών. Στη συγκεντρωτική προσέγγιση, κάθε MH απαιτεί κανάλια από έναν κεντρικό διαχειριστή, ο οποίος ονομάζεται Κέντρο Κινητών Μεταγωγών (Mobile Switching Center - MSC). Έτσι ο καταμερισμός γίνεται από το MSC, με τέτοιο τρόπο ώστε να μην υπάρχουν ομοκαναλικές παρεμβολές. Το μειονέκτημα αυτής της προσέγγισης είναι ότι το Κέντρο Κινητών Μεταγωγών μπορεί να γίνει μια στενωπός στο δίκτυο. Εκτός αυτού, μια βλάβη στο MSC συνεπάγεται άμεσα και την επίδραση της σε ολόκληρο το δίκτυο. Αυτή η προσέγγιση δεν είναι διόλου κλιμακούμενη και υγιής.

Στην καταναμημένη προσέγγιση δεν υπάρχει κάποιος κεντρικός διαχειριστής και κάθε κυψέλη έχει και το δικό της MSS. Είναι η δουλειά του MSS να μοιράσει τα κανάλια και να καταστήσει σίγουρη την έλλειψη παρεμβολών στα κανάλια. Στην προσέγγιση αυτή, η κυψέλη ενημερώνει τους γείτονες της για την χρήση των καναλιών που κάνει, και τους πληροφορεί όποτε απελευθερώνει κάποιον δίαυλο, έτσι ώστε οι άλλοι να μπορούν να τον χρησιμοποιήσουν. Όταν μια κυψέλη χρειάζεται ένα κανάλι, στέλνει ένα μήνυμα έλεγχου σε όλους τους γείτονες της και όταν πάρει την απάντηση απ' όλους τους γείτονές της ότι το συγκεκριμένο κανάλι δεν χρησιμοποιείται, τότε αυτή χρησιμοποιεί το κανάλι.

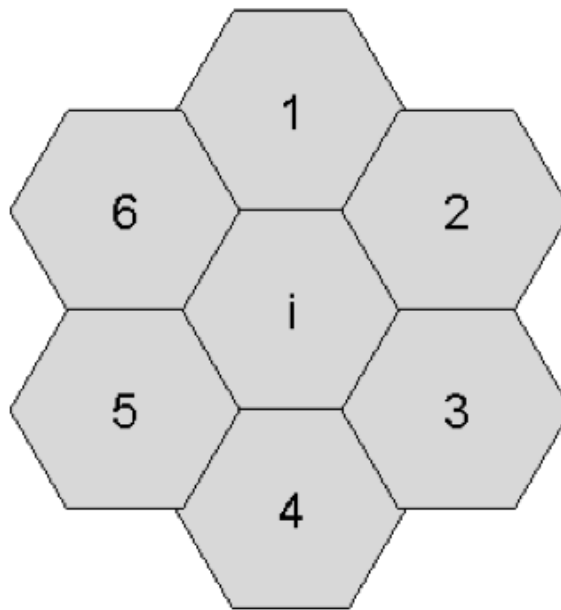
2.5.1.1. Μοντέλο καταναμημένου και ανθεκτικού σε λάθη συστήματος

Στο κεφάλαιο αυτό, θα αναφερθούμε στο μοντέλο συστήματος αυτής της καταναμημένης ανεκτικής στα λάθη προσέγγισης. Τα κανάλια μοιράζονται δυναμικά και δε δίνονται εξ' αρχής σε καμία κυψέλη. Ένα μοντέλο κυψέλης τριών τομέων χρησιμοποιείται και έτσι κάθε κυψέλη έχει έξι γείτονες [17]. Εάν ένα κανάλι χρησιμοποιείται από μια συγκεκριμένη κυψέλη, τότε κανείς από τους γείτονες δεν μπορεί να χρησιμοποιήσει το συγκεκριμένο κανάλι και έτσι αποφεύγεται η ομοκαναλική παρεμβολή. Μια κυψέλη μπορεί να δανειστεί ένα κανάλι για να χρησιμοποιήσει, αν δεν έχει κάποιο άλλο κανάλι ελεύθερο για επικοινωνία. Η αίτηση πρέπει να ικανοποιηθεί από οποιοδήποτε γείτονα της κυψέλης, ο οποίος μπορεί να δανείσει ένα κανάλι και παρ' όλα αυτά να αποφύγει την ομοκαναλική παρεμβολή.

Υπάρχουν δύο διαφορετικά στάδια λειτουργίας, τα οποία είναι το στάδιο ενημέρωσης και το στάδιο αναζήτησης. Στο στάδιο ενημέρωσης, η κυψέλη πληροφορεί τους γείτονές της για την αλλαγή της χρήσης των καναλιών της, όποτε δανείζεται ή απελευθερώνει ένα κανάλι. Έτσι ενημερώνει όλους τους γείτονες για την κατάσταση. Στο στάδιο αναζήτησης, όταν μια κυψέλη χρειάζεται να δανειστεί ένα κανάλι, στέλνει μήνυμα αίτησης σε όλους τους γείτονες. Βασισμένη στην απάντηση που λαμβάνει από αυτούς δανείζεται το κανάλι.

2.5.1.2. Αλγόριθμος Καταναμημένου Διαμοιρασμού Καναλιών

Στο μοντέλο αυτό χρησιμοποιείται ο τύπος μιας κυψέλης τριών τομέων, και έτσι η κυψέλη έχει 6 γείτονες, όπου κάθε ένας από αυτούς έχει μια μοναδική ταυτότητα. Τα κανάλια δεν είναι προδιαμοιρασμένα σε καμία κυψέλη. Το κανάλι με την μικρότερη συχνότητα έχει τη μικρότερη κατάταξη και το κανάλι με τη μεγαλύτερη συχνότητα έχει τη μεγαλύτερη κατάταξη. Μια κυψέλη C_i επιλέγει ένα κανάλι ανάμεσα στις υψηλότερες τάξεις, για να υποστηρίξει αν είναι δυνατή μια κλήση.



Εικόνα 4. Ομάδα Κυψελών για την αναπαράσταση Κατανομής Καναλιών
Πηγή: www.cse.wustl.edu

Στην παραπάνω εικόνα (Εικόνα 4) κάθε κυψέλη αριθμείται από το ένα έως το έξι. Εμείς θα χρησιμοποιήσουμε την κυψέλη C_i , για να παραστήσουμε την επιθυμητή κυψέλη. Όταν η C_i αιτείται για τη χρήση ενός καναλιού, είναι σε κατάσταση αναζήτησης και καλείται δανειζόμενη (borrower). Στέλνει ένα ευρυζωνικό μήνυμα σε όλους τους γείτονες για το κανάλι και ξεκινά ένα χρονομετρητή, περιμένοντας μέχρι ο χρονομετρητής να μηδενίσει. Όταν μία κυψέλη είναι σε κατάσταση αίτησης δανεισμού, δεν απαντά σε καμία άλλη αίτηση δανεισμού από τις γειτονικές κυψέλες. Αυτό είναι λογικό, αφού η ίδια η δανειζόμενη κυψέλη είναι σε κατάσταση αναζήτησης και έτσι δεν μπορεί να προσφέρει βοήθεια σε γειτονικές κυψέλες που αναζητούν δανεισμό. Το μοντέλο αυτό είναι καλό, από την πλευρά ότι απαγορεύει την ανεπιθύμητη συμφόρηση του δικτύου. Όταν ο χρονομετρητής λήξει, η κυψέλη C_i θα δανειστεί ένα κανάλι, βασισόμενη στις απαντήσεις που έχει λάβει από τους γείτονες της.

Για την απόδειξη της ορθής λειτουργίας του συγκεκριμένου αλγόριθμου, ας υποθέσουμε ότι η κυψέλη C_i έχει απαντήσεις μόνο από δύο γείτονες, από τον 1 και τον 4 και ότι είναι το κανάλι r το οποίο είναι μοιρασμένο και στις δύο κυψέλες 1 και 4. Ο λόγος είναι ότι, αφού το κανάλι r είναι μοιρασμένο στις 1 και 4, δεν μπορεί να μοιραστεί στους γείτονες των κυψελών 1 και 4 (κυψέλες 2,6,3 και 5) για το λόγο της ομοκαναλικής παρεμβολής. Έτσι το κανάλι που η C_i μπορεί να δανειστεί προέκυψε από τις απαντήσεις που τις έδωσαν οι άλλες δύο κυψέλες. Αυτό αποδεικνύει ότι ο αλγόριθμος είναι ανεκτικός σε λάθη.

Λαμβάνοντας υπ' όψιν τα παραπάνω, ο αλγόριθμος που περιγράψαμε είδαμε ότι είναι ανεκτικός σε λάθη και ότι δεν χρειάζεται να περιμένει απαντήσεις απ όλα τα κανάλια. Αυτό επίσης σημαίνει ότι ο αλγόριθμος είναι πιο κλιμακούμενος. Χρησιμοποιεί το μοντέλο κατανεμημένου διαμοιρασμού καναλιών στο οποίο τα κανάλια δίδονται βασισμένα στην

ζήτηση. Ο αλγόριθμος αυτός επίσης επιτρέπει την επαναχρησιμοποίηση των καναλιών. Αυτό σημαίνει ότι ο δανειστής μπορεί να δανείσει ένα κανάλι σε δύο ή περισσότερες κυψέλες συγχρόνως δεδομένου πάντα ότι δεν συμβαίνει ομοκαναλική παρεμβολή.

2.5.2. Μοντέλο Έλεγχου Εισόδου Κλήσεων

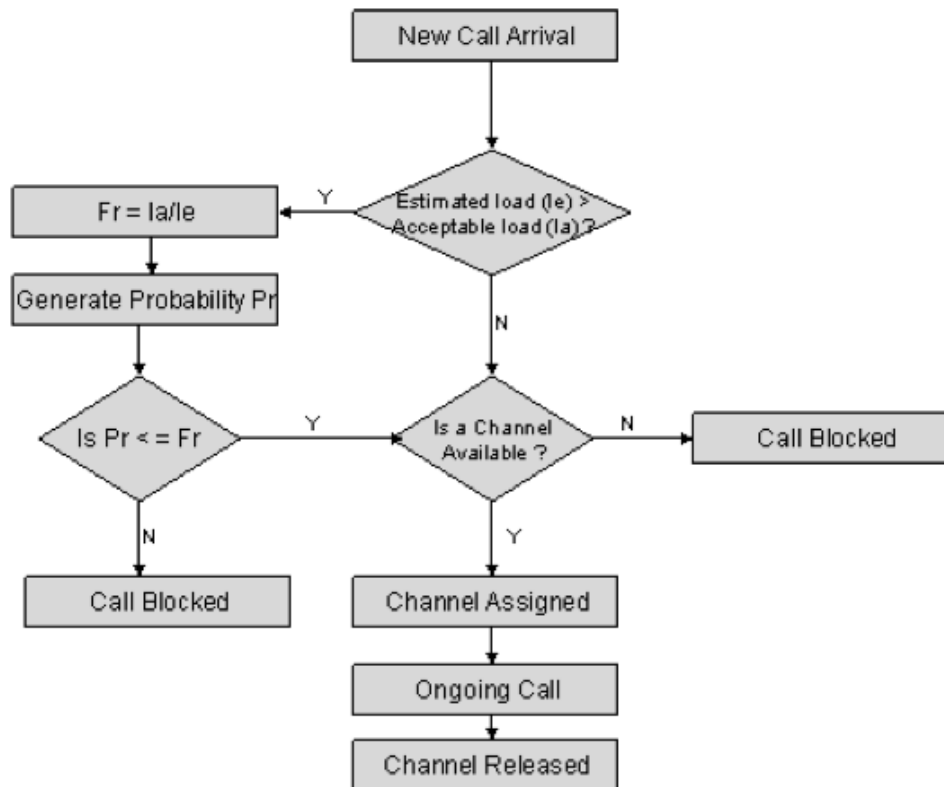
Στον αλγόριθμο CAC (Call Admission Control) [23] νέοι ρυθμοί αφίξεων εκτιμώνται συνεχώς και εάν είναι υψηλότεροι από ένα αποδεκτό επίπεδο μερικές κλήσεις φράσσονται ανεξαρτήτως απ τη διαθεσιμότητα του καναλιού. Ο στόχος του μοντέλου αυτού είναι να διατηρήσει τον νέο ρυθμό αφίξεων κλήσεων μικρότερο από ένα προκαθορισμένο επίπεδο. Σε αυτό το μοντέλο γίνεται μια σύγκριση ανάμεσα στα δύο ήδη υπάρχοντα μοντέλα τα οποία ονομάζονται pre-request scheme (μοντέλο πρότερης αίτησης) και το guard scheme (μοντέλο προστασίας).

Υπάρχουν πολλά πλεονεκτήματα και μειονεκτήματα τα οποία δίνονται για τα δύο αυτά μοντέλα και τότε ένας CAC αλγόριθμος αναπτύσσεται, ο οποίος παρέχει μια καλύτερη ποιότητα υπηρεσίας από τα δύο αυτά μοντέλα. Τα δύο μετρούμενα μεγέθη που χρησιμοποιούνται για την QoS στον αλγόριθμο αυτό είναι η Πιθανότητα Εξαναγκαστικού Τερματισμού (**Force Termination Probability FTP**), που ορίζεται ως η αναλογία του αριθμού των κλήσεων που εξαναγκάζονται να τερματιστούν λόγω ελαττωματικής παράδοσης προς τον πλήθος των κλήσεων που εισήχθησαν επιτυχώς στο δίκτυο. Άλλο ένα μετρικό μέγεθος είναι ο Ρυθμός Επιτυχούς Ολοκλήρωσης Κλήσης (**Successful Call Completion Rate SCCR**), ο οποίος ορίζεται σαν ο αριθμός των κλήσεων οι οποίες είναι επιτυχημένες σε μια μονάδα χρόνου για την κάθε κυψέλη. Έτσι μικρότερη FTP και μεγαλύτερος SCCR είναι το ιδανικό που οι αλγόριθμοι προσπαθούν να επιτύχουν και αυτός ο αλγόριθμος κατάφερε και τα δύο.

2.5.2.1. Μοντέλο Προτεραιότητας Καναλιών

Στο μοντέλο Προστασίας Καναλιού (Guard Channel Scheme), κάποια κανάλια είναι αποκλειστικά δεσμευμένα για την παράδοση κλήσεων και αυτά τα κανάλια καλούνται κανάλια προστασίας. Ένας συγκεκριμένος αριθμός από κανάλια ας πούμε G από τα C κανάλια είναι αποκλειστικά δεσμευμένα για παράδοση κλήσεων σε πελάτες συγκεκριμένων χαρακτηριστικών. Τα εναπομείναντα κανάλια μοιράζονται για όλους τους τύπους κλήσεων. Αυτά τα κανάλια περιλαμβάνουν νέες κλήσεις, παράδοση κλήσεων συγκεκριμένων χρηστών και παράδοση κλήσεων απλών χρηστών. Αυτό είναι ένας στατικός καταμερισμός καναλιών. Στο μοντέλο πρωθύστερης αίτησης καναλιού, τα κανάλια ζητούνται εκ των προτέρων, πριν η παράδοση των κλήσεων αρχίσει να συμβαίνει. Οι πληροφορίες γύρω από τα πρότυπα φορητότητας των ειδικών χρηστών είναι καταχωρημένες σε Home Location Register (HLR). Χρησιμοποιώντας τις πληροφορίες αυτές, η συμπεριφορά παράδοσης που μπορεί να έχουν οι ειδικοί χρήστες μπορεί να προβλεφθεί. Η πρωθύστερη αίτηση καναλιών ζητείται από την γειτονική κυψέλη, για ένα συγκεκριμένο χρονικό διάστημα, που ονομάζεται σαν περίοδος «κατάληψης». Αυξάνοντας την περίοδο κατάληψης, η πιθανότητα για εξαναγκασμένο τερματισμό μπορεί να μειωθεί.

2.5.2.2. Αλγόριθμος Εισόδου Κλήσεων



Εικόνα 5. Διάγραμμα Ροής για τον αλγόριθμο CAC
Πηγή: www.cse.wustl.edu

Στον αλγόριθμο CAC, το ανεκτό φορτίο υπολογίζεται με βάση τα αποτελέσματα της προσομοίωσης και η τιμή αυτή χρησιμοποιείται για λόγους σύγκρισης. Το εκτιμώμενο φορτίο υπολογίζεται επίσης και ελέγχεται με το ανεκτό φορτίο. Εάν το εκτιμώμενο φορτίο είναι μικρότερο ή ίσο με το ανεκτό φορτίο, τότε γίνονται προσπάθειες για τον καταμερισμό των καναλιών για όλες τις εισερχόμενες κλήσεις. Εάν το εκτιμώμενο φορτίο είναι μεγαλύτερο από το ανεκτό φορτίο, τότε μόνο ένα τμήμα από τις εισερχόμενες κλήσεις θα οδηγηθεί σε καταμερισμένα κανάλια και το τμήμα που περισσεύει θα απορριφθεί ακόμη και αν υπάρχουν διαθέσιμα κανάλια. Αυτό ονομάζεται πρωθύστερος φραγμός καναλιών και το μοντέλο βελτιώνει το FTP και το SCCR των ειδικών χρηστών.

2.5.3. Ποιότητα Υπηρεσίας Βασισμένη σε Πρόβλεψη Κινητικότητας

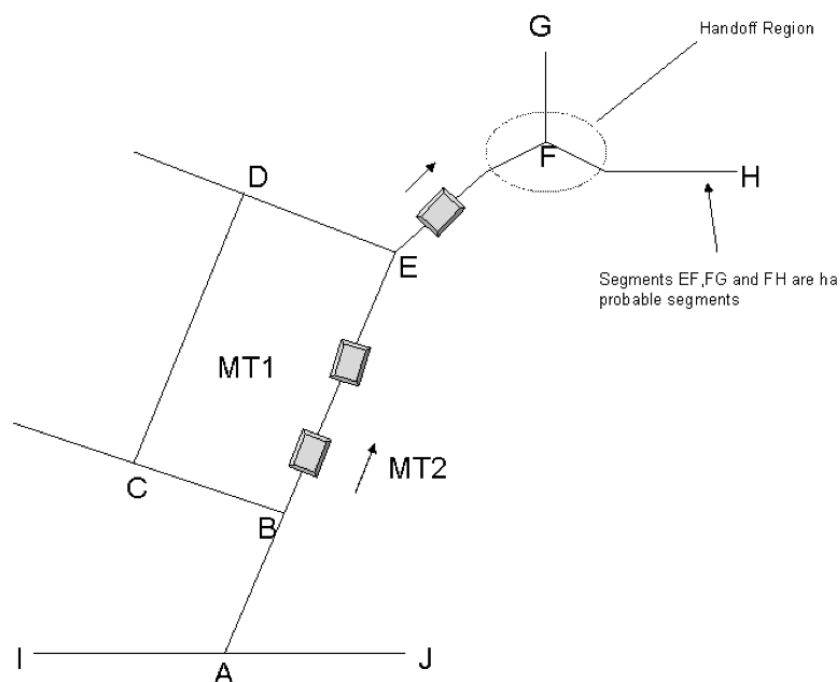
Οι τεχνικές πρόβλεψης κινητικότητας χρησιμοποιούνται για την εύρεση του μονοπατιού ή της τροχιάς ενός κινητού κόμβου, η οποία και αποθηκεύεται σε μια βάση δεδομένων από χρόνο σε χρόνο. Αυτή η τεχνική βοηθά στην εξοικονόμηση πόρων για ΜΗ, πριν την παράδοση μιας κλήσης. Έτσι η απονομή προτεραιότητας στους πόρους ενός κόμβου γίνεται πριν την παράδοσή του και έτσι έχουμε μείωση του ρυθμού φραγής των κλήσεων, κατά τη διάρκεια της παράδοσης [39]. Παρ' όλα αυτά, ο νέος ρυθμός εισόδου κλήσεων μειώνεται όσο δεσμεύονται πόροι για παράδοση κλήσεων. Η εξαναγκασμένη λήξη μπορεί να μειωθεί

με την αύξηση της πιθανότητας φραγής μιας νέας κλήσης. Όμως, δεν αποτελούν αυτά αποδοτική χρήση των πόρων ραδιοφάσματος.

Χρησιμοποιώντας τις τεχνικές πρόβλεψης φορητότητας, μπορούμε να ξέρουμε εκ των προτέρων πότε μια παράδοση θα συμβεί, και έτσι η δυναμική κατανομή των πόρων μπορεί να γίνει για την παράδοση. Αυτό συνεπάγεται, ότι η πιθανότητα φραγής νέων κλήσεων δε θα αυξηθεί τόσο πολύ, ώστε να δεσμεύσει κρίσιμους πόρους για την παράδοση κλήσεων. Υπάρχουν πολλές τεχνικές πρόβλεψης φορητότητας, οι οποίες έχουν χρησιμοποιηθεί στο παρελθόν και οι οποίες περιλαμβάνουν μεθόδους εύρεσης θέσης με χρήση GPS, στις οποίες κάθε ΜΗ είναι ενιαίος με ένα δέκτη GPS, για να μπορεί να ανιχνευτεί το μονοπάτι του ΜΗ.

2.5.3.1. Τοπολογία Δρόμου Χρησιμοποιώντας Τεχνικές Πρόβλεψης Κινητών

Η τεχνική αυτή βασίζεται στο γεγονός ότι οι ΜΗ σε οχήματα θα αντιμετωπίσουν τις πιο συνηθισμένες παραδόσεις, γι αυτό και η μελέτη των χαρακτηριστικών τους θα αποδειχθεί ότι είναι πιο ωφέλιμη. Από τη στιγμή που τα οχήματα ταξιδεύουν επί του δρόμου, η τοπολογία του δρόμου μελετάται και χρησιμοποιείται σε αλγόριθμο πρόβλεψης. Στο μοντέλο αυτό, ο σταθμός βάσης θα κάνει την πρόβλεψη του κινητού, βασισμένος στην τοπολογία του δρόμου.



Εικόνα 6. Πληροφορίες τοπολογίας δρόμου για πρόβλεψη φορητότητας. Πηγή: www.cse.wustl.edu

Από τη στιγμή που ο σταθμός βάσης έχει μεγαλύτερο αποθηκευτικό χώρο, αυτός θα χρησιμοποιηθεί για την αποθήκευση μετρήσεων υψηλότερης ακρίβειας.

Στην τεχνική αυτή υπάρχουν πολλοί σταθμοί βάσης, που ενημερώνονται με την τελευταία θέση του ΜΗ, σε τακτά χρονικά διαστήματα. Κάθε σταθμός βάσης θα διατηρεί μια βάση δεδομένων, που θα περιέχει πληροφορίες γύρω από την τοπολογία του δρόμου. Όπως φαίνεται και στην παραπάνω εικόνα (Εικόνα 6), ο δρόμος αποτελείται από πολλές στροφές,

οι οποίες εμφανίζονται θρυμματισμένες σε ευθύγραμμα τμήματα, και αποθηκεύονται ξεχωριστά. Η βάση δεδομένων έχει πληροφορίες όπως, την μέση τιμή χρόνου, για την διέλευση ενός τμήματος, για γειτονικά τμήματα σε κάθε διασταύρωση, αλλά και την πιθανότητα του ΜΗ να κάνει μια παράδοση και να πάει στο επόμενο γειτονικό τμήμα. Η βάση δεδομένων ανανεώνεται περιοδικά και κάθε φορά οι πληροφορίες για την θέση του ΜΗ ανακτάται. Τα τμήματα στα οποία έχουν γίνει παραδόσεις αναφέρονται σαν τμήματα πιθανών παραδόσεων. Χρησιμοποιώντας το μοντέλο αυτό, γίνεται ακριβής πρόβλεψη, η οποία είναι ωφέλιμη για να μπορεί να δοθεί προτεραιότητα σε πόρους για την παράδοση ενός ΜΗ.

2.5.4. Μοντέλο Δυναμικού Καταμερισμού με την χρήση Επαναδιαπραγμάτευσης

Σε ένα μοντέλο δυναμικού καταμερισμού, που χρησιμοποιεί επαναδιαπραγμάτευση, οι μη χρησιμοποιούμενοι πόροι του δικτύου εξερευνώνται και μοιράζονται σε υπηρεσίες που έχουν μικρότερο εύρος ζώνης, την στιγμή της εισόδου, ενώ αυτές για την σωστή λειτουργία τους χρειάζονται περισσότερο. Με άλλα λόγια, μια επαναδιαπραγμάτευση του εύρους ζώνης γίνεται για μια υπηρεσία μικρότερης προτεραιότητας, όταν το μέσο είναι ελεύθερο. Έτσι αυξάνεται το συνολικό εύρος ζώνης για τις υπηρεσίες μειωμένης προτεραιότητας. Το μοντέλο αυτό θα πρέπει να σημειωθεί πως διατηρεί πάντα εύρος ζώνης για την υπηρεσίες υψηλής προτεραιότητας.

2.5.4.1. Μοντέλο Επαναδιαπραγμάτευσης

Στο μοντέλο επαναδιαπραγμάτευσης, στις κλάσεις συνομιλίας έχει δοθεί η μεγαλύτερη προτεραιότητα, και είναι κλάσεις προτεραιότητας βαθμού 1. Η κλάση streaming έχει επίσης μεγάλη προτεραιότητα και ανήκει στις κλάσεις βαθμού 2. Τώρα οι κλάσεις βαθμού 1 και βαθμού 2 θα δίνονται μό σ εάν υπάρχει ν αρκετοί πόροι (εύρος ζώνης), για να ικανοποιηθούν οι απαιτήσεις τους. Σε αντίθετη περίπτωση δεν θα διαθέτονται [34].

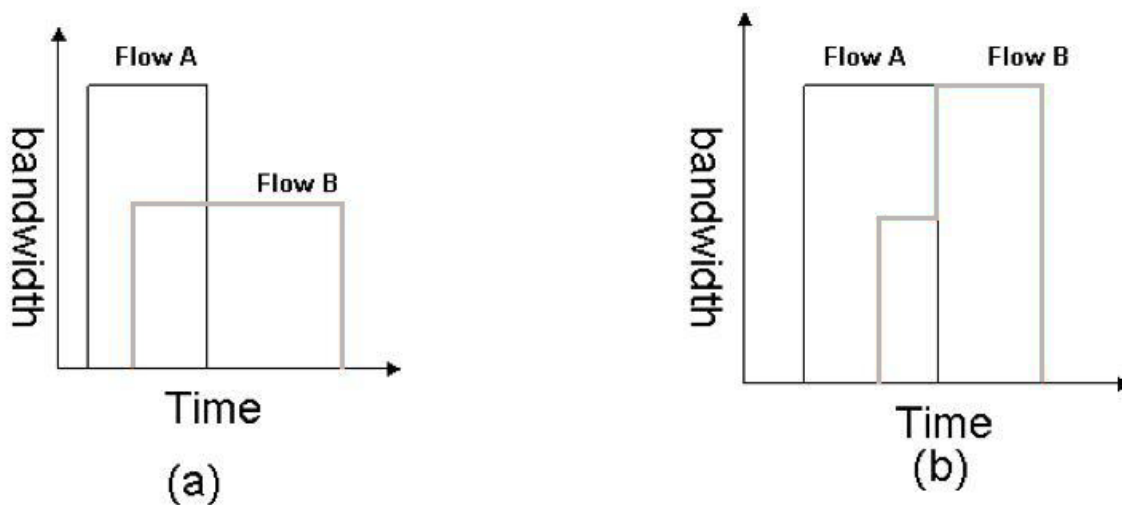
Στην διαδραστική υπηρεσία έχει δοθεί η μικρότερη δυνατή προτεραιότητα και είναι βαθμού 3. Έτσι, οι αιτήσεις από αυτές τις κλάσεις θα γίνονται δεκτές ακόμη και αν το δίκτυο έχει μικρότερο εύρος ζώνης από αυτό που έχουν ζητήσει (οι κλάσεις). Το πλεονέκτημα για το μοντέλο Επαναδιαπραγμάτευσης, όταν συγκρίνεται με το μοντέλο CAC, είναι πως στο μοντέλο CAC το εύρος ζώνης μοιράζεται σε κλάσεις προτεραιότητας 1 και 2, ενώ δεν μπορεί να μεταφερθεί σε μικρότερης τάξης κλάσεις υπηρεσιών, ακόμη και όταν οι υψηλής τάξης κλάσεις εγκαταλείψουν το δίκτυο. Στο μοντέλο επαναδιαπραγμάτευσης, μια εφαρμογή προτεραιότητας 3 μπορεί να χρησιμοποιήσει περισσότερο εύρος ζώνης από αυτό που της έχει μοιραστεί. Αυτό γίνεται εφικτό, επειδή όλοι οι μη χρησιμοποιούμενοι πόροι από τις τάξεις υψηλότερης προτεραιότητας, μπορούν να μεταφερθούν σε αυτή την κλάση.

Από την άλλη πλευρά βέβαια, όταν μια εφαρμογή υψηλότερης προτεραιότητας εκκινήσει, τότε δεν θα φραχθεί από το σύστημα εάν αυτό έχει λίγους πόρους. Την στιγμή αυτή το εύρος ζώνης που δίνεται στην μικρότερης τάξης υπηρεσία, θα αφαιρεθεί και θα δοθεί στην υψηλότερης. Έτσι οι βασικές υπηρεσίες δεν επηρεάζονται καθόλου.

2.5.4.2. Επαναδιαπραγμάτευση με Διακοπή Ροής

Η επαναδιαπραγμάτευση με διακοπή της ροής είναι ένα μέσο, έτσι ώστε περισσότερο εύρος ζώνης να μοιράζεται σε μικρότερης προτεραιότητας ροές, όταν μια υψηλής προτεραιότητας ροή τελειώσει ή όταν υπάρχουν περισσότεροι πόροι στο δίκτυο [17].

Στην παρακάτω εικόνα (Εικόνα 7), φαίνεται η συμπεριφορά δύο διαφορετικών ροών τη μία φορά με χρήση επαναδιαπραγμάτευσης και την άλλη χωρίς. Στην περίπτωση (a) (χωρίς επαναδιαπραγμάτευση) φαίνονται τα εύρη ζωνών των A και B. Ακόμη και μετά τον τερματισμό της ροής A, το εύρος ζώνης παραμένει το ίδιο. Στην περίπτωση (b), μετά τον τερματισμό της ροής A, το εύρος ζώνης της ροής B αυξάνεται λόγω της αύξησης διαθέσιμων πόρων.



Εικόνα 7. Συμπεριφορά των δύο ροών Χωρίς (a) και Με (b) Επαναδιαπραγμάτευση
Πηγή: www.cse.wustl.edu

2.5.5. Ατομική Ποιότητα Υπηρεσίας για Υπηρεσίες Ομιλίας

Η ατομική ποιότητα υπηρεσίας (Individual QoS) είναι μια πολύ σημαντική μέτρηση της QoS και η αξία της συχνά παραβλέπεται [24]. Μέχρι στιγμής, η ποιότητα υπηρεσίας που βασίζεται σε επίπεδο κυψέλης ή δικτύου είναι αυτό που πραγματεύονται οι μελέτες πάνω στην QoS. Η Individual QoS (iQoS) μετρά τον ρυθμό ικανοποίησης ανά πελάτη και αυτή είναι μια από τις πιο σημαντικές παραμέτρους που θα πρέπει να λαμβάνεται υπ' όψιν. Γίνεται ξεχωριστή μέτρηση σε κάθε ένα πελάτη - τερματικό.

Η εκτίμηση της συνολικής QoS στο δίκτυο δεν δίνει μια πολύ ακριβή εικόνα για το τι συμβαίνει σε κάθε έναν χρήστη ξεχωριστά. Για παράδειγμα ο μέσος ρυθμός χαμένων κλήσεων μέσα σε όλο το δίκτυο μπορεί να είναι τόσο χαμηλός όσο 1-2%. Παρ' όλα αυτά, μερικοί χρήστες μπορεί να βιώσουν μεγαλύτερη ποσότητα, αλλά και συχνότητα στις χαμένες κλήσεις, ακόμη και το 30% των κλήσεών τους μπορεί να χαθούν. Αυτό σημαίνει ότι το καθολικό επίπεδο QoS είναι υψηλό (μεγαλύτερο από 95%), αλλά μερικοί χρήστες έχουν γίνει

θύματα μιας μέτριας διαδικασίας και χρεώνονται από τους διαχειριστές τους για μια QoS την οποία δεν λαμβάνουν.

Αποδεικνύεται, ότι οι χρήστες με μικρότερη κάλυψη σήματος συχνά βιώνουν τέτοιου είδους απώλειες και τότε ο μέσος όρος δεν μπορεί ή δεν πρέπει να λαμβάνεται υπ' όψιν και η μη ικανοποίηση αυτών των χρηστών να κρύβεται στα νοήματα του μέσου όρου. Έτσι, είναι πολύ σημαντικό να μετράμε την ικανοποίηση του κάθε χρήστη σε ένα δίκτυο.

2.5.5.1. Λόγος Μέτρησης Ατομικής Ποιότητας Υπηρεσίας

Μπορεί να έχει παρατηρηθεί ότι, ενώ το παγκόσμιο σχέδιο για την διαχείριση της QoS πραγματοποιείται σε ένα δίκτυο, εξ' ορισμού δεν μπορεί να παράξει ποτέ την ίδια ποσότητα ικανοποίησης πελατών στο ίδιο επίπεδο. Η ισχύς του σήματος που λαμβάνει ένας χρήστης, εξαρτάται από την φυσική του θέση σε σχέση με το σταθμό βάσης. Σε χρήστες οι οποίοι είναι κοντά στον Σταθμό Βάσης, μπορεί να παρατηρείται πολύ καλή ισχύ σήματος και ποιότητα υπηρεσιών φωνής, αλλά από χρήστες που είναι πολύ μακριά από τον σταθμό βάσης, ίσως βιώνεται μια πολύ φτωχή σύνδεση στο δίκτυο και επίσης πολλές χαμένες κλήσεις.

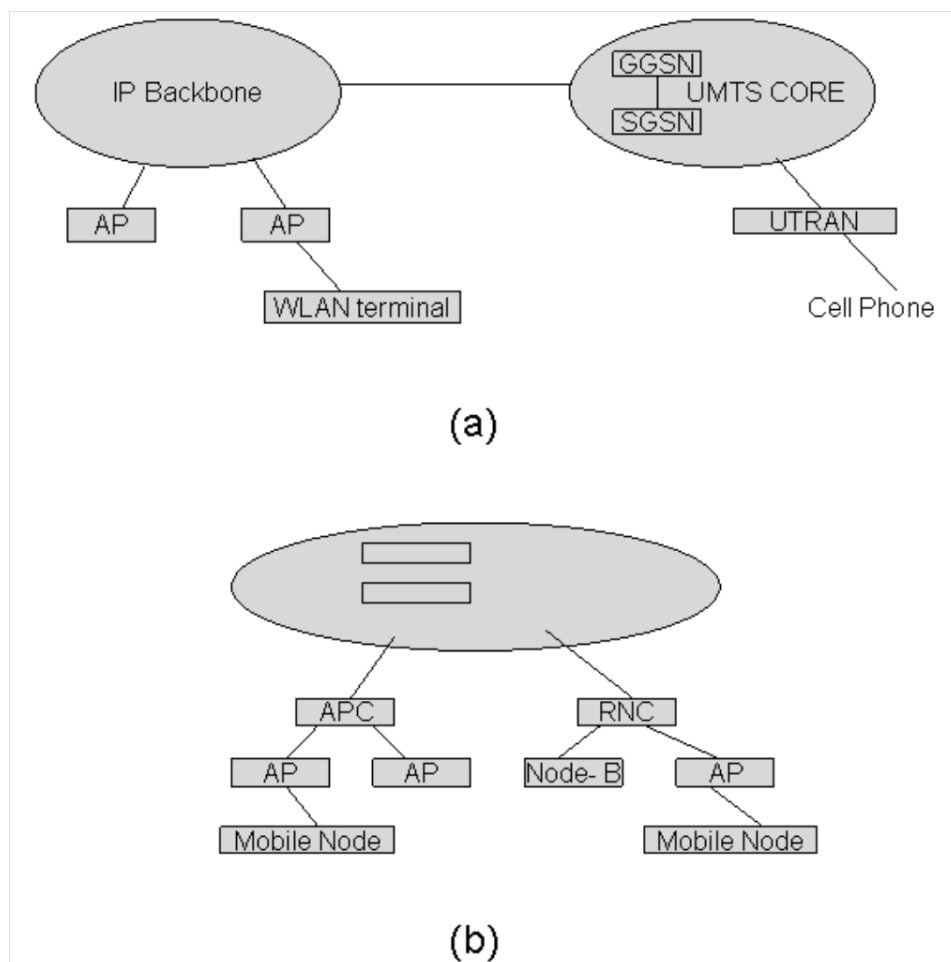
Ο ενδιάμεσος χαρακτηρισμός του επιπέδου σήματος ενός ραδιοδιαύλου για ένα συγκεκριμένο σημείο μπορεί να ποικίλει από 10-20dB, ακόμη και μέσα σε μια μικρή απόσταση μερικών δεκάδων μέτρων. Έτσι θα ήταν πολύ καλύτερο, εάν η iQoS περιλαμβανόταν στον λογαριασμό, έτσι ώστε κάθε ξεχωριστός χρήστης να είναι ενήμερος για το τι QoS λαμβάνει και να ξέρει φυσικά αν η ποιότητα που παίρνει είναι και αυτή που πληρώνει.

2.5.5.2. Ιδέα της Αξιολόγησης της Ατομικής Ποιότητας Υπηρεσίας

Είναι προφανές ότι από την οπτική γωνία του χρήστη, η απώλεια κλήσεων, καθώς και η βλάβη του συστήματος, είναι το πιο ενοχλητικό γεγονός το οποίο θα πρέπει να πληρώσει εάν αυτό συμβεί κατά τη διάρκεια του χρόνου χρέωσής του. Έτσι, για τη μέτρηση της iQoS πρώτα θα πρέπει να γίνει διαχωρισμός ανάμεσα στους χρήστες, όπου δεν έχουν λάβει κάποια χαμένη κλήση και στους χρήστες, όπου λόγω συνθηκών δεν μπορέσει να γίνει κλήση. Η Ποιότητα Υπηρεσίας μπορεί να μετρήσει μόνο αυτούς που βιώνουν κακή εξυπηρέτηση όπως χαμένες κλήσεις και φραγμό κλήσεων

2.6. Συνδυασμός Κυβελωτών και Ασύρματων δικτύων για επίτευξη καλύτερης Ποιότητας Υπηρεσίας

Τα Wireless LAN συστήματα παρέχουν υψηλής ταχύτητας ρυθμούς μετάδοσης [25], αλλά μικρότερη ικανότητα φορητότητας, ενώ από την άλλη πλευρά τα κυβελωτά συστήματα παρέχουν υψηλότερη φορητότητα, αλλά όχι τόσο υψηλούς ρυθμούς μετάδοσης. Έτσι ο συνδυασμός αυτών των δύο ειδών δικτύων έχει ως αποτέλεσμα την επίτευξη υψηλών ταχυτήτων μετάδοσης, αλλά και υψηλού επιπέδου φορητότητας. Το θέμα που εγείρεται με τον συγκεκριμένο συνδυασμό δικτύων είναι πως η ποιότητα υπηρεσιών των κυβελωτών συστημάτων είναι ευκολότερα παρεχόμενη, αλλά επίσης έχει οριστεί πολύ καλύτερα σε σχέση με την ποιότητα υπηρεσιών σε ένα IP δίκτυο. Δεν υπάρχει εγγυημένη QoS σε ένα δίκτυο IP και έτσι με τον συνδυασμό των δύο αυτών δικτύων θα προκύψουν θέματα σχετικά με την QoS, τα οποία θα είναι πολύ δύσκολο να επιλυθούν σε ένα ενιαίο σύστημα δικτύων.

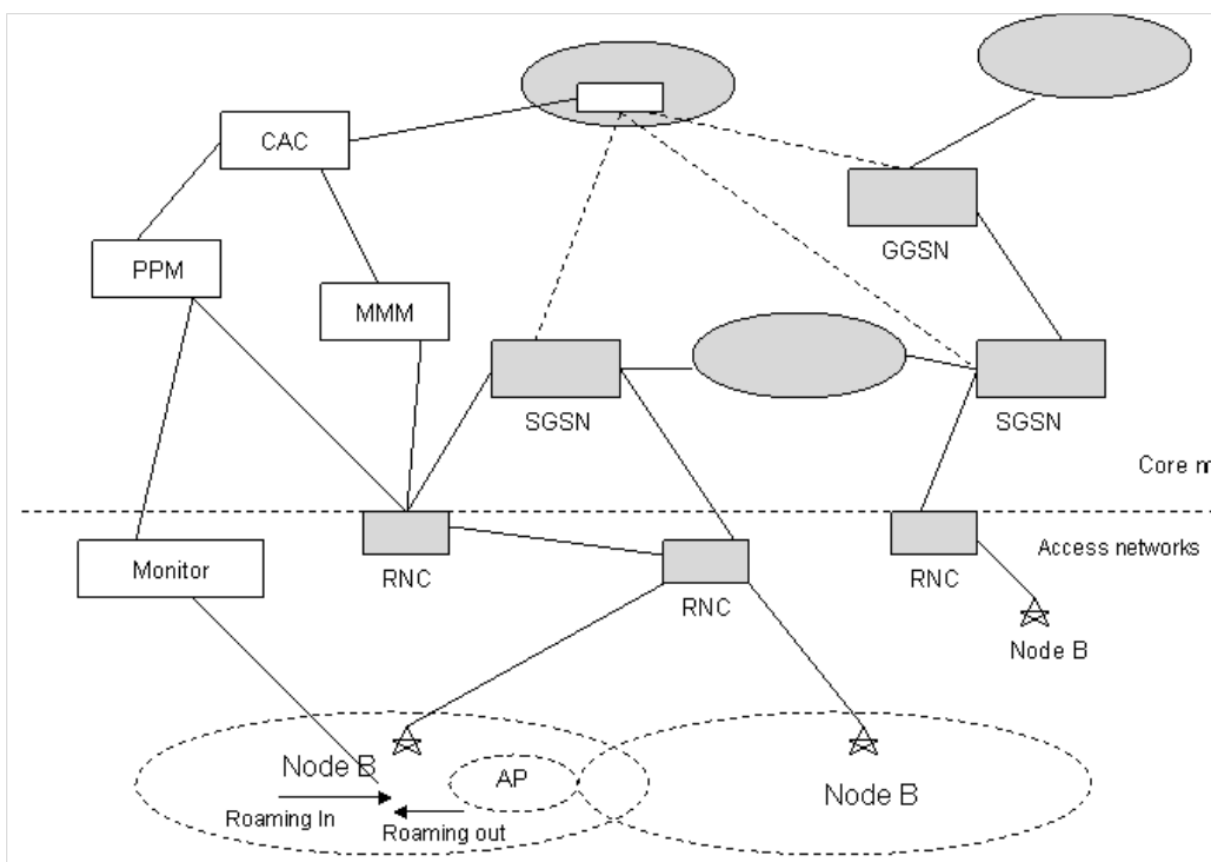


Εικόνα 8. Χαλαρή διασύνδεση (a) και Αυστηρή Διασύνδεση (b)
Πηγή: www.cse.wustl.edu

Στην αυστηρή διασύνδεση, η τεχνολογία WLAN χρησιμοποιείται σαν μια νέα τεχνολογία ράδιο-εισόδου μέσα σε ένα κυψελωτό δίκτυο. Άσχετα από την τεχνολογία εισόδου, θα υπάρχουν κοινά κυψελωτά δίκτυα πυρήνα. Αυτό μπορεί να γίνει συνδέοντας το WLAN AP(Access Point) στον Radio Network Controller (RNC) [38]. Η χαλαρή διασύνδεση από την άλλη πλευρά περιλαμβάνει μικρότερης έκτασης σύνδεση ανάμεσα στα δύο δίκτυα. Το δίκτυο εισόδου WLAN είναι προσκολλημένο στην ραχοκοκαλιά του Internet και το κυψελωτό δίκτυο στον πυρήνα του κυψελωτού δικτύου.

2.6.1. Δομή Μοντέλου Ποιότητας Υπηρεσίας Αυστηρής Διασύνδεσης

Στην αρχιτεκτονική Αυστηρής Διασύνδεσης, το κυψελωτό δίκτυο βασίζεται σε ένα δίκτυο μεταγωγής πακέτων IP. Η αρχιτεκτονική αυτή προϋποθέτει έναν πυρήνα δικτύου μεταγωγής πακέτου, το οποίο βασίζεται πάνω στην αρχιτεκτονική δικτύου UMTS. Η παρακάτω εικόνα (Εικόνα 9) δείχνει τα μέλη της αρχιτεκτονικής Αυστηρής Διασύνδεσης.



Εικόνα 9. Αρχιτεκτονική Μοντέλου QoS Αυστηρής Διασύνδεσης
Πηγή: www.cse.wustl.edu

Policy Provisioning Module (PPM). Το PPM παίρνει την ποιότητα υπηρεσιών των χρηστών και αποφασίζει σε ποια κλάση ανήκει η κίνηση όσον αφορά την προτεραιότητα και τότε παραδίδει την αίτηση του χρήστη στον CAC για περαιτέρω επεξεργασία, δηλαδή για το αν θα επιτραπεί η πρόσβαση ή όχι.

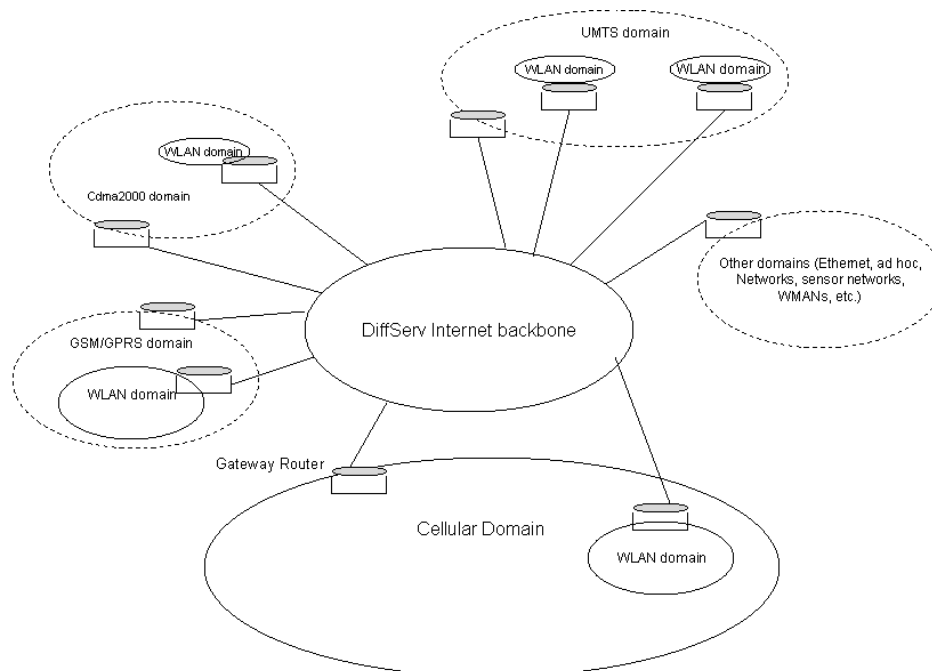
Connection Admission Control Module (CAC). Το μοντέλο CAC δέχεται τις ροές κίνησης και διανέμει το εύρος ζώνης σε όλες τις ροές. Μπορεί επίσης να απορρίψει ροές εάν δεν υπάρχει διαθέσιμο εύρος ζώνης για μια συγκεκριμένη ροή και να διατηρεί την ποιότητα

υπηρεσίας για τις ήδη υπάρχουσες συνδέσεις. Το μοντέλο CAC αρχικά λαμβάνει μια αίτηση σύνδεσης από το PPM και στη συνέχεια συμβουλευτεί το Μοντέλο Διαχείρισης Κινητικότητας (Mobility Management Module – MMM) για να ενημερωθεί για την κατάσταση φορητότητας και τις πληροφορίες των κόμβων. Τότε χρησιμοποιεί πρωτόκολλα κράτησης, όπως είναι το RSVP, για να δεσμεύσει συνδέσεις.

QoS Mobility Management Module (MMM). Το MMM παρακολουθεί τα τερματικά τα οποία δεν είναι πλέον συνδεδεμένα στο δίκτυο, τα τερματικά που είναι ανενεργά, καθώς και αυτά που είναι συνδεδεμένα στο δίκτυο. Επιπλέον καταγράφει τους συνδεδεμένους κόμβους, οι οποίοι κινούνται σε υψηλές ταχύτητες. Θέματα, όπως είναι η αλλαγή από ένα κυψελωτό σε ένα WLAN δίκτυο, καθώς και το αντίθετο, επιλύονται από το μοντέλο αυτό. Μια επίσης σημαντική πλευρά του MMM είναι πως καταγράφει την ποιότητα υπηρεσίας των χρηστών και εάν αυτή δεν συμφωνεί με τις προδιαγραφές που έχουν θεσπιστεί για τον συγκεκριμένο χρήστη έχει μηχανισμό ανατροφοδότησης με τον οποίο πληροφορεί τον CAC για την τρέχουσα QoS του χρήστη και προκαλεί ενέργειες για τη διόρθωση και την βελτίωση της.

2.6.2. Χαλαρή Διασύνδεση Κυψελωτού με Ασύρματο Τοπικό Δίκτυο

Σε αυτή την χαλαρή διασύνδεσης αρχιτεκτονική δεν υπάρχει άμεσος σύνδεσμος ανάμεσα στα WLAN και στα κυψελωτά δίκτυα. Η πύλη (gateway) συνδέει τα WLAN με τη ραχοκοκαλιά του Internet [17]. Το πλεονέκτημα αυτής της προσέγγισης είναι πως δεν υπάρχει εξάρτηση ανάμεσα στα δύο δίκτυα. Το κάθε ένα ξεχωριστά μπορεί να λειτουργήσει ανεξάρτητα και το ένα δίκτυο δεν θα είναι τροχοπέδη για το άλλο. Μια καθολική IP DiffServ πλατφόρμα είναι η πιο δημοφιλής αρχιτεκτονική για αυτό το μοντέλο χαλαρής διασύνδεσης.



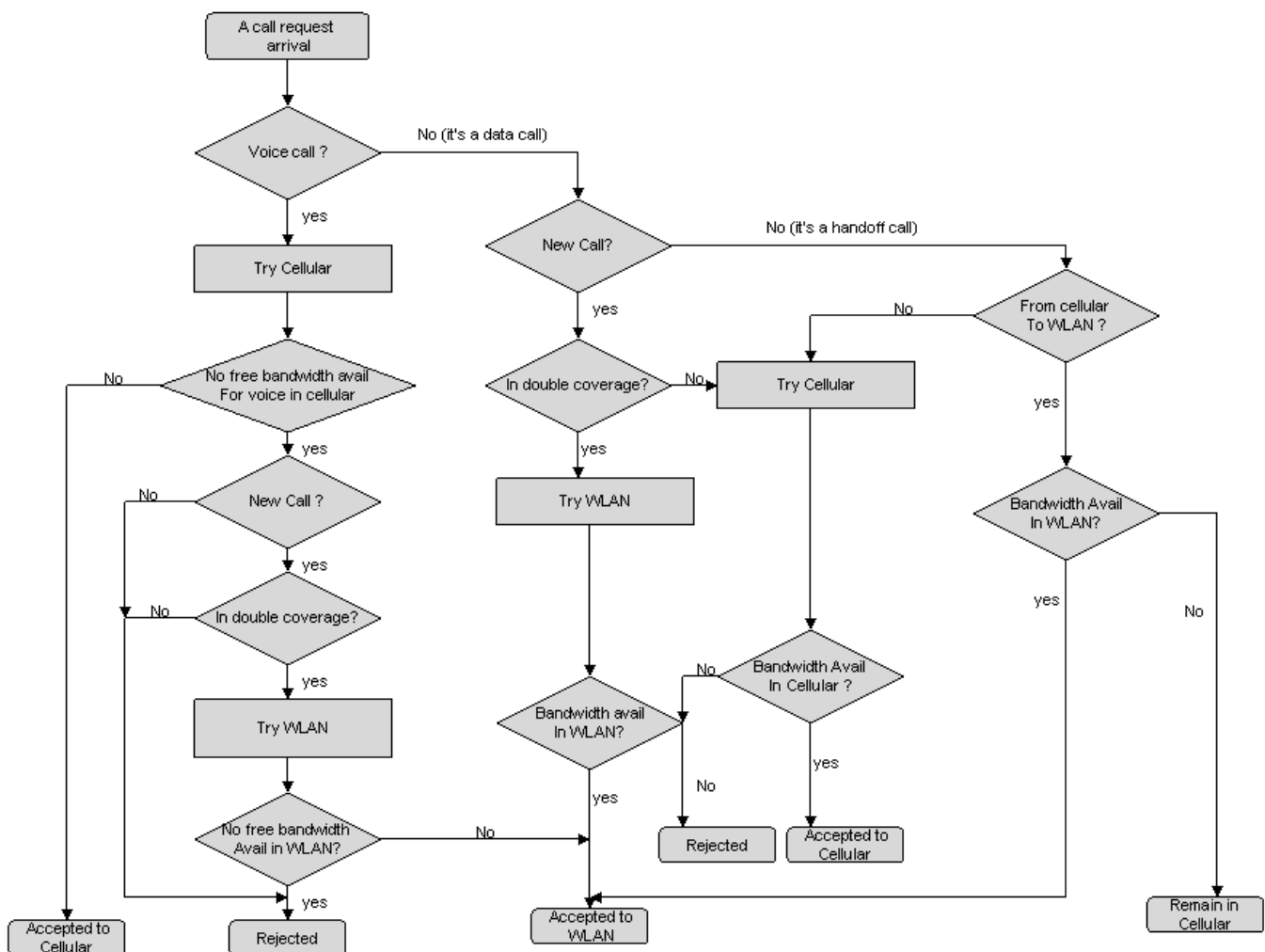
Εικόνα 10. Αρχιτεκτονική Χαλαρής Διασύνδεσης WLAN/κυψελωτών σε πλατφόρμα DiffServ
Πηγή: www.cse.wustl.edu

Η παραπάνω εικόνα (Εικόνα 10) [25] απεικονίζει την Αρχιτεκτονική Χαλαρής Διασύνδεσης WLAN/κυψελωτών σε πλατφόρμα DiffServ. Αυτό παρουσιάζει την ολοκλήρωση ενός

WLAN με διαφορετικές κυψελωτές αρχιτεκτονικές, όπως είναι το UMTS, η Code Division Multiple Access (CDMA2000) και το GSM/GPRS. Η DiffServ πλατφόρμα είναι ένα domain, πάνω στο οποίο διάφορα άλλα domains μπορούν ελεύθερα να επιλέξουν τους μηχανισμούς συστήματος που θα αξιοποιήσουν, αλλά και θα τηρήσουν τους όρους των Συμφωνητικών Επιπέδων Εξυπηρέτησης (βλ. επόμενο κεφάλαιο), τα οποία έχουν γίνει με γειτονικά domain.

2.6.2.1. Η αποδοχή κλήσεων για υπηρεσίες ομιλίας και δεδομένων

Δύο από τις βασικότερες υπηρεσίες που παρέχονται σε δίκτυο κινητής τηλεφωνίας περιγράφονται εδώ. Η κίνηση σε υπηρεσίες ομιλίας είναι εξαιρετικά ευαίσθητη στην καθυστέρηση και απαιτεί μετάδοση σε πραγματικό χρόνο. Από την άλλη πλευρά, είναι αρκετά ανθεκτική στην απώλεια δεδομένων. Αντιθέτως, η κίνηση δεδομένων δεν είναι τόσο ευαίσθητη στην καθυστέρηση, αλλά είναι εξαιρετικά ευαίσθητη στην απώλεια πακέτων και στους ρυθμούς σφαλμάτων. Οι υπηρεσίες ομιλίας πάντα λάμβαναν προτεραιότητα μεγαλύτερη από αυτές των υπηρεσιών δεδομένων σε κυψελωτά συστήματα. Από την άλλη πλευρά, η κίνηση πακέτων είναι πολύ πιθανό να καταλήξει σε κάλυψη WLAN, αφού επιτυγχάνεται καλύτερο εύρος ζώνης σε αυτό το πεδίο.



Εικόνα 11. Διαδικασία αποδοχής κλήσης για κυψελωτά/WLAN αλληλένδετα δίκτυα. Πηγή: www.cse.wustl.edu

Στην παραπάνω εικόνα (Εικόνα 11) φαίνεται με κάθε λεπτομέρεια η διαδικασία που εκτελείται, για την αποδοχή μιας κλήσης ή δεδομένων σε ένα σύστημα αλληλεξάρτησης WLAN και κυψελωτών δικτύων. Μια περιοχή, στην οποία η κάλυψη είναι μόνο από το κυψελωτό δίκτυο, ονομάζεται cellular-only area, ενώ εάν έχει κάλυψη τόσο κυψελωτού δικτύου όσο και WLAN ονομάζεται double coverage area. Για τις κλήσεις ομιλίας υπάρχουν δύο επιλογές. Μπορεί να ελεγχθεί και η cellular area και η WLAN area. Εάν μια κλήση ομιλίας απορριφθεί σε μια cellular area, τότε θα επιχειρήσει την πλήρωσή της από μια WLAN area. Η αίτηση απορρίπτεται αν δεν υπάρχει ικανοποιητικό διαθέσιμο εύρος ζώνης, για να υποστηρίξει κλήσεις σε αυτή την περιοχή.

Μια κλήση δεδομένων έχει μοναδική επιλογή την WLAN area. Εάν απορριφθεί, τότε δε θα υπάρξει αναζήτηση στην cellular area, αφού η υπερχειλίση των δεδομένων θα προκαλέσει πτώση της ποιότητας υπηρεσίας για τις υπηρεσίες ομιλίας, γεγονός που δεν είναι επιθυμητό. Για την παροχή προτεραιότητας σε μια περιοχή που καλύπτεται μόνο από κυψελωτό δίκτυο, θα πρέπει να υιοθετηθεί πολιτική Limited Fractional Guard Channel (LFG).

2.7. Ανάλυση δικτύων 3.5G και 4G.

Οι διαχειριστές κυψελωτών δικτύων βίωσαν μια εκρηκτική ανάπτυξη στην χρήση των ευρυζωνικών δικτύων. Η αναλογία του φορτίου κίνησης ανά συνδρομητή έχει αρχίσει να αυξάνεται μέρα με την μέρα, ειδικότερα μετά την είσοδο των νέας γενιάς κινητών στην αγορά (smart phones). Επιπλέον οι υπηρεσίες που παρέχονται ακολουθούν ένα διαφορετικό μοτίβο και από ανεξάρτητες και μονομερείς έχουν γίνει πολυμερείς και αντίστοιχα εξαρτώμενες από άλλες υπηρεσίες. Βασικό παράδειγμα πολυμερών υπηρεσιών είναι η multimedia telephony καθώς και η mobile-TV ενώ το real time multimedia gaming κερδίζει συνεχώς έδαφος. Οι υπηρεσίες αυτές έχουν διαφορετικές απαιτήσεις λειτουργίας, όπως για παράδειγμα αυξημένο ρυθμό μετάδοσης bit αλλά και μειωμένη καθυστέρηση πακέτων. Για τον λόγο αυτό η τεχνολογική κοινότητα στράφηκε στην εξέλιξη των ήδη υπάρχοντων δικτύων αλλά και στην εισαγωγή νέων τεχνο λογιών. Αποτέλεσμα ήταν η δημιουργία τεχνολογιών δικτύων που ανήκουν στα λεγόμενα 3.5G και 4G [42].

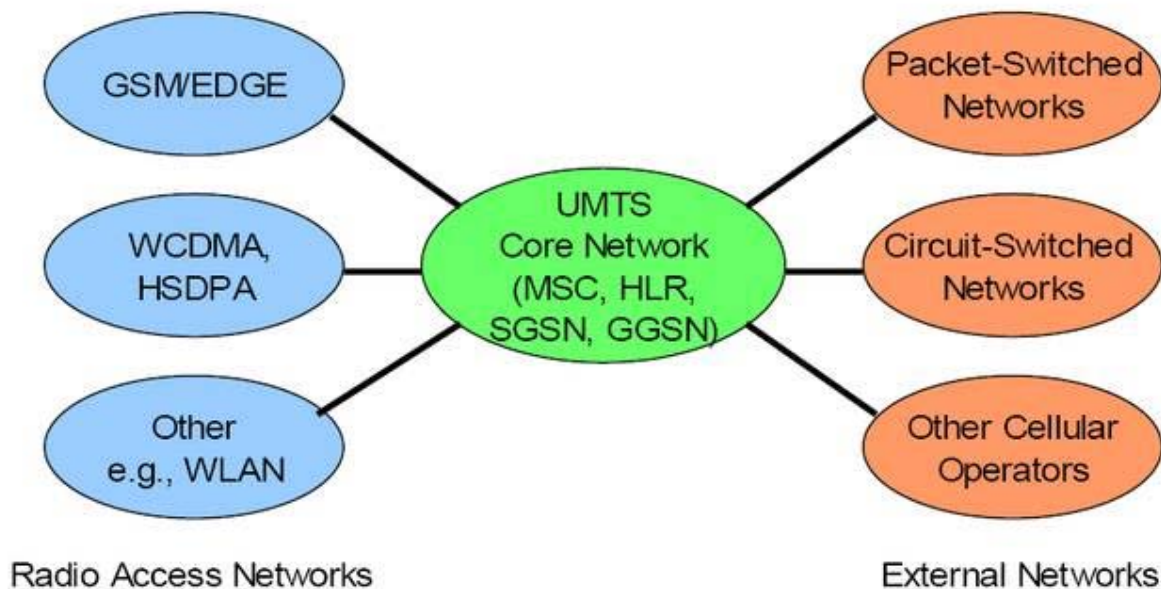
Οι τεχνολογίες 3.5G έχουν κυρίως βασιστεί στις υποδομές των δικτύων τρίτης γενιάς. Χαρακτηριστικό παράδειγμα είναι το πρωτόκολλο HSPA (High Speed Packet Access) το οποίο στηριζόμενο στις υποδομές του δικτύου UMTS δίνει μεγαλύτερους ρυθμούς μετάδοσης και καλύτερη ποιότητα υπηρεσιών.

Οι τεχνολογίες τέταρτης γενιάς (4G) όπως είναι οι IEEE 802.16e/m και Third Generation Partnership Project (3GPP) – Long Term Evolution (LTE) έχουν σχεδιαστεί με καθαρά διαφορετικό μοντέλο για την παροχή QoS με σκοπό την υποστήριξη όχι μόνο των ήδη υπάρχουσών εφαρμογών και απαιτήσεων αλλά και των επερχόμενων εφαρμογών διαδικτύου και των απαιτήσεων τους. Η ποιότητα υπηρεσιών είναι ένα καθοριστικό στοιχείο για τις αναπτυσσόμενες εφαρμογές διαδικτύου αφού αποτελεί κλειδί ανάμεσα στην ικανοποίηση των πελατών αλλά και στην εξοικονόμηση πόρων δικτύου.

2.7.1. High Speed Packet Access

Πρόκειται για τον συνδυασμό δύο πρωτοκόλλων κινητής τηλεφωνίας του High Speed Downlink Packet Access και του High Speed Uplink Packet Access τα οποία επεκτείνουν και βελτιώνουν την απόδοση των πρωτοκόλλων WCDMA [43] χρησιμοποιώντας τον πυρήνα του δικτύου UMTS. Πολλές φορές το εν λόγω πρωτόκολλο αναφέρεται και ως τεχνολογία 3.5G αφού ουσιαστικά αναβαθμίζει τις υπηρεσίες τρίτης γενιάς παρέχοντας στους χρήστες αυξημένες ταχύτητες λήψης πληροφοριών. Σε πρώτη φάση η ταχύτητα λήψης των πληροφοριών με την χρήση της συγκεκριμένης τεχνολογίας είναι έως και τρεις φορές μεγαλύτερη από αυτήν που παρέχουν τα δίκτυα 3G - και πλέον μπορεί να συγκριθεί με αυτή μιας σχετικά γρήγορης σταθερής «γραμμής» ADSL.

Τα πρωτόκολλα αυτά μπορούν να υποστηρίξουν ρυθμούς μετάδοσης για κινήσεις στο τμήμα downlink με μέγιστη τιμή 14Mbps και 5.76Mbps στις κινήσεις τμήματος uplink. Επιπλέον έχει μειωθεί η καθυστέρηση και παρέχεται περίπου πέντε φορές μεγαλύτερη χωρητικότητα στο downlink και περισσότερο από δύο φορές μεγαλύτερη χωρητικότητα στο uplink σε σχέση με τα αντίστοιχα μεγέθη που επιτυγχάνουμε με τα πρωτόκολλα WCDMA.



Εικόνα.12 Δομή και σύνδεση των δικτύων με τον πυρήνα του UMTS.

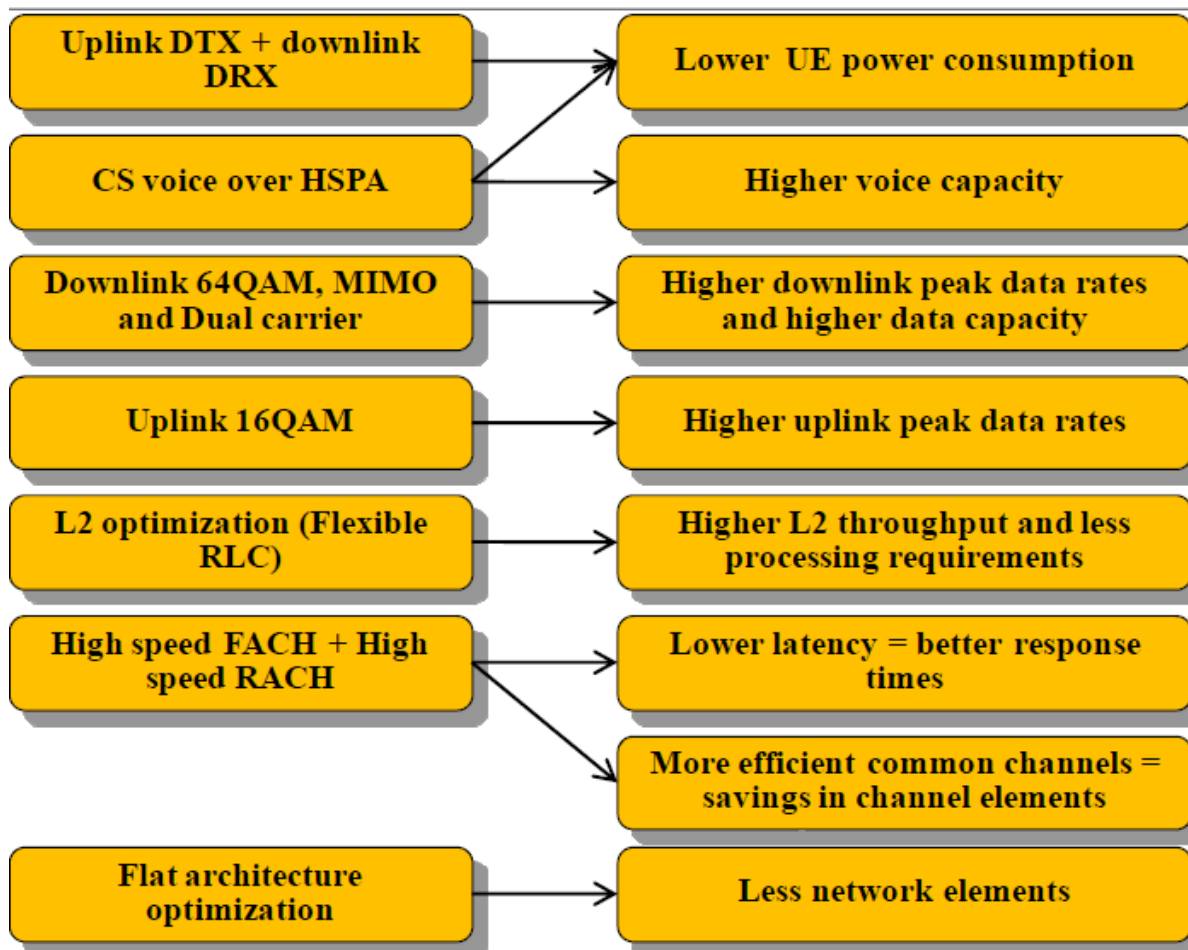
2.7.1.1. High Speed Downlink Packet Access

Αναλυτικότερα το HSDPA ή αλλιώς High Speed Downlink Packet Access είναι η τεχνολογία που αναβαθμίζει τα δίκτυα κινητής τηλεφωνίας WCDMA. Για την επίτευξη αυτών των ταχυτήτων το HSDPA εισάγει ένα High Speed Downlink Shared Channel στα δίκτυα WCDMA, που λειτουργεί διαφορετικά από τα υπάρχοντα κανάλια. Το κανάλι HS-DSCH εισάγει την χρήση προσαρμοστικής διαμόρφωσης και κωδικοποίησης των μεταδιδόμενων πληροφοριών (Adaptive Modulation and Coding, AMC), τον γρήγορο χρονοπρογραμματισμό στον σταθμό βάσης καθώς και τις ταχύτερες επανεκπομπές των πληροφοριών. Με το ν τρόπο αυτό γίνεται εφικτή η σημαντική αύξηση του ρυθμού μεταφοράς των δεδομένων καθώς και η βελτιστοποίηση της διεκπεραίωσης των μεταδόσεων.

2.7.1.2. High Speed Uplink Packet Access

Είναι η τεχνολογία που επιτρέπει την μεταφορά δεδομένων από το τερματικό στον σταθμό βάσης σε ταχύτητες που θεωρητικά φτάνουν τα 5.76Mbps. Πρακτικά το HSUPA επιτρέπει στον τελικό χρήστη να στέλνει δεδομένα με υψηλές ταχύτητες. Αναφέρεται συχνά και ως τεχνολογία 3.75G. Η χρήση του καθιστά δυνατή την πρόσβαση σε συμμετρικές εφαρμογές υψηλής ταχύτητας, όπως είναι οι υπηρεσίες Voice over IP και interactive multimedia, παρέχοντας.

Στην εικόνα (Εικόνα 13) που ακολουθεί φαίνονται συγκεντρωτικά οι λειτουργίες του HSPA καθώς και τα οφέλη που έχει ο χρήστης όσον αφορά την ποιότητα υπηρεσίας που λαμβάνει.



Εικόνα 13. Συγκεντρωτική παράθεση των Λειτουργιών του HSPA και των Οφελών του.

2.7.2. 3GPP - Long Term Evolution

3GPP Long Term Evolution είναι το όνομα που δόθηκε σε ένα νέο επίπεδο τεχνολογιών που αναπτύχθηκε από την 3GPP για να ανταπεξέλθει στην συνεχώς αυξανόμενη απαίτηση διέλευσης της αγοράς. Το LTE είναι η εξέλιξη των συστημάτων δεύτερης και τρίτης γενιάς, ενώ ταυτόχρονα είναι και το επόμενο βήμα για την επίτευξη και παροχή ασύρματων ρυθμών μεταφοράς δεδομένων παραπλήσιων με αυτούς που συναντάμε στα ενσύρματα μέσα [42]. Οι κύριοι στόχοι που τέθηκαν κατά το σχεδιασμό του ήταν:

- Η υποστήριξη υψηλών ρυθμών μετάδοσης, οι οποίοι έχουν μέγιστη τιμή για το Downlink 100Mbps και για το Uplink 50Mbps.
- Εξοικονόμηση τους εύρους ζώνης, για το λόγο αυτό οι ρυθμοί αυτοί υλοποιούνται μέσα σε εύρος ζώνης 20MHz.
- Υποστηριζόμενη mobility μέχρι και 350km/h.
- Ευελιξία φάσματος με συνύπαρξη των ήδη υπάρχουσών τεχνολογιών αλλά και μειωμένη πολυπλοκότητα και κόστος.

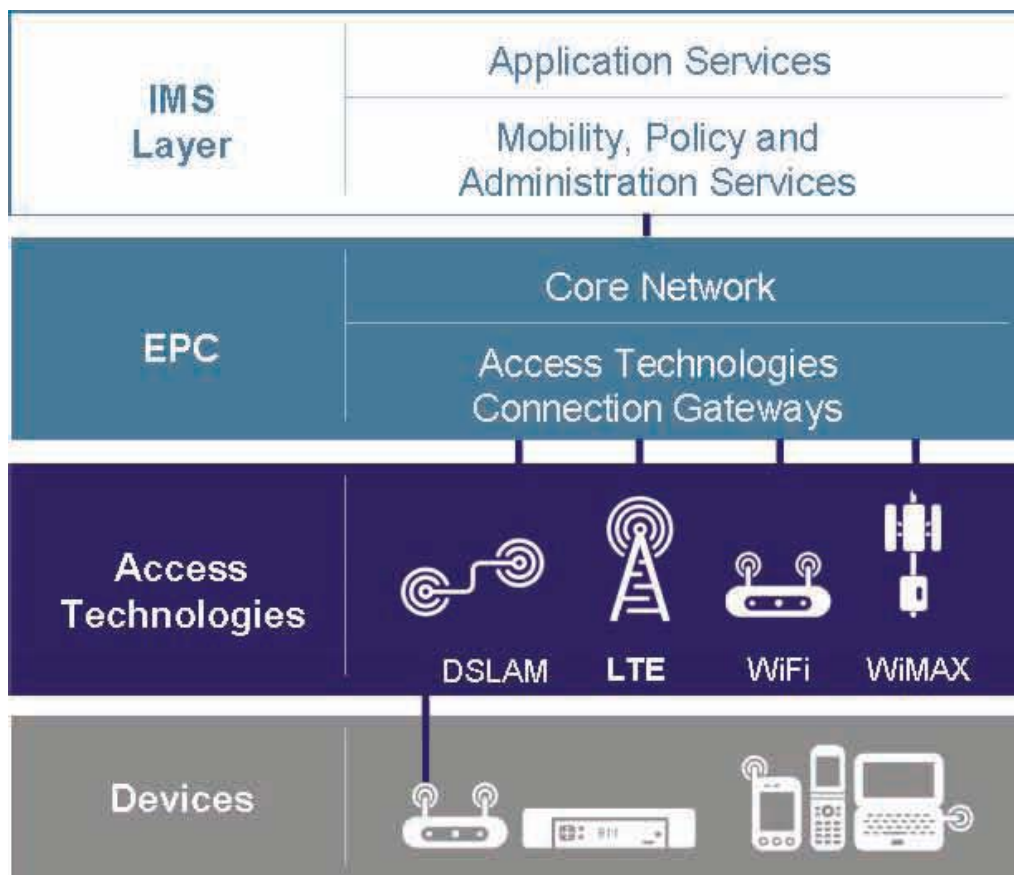
Για την επίτευξη των στόχων αυτών ήταν αναγκαίο να γίνει ένας συνδυασμός μιας νέας αρχιτεκτονικής συστήματος μαζί με μια ενδυναμωμένη τεχνολογία ασύρματης πρόσβασης. Υπάρχουν πολλοί διαφορετικοί τύποι λειτουργιών μέσα σε ένα κυψελωτό σύστημα. Βασισμένοι σε αυτούς μπορούμε να χωρίσουμε το δίκτυο σε δύο βασικά μέρη.

- Τμήμα Ασύρματης Πρόσβασης στο δίκτυο.
- Τμήμα Πυρήνα του δικτύου.

Το **Δίκτυο Ασύρματης Πρόσβασης** για τα LTE ονομάζεται E-UTRAN (Evolved UMTS Radio Access Network) και αναμένεται να βελτιώσει τους ρυθμούς διέλευσης των τελικών χρηστών, την χωρητικότητα των τομέων κυψελών αλλά και να μειώσει την καθυστέρηση που αντιλαμβάνεται ένας χρήστης. Ένα από τα κύρια χαρακτηριστικά του είναι πως όλες οι υπηρεσίες, συμπεριλαμβανομένου και των υπηρεσιών πραγματικού χρόνου, υποστηρίζονται πλέον με την χρήση πρωτοκόλλων IP. Η κίνηση για υπηρεσίες ομιλίας θα υποστηρίζεται πλέον αποκλειστικά ως Voice over IP (VoIP) δίνοντας την δυνατότητα για μεγαλύτερη κλιμάκωση των υπηρεσιών πολυμέσων. Η προσέγγιση αυτή επιτυγχάνει αυξημένη φασματική απόδοση, η οποία με τη σειρά της οδηγεί σε αυξημένη χωρητικότητα δικτύου.

Κάποια βασικά χαρακτηριστικά του E-UTRAN είναι:

- Downlink OFDM 100Mbps (20MHz φάσμα)
- Uplink SC-FDMA 50 Mbps (20MHz φάσμα)
- Χρησιμοποιούνται τεχνολογίες βασισμένες στην OFDM με αποτέλεσμα να υπάρχει διάσπαση των δεδομένων σε πολλά φέροντα σήματα, γεγονός που παρέχει ανοσία στην εξασθένηση και οδηγεί στην αύξηση της αξιοπιστίας παράδοσης.
- Καθυστέρηση Τελικού Χρήστη μικρότερη από 10ms
- Ευέλικτο και Κλιμακούμενο Εύρος Ζώνης
- Επιλογή φάσματος συχνότητας για υποστήριξη προγενέστερων τεχνολογιών ίσως διαφορετικών και από χώρα σε χώρα.
- Θα υποστηρίζεται η κίνηση ακόμη και σε ταχύτητες που φτάνουν τα 500kmph.
- Η κάλυψη θα επεκτείνεται από 5 – 100 km με κλιμάκωση μετά τα 30km
- Η χωρητικότητα για υπηρεσίες τηλεφωνίας που θα δώσει το VoIP θα είναι περίπου τριπλάσια από αυτή που μας δίνει το UMTS.
- Ποιότητα Υπηρεσίας απ' άκρο σ' άκρο το οποίο θα διαχωρίζει την προτεραιότητα για την κάθε κλάση υπηρεσίας.



Εικόνα 14. Δομή και αρχιτεκτονική των LTE από το τερματικό έως το επίπεδο υπηρεσιών

Ο **Πυρήνας Δικτύου** είναι πρακτικά η σταδιακή εξέλιξη των πυρήνων που συναντήσαμε και στα δίκτυα τρίτης γενιάς. Στην προκειμένη περίπτωση καλύπτει μόνο το τμήμα μεταγωγής πακέτων και για το λόγο αυτό έχει το όνομα Evolved Packet Core. Πρόκειται για έναν πυρήνα δικτύου υψηλής απόδοσης και χωρητικότητας, ο οποίος χρησιμοποιεί τεχνολογίες καθολικού IP. Έχει τη δυνατότητα να παρέχει ανεπτυγμένες υπηρεσίες πραγματικού χρόνου αλλά και υψηλών απαιτήσεων multimedia υπηρεσίες με αποτέλεσμα την ενδυνάμωση της ποιότητας υπηρεσιών που λαμβάνει ο χρήστης. Με την χρήση του υπάρχει βελτίωση στην απόδοση του δικτύου καθώς γίνεται διαχωρισμός ανάμεσα στον έλεγχο και στην διαχείριση των δεδομένων.

Οι βασικές αλλαγές που έφερε η χρήση του EPC και τα δίκτυα LTE στις επικοινωνίες είναι:

- Λήξη υπηρεσιών ομιλίας με μεταγωγή κυκλώματος. Ο νέος τρόπος μεταφοράς πληροφορίας φωνής γίνεται με την χρήση τεχνολογιών VoIP. Ο EPC αντιμετωπίζει την φωνή σαν μία από τις πολλές εφαρμογές που βασίζονται στο IP.
- Αυξημένη Ασύρματη Ευρυζωνικότητα. Το LTE πρέπει να πετύχει την ποιότητα υπηρεσιών που παρέχουν τα ενσύρματα δίκτυα. Αυτό διαφέρει από την παροχή υπηρεσιών βέλτιστης προσπάθειας αλλά και χαμηλής ταχύτητας περιήγηση στο διαδίκτυο. Οι υπηρεσίες αυτές έχουν ήδη ικανοποιήσει τις απαιτήσεις τους από

προγενέστερα δίκτυα. Για το λόγο αυτό το LTE στρέφεται στην υποστήριξη πιο απαιτητικών υπηρεσιών.

- Η Ποιότητα Υπηρεσίας απ' άκρο σ' άκρο είναι πλέον απαραίτητη. Το LTE παρέχει υψηλή διαχείριση και ενδυνάμωση της ποιότητας υπηρεσίας απ' άκρο σ' άκρο με σκοπό την παράδοση υψηλού περιεχομένου και χαμηλής καθυστέρησης υπηρεσίες πραγματικού χρόνου. Υπάρχει η μετάβαση από τις τέσσερις κλάσεις υπηρεσιών που ορίστηκαν για τα δίκτυα τρίτης γενιάς (Conversational, Interactive, Streaming, Background) σε εννέα διαφορετικά προφίλ ποιότητας υπηρεσίας. Αυτό επιτυγχάνεται με την ταυτόχρονη κλιμάκωση των χρηστών, των υπηρεσιών και των συνόδων μεταφοράς δεδομένων.
- Παγκόσμια προσέγγιση. Το LTE είναι πλέον πλήρως αναγνωρισμένο από την ITU και αναγνωρίζει όλες τις ζώνες συχνοτήτων που χρησιμοποιούνται ανά τον κόσμο παρέχοντας έτσι δυνατότητες περιαγωγής παρόμοιες με αυτές που παρέχουν τα δίκτυα GSM και UMTS. Εξ αιτίας της συμβατότητας που θα έχουν οι συσκευές LTE με τα προγενέστερα δίκτυα οι συνδρομητές θα μπορούν να απολαύσουν συνεχή επικοινωνία ανεξαρτήτως της χώρας στην οποία βρίσκονται.
- Μικρότερο Κόστος ανά bit. Ο συνδυασμός της αυξημένης αποδοτικότητας και ευελιξίας του φάσματος που επιτυγχάνεται από ένα δίκτυο LTE με την προστιθέμενη χωρητικότητα είναι πιθανό να προκαλέσει μείωση του κόστους για την μεταδιδόμενη πληροφορία. Για παράδειγμα κάθε E-UTRAN κυψέλη θα μπορεί να εξυπηρετήσει μέχρι και 4 φορές μεγαλύτερο φορτίο σε σχέση με μια κυψέλη που χρησιμοποιεί HSxPA. Ο συνδυασμός της αυξημένης χωρητικότητας, με την βελτιωμένη κάλυψη που θα προκύψει από την χρήση ζωνών χαμηλότερων συχνοτήτων, αλλά και την χρήση εξελιγμένων κεραιών δίνει στα δίκτυα LTE την δυνατότητα να μειώσουν κατά πολύ το κόστος μεταδιδόμενου bit.
- Μειωμένη Καθυστέρηση. Επιπρόσθετα με την αύξηση των ρυθμών μετάδοσης και διέλευσης, τα LTE δίκτυα αναμένεται να παρέχουν βελτιωμένη ποιότητα υπηρεσίας στους τελικούς χρήστες με την μείωση της καθυστέρησης για την μετάδοση των πακέτων πληροφορίας. Εφαρμογές πραγματικού χρόνου αλλά και υπηρεσίες αλληλεπίδρασης όπως είναι το online gaming και το VoIP θα παρέχουν υψηλότερης ποιότητας εμπειρία για τους τελικούς χρήστες έχοντας μειωμένη καθυστέρηση μετάδοσης.

2.7.2.1. Περιγραφή της Αρχιτεκτονικής του συστήματος

Evolved Radio Access Network (RAN)

Το Evolved – RAN στα LTE αποτελείται από έναν κόμβο, τον κόμβο eNodeB. Ο κόμβος eNB φιλοξενεί πρωτόκολλα Φυσικού στρώματος, στρώματος MAC, Radio Link Control πρωτόκολλα και Packet Data Control Protocol. Επίσης παρέχει Radio Resource Control λειτουργίες οι οποίες εντάσσονται στο πεδίο ελέγχου. Εκτελεί λειτουργίες διαχείρισης ασύρματων πόρων, ελέγχου εισόδου, χρονοπρογραμματισμού, βελτίωσης ποιότητας υπηρεσίας, μετάδοσης πληροφοριών κυψέλης, κρυπτογράφησης και αποκρυπτογράφησης

πληροφοριών στο επίπεδο των χρηστών καθώς και συμπίεση και από-συμπίεση επικεφαλίδων κατά το download – upload.

Serving Gateway (SGW)

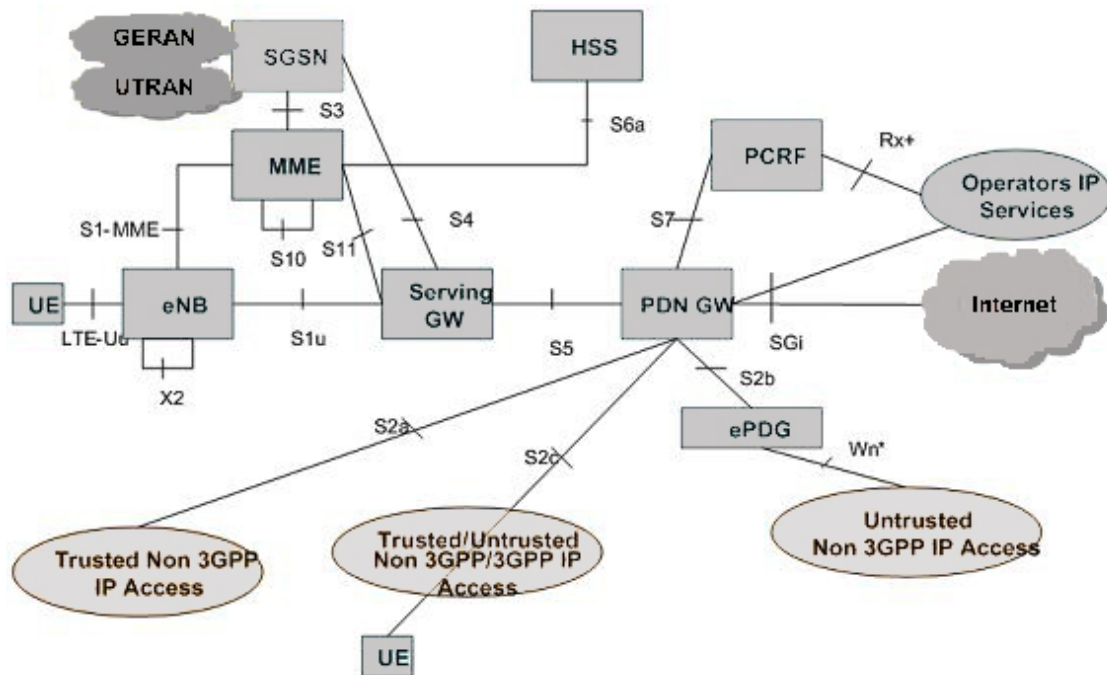
Η SGW δρομολογεί και προωθεί τα πακέτα χρηστών, ενώ λειτουργεί ως μεσολαβητής ανάμεσα στο πεδίο των χρηστών και στον eNodeB για τις εσωτερικές διαπομπές αλλά και ανάμεσα στο LTE και στα προγενέστερα δίκτυα για τις διαπομπές που μπορεί να συμβούν με την κίνηση του τερματικού. Όταν το τερματικό μπει σε κατάσταση αναμονής τότε η SGW τερματίζει την μεταφορά δεδομένων. Διαχειρίζεται και αποθηκεύει καταγραφές για τον χρήστη, οι οποίες περιέχουν παραμέτρους για τις υπηρεσίες μεταφοράς IP και πληροφορίες εσωτερικής δρομολόγησης. Επίσης διενεργεί μια αντιγραφή της κίνησης του χρήστη για την περίπτωση που υπάρξει κάποια παρεμβολή σφάλματος.

Mobility Management Entity (MME)

Η MME είναι το ο κώμβος κλειδί για την πρόσβαση σε ένα δίκτυο LTE. Είναι υπεύθυνη για τον εντοπισμό τερματικών που βρίσκονται σε κατάσταση αναμονής και για την διαδικασία επαναμετάδοσης. Περιλαμβάνεται στην διαδικασία ενεργοποίησης και απενεργοποίησης της μεταφοράς και είναι υπεύθυνη για την επιλογή της SGW από τον τερματικό εξοπλισμό κατά την αρχική προσέγγιση. Επίσης, στο τμήμα αυτό γίνεται και η εξακρίβωση των στοιχείων του χρήστη. Ελέγχει την πιστοποίηση ενός χρήστη για την χρήση μιας υπηρεσίας ενώ ενδυναμώνει τους περιορισμούς περιαγωγής. Η λανθασμένη παρεμβολή σηματοδοσίας είναι επίσης ένα γεγονός το οποίο διαχειρίζεται από την MME. Είναι επίσης αρμοδιότητά της η διατήρηση της επικοινωνίας κατά τη μετάβαση από το LTE δίκτυο σε κάποιο από τα δίκτυα δεύτερης ή τρίτης γενιάς τα οποία λειτουργούν παράλληλα.

Packet Data Network Gateway (PDN GW)

Δίνει τη δυνατότητα σύνδεσης στον χρήστη με εξωτερικά δίκτυα για την μεταφορά δεδομένων λειτουργώντας ως η πύλη εισόδου ή εξόδου (αντίστοιχα για την κάθε κίνηση) για τον χρήστη. Ο τερματικός εξοπλισμός ενός χρήστη μπορεί να έχει ταυτόχρονη σύνδεση με περισσότερες της μίας PDN GW, λόγω της σύνδεσής του με πολλαπλά PDNs. Η PDN GW εκτελεί εφαρμογή της πολιτικής κίνησης του παρόχου για κάθε χρήστη αλλά και φιλτράρισμα των πακέτων. Επίσης λειτουργεί ως συνδετικός κρίκος ανάμεσα σε δίκτυα 3GPP και σε δίκτυα non-3GPP όπως είναι τα δίκτυα WiMAX, EvDO κτλ.



Εικόνα 15. Αρχιτεκτονική Δικτύου LTE

Κοιτώντας υπηρεσίες που υποστηρίζονται σε ένα δίκτυο LTE μπορούμε να καταλάβουμε πως χωρίζονται σε υπηρεσίες και αντίστοιχα σε κινήσεις όπου θα είναι απαραίτητη μια εγγυημένη ταχύτητα μετάδοσης πληροφορίας και σε κινήσεις (υπηρεσίες), στις οποίες η ταχύτητα μεταφοράς πληροφοριών δεν θα εγγυάται. Στην Εικόνα που ακολουθεί παρατηρούμε πως γίνεται ο διαχωρισμός των εννέα προφίλ ποιότητας υπηρεσίας που προαναφέρθηκαν και αντίστοιχα βασικά χαρακτηριστικά τους όπως η καθυστέρηση και ο ρυθμός σφαλμάτων.

QCI	Resource type	Priority	Packet delay budget	Packet error loss rate	Example services
1	GBR	2	100 ms	10^{-2}	Conversational voice
2		4	150 ms	10^{-3}	Conversational video (live streaming)
3		3	50 ms	10^{-3}	Real time gaming
4		5	300 ms	10^{-6}	Non-conversational video (buffered streaming)
5	Non-GBR	1	100 ms	10^{-3}	IMS signaling
6		6	300 ms	10^{-6}	Video (buffered streaming), TCP-based (e.g., www, e-mail, chat, ftp, p2p file sharing, progressive video, etc.)
7		7	100 ms	10^{-5}	Voice, Video (live streaming), Interactive gaming
8		8	300ms	10^{-3}	Video (buffered streaming), TCP-based (e.g., www, e-mail, chat, ftp, p2p file sharing, progressive video, etc.)
9		9		10^{-6}	

Εικόνα 16. Διαχωρισμός των εννέα διαφορετικών προφίλ ποιότητας υπηρεσίας

2.7.3. LTE Advanced

Αν και τα δίκτυα LTE επιτυγχάνουν την επίτευξη υψηλής ποιότητας υπηρεσιών παρέχοντας την δυνατότητα για ανάπτυξη υψηλών ρυθμών μετάδοσης πληροφοριών δεν είναι δίκτυα τέταρτης γενιάς καθαρά σύμφωνα με τις προδιαγραφές που έχει θεσπίσει η ITU-R. Η κατάταξή τους ίσως ήταν ανάμεσα στην γενιά 3.9G και στην γενιά 4G. Παρ' όλα αυτά έχει εδώ και καιρό δημιουργηθεί και ελεγχθεί η εξέλιξη των δικτύων LTE η οποία και έχει το όνομα LTE Advanced [42]. Κύριος στόχος της νέας αυτής εξελιγμένης έκδοσης των LTE είναι η περαιτέρω βελτίωση της LTE ασύρματης πρόσβασης. Όπως γίνεται αντιληπτό τα δίκτυα LTE advanced χρησιμοποιούν βασικές δομές και τεχνολογίες των δικτύων LTE αλλά ταυτόχρονα τις συνδυάζουν με καινοτόμες τεχνικές για την βελτίωση της ποιότητας υπηρεσίας. Οι βασικές απαιτήσεις της ITU-R για τα δίκτυα LTE Advanced είναι οι εξής:

- Μέγιστος Ρυθμός Μετάδοσης για την κίνηση downlink 1Gbps, και για την κίνηση uplink 500Mbps.
- Εύρος Ζώνης Μετάδοσης μεγαλύτερο από περίπου 70MHz στην κίνηση downlink και 40MHz στην κίνηση uplink.
- Καθυστέρηση από ανενεργή κατάσταση σε κατάσταση πλήρους σύνδεσης μικρότερη των 5ms.
- Ρυθμός διέλευσης ακραίου χρήστη δύο φορές μεγαλύτερος από τον αντίστοιχο στα LTE.
- Μέσος ρυθμός διέλευσης χρήστη σχεδόν τρεις φορές μεγαλύτερος από αυτόν των LTE δικτύων.
- Χωρητικότητα (Απόδοση Φάσματος) τρεις φορές μεγαλύτερη από αυτή των LTE.
- Μέγιστη αποδοτικότητα φάσματος για την κίνηση downlink 30bps/Hz και για την κίνηση uplink 15bps/Hz.
- Η φορητότητα παραμένει στα ίδια επίπεδα με τα LTE.

Item	Sub-category	LTE target	LTE-Advanced (4G) target	IMT-Advanced (4G) requirement
Peak spectral efficiency (b/s/Hz)	Downlink	16.3 (4x4 MIMO)	30 (up to 8x8 MIMO)	15 (4x4 MIMO)
	Uplink	4.32 (64QAM SISO)	15 (up to 4x4 MIMO)	6.75 (2x4 MIMO)
Downlink cell spectral efficiency b/s/Hz/user Microcellular 3 km/h, 500 m ISD	(2x2 MIMO)	1.69	2.4	
	(4x2 MIMO)	1.87	2.6	2.6
	(4x4 MIMO)	2.67	3.7	
Downlink cell-edge user spectral efficiency (b/s/Hz/user) (5 percentile, 10 users), 500m ISD	(2x2 MIMO)	0.05	0.07	
	(4x2 MIMO)	0.06	0.09	0.075
	(4x4 MIMO)	0.08	0.12	

Εικόνα 17. Στόχοι απόδοσης για LTE και LTE Advanced

3. Μηχανισμοί και Παράμετροι Ποιότητας Υπηρεσίας

3.1. Εισαγωγή

Η αύξηση της κίνησης που καλείται να εξυπηρετήσει ένα δίκτυο αλλά και γενικότερα ή συνολική υποστήριξη των υπηρεσιών που βασίζονται στο εν λόγω δίκτυο απαιτούν την χρήση αλλά και την ύπαρξη μηχανισμών διαχείρισης. Αποτελούν το εκτελεστικό τμήμα ενός δικτύου, το οποίο καλείται να λάβει αποφάσεις για διαχείριση κινήσεων και ροών δεδομένων μηχανισμοί για την υποστήριξη της QoS σε ένα δίκτυο, μπορούμε να πούμε, ότι χωρίζονται σε δύο ομάδες έχοντας ως βάση την διαχείριση της κίνησης [7].

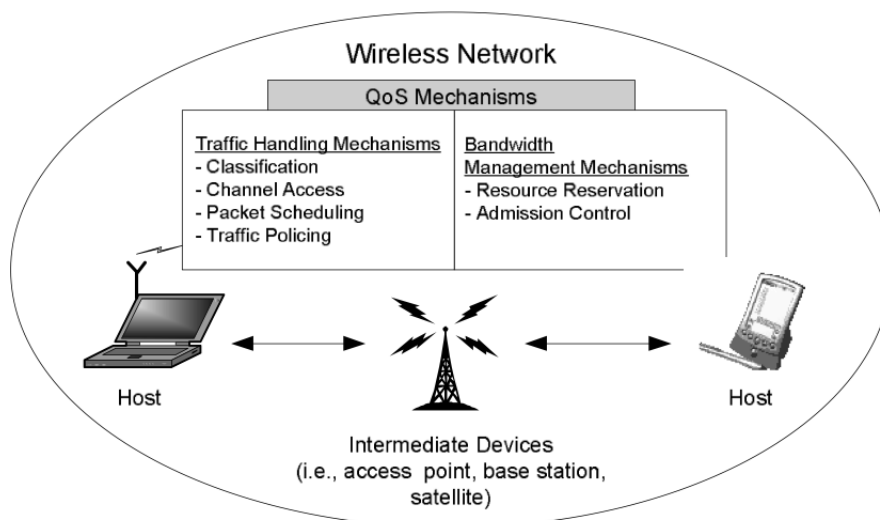
- Μηχανισμοί χειρισμού Κίνησης
- Μηχανισμοί διαχείρισης εύρους ζώνης (βλ. Εικόνα 18)

Μηχανισμοί Χειρισμού Κίνησης (πολλές φορές καλούνται και In-traffic μηχανισμοί) είναι οι μηχανισμοί που ταξινομούν, χειρίζονται, αστυνομεύουν και καταγράφουν την κίνηση μέσα στο δίκτυο. Οι κύριοι μηχανισμοί που ανήκουν σε αυτή την κατηγορία είναι:

- 1) Ταξινόμηση
- 2) Πρόσβαση Καναλιού
- 3) Προγραμματισμός Πακέτου
- 4) Αστυνόμηση Κίνησης

Μηχανισμοί Διαχείρισης Εύρους Ζώνης (μερικές φορές καλούνται Out-traffic μηχανισμοί) είναι οι μηχανισμοί που χειρίζονται τους πόρους του δικτύου (όπως είναι το εύρος ζώνης), συντονίζοντας τις συσκευές που είναι συνδεδεμένες στο δίκτυο (hosts, σταθμούς βάσης, access point). Οι κύριοι Out – traffic μηχανισμοί είναι:

- Σηματοδοσία Δέσμευσης Πόρων
- Έλεγχος Εισόδου

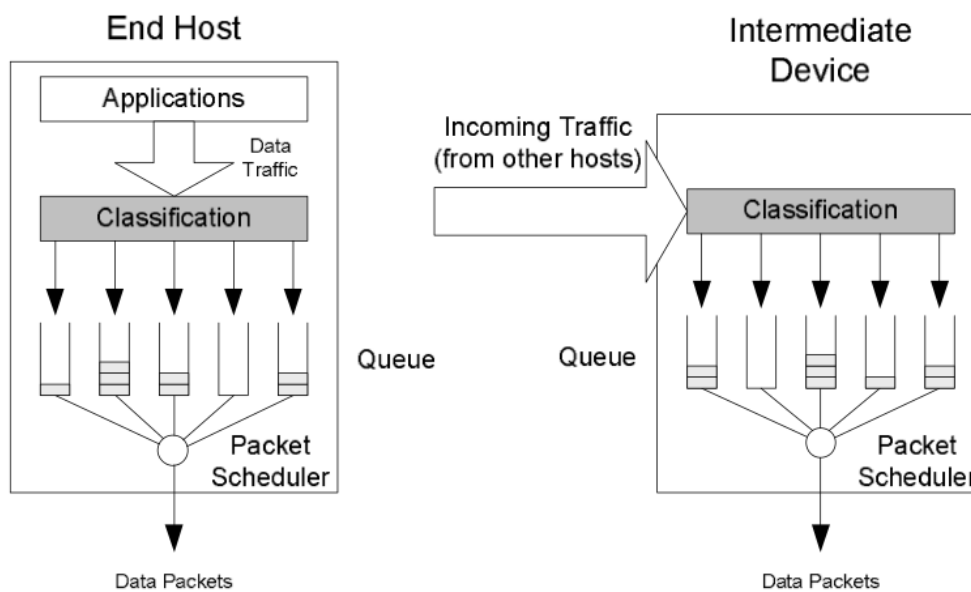


Εικόνα 18. Μηχανισμοί ποιότητας υπηρεσίας για ασύρματα δίκτυα

3.2. Ταξινόμηση

Το μικρότερο επίπεδο υπηρεσιών που μπορεί να παρέχει ένα δίκτυο είναι οι υπηρεσίες βέλτιστης προσπάθειας, οι οποίες όμως δεν παρέχουν υποστήριξη ποιότητας υπηρεσίας. Σε μια υπηρεσία βέλτιστης προσπάθειας, όλη η κίνηση χειρίζεται εξίσου, άσχετα από την εφαρμογή ή τον host που δημιούργησε την κίνηση. Παρ' όλα αυτά μερικές εφαρμογές χρειάζονται υποστήριξη QoS, απαιτώντας υπηρεσίες καλύτερες από αυτές της βέλτιστης προσπάθειας. Για να είναι δυνατή η παροχή επιλεκτικών υπηρεσιών σε συγκεκριμένες εφαρμογές για ένα δίκτυο πρέπει, πρώτα απ' όλα το δίκτυο να έχει ένα μηχανισμό ταξινόμησης ο οποίος μπορεί να διαφοροποιηθεί ανάμεσα σε διαφορετικές εφαρμογές. Ο μηχανισμός ταξινόμησης αναγνωρίζει και ξεχωρίζει διαφορετική κίνηση σε ροές ή ομάδες ροών (συγκεντρωτικές ροές ή τάξεις ροών). Έτσι, κάθε ροή ή συγκεντρωτική ροή μπορεί να χειριστεί ξεχωριστά.

Ο μηχανισμός ταξινόμησης μπορεί να ενσωματωθεί σε διαφορετικές συσκευές δικτύου (ακραίους host, ενδιάμεσες συσκευές, όπως είναι οι μεταγωγείς, οι δρομολογητές και τα access point). Η Εικόνα 19 δείχνει ένα απλουστευμένο διάγραμμα ενός στοιχείου που ανήκει σε έναν ακραίο host ή σε μια ενδιάμεση συσκευή.



Εικόνα 19. Ταξινόμηση

Η κίνηση της εφαρμογής (στον ακραίο host) ή στην εισερχόμενη κίνηση από άλλους host (σε μια ενδιάμεση συσκευή) αναγνωρίζεται από το μηχανισμό ταξινόμησης και προωθείται σε μια κατάλληλη ουρά, που αναμένει την υπηρεσία από άλλους μηχανισμούς, όπως είναι ο χρονοπρογραμματιστής πακέτου. Το πιο μικρό στοιχείο του μηχανισμού ταξινόμησης μπορεί να είναι ανά χρήστη, ανά ροή, ανά τάξη εξαρτώμενο πάντα από τον τύπο της QoS που παρέχεται. Για παράδειγμα, η ανά ροή QoS απαιτεί ταξινόμηση ανά ροή, ενώ η ανά τάξη QoS απαιτεί ταξινόμηση ανά κλάση [9].

Για να αναγνωρίσουμε και να ταξινομήσουμε την κίνηση, οι μηχανισμοί ταξινόμησης της κίνησης απαιτούν μια μορφή ετικέτας ή μαρκαρίσματος στα πακέτα. Υπάρχουν πολλές προσεγγίσεις της ταξινόμησης κίνησης. Μερικές από τις προσεγγίσεις είναι κατάλληλες για ακραίους host και μερικές για ενδιάμεσους host. Στην Εικόνα 20 φαίνεται ένα παράδειγμα

μερικών προσεγγίσεων ταξινόμησης της κίνησης οι οποίες εισάγονται σε διάφορα Ανοιχτά επίπεδα Διασύνδεσης Συστήματος. (Open System Interconnection – OSI)

OSI Layer	Classification Techniques
Application	User/Application Identification
Transport	Flow (5-tuplet IP Address)
Network	IPTOS, DSCP
Data Link	802.1p/Q Classification
Physical Layer	

Εικόνα 20. Παραδείγματα υπάρχουσας Ταξινόμησης για κάθε επίπεδο του μοντέλου OSI

3.2.1. Επίπεδο Ζεύξης Δεδομένων.

Στο επίπεδο data link ή Ζεύξης Δεδομένων, η κίνηση ταξινομείται με βάση μια ετικέτα που υπάρχει στην επικεφαλίδα των πλαισίων data link. Σύμφωνα με την IEEE, μια επικεφαλίδα επιπέδου 2 περιέχει μια ετικέτα 3 bit, η οποία καθορίζει την προτεραιότητα του πακέτου ανάμεσα σε 8 κλάσεις προτεραιότητας. Ο λόγος για την ύπαρξη αυτής την ετικέτας είναι η υποστήριξη των διάφορων υπηρεσιών που βασίζονται στο επίπεδο 2, όπως είναι το LAN.

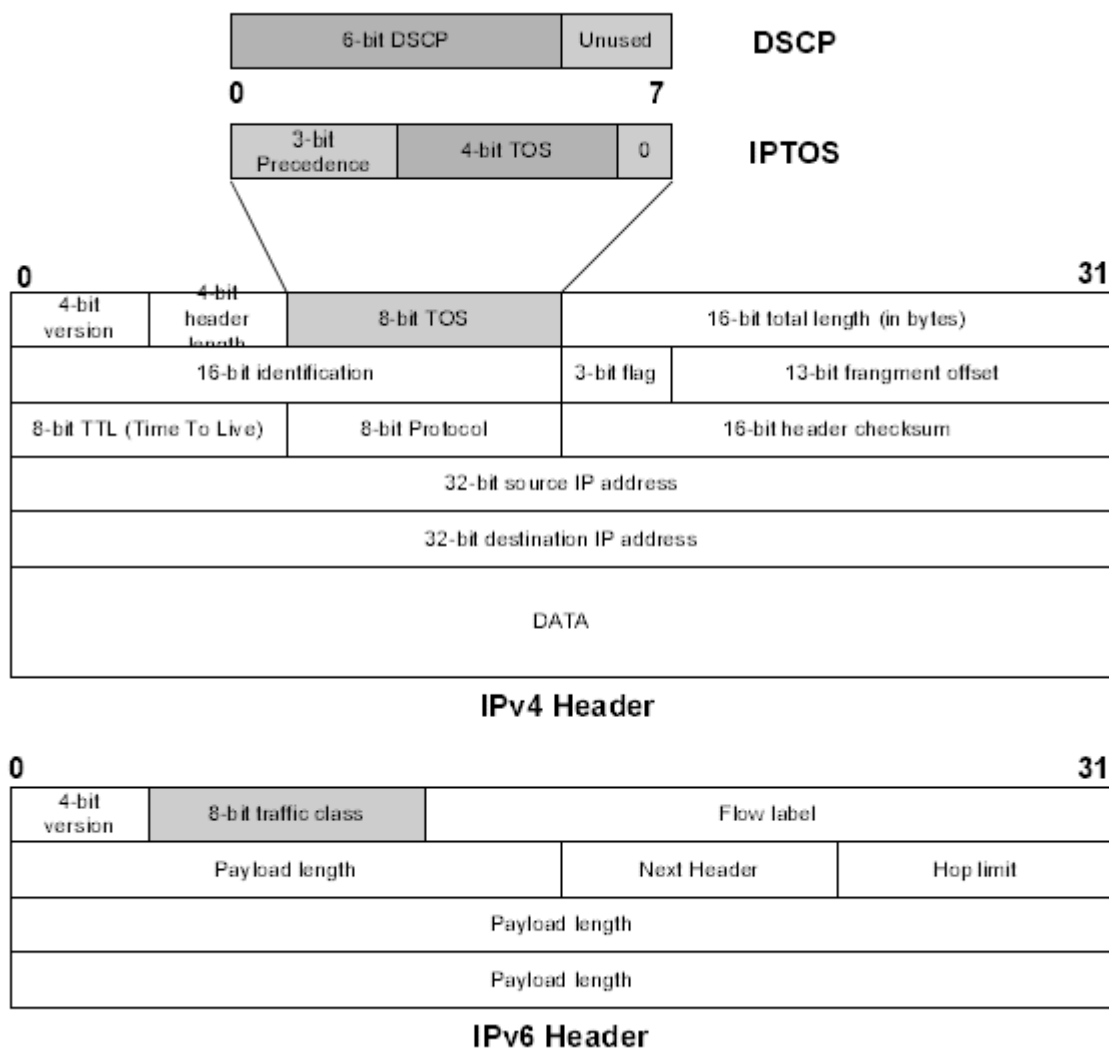
Ο ακραίος host ή κάποιος από τους ενδιάμεσους εντοπίζει την προτεραιότητα της υπηρεσίας που εξυπηρετεί το συγκεκριμένο πακέτο και αλλάζει το πεδίο προτεραιότητας στην επικεφαλίδα του πακέτου. Ένας μηχανισμός ταξινόμησης εντοπίζει τα πακέτα αυτά και τα προωθεί, ανάλογα όταν αυτά βρεθούν σε ουρές δρομολογητών μέσα σε ένα δίκτυο. Η IEEE έχει δώσει ένα βασικό κανόνα για τις περιπτώσεις και τις τιμές που θα πρέπει να πάρει το priority field στην επικεφαλίδα [7]. Αυτές οι περιπτώσεις φαίνονται παρακάτω στον Πίνακα 1.

Προτεραιότητα	Υπηρεσία
0	Προκαθορισμένο, είναι για υπηρεσίες βέλτιστης προσπάθειας
1	Κάτι λιγότερο από υπηρεσίες βέλτιστης προσπάθειας
2	Δεσμευμένο
3	Δεσμευμένο
4	Υπηρεσίες με Ευαισθησία στην καθυστέρηση, χωρίς όριο
5	Υπηρεσίες με Ευαισθησία στην καθυστέρηση, όριο 100ms
6	Υπηρεσίες με Ευαισθησία στην καθυστέρηση, όριο 10ms
7	Έλεγχος Δικτύου

Πίνακας 1. Αντιστοίχιση τιμής πεδίου προτεραιότητας και υπηρεσίας που εξυπηρετεί.

3.2.1.1. Ταξινόμηση στο Επίπεδο Δικτύου

Η ταξινόμηση στο επίπεδο δικτύου ή ταξινόμηση επιπέδου 3 χρησιμοποιεί ένα πεδίο στην επικεφαλίδα του πρωτοκόλλου επιπέδου 3. Ένα παράδειγμα ταξινόμησης επιπέδου 3 είναι το IPTOS (Internet Protocol Type of Service) και το DSCP (Internet protocol differential service code point). Το μοντέλο διευθυνσιοδότησης IPv4 και IPv6 έχει προκαθορίσει ένα πεδίο στην επικεφαλίδα πακέτων IP, το οποίο χρησιμοποιείται για ταξινόμηση των πακέτων. Το πεδίο τύπου υπηρεσίας αποτελείται από ένα υπό-πεδίο 3 bit προτεραιότητας, ένα πεδίο TOS 4 bit, στο οποίο ανάλογα με το περιεχόμενο αντιστοιχίζονται 16 διαφορετικές τάξεις υπηρεσιών, και άλλο ένα bit το οποίο μένει μηδενικό και δεν χρησιμοποιείται.



Εικόνα 21. Επικεφαλίδες πρωτοκόλλων IPv4 και IPv6

3.2.1.2. Ταξινόμηση στο επίπεδο μεταφοράς

Η χρήση 5 πεδίων της επικεφαλίδας IP μπορούν να χρησιμοποιηθούν για την ταξινόμηση των πακέτων σε επίπεδο μεταφοράς. Τα πεδία αυτά είναι τα εξής:

- Source IP
- Destination IP
- Source Port
- Destination Port
- Protocol IP

Η πεντάδα αυτών των πεδίων μπορεί να αναγνωρίσει με μοναδικό τρόπο την εφαρμογή ή την ροή, η οποία δημιούργησε ή στην οποία απευθύνονται αυτά τα πακέτα. Η ταξινόμηση αυτή προσφέρει την καλύτερη κοκκοποίηση και υποστηρίζει υπηρεσίες QoS ανά ροή. Παρ' όλα αυτά, η χρήση αυτής της πεντάδας για ταξινόμηση έχει ορισμένους περιορισμούς:

1. Είναι κατάλληλη για ακραία δίκτυα, αλλά δεν κρίνεται ακατάλληλη για δίκτυα τα οποία μεταφέρουν μεγάλο φόρτο κίνησης. Η διατήρηση των ουρών καθεμιάς από τις διαφορετικές ροές πολλές φορές είναι μια διαδικασία που μπορεί να συντρίψει το σύστημα.
2. Εάν η κίνηση περάσει μέσα από ένα firewall, το οποίο χρησιμοποιεί NAT (Network Address Translation), η πραγματική διεύθυνση IP είναι κρυφή για δίκτυα έξω από το δίκτυο. Έτσι, δεν μπορεί η πεντάδα αυτή να αντιστοιχίσει μοναδικά μια εφαρμογή.

3.2.1.3. Ταξινόμηση χρήστη ή εφαρμογής

Η εφαρμογή ή ο χρήστης μπορεί να αναγνωριστεί, χρησιμοποιώντας και μια μοναδική ταυτότητα (ID). Η απόδοση αυτής της ταυτότητας μπορεί να είναι στατική ή μπορεί να είναι δυναμική. Για παράδειγμα, δυναμική απόδοση συνήθως χρησιμοποιείται σε περίπτωση σηματοδότησης, για την έναρξη μιας νέας σύνδεσης. Στην περίπτωση αυτή, υπάρχει ένας κεντρικός σταθμός ή μια οντότητα, η οποία είναι υπεύθυνη για τη λήψη απόφασης δημιουργίας μιας νέας σύνδεσης στο ήδη υπάρχον δίκτυο. Αρχικά, η εφαρμογή ή ο χρήστης στέλνουν μια αίτηση για εγκατάσταση σύνδεσης στον κεντρικό σταθμό. Στη συνέχεια, εάν επιτραπεί η νέα σύνδεση, θα ανατεθεί στον χρήστη ή στην εφαρμογή μια μοναδική ταυτότητα. Όλα τα πακέτα, που θα προέλθουν από τη συγκεκριμένη εφαρμογή, θα συσχετιστούν με τον αριθμό αυτό.

3.3. Μηχανισμός Πρόσβασης Καναλιού

Στα ασύρματα δίκτυα, όλοι οι hosts επικοινωνούν, μέσα από ένα διαμοιραζόμενο ασύρματο μέσο. Όταν πολλαπλοί hosts προσπαθούν να εκπέμψουν πακέτα στο κοινό κανάλι επικοινωνίας μπορεί να προκληθεί σύγκρουση. Για το λόγο αυτό, τα ασύρματα δίκτυα χρειάζονται ένα μηχανισμό, ο οποίος θα ελέγχει την πρόσβαση στο διαμοιραζόμενο κανάλι.

Υπάρχουν δύο τύποι μηχανισμών ελέγχου πρόσβασης και αυτοί είναι:

- Μηχανισμός πρόσβασης στο κανάλι με βάση τις συγκρούσεις (Collision Based Channel Access)
- Μηχανισμός πρόσβασης στο κανάλι αδιάφορος ως προς τις συγκρούσεις (Collision Free Channel Access)

3.3.1. Μηχανισμός Πρόσβασης σε Κανάλι με Βάση τις Συγκρούσεις

Η Collision Based Channel Access είναι μια διαμοιρασμένη μέθοδος πρόσβασης στο κανάλι, η οποία παρέχει τους απαραίτητους μηχανισμούς για τον εντοπισμό, αλλά και την επίλυση συγκρούσεων σε περίπτωση που αυτές συμβούν [7]. Ένα κλασικό κανάλι με μηχανισμούς πρόσβασης που λαμβάνουν υπ' όψιν τους τις συγκρούσεις είναι το CSMA/CD (Carrier Sense Multiple Access with Collision Detection), το οποίο χρησιμοποιείται κυρίως σε ενσύρματα LAN και έχει ενσωματωθεί στο Ethernet.

Η πιθανότητα για δημιουργία συγκρούσεων συνήθως εξαρτάται από τον αριθμό των χρηστών που είναι ενεργοί, δηλαδή μεταδίδουν πακέτα, εκείνη τη στιγμή στο δίκτυο. Τα υψηλά φορτία κίνησης αυξάνουν τον αριθμό των συγκρούσεων και των επανεκπομπών, το οποίο με τη σειρά του αυξάνει την καθυστέρηση. Δεχόμενοι πως η κίνηση που μελετάμε έχει στοχαστικό χαρακτήρα είναι προφανές πως ο αριθμός τόσο των συγκρούσεων όσο και των επαναμεταδόσεων είναι ένας τυχαίος αριθμός, ο οποίος με τη σειρά του οδηγεί σε μια αόριστη καθυστέρηση. Για τον λόγο αυτό, τα μοντέλα καναλιών, βασισμένα στις συγκρούσεις, μπορούν να παρέχουν υπηρεσίες βέλτιστης προσπάθειας. Όλοι οι host στο δίκτυο λαμβάνουν ίσα μερίδια εύρους ζώνης και δέχονται την ίδια αόριστη καθυστέρηση.

3.3.2. Μηχανισμός Πρόσβασης σε Κανάλι Αδιάφορος προς τις Συγκρούσεις

Σε ένα μοντέλο μηχανισμού πρόσβασης στο κανάλι χωρίς συγκρούσεις, το κανάλι διαιτητεύεται με τέτοιο τρόπο, ώστε να μην υπάρχουν συγκρούσεις. Μόνο ένας χρήστης είναι ελεύθερος να μεταδώσει πακέτα στο κανάλι σε κάθε δεδομένη στιγμή. Για τον λόγο αυτό, συγκρούσεις δεν πρόκειται να συμβούν. Παραδείγματα τεχνικών για κανάλια χωρίς συγκρούσεις, είναι οι τεχνικές πόλωσης και οι τεχνικές TDMA.

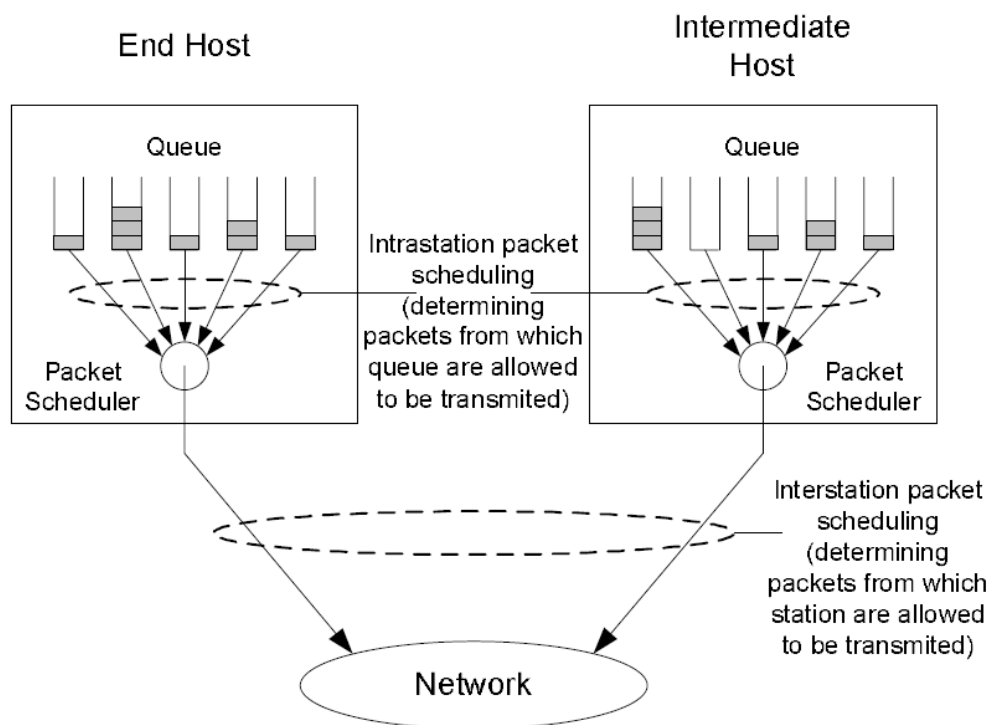
3.4. Μηχανισμοί Προγραμματισμού Πακέτου

Ο προγραμματισμός πακέτων είναι ο μηχανισμός, ο οποίος επιλέγει τα πακέτα για μετάδοση από τα πακέτα τα οποία βρίσκονται στην ουρά μετάδοσης. Αποφασίζει ποιο πακέτο από ποια ουρά και από ποιον σταθμό είναι προγραμματισμένο για μετάδοση σε μια συγκεκριμένη περίοδο χρόνου [9]. Ο προγραμματισμός πακέτων ελέγχει τον καταμερισμό του εύρους ζώνης στους σταθμούς, στις τάξεις υπηρεσιών αλλά και στις εφαρμογές. Υπάρχουν δύο είδη μηχανισμών για τον προγραμματισμό πακέτων

- Intra-station προγραμματισμός πακέτων: Ο μηχανισμός προγραμματισμού πακέτων επιλέγει το πακέτο προς αποστολή από την ουρά του ίδιου του host.
- Inter-station προγραμματισμός πακέτων: Ο μηχανισμός προγραμματισμού πακέτων επιλέγει πακέτα για αποστολή από ουρές διαφορετικών host.

Ο προγραμματισμός πακέτων μπορεί να ενσωματωθεί, χρησιμοποιώντας ιεραρχικές ή οριζόντιες προσεγγίσεις.

Ιεραρχικός Προγραμματισμός Πακέτων: Το εύρος ζώνης μοιράζεται στους σταθμούς και ο κάθε σταθμός είναι ελεύθερος να μεταδώσει πακέτα σε μια συγκεκριμένη χρονική περίοδο. Το μέγεθος του εύρους ζώνης, το οποίο ορίζεται για τον κάθε σταθμό, ελέγχεται από την πολιτική που διέπει τις σχέσεις των σταθμών, αλλά και από το είδος του μοντέλου διαμοιρασμού που χρησιμοποιείται. Η προσέγγιση αυτή είναι κλιμακούμενη, επειδή στηρίζεται στη διατήρηση μιας κατάστασης ανά σταθμό. Έτσι, το εύρος ζώνης μοιράζεται σε κάθε σταθμό ή ομάδα σταθμών ξεχωριστά και στη συνέχεια είναι αυτοί (οι σταθμοί) υπεύθυνοι για τον τρόπο με τον οποίο θα το εκμεταλλευτούν.



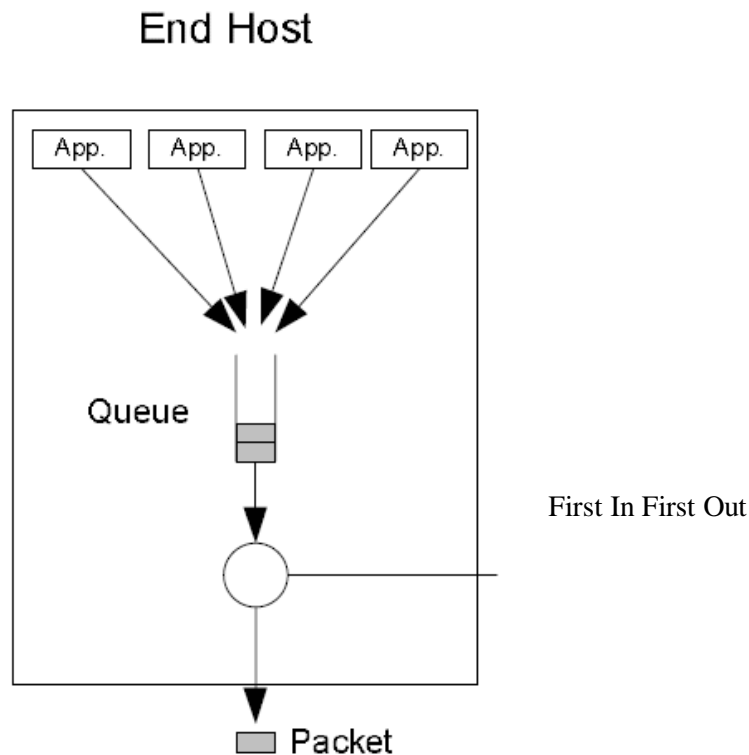
Εικόνα 22. Απεικόνιση Inter-station και Intra-station επιπέδου λειτουργίας.

Οριζόντιος Προγραμματισμός Πακέτων: Ο προγραμματισμός πακέτων βασίζεται σε όλες τις ουρές όλων των σταθμών. Κάθε μια ουρά λαμβάνει μια ξεχωριστή υπηρεσία από το δίκτυο.

Ο προγραμματισμός πακέτων σχετίζεται άμεσα με τον τρόπο ανάκτησης πακέτων από ουρές. Τόσο για μηχανισμούς σε intra-station επίπεδο, όσο και για μηχανισμούς σε inter-station επίπεδο, παρατηρούμε ότι η ανάκτηση πακέτων από ουρές παρουσιάζει πολλές ομοιότητες με την προσθήκη πακέτων σε ουρά. Για το λόγο αυτό, πολλές φορές για την ανάκτηση χρησιμοποιούνται τεχνικές που έχουν γίνει γνωστές, κυρίως για την είσοδο πακέτων σε ουρές και την εξυπηρέτησή τους. Κάποιες από αυτές είναι η FIFO (First In First Out – Το πρώτο που μπαίνει είναι και το πρώτο που υ θα βγει), Αυστηρής Προτεραιότητας, Τεχνική Δικαίου Βάρους. Για να δούμε τον τρόπο με τον οποίο αυτές εξυπηρετούν, αλλά και επηρεάζουν την ποιότητα υπηρεσιών, θα προσπαθήσουμε να τις αναλύσουμε παρακάτω.

3.4.1. Εξυπηρέτηση Πρώτα του Πρώτου Πακέτου Εισόδου

Η τεχνική FIFO είναι ο πιο απλός μηχανισμός για την εξυπηρέτηση μιας ουράς. Όλα τα πακέτα εισέρχονται στην ουρά μιας και μόνο σειράς. Τα πακέτα στοιβάζονται με βάση τον χρόνο εισόδου τους.



Εικόνα 23. Προγραμματισμός πακέτων με χρήση FIFO

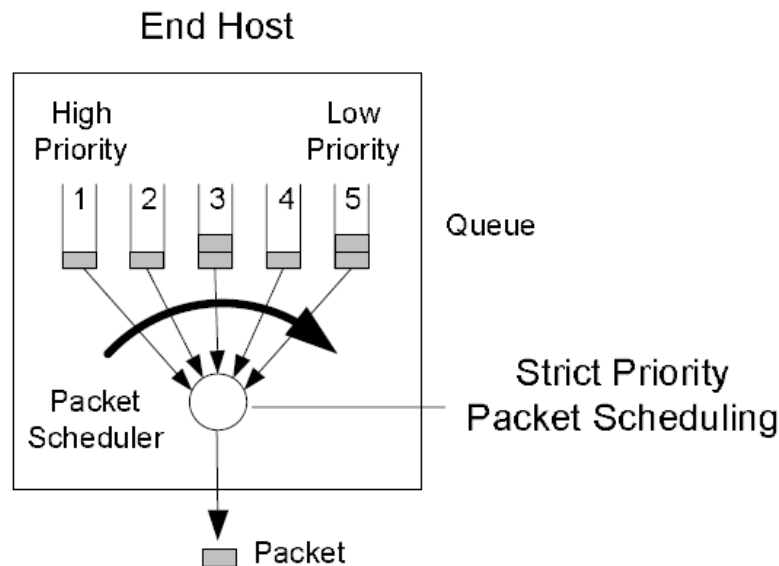
Η FIFO προσφέρει υπηρεσίες βέλτιστης προσπάθειας, με άλλα λόγια δηλαδή δεν προσφέρει υπηρεσίες, οι οποίες να υποστηρίζουν διαφοροποίηση στο εύρος ζώνης και στην καθυστέρηση. Οι ροές δεδομένων υψηλού εύρους ζώνης θα πάρουν μεγαλύτερο ποσοστό εύρους ζώνης από τις μικρότερες ροές. Σε γενικές γραμμές, όλες οι ροές θα δεχθούν την ίδια μέση καθυστέρηση. Εάν μια ροή αυξήσει το εύρος ζώνης της με γρήγορο ρυθμό, τότε οι υπόλοιπες ροές δεν είναι προστατευμένες, αλλά θα δεχτούν μείωση του εύρους ζώνης που είναι διαθέσιμο. Ένα τέτοιο γεγονός πρακτικά προκαλεί αυξημένη μέση καθυστέρηση για όλες τις ροές. Είναι δυνατό να βελτιωθεί η ποιότητα υπηρεσίας προσθέτοντας:

- Πολιτική Χειρισμού Κίνησης για να περιοριστεί ο ρυθμός μετάδοσης κάθε ροής.
- Έλεγχος Εισόδου.

3.4.2. Αυστηρής Προτεραιότητας

Στις ουρές έχει καθοριστεί μια σειρά προτεραιότητας. Ο προγραμματισμός πακέτων με χρήση αυστηρής προτεραιότητας οργανώνει τα πακέτα, με βάση αυτήν τη σειρά προτεραιότητας που έχει θεσπιστεί. Τα πακέτα σε ουρές υψηλής προτεραιότητας μεταδίδονται πάντα πρώτα από τα πακέτα σε ουρές μικρότερης προτεραιότητας, ενώ για να

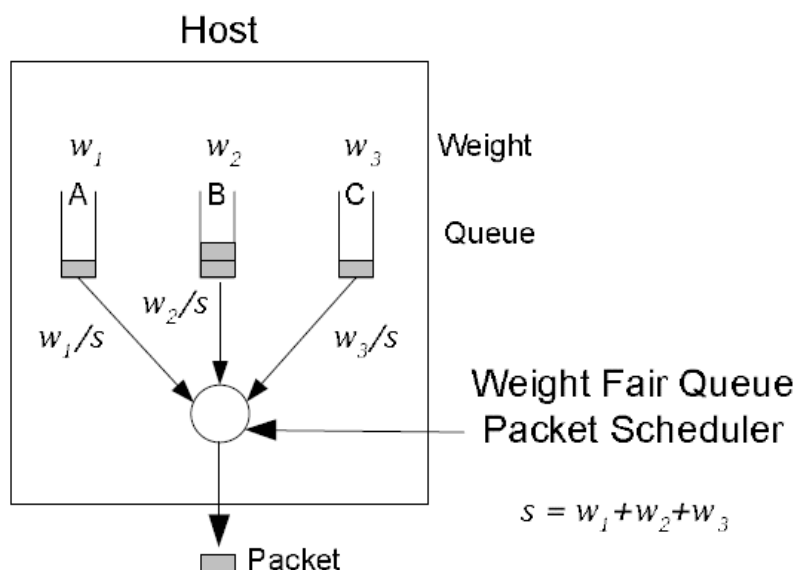
μεταδοθεί ένα πακέτο σε ουρά μικρής προτεραιότητας θα πρέπει να έχουν εξαντληθεί όλα τα πακέτα στις ουρές υψηλής προτεραιότητας. Μια επιθετική αύξηση του εύρους ζώνης, που χρησιμοποιούν οι ουρές υψηλής προτεραιότητας, μπορεί να οδηγήσει σε ανεπάρκεια πόρων για τις υπόλοιπες ουρές. Είναι επίσης εφικτή η βελτίωση της QoS με τους τρόπους που προαναφέραμε.



Εικόνα 24. Προγραμματισμός Πακέτου με χρήση Αυστηρής Προτεραιότητας-Strict Priority

3.4.3. Ουρές με χρήση Βάρους

Οι ουρές με χρήση βάρους ταξινομούν τα πακέτα, με βάση ένα ποσοστό βάρους. Ένα βάρος w_i έχει ανατεθεί σε κάθε ουρά, με βάση την πολιτική ταξινόμησης ή τις ανάγκες του συστήματος. Για παράδειγμα, εάν υπάρχουν τρεις ουρές A,B,C και οι οποίες έχουν βάρη w_1 , w_2 , w_3 αντίστοιχα τότε οι ουρές A,B,C θα δεχτούν ποσοστά του εύρους ζώνης τα οποία θα είναι αντίστοιχα: $\frac{w_1}{w_1 + w_2 + w_3}$, $\frac{w_2}{w_1 + w_2 + w_3}$, $\frac{w_3}{w_1 + w_2 + w_3}$.



Εικόνα 25.
Προγραμματισμός
πακέτων με χρήση
Ουρών Βάρους

Η κατάχρηση εύρους ζώνης από μία ροή δεν θα επηρεάσει τις υπόλοιπες. Το εύρος ζώνης και η καθυστέρηση, που μπορεί να παρέχει η WFQ, είναι άρρηκτα συνδεδεμένα με το εύρος ζώνης που της έχει δοθεί. Μια ουρά, που της έχει δοθεί μεγάλο ποσοστό εύρους ζώνης, θα υποστεί μικρότερη καθυστέρηση.

3.5. Μηχανισμός Επίβλεψης Κίνησης

Ο μηχανισμός επίβλεψης της κίνησης καταγράφει την κίνηση των συνόδων, έτσι ώστε οι σύνοδοι να μην παραβιάσουν τις υποχρεώσεις τους για την ποιότητα υπηρεσίας. Επίσης, σιγουρεύει πως όλη η κίνηση που περνά μέσα από ένα δίκτυο συμβαδίζει με τις συμφωνηθείσες παραμέτρους κίνησης. Όταν εντοπιστεί κάποια παραβίαση (πχ. Στέλνεται μεγαλύτερη κίνηση από αυτή που έχει συμφωνηθεί), τότε ο μηχανισμός είναι υπεύθυνος να αναμορφώσει την κίνηση. Επειδή ο μηχανισμός διαμορφώνει την κίνηση, βασισμένος σε κβαντοποιημένες παραμέτρους κίνησης, υπηρεσίες, όπως οι υπηρεσίες πολυμέσων, είναι συμβατές με τέτοιους είδους αναμορφώσεις. Το μεγαλύτερο ποσοστό κίνησης πολυμέσων (ομιλία - voice, video) δημιουργείται με βάση τις παραμέτρους αυτές της κίνησης. Ο μηχανισμός επίβλεψης μπορεί να εφαρμοστεί, ακόμη και σε μια μεμονωμένη ροή.

Η κίνηση μη πραγματικού χρόνου δεν υπόκειται σε αυτές τις παραμέτρους κίνησης και συνήθως απαιτεί το δυνατό μεγαλύτερο ποσοστό εύρους ζώνης. Για το λόγο αυτό, ο μηχανισμός επίβλεψης επιβάλλει σε αυτές τις υπηρεσίες να οριοθετήσουν το εύρος ζώνης που δεσμεύουν, κάτι το οποίο γίνεται πάντα σε συμφωνία με τις γενικότερες αρχές που διέπουν το δίκτυο. Οι μηχανισμοί επίβλεψης μπορούν να ενσωματωθούν, τόσο σε ακραίο host όσο και σε ενδιάμεσο.

3.6. Μηχανισμός Δέσμευσης Πόρων με Σηματοδοσία

Οι μηχανισμοί χειρισμού κίνησης (ταξινόμησης, πρόσβασης καναλιού, προγραμματισμού πακέτων και επίβλεψης κίνησης), που ήδη περιγράψαμε, κάνουν εφικτή την QoS σε κάθε συσκευή. Παρ' όλα αυτά, ο συντονισμός ανάμεσα στις συσκευές είναι ζωτικός για την παράδοση υπηρεσιών QoS απ' άκρο σ' άκρο. Οι μηχανισμοί δέσμευσης πόρων με σηματοδοσία πληροφορούν τις δικτυακές οντότητες για τις απαιτήσεις QoS των εφαρμογών πολυμέσων, χρησιμοποιώντας πόρους του συστήματος. Οι δικτυακές συσκευές θα χρησιμοποιήσουν τις πληροφορίες αυτές, για να διαχειριστούν τους δικτυακούς πόρους (όπως είναι το εύρος ζώνης) με τρόπο τέτοιο, ώστε να ανταπεξέλθουν σε αυτές τις απαιτήσεις.

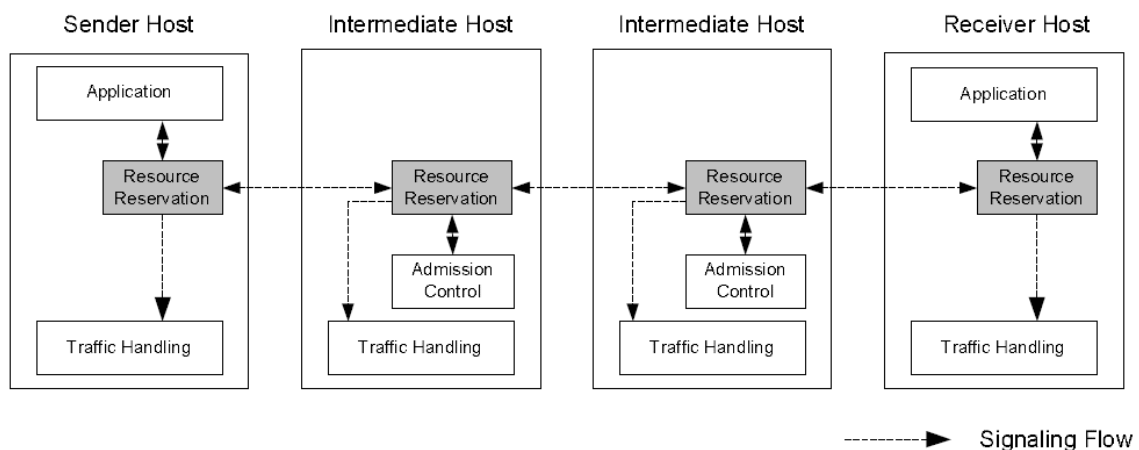
Οι δικτυακές συσκευές ελέγχουν τους δικτυακούς πόρους και παρέχουν ποιότητα υπηρεσίας, χρησιμοποιώντας τους μηχανισμούς χειρισμού κίνησης. Η δέσμευση πόρων μπορεί να εφαρμοστεί σε ξεχωριστές ροές ή σε ενοποιημένες ροές. Στην παρακάτω εικόνα (Εικόνα 26), φαίνεται ο συντονισμός ανάμεσα στους μηχανισμούς δέσμευσης πόρων και των μηχανισμών ελέγχου εισόδου.

Οι μηχανισμοί δέσμευσης πόρων περιλαμβάνουν τις παρακάτω συναρτήσεις:

- i. Παροχή σηματοδότησης για την δέσμευση πόρων, που ενημερώνει όλες τις συσκευές διαμέσου του τηλεπικοινωνιακού μονοπατιού για τις απαιτήσεις QoS που έχει η εφαρμογή πολυμέσων.
- ii. Παράδοση των απαιτήσεων QoS στον μηχανισμό ελέγχου πρόσβασης, ο οποίος με τη σειρά του αποφασίζει αν υπάρχουν διαθέσιμοι πόροι για να μπορέσει να ανταπεξέλθει σ' αυτές.
- iii. Ειδοποίηση της εφαρμογής για το αποτέλεσμα της αίτησης εισόδου.

Το Πρωτόκολλο Δέσμευσης Πόρων (Resource Reservation Protocol – RSVP) είναι ένας από τους πιο γνωστούς μηχανισμούς δέσμευσης πόρων. Το RSVP λειτουργεί πάνω από το IP, στο επίπεδο μεταφοράς, και έτσι είναι συμβατό με τους μηχανισμούς TCP/IP και μπορεί να μεταφερθεί διαμέσου πολλών δικτύων. Η κύρια λειτουργία του RSVP είναι να μεταφέρει πληροφορίες για τις απαιτήσεις QoS, ανάμεσα στον host-αποστολέα και στον host – παραλήπτη καθώς και στις ενδιάμεσες συσκευές. Με την χρήση αυτών των πληροφοριών, κάθε δικτυακή συσκευή θα δεσμεύσει τους απαραίτητους πόρους και θα χρησιμοποιήσει κατάλληλα τους μηχανισμούς χειρισμού κίνησης για να παρασχεθεί η επιθυμητή ποιότητα υπηρεσίας [35].

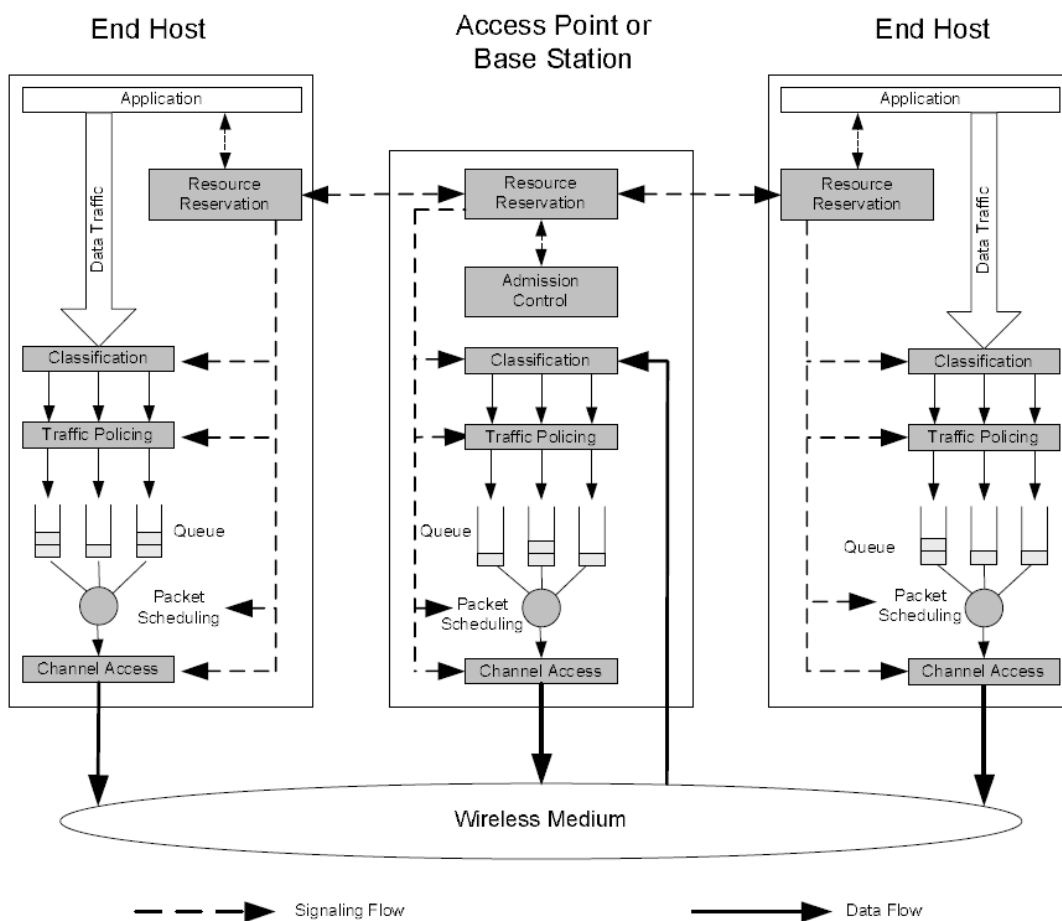
Από την στιγμή που η διαδικασία δέσμευσης ολοκληρωθεί, ο host-αποστολέας είναι σε θέση να μεταδώσει δεδομένα σύμφωνα με ένα συγκεκριμένο προφίλ κίνησης. Εάν κάποια δικτυακή συσκευή, κατά μήκος του τηλεπικοινωνιακού μονοπατιού, δεν έχει ικανοποιητικούς πόρους για να μπορέσει να ικανοποιήσει την κίνηση που θα δημιουργηθεί, τότε στέλνεται μήνυμα ειδοποίησης στην εφαρμογή για την ανικανότητα του δικτύου να εκπληρώσει τις απαιτήσεις QoS που ζητά η εφαρμογή. Για να κρατηθούν πόροι απ' άκρη σ' άκρη στο δίκτυο θα πρέπει όλα τα δικτυακά στοιχεία που βρίσκονται στο τηλεπικοινωνιακό μονοπάτι, να είναι ενεργά αλλά και διαθέσιμα να υποστηρίξουν το πρωτόκολλο RSVP.



Εικόνα 26. Μηχανισμός Δέσμευσης Πόρων με Σηματοδότηση

3.7. Αρχιτεκτονική Ποιότητας Υπηρεσίας για Υποδομές Ασύρματων Τοπικών Δικτύων.

Στις υποδομές των ασύρματων δικτύων υπάρχουν δύο ειδών σταθμοί: οι τερματικοί σταθμοί (hosts) και ένας κεντρικός σταθμός (access point, σταθμός βάσης). Ο κεντρικός σταθμός ρυθμίζει όλες τις επικοινωνίες στο δίκτυο, με αυτό να σημαίνει πως δεν υπάρχει peer to peer επικοινωνία απευθείας ανάμεσα στους hosts. Η κίνηση από τον host-πηγή στέλνεται στον κεντρικό σταθμό και από εκεί προωθείται στον host-προορισμό. Όλος ο χειρισμός της κίνησης (ταξινόμηση, επίβλεψη κίνησης, προγραμματισμός πακέτων και πρόσβαση καναλιού) και όλοι οι μηχανισμοί δέσμευσης πόρων καταλήγουν σε όλους τους σταθμούς (ακραίους host και κεντρικό σταθμό).



Εικόνα 27. Αρχιτεκτονική QoS για ασύρματα δίκτυα

3.8. Παράμετροι Ποιότητας Υπηρεσίας για Συγκεκριμένες Υπηρεσίες

Για να μπορέσουμε να ερευνήσουμε αναλυτικά τις παραμέτρους που παρέχουν την ποιότητα υπηρεσίας σε ένα δίκτυο, θα πρέπει να επικεντρωθούμε στη μελέτη κάποιων βασικών και πολύ συγκεκριμένων υπηρεσιών. Οι υπηρεσίες αυτές υποστηρίζονται πλήρως από τα δίκτυα 3G και αναμένεται να επεκταθούν αλλά και να εξελιχθούν με την έλευση των δικτύων επόμενης γενιάς. Οι παράμετροι που παραθέτουμε παίρνουν ενδεικτικές καθορισμένες τιμές, οι οποίες προσφέρουν ένα καθολικά αποδεκτό επίπεδο ποιότητας, χωρίς όμως να εξαντλούν πόρους από το δίκτυο. Στο μέλλον όπως είναι φυσικό, πολλές από αυτές τις τιμές θα μεταβληθούν καθώς θα αλλάξουν τόσο τα δεδομένα των δικτυακών υποδομών όσο και οι απαιτήσεις των χρηστών από τις υπηρεσίες.

Για τον προσδιορισμό των άνω ορίων των παραμέτρων κάθε υπηρεσίας είναι αναγκαία η διάκριση των χρηστών ανάλογα με το προφίλ τους. Όπως προαναφέραμε, ο μέγιστος ρυθμός μετάδοσης δεν είναι ο ίδιος για όλους τους χρήστες καθώς κάτι τέτοιο έρχεται σε αντίθεση με την διαφοροποίηση των χρηστών ανάλογα με τις οικονομικές τους υποχρεώσεις έναντι του παρόχου. Θα πρέπει να αναφέρουμε πως ο προσδιορισμός και η παρακολούθηση των παραμέτρων RAB (Radio Access Bearer), αποτελεί ένα σημαντικό γεγονός το οποίο είναι χρήσιμο εργαλείο για τις διαχείριση αλλά και εξέλιξη του δικτύου. Για τον λόγο αυτό θα αναφέρουμε παρακάτω παραμέτρους για τις βασικές και πιο χαρακτηριστικές υπηρεσίες, όπως:

- Ομιλία (Voice/Rich Voice)
- Βίντεο – Τηλεφωνία (Video Telephony)
- Βίντεο Συνδιάσκεψη (Video Conference)
- Multimedia Messaging Service
- Εφαρμογές Ροής Δεδομένων (Media Streaming Applications)
- Content Download
- Πρόσβαση στο Internet και Πλοήγηση
- Εταιρική Πρόσβαση (Corporate Access)

3.8.1. Ομιλία

Η φωνή στα δίκτυα δεύτερης γενιάς μεταδίδεται με την χρήση μεταγωγής κυκλώματος. Αυτό στα δίκτυα τρίτης γενιάς, σε ορισμένες περιπτώσεις, συνεχίζεται (έχουμε την ταυτόχρονη λειτουργία τεχνολογιών δεύτερης και τρίτης γενιάς). Καθώς όμως μεταφερόμαστε στην εποχή του καθολικού IP, θα έχουμε μια καθολική κυριάρχηση της μεταγωγής πακέτου και έτσι η φωνή θα μεταφέρεται αποκλειστικά και μόνο πάνω από πακέτα IP (voice over IP). Η λύση αυτή αν και μειώνει τα έξοδα για τους παρόχους και για τους χρήστες, εγείρει βασικά ερωτήματα αλλά και προβλήματα για την υποστήριξη καλής ποιότητας ομιλίας λόγω της καθυστέρησης μετάδοσης των φωνητικών πακέτων.

Πολλά από τα προβλήματα αυτά θα επιλυθούν όταν καθιερωθεί πλήρως η χρήση των διευθύνσεων IPv6, όπου τότε θα γίνει και η αποκλειστική μεταφορά της φωνής μέσω πακέτων. Ο ρυθμός κωδικοποίησης μπορεί να διαφέρει ανάμεσα σε 4.75 και 12.2kbit/sec ο οποίος είναι και ο ρυθμός του GSM-EFR. Ο κωδικοποιητής λειτουργεί με frames των 20 ms στην συχνότητα δειγματοληψίας των 8kHz και εκτιμάται ότι θα είναι ιδιαίτερα δημοφιλής στις εφαρμογές VoIP.

Το επιθυμητό όριο καθυστέρησης για την φωνή είναι τα 150ms. Παρόλα αυτά καθυστερήσεις μέχρι τα 400ms θεωρούνται αποδεκτές παρά τον σχετικό μεγάλο υποβιβασμό της ποιότητας. Το jitter θα πρέπει να είναι λιγότερο από 1 ms και η χρήση ενός buffer είναι απαραίτητη. Η απώλεια πληροφορίας θα πρέπει να είναι λιγότερη από 3%. Μια τυπική τιμή BER για την φωνή είναι το 10^{-4} . Για την διατήρηση χαμηλής καθυστέρησης κάθε SDU περιέχει ένα πεδίο που ονομάζεται source statistics descriptor και λαμβάνει την τιμή "speech" εάν έχουμε την μετάδοση φωνής.

QoS Parameter	Parameter Value
Παράδοση Εσφαλμένων SDU	Ναι
Παράδοση με καθορισμένη σειρά	Ναι
Τάξη Κίνησης	Τάξη Συνομιλίας
Μέγιστο Μέγεθος SDU	1500 bytes
Εγγυημένος ρυθμός downlink	13kbps 4kbps (older codecs)
Μέγιστος ρυθμός downlink	24kbps
Εγγυημένος ρυθμός uplink	13kbps 4kbps (older codecs)
Μέγιστος ρυθμός uplink	24kbps
Residual BER	0.00001
SDU error ratio	0.007
Προτεραιότητα Χειρισμού Κίνησης	Βασισμένο στο προφίλ της QoS
Καθυστέρηση Κίνησης	150ms AMR Maximum 400ms
SDU format information	Not Used
Καταμερισμός/ Διατήρηση Προτεραιότητας	Βασισμένο στο προφίλ της QoS
Source Statistics Descriptor	Ομιλία

Πίνακας 2. Καθορισμός Παραμέτρων Voice Service

3.8.2. Βίντεο Τηλεφωνία

Όπως και η υπηρεσία ομιλίας, έτσι κι αυτή η υπηρεσία ανήκει στην τάξη υπηρεσιών Conversational. Και πάλι έχουμε επικοινωνία δύο άκρων αλλά αυτή τη φορά έχουμε και την παρουσία εικόνας. Για μια καλή ποιότητα η καθυστέρηση θα πρέπει να είναι μικρότερη από 150ms. Η μέγιστη όμως καθυστέρηση έχει καθοριστεί στα 400ms και έτσι οποιαδήποτε τιμή μικρότερη από αυτή θεωρείται ως αποδεκτή. Η εικόνα και ο ήχος θα πρέπει να συγχρονίζονται μέσα σε ορισμένα όρια ώστε να επιτυγχάνεται το "lip-synch", αλλιώς το αποτέλεσμα θα είναι ενοχλητικό στον τελικό χρήστη ο οποίος τελικά θα προτιμήσει την χρήση της απλής τηλεφωνίας. Το όριο αυτό καθορίζεται στα 100ms. Επίσης, καθώς το ανθρώπινο μάτι είναι ανεκτικό σε κάποιες μικρές απώλειες, επιτρέπεται η απώλεια δεδομένων να φτάνει το 1% της συνολικής πληροφορίας, οπότε BER είναι ίσο με 10^{-4} .

QoS Parameter	Parameter Value
Παράδοση Εσφαλμένων SDU	Ναι
Παράδοση με καθορισμένη σειρά	Ναι
Τάξη Κίνησης	Τάξη Συνομιλίας
Μέγιστο Μέγεθος SDU	1500 bytes
Εγγυημένος ρυθμός downlink	64kbps
Μέγιστος ρυθμός downlink	384kbps
Εγγυημένος ρυθμός uplink	64kbps
Μέγιστος ρυθμός uplink	384kbps
Residual BER	0.0001
SDU error ratio	0.001
Προτεραιότητα Χειρισμού Κίνησης	Βασισμένο στο προφίλ της QoS
Καθυστέρηση Κίνησης	100ms Maximum 400ms
SDU format information	Not Used
Καταμερισμός/ Διατήρηση Προτεραιότητας	Βασισμένο στο προφίλ της QoS
Source Statistics Descriptor	Unknown

Πίνακας 3. Καθορισμός Παραμέτρων Βίντεο Κλήσης

3.8.3. Βίντεο Συνδιάσκεψη

Η βίντεο συνδιάσκεψη είναι μια υπηρεσία βίντεο τηλεφωνίας, στην οποία όμως έχουμε περισσότερους από 2 χρήστες να εμπλέκονται και να μπορούν να έχουν ταυτόχρονη επικοινωνία. Η υπηρεσία αυτή ανήκει στην τάξη Streaming. Η καθυστέρηση δεν θα πρέπει να ξεπερνάει το όριο των 200ms. Είναι απαραίτητο να έχουμε υψηλούς ρυθμούς μετάδοσης καθώς ο κάθε χρήστης θα δέχεται πλέον πολλές ροές πληροφορίας. Οι προτεινόμενοι ρυθμοί μετάδοσης από εταιρείες είναι της τάξης των 384 και 768kbps. Εδώ είναι φυσική η ύπαρξη ασυμμετρίας ανάμεσα στο uplink και στο downlink αφού ο χρήστης θα χρειάζεται αυξημένο downlink αλλά όχι και τόσο αυξημένο uplink.

QoS Parameter	Parameter Value
Παράδοση Εσφαλμένων SDU	Ναι
Παράδοση με καθορισμένη σειρά	Ναι
Τάξη Κίνησης	Τάξη Συνομιλίας
Μέγιστο Μέγεθος SDU	
Εγγυημένος ρυθμός downlink	64kbps 128 for PS
Μέγιστος ρυθμός downlink	768kbps
Εγγυημένος ρυθμός uplink	64kbps 128kbps for PS
Μέγιστος ρυθμός uplink	768kbps
Residual BER	0.00001
SDU error ratio	0.007
Προτεραιότητα Χειρισμού Κίνησης	Βασισμένο στο προφίλ της QoS
Καθυστέρηση Κίνησης	200ms
SDU format information	Not Used
Καταμερισμός/ Διατήρηση Προτεραιότητας	Βασισμένο στο προφίλ της QoS
Source Statistics Descriptor	Άγνωστο

Πίνακας 4. Καθορισμός Παραμέτρων βίντεο συνδιάσκεψης

3.8.4. Υπηρεσία Μηνυμάτων Πολυμέσων

Η υπηρεσία αυτή αποτελεί την εξέλιξη της υπηρεσίας ανταλλαγής σύντομων μηνυμάτων SMS/EMS. Για την παροχή της, το δίκτυο θα πρέπει να χρησιμοποιεί τεχνολογίες υποδομής 2.5G και η μεταφορά να γίνεται τουλάχιστον μέσω GPRS. Το κύριο χαρακτηριστικό των μηνυμάτων που αποστέλλονται είναι πως περιέχουν εκτός από κείμενο και εικόνες, βίντεο και ήχο. Η καθυστέρηση που εμφανίζεται σε αυτή την υπηρεσία είναι μεγάλη και συνήθως εξαρτάται από το μέγεθος του μηνύματος, αλλά παρ' όλα αυτά η υπηρεσία αν και ανθεκτική στην καθυστέρηση είναι εξαιρετικά ευαίσθητη στην απώλεια δεδομένων. Ανήκει στην τάξη Background και επιπλέον η ύπαρξη ανθεκτικότητας στην καθυστέρηση την κατατάσσει στις υπηρεσίες βέλτιστης προσπάθειας. Οι ρυθμοί μετάδοσης μπορεί να είναι σχετικά χαμηλοί μέχρι τα 4 kbps και ο μέγιστος ρυθμός μετάδοσης δεν συνίσταται να ξεπεράσει τα 64 kbps, για λόγους υπερφόρτωσης του δικτύου.

QoS Parameter	Parameter Value
Παράδοση Εσφαλμένων SDU	Όχι
Παράδοση με καθορισμένη σειρά	Όχι
Τάξη Κίνησης	Τάξη Background
Μέγιστο Μέγεθος SDU	1500 bytes
Εγγυημένος ρυθμός downlink	4kbps
Μέγιστος ρυθμός downlink	64kbps
Εγγυημένος ρυθμός uplink	4kbps
Μέγιστος ρυθμός uplink	64kbps
Residual BER	10^{-7}
SDU error ratio	10^{-6}
Προτεραιότητα Χειρισμού Κίνησης	Best Effort
Καθυστέρηση Κίνησης	Some seconds
SDU format information	Not Used
Καταμερισμός/ Διατήρηση Προτεραιότητας	Βασισμένο στο προφίλ της QoS
Source Statistics Descriptor	Unknown

Πίνακας 5. Καθορισμός Παραμέτρων Multimedia Messaging Service

3.8.5. Εφαρμογές Ροής Πολυμέσων

Με τον όρο αυτό εννοούμε εφαρμογές που περιλαμβάνουν βίντεο, ήχο, εικόνα, μουσική αλλά και συνδυασμό αυτών. Όλες αυτές οι εφαρμογές και υπηρεσίες ανήκουν στην τάξη κίνησης Streaming και η αλληλεπίδραση είναι περιορισμένη. Βασικό χαρακτηριστικό που πρέπει να αναφερθεί είναι πως κάθε εφαρμογή έχει τις δικές τις παραμέτρους. Κυρίως μας ενδιαφέρουν οι υψηλοί ρυθμοί μετάδοσης, καθώς και το ελάχιστο jitter. Η καθυστέρηση συνήθως είναι ανάλογη του μεγέθους της εφαρμογής. Το jitter συνήθως περιορίζεται με την χρήση buffers ενώ όσον αφορά την απώλεια πληροφορίας είναι ανεκτή αλλά σε πολύ μικρό βαθμό.

Το προφίλ για την ποιότητα υπηρεσίας που έχει ο κάθε χρήστης παίζει ιδιαίτερα σημαντικό ρόλο εδώ καθώς οι χρήστες με περισσότερα προνόμια θα απολαύσουν υψηλότερους ρυθμούς μετάδοσης αλλά και καλύτερη ποιότητα.

QoS Parameter	Parameter Value
Παράδοση Εσφαλμένων SDU	Ναι
Παράδοση με καθορισμένη σειρά	Ναι
Τάξη Κίνησης	Τάξη Streaming
Μέγιστο Μέγεθος SDU	1500 bytes
Εγγυημένος ρυθμός downlink	64kbps
Μέγιστος ρυθμός downlink	2048kbps
Εγγυημένος ρυθμός uplink	64kbps
Μέγιστος ρυθμός uplink	2048kbps
Residual BER	10^{-6}
SDU error ratio	10^{-5}
Προτεραιότητα Χειρισμού Κίνησης	Βασισμένο στο προφίλ της QoS
Καθυστέρηση Κίνησης	200ms
SDU format information	Not Used
Καταμερισμός/ Διατήρηση Προτεραιότητας	Βασισμένο στο προφίλ της QoS
Source Statistics Descriptor	Unknown

Πίνακας 6. Παράμετροι Media Streaming Application

3.8.6. Λήψη Δεδομένων

Ο όρος content download αναφέρεται σε υπηρεσίες ροής αλλά η διαφορά με τις υπόλοιπες υπηρεσίες είναι πως εδώ μπορούμε να έχουμε αποθήκευση του περιεχομένου και μετέπειτα αναπαραγωγή της. Για τον λόγο αυτό η υπηρεσία αυτή ανήκει στην κατηγορία Background. Η υπηρεσία παρουσιάζει αρκετά μεγάλη αναισθησία στον τομέα της καθυστέρησης αλλά από την άλλη πλευρά ιδιαίτερη σημασία έχει η ακεραιότητα της πληροφορίας. Ο χρήστης μπορεί να επιλέξει την επιθυμητή ποιότητα υπηρεσίας του και σύμφωνα με το προφίλ του να απολαύσει μια είτε αργή είτε γρήγορη σύνδεση.

QoS Parameter	Parameter Value
Παράδοση Εσφαλμένων SDU	Όχι
Παράδοση με καθορισμένη σειρά	Όχι
Τάξη Κίνησης	Τάξη Background
Μέγιστο Μέγεθος SDU	1500 bytes
Εγγυημένος ρυθμός downlink	64kbps
Μέγιστος ρυθμός downlink	2000kbps
Εγγυημένος ρυθμός uplink	64kbps
Μέγιστος ρυθμός uplink	2000kbps
Residual BER	10^{-7}
SDU error ratio	10^{-6}
Προτεραιότητα Χειρισμού Κίνησης	Best Effort
Καθυστέρηση Κίνησης	-
SDU format information	Not Used
Καταμερισμός/ Διατήρηση Προτεραιότητας	Βασισμένο στο προφίλ της QoS
Source Statistics Descriptor	Unknown

Πίνακας 7. Παράμετροι Content Download

3.8.7. Πρόσβαση στο διαδίκτυο και πλοήγηση

Καθώς η ζήτηση για πρόσβαση στο Internet γνωρίζει μια διαρκή αύξηση, αυτή η υπηρεσία δεν θα μπορούσε να απουσιάζει από το κινητό δίκτυο. Η πλοήγηση στο Web (η οποία είναι η κυρίαρχη εφαρμογή του Internet), ανήκει στην τάξη Interactive, καθώς η ύπαρξη αλληλεπίδρασης είναι αναγκαία. Για την παρουσίαση των διαφόρων ιστοσελίδων είναι απαραίτητη η μηδενική απώλεια πληροφορίας, και ο χρόνος καθυστέρησης δεν θα πρέπει να υπερβαίνει τα 4sec για κάθε σελίδα που θα έχει συνολικό όγκο 10 KB.

Εξαιτίας της μεγάλης ζήτησης αυτής της υπηρεσίας, έχουν καθοριστεί δύο διαφορετικοί τύποι που θα δοθούν στο κοινό. Ο ένας είναι η βασική υπηρεσία (basic service) και ο άλλος είναι η εξαιρετική υπηρεσία (premium service). Αυτοί οι δύο τύποι μπορούν να συνυπάρχουν με τα υπάρχοντα προφίλ QoS, χωρίς να τα παρακάμπτουν όσον αφορά τις άλλες υπηρεσίες. Ένας χρήσης με χαμηλό QoS θα μπορεί να επιλέγει την επιλογή premium εάν τον ενδιαφέρει η πλοήγηση στον ιστό, και έτσι θα αποκτάει μεγαλύτερη προτεραιότητα από ένα χρήστη ανώτερου προφίλ που θα έχει επιλέξει τον βασικό τύπο σύνδεσης. Έτσι έχουμε μια διάκριση δύο επιπέδων όσον αφορά την υπηρεσία αυτή.

Η βασική υπηρεσία θα κάνει χρήση ρυθμών μετάδοσης που δεν θα είναι χαμηλότεροι από αυτούς του ISDN ή του GPRS, τα οποία λειτουργούν στα 64 Kbps. Η ταχύτητα αυτή είναι σχετικά χαμηλή αλλά είναι αποδεκτή ως το κατώτερο εγγυημένο όριο. Οι μέγιστοι ρυθμοί μετάδοσης εξαρτώνται από τον πάροχο, αλλά για την βασική υπηρεσία ρυθμοί της τάξης των 256 kbps κρίνονται ως ικανοποιητικοί.

QoS Parameter	Parameter Value
Παράδοση Εσφαλμένων SDU	Όχι
Παράδοση με καθορισμένη σειρά	Όχι
Τάξη Κίνησης	Τάξη Interactive
Μέγιστο Μέγεθος SDU	1500 bytes
Εγγυημένος ρυθμός downlink	64kbps
Μέγιστος ρυθμός downlink	256kbps
Εγγυημένος ρυθμός uplink	64kbps
Μέγιστος ρυθμός uplink	256kbps
Residual BER	10^{-7}
SDU error ratio	10^{-6}
Προτεραιότητα Χειρισμού Κίνησης	Βασισμένο στο προφίλ της QoS
Καθυστέρηση Κίνησης	<4sec/page
SDU format information	Not Used
Καταμερισμός/ Διατήρηση Προτεραιότητας	Βασισμένο στο προφίλ της QoS
Source Statistics Descriptor	Unknown

Πίνακας 8. Παράμετροι για υπηρεσία πλοήγησης στο Internet

3.8.8. Εταιρική Πρόσβαση

Μια ακόμη υπηρεσία με υψηλό δυναμικό είναι η εταιρική πρόσβαση σε υψηλές ταχύτητες. Αυτή η παρεχόμενη από το 3G υπηρεσία είναι πολύ σημαντική καθώς δίνει την δυνατότητα σε στελέχη επιχειρήσεων να έχουν πρόσβαση στο εταιρικό το π δίκτυο σε ο π οδήποτε σημείο και εάν βρίσκονται με απλή χρήση των κινητών τερματικών ή και φορητών υπολογιστών, οι οποίοι θα είναι συνδεδεμένοι με κάποιο 3G κινητό τηλέφωνο.

Μια τέτοια εταιρική σύνδεση θα πρέπει να είναι ασφαλής, καθώς είναι πιθανό τα δεδομένα που θα ανταλλαχθούν να είναι εντελώς εμπιστευτικά και θα πρέπει να προστατεύονται από πιθανούς εισβολείς. Η υπηρεσία αυτή θα χρησιμοποιεί το PS κομμάτι του δικτύου και η ταχύτητα του downlink θα είναι 384kbit/s. Το uplink θα είναι ασύμμετρο ως προς το downlink και θα έχει ελάχιστη ταχύτητα τα 64 kbit/s. Ο λόγος για αυτή την διάκριση είναι ό π σύμφωνα με τα στατιστικά, ένας χρήστης χρησιμοποιεί πολύ περισσότερο το d o π In k κανάλι και με έναν τέτοιο διαχωρισμό επιτυγχάνεται η απελευθέρωση σημαντικών για το δίκτυο πόρων και το ελεύθερο εύρος ζώνης μπορεί να αποδοθεί σε άλλους χρήστες χωρίς έτσι να αλλοιώνεται ο χαρακτήρας του QoS του δικτύου.

Σε μια πιθανή περίπτωση συμφόρησης ο συνδρομητής που θα χρησιμοποιεί μια τέτοια υπηρεσία θα αντιμετωπίσει δυσκολίες. Για να αποφευχθεί αυτό, η υπηρεσία αυτή μπορεί να διατεθεί και από το CS κομμάτι του δικτύου με εγγυημένο QoS και σταθερούς ρυθμούς μετάδοσης για το uplink και το d o π In k που θα είναι ίσοι με 64 kbit/s. Τέλος θα πρέπει να σημειωθεί ότι αυτή η υπηρεσία δεν επιτρέπει την απώλεια πληροφορίας.

QoS Parameter	Parameter Value
Παράδοση Εσφαλμένων SDU	Όχι
Παράδοση με καθορισμένη σειρά	Όχι
Τάξη Κίνησης	Τάξη Interactive
Μέγιστο Μέγεθος SDU	1500 bytes
Εγγυημένος ρυθμός downlink	64kbps for CS 384 for PS
Μέγιστος ρυθμός downlink	384kbps
Εγγυημένος ρυθμός uplink	64kbps for CS
Μέγιστος ρυθμός uplink	384kbps
Residual BER	10^{-7}
SDU error ratio	10^{-6}
Προτεραιότητα Χειρισμού Κίνησης	Βασισμένο στο προφίλ της QoS
Καθυστέρηση Κίνησης	200ms
SDU format information	Not Used
Καταμερισμός/ Διατήρηση Προτεραιότητας	Βασισμένο στο προφίλ της QoS
Source Statistics Descriptor	Unknown

Πίνακας 9. Παράμετροι Εταιρικής Πρόσβασης

3.9. Μέθοδοι Μέτρησης Ποιότητας Υπηρεσίας Ομιλίας

3.9.1. Ληφθείσα Ποιότητα Υπηρεσίας για Ομιλία πάνω από πρωτόκολλο Διαδικτύου

Ο στόχος της δοκιμής που περιγράφεται σε αυτό το τμήμα είναι να καθορίσει τα ληφθέντα QoS μιας κλήσης VoIP μεταξύ της πλευράς A και της πλευράς B [10]. Προτού περιγράψουμε τη δοκιμή με μεγαλύτερη λεπτομέρεια, αρχικά ανακεφαλαιώνουμε κάποιες απόψεις που σχετίζονται για την ληφθείσα ποιότητα των VoIP. Η συνολική ποιότητα της διαδραστικής ομιλίας αναφέρεται ως ομιλούμενη ποιότητα. Η ποιότητα συνομιλίας καθορίζεται από τρεις παράγοντες : την ακουστική ποιότητα (« πόσο καλά μπορεί το άλλο άτομο να μας ακούσει;»), την ποιότητα ομιλίας (« πόσο καλά μπορούμε να ακούσουμε τον εαυτό μας;») και τη διαδραστική ποιότητα («πόσο καλά είναι εφικτό να επικοινωνήσουμε διαδραστικά;»).

Κάθε παράγοντας ποιότητας εκφράζεται ως ένα Αποτέλεσμα Μέσης Γνωμάτευσης (Mean Opinion Score – MOS) σε μια κλίμακα πέντε επιπέδων, όπως απεικονίζεται στον Πίνακα 10.

Αποτέλεσμα Μέσης Γνωμάτευσης	Απόλυτη ποιότητα	Βλάβη
5	Άριστη	Ανεπαίσθητη
4	Καλή	αισθητή, μη ενοχλητική
3	Λιγότερο καλή	ελαφρώς ενοχλητική
2	Φτωχή	ενοχλητική
1	Κακή	πολύ ενοχλητική

Πίνακας 10 : Καθορισμός της κλίμακας MOS

Επειδή το MOS μπορεί να αναφερθεί σε διαφορετικές όψεις, προτείνεται η ακόλουθη σημειογραφία.

MOS-CQ = MOS συνολική ποιότητα συνομιλίας

MOS-LQ = MOS Ακουστική ποιότητα

MOS-TQ = MOS ποιότητα ομιλίας

MOS -IQ = MOS Διαδραστική ποιότητα

Επειδή η συνολική ποιότητα συνομιλίας καθορίζεται από άλλους τρεις παράγοντες ισχύει:

$MOS-CQ = \min\{ MOS-LQ, MOS-TQ, MOS-IQ\}$.

3.9.2. Μέθοδοι Εξέτασης

Η διάταξη που χρησιμοποιήθηκε για την δοκιμή απεικονίζεται στην εικόνα που ακολουθεί (Εικόνα 28.)

ΠΛΕΥΡΑ Α**ΠΛΕΥΡΑ Β**

Εικόνα 28 : Αρχιτεκτονική της δοκιμής VoIP

Για να καθοριστεί η ποιότητα συνομιλίας μιας φωνητικής σύνδεσης χρησιμοποιούμε μια συνδυασμένη μπαταρία αντικειμενικής/υποκειμενικής δοκιμής. Δηλώνουμε την ποιότητα συνομιλίας στην ΠΛΕΥΡΑ Α και ΠΛΕΥΡΑ Β με **A_MOS_CQ** και **B_MOS_CQ** αντίστοιχα. Όλες οι δοκιμές αφορούν δομές από άκρο σε άκρο μεταξύ των σημείων μέτρησης MP1 και MP4 στην Εικόνα 28.

Η ακολουθία της δοκιμής αποτελείται από τα ακόλουθα έξι βήματα :

- α) Μονόδρομη ακουστική, ποιότητα ομιλίας
- β) Μονόδρομη ποιότητα ομιλίας, side tone/echo quality
- γ) Μονόδρομη ποιότητα ομιλίας, ποιότητα ήχου, μέγιστη κρισιμότητα
- δ) Μονόδρομη ποιότητα ομιλίας, ποιότητα μακρινού θορύβου
- ε) Αμφίδρομη διαδραστική, πλήρη/μέση διπλή ποιότητα
- στ) Αμφίδρομη διαδραστική, καθυστέρηση

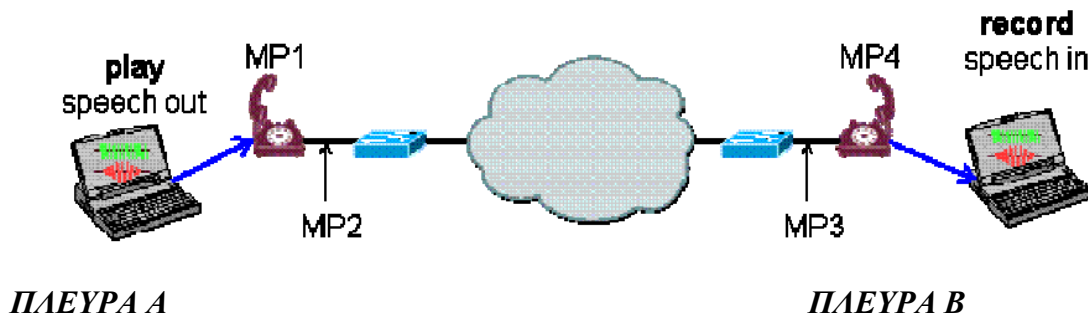
Η ποιότητα συνομιλίας καθορίζεται από το ελάχιστο MOS που αποκτάται από τις δοκιμές (α – στ). Στο επόμενο κεφάλαιο περιγράφονται τα έξι αυτά βήματα με μεγαλύτερη λεπτομέρεια. Επιπρόσθετα, στην προηγούμενη ακολουθία δοκιμής, κάποιος θα μπορούσε να χρησιμοποιήσει την ποιότητα δικτύου για να καθορίσει την ποιότητα συνομιλίας και ομιλίας. Αυτό μπορεί να επιτευχθεί μέσω μιας χαρτογράφησης.

3.9.3. Προδιαγραφή Δοκιμής σε Επίπεδο Εφαρμογών

Σε αυτό το κεφάλαιο περιγράφονται τα έξι βήματα της υποκειμενικής/ αντικειμενικής ακολουθίας δοκιμής. Είναι απαραίτητη η διαθέσιμη ύπαρξη μιας πρόσθετης τηλεφωνικής σύνδεσης κατά τη διάρκεια των δοκιμών ώστε να πραγματοποιείται η επικοινωνία κατά τη διάρκειά τους.

Δοκιμή Α : Μονόδρομη ακουστική, ποιότητα ομιλίας

Αυτή η δοκιμή χρησιμοποιεί την αντικειμενική μέθοδο PESQ. Για να εφαρμοστεί αυτή η μέθοδος χρησιμοποιείται ένα σήμα αναφοράς, που ονομάζεται *sample.wav*. Αυτό το δείγμα αρχείου περιέχει 30 δευτέρα ομιλίας τόσο από θηλυκό όσο και αρσενικό ομιλητή, που καταγράφεται στα 16kHz. Το αρχείο ξεκινά με ένα συγχρονισμένο σήμα. Το αρχείο ομιλίας πρέπει να ανοιχτεί από ένα μέρος της σύνδεσης VoIP και να καταγραφεί από ένα άλλο μέρος, βλέπε Εικόνα 29.



Εικόνα 29. Διάταξη για την εκκίνηση των μετρήσεων

Μέτρηση αρχιτεκτονικής PESQ

Οι φορητοί υπολογιστές πρέπει να είναι συνδεδεμένοι με τηλέφωνα IP μέσω RJ – 11 jacks. Για την αναπαραγωγή και καταγραφή του αρχείου, η χρήση του ελεύθερου προγράμματος (<http://audacity.sourceforge.net/>) συστήνεται. Κατά την αναπαραγωγή του sample.wav η λειτουργία του βρόχου πρέπει να είναι ενεργοποιημένη έτσι ώστε να επιτρέπει του καταγραφικού τμήματος να κάνει την καταγραφή από συγχρονισμένο σήμα σε συγχρονισμένο σήμα. Σε αυτή τη δοκιμή τόσο η ΠΛΕΥΡΑ Α όσο και η ΠΛΕΥΡΑ Β θα πρέπει να κάνουν την καταγραφή. Τα τελικά αρχεία ομιλίας ονομάζονται SIDE_A.wav και SIDE_B.wav αντίστοιχα. Έπειτα επεξεργάζονται από το TNO για να καθοριστεί η ακουστική ποιότητα μέσω του PESQ. Η τελική ακουστική ποιότητα στην ΠΛΕΥΡΑ Α και ΠΛΕΥΡΑ Β δηλώνεται με A_MOS_A_LQ και B_MOS_A_LQ αντίστοιχα.

Δοκιμή Β : Μονόδρομη ποιότητα ομιλίας, side tone/echo quality

Η υποκειμενική δοκιμή χρησιμοποιεί την εγκατάσταση που απεικονίζεται στην Εικόνα 29. Αφού η σύνδεση VoIP εγκατασταθεί ο ειδικός στην ΠΛΕΥΡΑ Α μιλά και ακούει για την ηχώ του τόνου ως παρεμβολή. Η αναφορά γίνεται για τον φυσιολογικό τόνο. Η ομιλία θα πρέπει να παράγεται σε δύο επίπεδα, φυσιολογικά και δυνατά. Ο ειδικός καταγράφει την side tone/echo quality χρησιμοποιώντας την κλίμακα Καταγραφής της Κατηγορίας διαβάθμισης (Degradation Category Rating) η οποία γίνεται με τον εξής τρόπο :

- 5 = καθόλου ηχώ και/ή τον τόνο ως παρεμβολή
- 4 = ηχητική αλλά όχι ενοχλητική
- 3 = ελαφρώς ενοχλητική
- 2 = ενοχλητική
- 1 = πολύ ενοχλητική

Η τελική ποιότητα ομιλίας σε αντιστοιχία με τον τόνο και την ηχώ στην ΠΛΕΥΡΑ Α δηλώνεται A_MOS_B_TQ. Η δοκιμή επίσης πραγματοποιείται στην ΠΛΕΥΡΑ Β και δηλώνεται ως B_MOS_B_TQ.

Δοκιμή Γ : Μονόδρομη ποιότητα ομιλίας, ποιότητα ήχου , με μέγιστη κρισιμότητα

Η υποκειμενική δοκιμή χρησιμοποιεί την εγκατάσταση στην Εικόνα 29. Αφού εγκατασταθεί η σύνδεση VoIP, ο ειδικός στην ΠΛΕΥΡΑ Α μιλάει και ακούει για ηχώ καθώς στην ΠΛΕΥΡΑ Β η θέση του τηλεφωνικού horn αλλάζει από την προεπιλεγμένη σε μια σκληρή επιφάνεια με ακουστικό προς αυτή την επιφάνεια. Η ομιλία θα πρέπει να παράγεται σε δύο επίπεδα, φυσιολογικά και δυνατά.

5 = ακουστική ηχώ αλλά όχι ενοχλητική

4 = ελαφρώς ενοχλητική

3 = ενοχλητική

Η τελική ποιότητα ομιλίας σε αντιστοιχία με την μέγιστη κρισιμότητα ήχου στην ΠΛΕΥΡΑ Α δηλώνεται ως **A_MOS_C_TQ**. Η δοκιμή επίσης διεξάγεται από την ΠΛΕΥΡΑ Β και το αποτέλεσμα δηλώνεται ως **B_MOS_C_TQ**

Δοκιμή Δ : Μονόδρομη ποιότητα ομιλίας, με μακρινό θόρυβο

Η υποκειμενική δοκιμή χρησιμοποιεί την εγκατάσταση που απεικονίζεται στην Εικόνα 29.

Αφού εγκατασταθεί η σύνδεση VoP, ο ειδικός στην ΠΛΕΥΡΑ Α μιλάει, ενώ στην ΠΛΕΥΡΑ Β ένα σύνολο από αρχεία ήχου με θόρυβο υποβάθρου (γραφείο, ομιλίες , θόρυβο αυτοκινήτων) βρίσκονται σε αναπαραγωγή. Οι καταγραφές πρέπει να περιέχουν ένα ελαφρώς μειωμένο και/ή ελαφρώς αυξημένο επίπεδο θορύβου υποβάθρου για να εκτιμηθεί το σύστημα που βρίσκεται υπό δοκιμή κάτω από ένα ευρύ φάσμα συνθηκών θορύβου υποβάθρου. Ο ρυθμός της αλλαγής πρέπει να βρίσκεται στη σειρά των περίπου 20 dB/s και να έχει τη διάρκεια των περίπου 20 δευτερολέπτων. Η ομιλία πρέπει να παράγεται σε δύο επίπεδα, φυσιολογικά και δυνατά.

Σχόλια : Οι πιο σημαντικές παρεμβολές που πραγματοποιούνται είναι χρονο-σύνδεση που συντελεί σε αλλαγή θορύβου υποβάθρου και αλλαγές στην καταστολή του ήχου που συντελούν σε μη φυσιολογικά επίπεδα και διακυμάνσεις στην ποικιλία ήχου ειδικά κατά τη διάρκεια της έναρξης/παύσης μιας ομιλίας στην ΠΛΕΥΡΑ Α.

Η κλίμακα (Degradation Category Rating) που χρησιμοποιείται, βρίσκεται σε συνάρτηση με

5 = μη καταστολή του θορύβου του υποβάθρου και άλλων ηχητικών ενοχλητικών αλλαγών

4 = ακουστική αλλά όχι ενοχλητική

3 = ελαφρώς ενοχλητική

2 = ενοχλητική

1 = πολύ ενοχλητική

Η τελική ποιότητα ομιλίας σε αντιστοιχία με το θόρυβο υποβάθρου στην ΠΛΕΥΡΑ Α δηλώνεται με **A_MOS_D_TQ**. Η δοκιμή επίσης διεξάγεται από την ΠΛΕΥΡΑ Β και συντελεί στην **B_MOS_D_TQ**.

Δοκιμή Ε : Αμφίδρομη διαδραστική, πλήρη/μέση διπλή ποιότητα

Η υποκειμενική δοκιμή χρησιμοποιεί την εγκατάσταση που απεικονίζεται στην Εικόνα 29.

Αφού η σύνδεση VoP εγκατασταθεί, ο ειδικός στην ΠΛΕΥΡΑ Β μετρά γρήγορα και συνεχώς : 1, 2, 3, 4, 5, 1, 2, 3, 4, 5, 1, 2, 3, 4, 5, 1, 2, 3, 4, 5...κλπ (παρατηρούνται τα δύο διαφορετικά επίπεδα ομιλίας). Ταυτόχρονα ο ειδικός στην ΠΛΕΥΡΑ Α μιλά χωρίς παύσεις

χρησιμοποιώντας μεμω κωμένα γράμματα : α, παύση, β, παύση, γ, παύση, ... κλπ. Στην εκτίμηση, η ποιότητα της φωνής συνεχούς μέτρησης από την πλευρά- Β (1, 2, 3, 4, 5) κρίνεται μαζί με την ποιότητα με την οποία η ίδια φωνή γίνεται αντιληπτή. Κατά τη διάρκεια διπλής ομιλίας, καμία παρεμβολή και ηχώ από τη φωνή του ενός θα πρέπει να γίνει ακουστή. Η ομιλία πρέπει να παράγεται σε δύο επίπεδα φυσιολογικά και δυνατά.

Η κλίμακα (Degradation Category Rating) που χρησιμοποιείται, βρίσκεται σε συνάρτηση με

5 = καμία υποβάθμιση της πλήρους/μέσης διπλής ποιότητας γίνεται αντιληπτή

4 = αντιληπτή αλλά όχι ενοχλητική

3 = ελαφρώς ενοχλητική

2 = ενοχλητική

1 = πολύ ενοχλητική

Η τελική ποιότητα ομιλίας σε αντιστοιχία με τη διπλή ποιότητα στην ΠΛΕΥΡΑ Α δηλώνεται με **A_MOS_E_IQ**. Η δοκιμή επίσης διεξάγεται από την ΠΛΕΥΡΑ Β και συντελεί στην **B_MOS_E_IQ**.

Δοκιμή Στ : Αμφίδρομη διαδραστικότητα, καθυστέρηση

Η δοκιμή πρώτα καθορίζει την μονόδρομη καθυστέρηση σε μία σύνδεση VoIP. Έπειτα αυτή η εκτίμηση χρησιμοποιείται για να καθορίσει την αμφίδρομη διαδραστική ποιότητα σε αντιστοιχία με την καθυστέρηση. Για αυτή τη δοκιμή ένα χρονόμετρο χρειάζεται στην ΠΛΕΥΡΑ Α. Αφού η σύνδεση VoIP εγκατασταθεί, ο ειδικός στην ΠΛΕΥΡΑ Α ξεκινά τη διαδικασία με τη λέξη μέτρησης «ένα» ενώ την ίδια στιγμή ξεκινά το χρονόμετρο. Στη συνέχεια ο ειδικός στην ΠΛΕΥΡΑ Β μετρά «δύο», αφού έχει λάβει το «ένα» κλπ έως ότου η ΠΛΕΥΡΑ Α έχει ακούσει την ΠΛΕΥΡΑ Β να μετράει «δέκα» μετά από το οποίο η ΠΛΕΥΡΑ Α σταματά τη χρονομέτρηση. Ο παρερχόμενος χρόνος (που εκφράζεται σε ms) δηλώνεται σε **T1**. Αυτή η διαδικασία επαναλαμβάνεται για 4 φορές συντελώντας στα **T2, T3, T4** και **T5**. Αν δηλώσουμε το μέσο όρο των πέντε μετρήσεων με **T** τότε η μέση μονόδρομη καθυστέρηση **D** της φωνητικής σύνδεσης εκτιμάται στα **0.1*(T-4500)ms**. Η τελική αμφίδρομη διαδραστική ποιότητα σε αντιστοιχία με την καθυστέρηση δηλώνεται με **MOS_F_IQ** και δίνεται από **min{5, 7 - 0.02*D}**. Ιδανικά η διαδικασία βαθμονομείται πρόσωπο με πρόσωπο μέχρι η ΕΝΑΡΞΗ-ΠΑΥΣΗ να είναι περίπου 4500ms.

Προσοχή : Ένα είναι δυνατό να διεξαχθούν επίπεδα μετρήσεων με πακέτα από άκρη σε άκρη τότε είναι δυνατό να αποκτηθεί μια πιο ακριβή εκτίμηση της μέσης μονόδρομη καθυστέρησης.

Περίληψη των δοκιμών σε επίπεδο εφαρμογής

Το αποτέλεσμα αυτού του κεφαλαίου είναι η συμπλήρωση του Πίνακα 11.

Όνομα	Περιγραφή	Καθορίζεται από	Μονάδα
A_MOS_A_LQ	Ακουστική ποιότητα στην ΠΛΕΥΡΑ Α	TNO	MOS
B_MOS_A_LQ	Ακουστική ποιότητα στην ΠΛΕΥΡΑ Β	TNO	MOS
A_MOS_B_TQ	Ποιότητα ομιλίας : side tone/echo στην ΠΛΕΥΡΑ Α	ΠΛΕΥΡΑ Α	MOS
B_MOS_B_TQ	Ποιότητα ομιλίας : side tone/echo	ΠΛΕΥΡΑ Β	MOS

	στην ΠΛΕΥΡΑ Β		
A_MOS_C_TQ	Ποιότητα ομιλίας : μέγιστη ηχώ στην ΠΛΕΥΡΑ Α	ΠΛΕΥΡΑ Α	MOS
B_MOS_C_TQ	Ποιότητα ομιλίας : μέγιστη ηχώ στην ΠΛΕΥΡΑ Β	ΠΛΕΥΡΑ Β	MOS
A_MOS_D_TQ	Ποιότητα ομιλίας : θόρυβος υπόβαθρου στην ΠΛΕΥΡΑ Α	ΠΛΕΥΡΑ Α	MOS
B_MOS_D_TQ	Ποιότητα ομιλίας : θόρυβος υπόβαθρου στην ΠΛΕΥΡΑ Β	ΠΛΕΥΡΑ Β	MOS
A_MOS_E_IQ	Αμφίδρομη διαδραστικότητα στην ΠΛΕΥΡΑ Α	ΠΛΕΥΡΑ Α	MOS
B_MOS_E_IQ	Αμφίδρομη διαδραστικότητα στην ΠΛΕΥΡΑ Β	ΠΛΕΥΡΑ Β	MOS
T1	Παρερχόμενος χρόνος Δοκιμή Στ1	ΠΛΕΥΡΑ Α	ms
T2	Παρερχόμενος χρόνος Δοκιμή Στ2	ΠΛΕΥΡΑ Α	ms
T3	Παρερχόμενος χρόνος Δοκιμή Στ3	ΠΛΕΥΡΑ Α	ms
T4	Παρερχόμενος χρόνος Δοκιμή Στ4	ΠΛΕΥΡΑ Α	ms
T5	Παρερχόμενος χρόνος Δοκιμή Στ5	ΠΛΕΥΡΑ Α	ms
T	Μέσος παρερχόμενος χρόνος Δοκιμής Στ	ΠΛΕΥΡΑ Α	ms
D	Μέση μονόδρομη καθυστέρηση	ΠΛΕΥΡΑ Α	ms
MOS_F_IQ	Αμφίδρομη διαδραστική ποιότητα σε αντιστοιχία με την καθυστέρηση	ΠΛΕΥΡΑ Α	MOS
A_MOS_CQ	Ποιότητα συνομιλίας στην ΠΛΕΥΡΑ Α	ΠΛΕΥΡΑ Α	MOS
B_MOS_CQ	Ποιότητα συνομιλίας στην ΠΛΕΥΡΑ Β	ΠΛΕΥΡΑ Β	MOS

Πίνακας 11 : Αποτελέσματα δοκιμών σε επίπεδα εφαρμογής για VoIP

Στον Πίνακα 11 η ποιότητα συνομιλίας στην ΠΛΕΥΡΑ Α και την ΠΛΕΥΡΑ Β αποκτάται μέσω :

$$A_MOS_CQ = \min\{ A_MOS_A_LQ, A_MOS_B_TQ, A_MOS_C_TQ, A_MOS_D_TQ, A_MOS_E_TQ, MOS_F_IQ\}$$

και

$$B_MOS_CQ = \min\{ B_MOS_A_LQ, B_MOS_B_TQ, B_MOS_C_TQ, B_MOS_D_TQ, B_MOS_E_TQ, MOS_F_IQ\}$$

Προδιαγραφή δοκιμής σε επίπεδο δικτύου

Η δοκιμή που περιγράφεται προηγουμένως δίνει μία μέτρηση του ληφθέντος QoS από μια κλήση VoIP μεταξύ της ΠΛΕΥΡΑΣ Α και της ΠΛΕΥΡΑΣ Β χρησιμοποιώντας τα τηλέφωνα IP. Εκτός από τις παραπάνω υποκειμενικές και αντικειμενικές μεθόδους ποιότητας κάποιος μπορεί να πραγματοποιήσει μια σειρά μετρήσεων δικτύου, για παράδειγμα χρησιμοποιώντας RTAnalyzer λογισμικό/λειτουργικό. Οι μέθοδοι βασίζονται στην έγχυση γνωστής ροής πακέτων μέσα στο δίκτυο IP. Αυτές οι ροές πακέτων αναλύονται με τη λήψη τους και το αποτέλεσμα αποθηκεύεται σε ένα αρχείο καταγραφής ιστορικού.

Η εγκατάσταση της δοκιμής απεικονίζεται στην Εικόνα 29. Για τη μέτρηση δύο ελέγχων δικτύου IP χρησιμοποιείται ένας δρομολογητής στην ΠΛΕΥΡΑ Α και ένας στην ΠΛΕΥΡΑ Β. Αναφορικά στην Εικόνα 29, αυτό υποδηλώνει ότι οι μετρήσεις πραγματοποιούνται μεταξύ του MP2 και MP3.

Η ροή πακέτων είναι προ-ρυθμισμένη π.χ. η ροή VoIP χρησιμοποιώντας RTP. Οι παράγοντες της ροής μπορούν να ρυθμιστούν έτσι ώστε να μιμηθούν την κλήση VoIP. Η μετρημένη ροή πακέτων εκχύνεται στο δίκτυο και συγχωνεύεται στην κυκλοφορία της πραγματικής εφαρμογής VoIP.

Η μέτρηση μπορεί να ρυθμιστεί για να τρέξει για π.χ. μία εβδομάδα έτσι ώστε να εξάγει την ώρα αιχμής του IP και το χαμηλό χρόνο φόρτωσης. Οι έλεγχοι δικτύου γίνονται από ένα κεντρικό κόμβο. Το αποτέλεσμα αυτόματα αναλύεται και παρουσιάζεται με γράφους και αναφορές. Παράμετροι όπως το εύρος ζώνης, η καθυστέρηση από άκρο σε άκρο, η διασπορά σήματος από άκρο σε άκρο, η απώλεια πακέτων και η λάθος σειρά των πακέτων καθορίζονται.

Η μέτρηση του δικτύου πρέπει να διεξάγεται σε παραλληλία με τη δοκιμή που σχετίζεται με την αντίληψη, ούτως ώστε να μπορεί να εξαχθεί ο συσχετισμός ανάμεσα στα δύο. Για την ακρίβεια, ο RTAnalyzer χαρτογραφεί την ποιότητα δικτύου απευθείας στο MOS_CQ. Όπως και παραπάνω, αυτή η εκτίμηση πρέπει να καθορίζεται και στις δύο κατευθύνσεις, δηλαδή και στην Πλευρά Α και στην Πλευρά Β.

Το τελικό αποτέλεσμα αυτού του κεφαλαίου φαίνεται στη συμπλήρωση του Πίνακα 12

Όνομα	Περιγραφή	Καθορίζεται από	Μονάδα
A_delay	Καθυστέρηση από την πλευρά Β στην Πλευρά Α	ΠΛΕΥΡΑ Α	ms
B_delay	Καθυστέρηση από την πλευρά Α στην Πλευρά Β	ΠΛΕΥΡΑ Β	ms
A_jitter	Διασπορά σήματος από την Πλευρά Β στην Πλευρά Α	ΠΛΕΥΡΑ Α	ms
B_jitter	Διασπορά σήματος από την Πλευρά Α στην Πλευρά Β	ΠΛΕΥΡΑ Β	ms
A_loss	Απώλεια πακέτων από την Πλευρά Β στην Πλευρά Α	ΠΛΕΥΡΑ Α	%
B_loss	Απώλεια πακέτων από την Πλευρά Α στην Πλευρά Β	ΠΛΕΥΡΑ Β	%
A_MOS_CQ	Ποιότητα συνομιλίας Πλευρά Α	TNO	MOS
B_MOS_CQ	Ποιότητα συνομιλίας Πλευρά Β	TNO	MOS

Πίνακας 12 : Αποτελέσματα δοκιμών σε επίπεδο δικτύου για VoIP

4. Σύμφωνα Επιπέδων Εξυπηρέτησης

4.1. Εισαγωγή

Όπως είδαμε και στα προηγούμενα κεφάλαια η εγγύηση της ποιότητας υπηρεσίας είναι ίσως η μεγαλύτερη πρόκληση που αντιμετωπίζουν οι πάροχοι υπηρεσιών αλλά και οι διαχειριστές δικτύων στις μέρες μας. Το άνοιγμα της αγοράς των υπηρεσιών αλλά και η συνεργασία εταιρειών διαφορετικών τομέων για την παροχή ολοκληρωμένων υπηρεσιών έγειρε την ανάγκη για αυστηρή εγγύηση των υπηρεσιών που παρέχονται ανάμεσα σε δύο συμβαλλόμενα μέλη. Για το λόγο αυτό δημιουργήθηκαν τα Σύμφωνα Επιπέδων Εξυπηρέτησης (Service Level Agreements) τα οποία και καθορίζουν τον τρόπο αλλά και τον χαρακτήρα μιας διμερούς συνεργασίας κυρίως όσον αφορά τον τεχνολογικό τομέα.

Στο κεφάλαιο αυτό θα προσπαθήσουμε να περιγράψουμε τα συμφωνητικά αυτά αλλά και να κατανοήσουμε τον ρόλο που διαδραματίζουν για την σωστή συνεργασία πελάτη και παρόχου.

4.2. Ορισμός και Γενικές Πληροφορίες

Τα SLAs (Service Level Agreements) ή αλλιώς τα συμφωνητικά επιπέδου εξυπηρέτησης είναι ο ορισμός των σχέσεων ανάμεσα σε έναν φορέα παροχής υπηρεσιών και σε έναν πελάτη [22]. Τα SLAs αποτελούν μια γραπτή και σε μορφή συμβολαίου διατύπωση μιας συμφωνίας ανάμεσα σε μια επιχείρηση ή οργανισμό και σε έναν πάροχο, όπου καθορίζεται το είδος υπηρεσιών αλλά και σε ποιο επίπεδο θα τροφοδοτεί ο πάροχος με αυτές την επιχείρηση - πελάτη κατά τη διάρκεια μιας συνεργασίας μακράς διάρκειας. Τα SLAs χρησιμοποιούνται σαν ασφαλιστικές δικλίδες για τις μικρομεσαίες επιχειρήσεις αφού διασφαλίζουν κατά μεγάλο βαθμό την ποιότητα των υπηρεσιών αλλά και τις εξυπηρέτησης που θα λάβει μια επιχείρηση - πελάτης. (Οι πληροφορίες που ακολουθούν και αναφέρονται σε χαρακτηριστικά αλλά και ιδιότητες των SLA προέκυψαν από διασταύρωση συμφωνητικών από διάφορες εταιρείες τεχνολογίας.)

Ένα τέτοιο συμφωνητικό μπορεί να οριστεί και να διατυπωθεί στα πλαίσια οποιασδήποτε επιχείρησης και μέσω αυτού να διευκρινιστεί τι μπορεί να αναμένει ο πελάτης από τον προμηθευτή για την συγκεκριμένη υπηρεσία, ποιές είναι οι υποχρεώσεις των δύο μερών (τόσο του πελάτη όσο και του προμηθευτή) αλλά και ζήτημα πιο τεχνικά όπως είναι η απόδοση, η διαθεσιμότητα αλλά και η ασφάλεια της υπηρεσίας.

Επιπλέον, διασφαλίζονται όροι ανεκτικότητας πάνω σε αποκλίσεις από το SLA καθώς και οι προϋποθέσεις για τον ντεντατισμό της συνεργασίας και την ακύρωση του εν λόγω συμφωνητικού ανάμεσα στις 2 πλευρές. Πολύ συχνή είναι η χρήση SLAs στις μέρες μας σε περιπτώσεις που οι εταιρείες χρησιμοποιούν έξω-εταιρικούς συνεργάτες ή παρόχους υπηρεσιών για την υλοποίηση ορισμένων βασικών λειτουργιών της εταιρείας(outsourcing). Βασικό παράδειγμα outsourcing είναι περιπτώσεις συντήρησης δικτύων υπολογιστών όπου αναθέτονται σε τρίτους προμηθευτές δικτύων.

Συνήθως ένα SLA περιλαμβάνει και πλαίσιο ποινικής ρήτρας καθώς και ανταμοιβής σε περίπτωση αθέτησης των όρων που έχουν υπογραφεί και συμφωνηθεί. Έτσι για τη αθέτηση

των όρων ο πελάτης ή ο πάροχος δεσμεύονται για την επιστροφή υπηρεσιών ή χρημάτων στον δεύτερο μέλος.

Για την αποφυγή παρερμηνειών τα SLAs δεν χρησιμοποιούνται για την μείωση των δαπανών μιας επιχείρησης αλλά ως μηχανισμοί διαχείρισης κινδύνων, όπου σαν στόχο έχουν να μετριάσουν τις συνέπειες από μια μη ορθή συμπεριφορά τόσο του πελάτη όσο και του παρόχου. Είναι γεγονός ότι πολλά SLAs δίνουν τη δυνατότητα στους πελάτες να τιμωρούν τους προμηθευτές τεχνολογιών που χρησιμοποιούν, για τη μη σωστή απόδοσή τους. Παρ' ότι οι ποινικές ρήτρες μειώνουν τις δαπάνες και διασφαλίζουν τη σωστή συμπεριφορά των παρόχων υποχρεώνοντας τους πολλές φορές να εξελίξουν ή να βελτιώσουν τις υπηρεσίες τους, δεν είναι προς όφελος καμίας πλευράς αν λείπει μια τέτοιου είδους συμφωνία. Θα πρέπει να θεωρηθεί ότι το SLA περιγράφει τους κοινούς στόχους ανάμεσα στον πάροχο και στον πελάτη και όχι ένα εργαλείο εκβιασμού του ενός από τον άλλο.

Παραδείγματα:

Ένα συμφωνητικό επιπέδων εξυπηρέτησης είναι μια συμβατική υποχρέωση για την επίτευξη συγκεκριμένων στόχων. Εάν για παράδειγμα υπογραφεί μια σύμβαση φιλοξενίας (hosting contract) με έναν προμηθευτή το SLA θα καθορίζει το χρονικό διάστημα που θα είναι “ανεβασμένος” ο ιστοχώρος στο internet. Στην περίπτωση που μια επιχείρηση θέλει να κάνει outsourcing για ένα γραφείο εξυπηρέτησης τότε ίσως χρειαστεί ένα SLA που να καθορίζει το χρόνο που χρειάζεται για να απαντηθεί το τηλέφωνο [36].

4.2.1. Ιστορία και Εξέλιξη

Τα συμφωνητικά επιπέδων εξυπηρέτησης δεν βρίσκονται στην επικαιρότητα για μεγάλο χρονικό διάστημα. Τα πρώτα δείγματα συμφωνητικών επιπέδων εξυπηρέτησης θα μπορούσαμε να πούμε ότι φαίνονται στην δεκαετία του 60, όπου χρησιμοποιούνταν συμφωνητικά για να διατυπώσουν τις γενικές λειτουργικές διαδικασίες καθώς και το επίπεδο εξυπηρέτησης για μια ομάδα χρηστών για την αγορά ή για την ενοικίαση μηχανών. Σε αυτά τα συμφωνητικά φαίνεται αδρά ο χαρακτήρας των SLA όπως τα ξέρουμε σήμερα.

Μεγάλο βήμα για την ανάπτυξη και εξέλιξη των συμφωνιών αυτού του χαρακτήρα ήταν η στιγμή που ο πελάτης/εξυπηρετητής και τα δικτυωμένα υπολογιστικά συστήματα εισήλθαν στον κόσμο των υπολογιστών. Δημιουργήθηκε έτσι η έννοια του κατανεμημένου συστήματος δικτύων.

Στη συνέχεια ήλθε η εξέλιξη των συστημάτων αυτών στα ευρέως επιχειρησιακά συστήματα που είχαν ως σκοπό το ψ τον προγραμματισμό των επιχειρησιακών πόρων (ERP), την διαχείριση της εφοδιαστικής αλυσίδας (SCM) καθώς και τη διαχείριση των σχέσεων πελατών του δικτύου.

Κατά τη διάρκεια της εξέλιξης αυτής υπήρξε και η ταυτόχρονη ανάπτυξη της εξάρτησης των επιχειρήσεων από το διαδίκτυο, η οποία σαν αποτέλεσμα είχε να αυξήσει διαφάνεια των αποτελεσμάτων του δικτύου μιας ομάδας εφαρμογών της εν λόγω εταιρείας. Συγχρόνως, ο χρήστης (δηλαδή ο πελάτης) έχει δεσμευτεί σε ορισμένες εγγυήσεις ποιότητας εξυπηρέτησης που καλύπτουν τη διαθεσιμότητα, την αξιοπιστία, και το χρόνο απόκρισης, να εξασφαλίσει συνεχείς επιχειρηματικές λειτουργίες, και έχει στηριχτεί σε εξωτερικούς φορείς παροχής υπηρεσιών για να παρέχει την εφαρμογή, το Διαδίκτυο, το δίκτυο, και άλλες υπηρεσίες.

Κατά συνέπεια, τα SLAs έχουν γίνει πιο σύνθετα και ευρύτερου πεδίου, με το χρήστη να έχει διάφορα SLAs από διαφορετικούς προμηθευτές. Ένας προμηθευτής, με τη σειρά του, μπορεί να έχει το SLAs του με άλλους προμηθευτές, κάθε ένα με ένα διαφορετικό σύνολο απαιτήσεων, κριτηρίων μέτρησης, και εξαιρέσεων.

4.2.2. Νέες Κατευθύνσεις και εφαρμογές

Η ανάπτυξη του διαδικτύου καθώς και οι νέες κατευθύνσεις στις οποίες αυτό κινείται, ειδικά στα εταιρικά Intranets υποχρεώνουν αλλά ταυτόχρονα δίνουν τη δυνατότητα να ενσωματωθούν και να συγκλίνουν για την επίτευξη ενός στόχου νέα μέσα και ευκαιρίες καθώς και να συνεργαστούν ανόμοια συστήματα από διαφορετικές υπηρεσίες και διαφορετικές πηγές [22]. Οι υπηρεσίες Ιστού έχουν φέρει μια νέα τάξη πραγμάτων στα SLAs καθώς τα έχουν καταστήσει πιο προκλητικά και πιο περίπλοκα, αφού οι σχέσεις ανάμεσα στους προμηθευτές και στους πελάτες γίνονται όλο και πιο δεσμευτικές όλο και πιο πολύπλοκες και πολύπλευρες. Τα συμφωνητικά αυτά χρησιμοποιούνται σαν κάτι παραπάνω από απλές εγγυήσεις για την απόδοση και τη συνεχή διαθεσιμότητα των δικτύων. Χρησιμοποιούνται ως μια αυστηρή εγγύηση για την απόδοση μιας εφαρμογής παίρνοντας ως δεδομένο το γεγονός ότι κάθε υπηρεσία Ιστού έχει διαφορετικά χαρακτηριστικά και κατ'επέκταση έχει και διαφορετικές απαιτήσεις από δικτυακούς πόρους. Έτσι θα μπορούσαμε να ισχυριστούμε ότι ακόμη και το ίδιο το SLA αποτελεί μια δημόσια υπηρεσία του Ιστού.

Όλες οι υπηρεσίες που αφορούν τον ιστό παρέχουν μια ευελιξία για την ενσωμάτωση και την τροποποίησή κάποιων τμημάτων τους μέσω του Ιστού, για να επιτραπεί στον χρήστη η βέλτιστη εξυπηρέτηση [32]. Έτσι ο χρήστης έχει τη δυνατότητα να αλλάξει τις απαιτήσεις του ώστε να αντιμετωπίσει τον ανταγωνισμό για τους δικτυακούς πόρους που τον αφορούν,

Simple Object
Access Protocol (SOAP) και του Universal Description and Discovery Interface (UDDI)

α μια υπηρεσία Ιστού προτού να προωθηθεί στο περιβάλλον παράγωγης και εκτεθεί ως δημόσια υπηρεσία σε UDDI ή σε άλλη δημόσια καταχώρηση. Ένα κάλο παράδειγμα θα ήταν ένα ενιαίο SLA που ισχύει για κάθε κομμάτι μιας υποδομής Ιστού, από το Διαδίκτυο έως τις εφαρμογές υπηρεσιών Ιστού.

Όσον αφορά τα γνωρίσματα μιας υπηρεσίας ιστού που αφορά τόσο σταθερά όσο και κινητά δίκτυα και τα οποία πρέπει να εξεταστούν προτού δημοσιοποιηθεί μια υπηρεσία φαίνονται παρακάτω (Πίνακας 13) :

Εξεταζόμενος Τύπος	Ερώτημα
Statefullness	Όταν γίνεται χρήση SOAP για να τεθεί μια τιμή στον εξυπηρετητή, ο εξυπηρετητής ανταποκρίνεται σωστά στις επόμενες καταστάσεις;

Εξεταζόμενος Τύπος	Ερώτημα
Πρόσβαση	Μπορεί ένας αναρμόδιος χρήστης να έχει πρόσβαση επιτυχώς σε έλεγχο που μόνο οι χειριστές εξουσιοδοτούνται να χρησιμοποιήσουν;
Χρόνος Απόκρισης	Πόσο χρόνο παίρνει η υπηρεσία για να αποκριθεί;
Τέλος Χρόνου	Τι συμβαίνει όταν μια υπηρεσία είναι εκτός χρόνου;
Εκδόσεις (Versioning)	Μπορεί μια νέα υπηρεσία να σπάσει μια υπάρχουσα δικτυακή λειτουργία;

Πίνακας 13. Πηγή: Judith Myerson, Use SLAs in a Web services context, Part 1: Guarantee your Web service with a SLA, 2004

4.2.3. Κοινά Μεγέθη Μέτρησης

Τα SLA μπορούν να περιέχουν πολλαπλά μεγέθη μέτρησης της απόδοσης μιας συγκεκριμένης υπηρεσίας, με αντίστοιχους στόχους την λειτουργία της σε ένα καθορισμένο επίπεδο. Μια κοινή περίπτωση σε ένα IT τμήμα διαχείρισης της υπηρεσίας, είναι ένα τηλεφωνικό κέντρο ή ένα γραφείο υπηρεσιών [33]. Τα μεγέθη μέτρησης που συχνά συμφωνούνται περιλαμβάνουν τα εξής:

ABA (Abandonment Rate - Ρυθμός Απόρριψης - Εγκατάλειψης): Είναι το ποσοστό των κλήσεων που εγκαταλείπονται ενώ περιμένουν για να απαντηθούν.

ASA (Average Speed to Answer - Μέση Ταχύτητα Απάντησης): Είναι ο μέσος χρόνος (συνήθως σε δευτερόλεπτα) που χρειάζεται μια κλήση για να απαντηθεί από έναν γραφείο υπηρεσιών.

TSF (Time Service Factor - Παράγοντας Χρόνου Υπηρεσίας): Το ποσοστό των κλήσεων που απαντώνται σε ένα συγκεκριμένο χρονικό διάστημα (παράθυρο).

FCR (First Call Resolution - Ανάλυση Πρώτης Κλήσης): Είναι το ποσοστό των κλήσεων που μπορούν να επιλυθούν χωρίς την ύπαρξη επανάκλησης ή χωρίς την επανάκληση από τον χρήστη για την ολοκλήρωση της επίλυσης του προβλήματος.

TAT (Turn Around Time - Χρόνος Ολοκλήρωσης): Είναι ο χρόνος που χρειάζεται για να ολοκληρωθεί μια διαδικασία.

4.3. Συμφωνητικά Επιπέδου Εξυπηρέτησης - Είδη

Τα SLA είναι επίσης ορισμένα σε διαφορετικά επίπεδα [30]:

SLA με Βάση στον Πελάτη (Customer Based SLA): Μια συμφωνία με μια μεμονωμένη ομάδα πελατών που περιλαμβάνει όλες τις υπηρεσίες που χρησιμοποιούν. Για παράδειγμα ένα SLA μεταξύ σε έναν προμηθευτή (πάροχο τεχνικό υπηρεσιών) και το οικονομικό τμήμα μιας μεγάλης επιχείρησης για υπηρεσίες όπως είναι το οικονομικό σύστημα, το σύστημα μισθοδοσίας, το σύστημα τιμολόγησης το σύστημα πωλήσεων και αγορών κτλ.

SLA με Βάση την Υπηρεσία (Service Based SLA): Μια συμφωνία για όλους τους πελάτες που χρησιμοποιούν τις υπηρεσίες που παραδίδονται από τον πάροχο υπηρεσιών.

SLA Πολυεπίπεδο (Multi-Level SLA): Το SLA αυτής της κατηγορίας διασπάται σε διαφορετικά επίπεδα, καθένα από τα οποία απευθύνεται σε διαφορετικές ομάδες πελατών για τις ίδιες υπηρεσίες του ίδιου SLA.

SLA εταιρικού επιπέδου (Corporate Level SLA): Το συμφωνητικό αυτό περιλαμβάνει όλες τις γενικές αρχές της διαχείρισης του επιπέδου υπηρεσιών που αφορούν το κάθε πελάτη για έναν οργανισμό. Αυτές οι αρχές είναι πιθανό να είναι λιγότερο ευμετάβλητα και έτι ενημερώσεις πάνω στο αρχικό συμφωνητικό είναι σπάνιες.

SLA επιπέδου πελάτη (Customer Level SLA): Καλύπτει όλες τις γενικές αρχές διαχείρισης που είναι σχετικές με τη συγκεκριμένη ομάδα πελατών ανεξαρτήτως του είδους της υπηρεσίας που χρησιμοποιείται.

SLA επιπέδου Υπηρεσίας (Service Level SLA): Περιλαμβάνει όλες τις γενικές αρχές διαχείρισης επιπέδου υπηρεσιών που είναι σχετικές με τη συγκεκριμένη υπηρεσία, πάντα σε σχέση με τη συγκεκριμένη ομάδα πελατών.

4.4. Οντότητες στα Συμφωνητικά Επιπέδου Εξυπηρέτησης

Ο πάροχος μιας υπηρεσίας δικτύου μπορεί να είναι ένας χειριστής, ένας μεταφορέας, ένας ISP ή μια εταιρεία με χαρακτήρα ASP. Βέβαια στη δική μας περίπτωση ο πάροχος μπορεί να έχει και τον χαρακτήρα μιας εταιρείας κινητής τηλεφωνίας, η οποία παρέχει τα ανάλογα για το δίκτυο της (Υπηρεσίες Internet σε κινητό τερματικό, Υπηρεσίες τηλεφωνίας κτλ). Από την πλευρά του πελάτη είναι πιθανό να έχουμε πάλι μεταφορείς, ISPs αλλά πιο συχνά μια επιχείρηση ή έναν συνδρομητή (τελικός χρήστης) [22]. Έτσι τα 2 μέρη μια τέτοιας συμφωνίας συνδέονται συνήθως με τον τρόπο που φαίνεται στον παρακάτω πίνακα (Πίνακας 14).

Καταναλωτής		Πάροχος
Μεταφορέας	<->	Μεταφορέας
ISP/μεταφορά	<->	Μεταφορέας
Επιχείρηση/χειριστής	<->	ISP
Επιχείρηση	<->	ASP

Καταναλωτής		Πάροχος
Συνδρομητής	<->	ISP

Πηγή: Emmanuel Marilly, Olivier Martinot, H el ene Papini, Danny Goderis, SERVICE LEVEL AGREEMENTS: A MAIN CHALLENGE for Next Generation Networks

Πίνακας 14

Όπως είναι προφανές, είναι εξαιρετικά δύσκολο για τους φορείς παροχής υπηρεσιών να προσφέρουν διαχείριση επιπέδων εξυπηρέτησης και κατ'επέκταση περίπλοκα SLAs. Αυτή η δυσκολία έγκειται στο γεγονός ότι οι φορείς παροχής υπηρεσιών πρέπει να διαχειρίζονται πολλές διαφορετικές τεχνολογίες και στοιχεία δικτύων. Επίσης, ένα ακόμη μεγάλο και βασικό πρόβλημα είναι ότι πολλές φορές μέρη και τμήματα δικτύου που αφορούν τους πελάτες ενός παρόχου, ίσως να ανήκουν σε κάποιον άλλο πάροχο, με αποτέλεσμα την επικάλυψη εξουσιοδότησης, σε περίπτωση παρέμβασης για επίλυση προβλήματος.

4.4.1. Συστατικά Συμφωνητικών Επιπέδου Υπηρεσιών

Παρ'ότι υπάρχουν SLA για πολλούς διαφορετικούς τύπους υπηρεσιών, υπάρχουν κάποια προκαθορισμένα πεδία στο συμφωνητικό, τα οποία θα πρέπει να περιλαμβάνονται. Κάποια από αυτά είναι η περιγραφή της υπηρεσίας, τα πρότυπα των υπηρεσιών, η διάρκεια, οι ρόλοι και οι ευθύνες των δύο μερών καθώς και κάποια κριτήρια αξιολόγησης [37]. Όλα αυτά τα χαρακτηριστικά και τα περιεχόμενα αναφέρονται παρακάτω:

- **Περιγραφή της υπηρεσίας που παρέχεται.** Στο πεδίο αυτό περιγράφεται ο τύπος υπηρεσίας που παρέχεται και τα χαρακτηριστικά του στοιχεία (δυνατότητες ή ανάγκες). Τα χαρακτηριστικά αυτά συνήθως φαίνονται σε έναν κατάλογο τον οποίο δημιουργούν οι πάροχοι των υπηρεσιών και ο οποίος περιλαμβάνει συγκεντρωτικά όλα τα χαρακτηριστικά της υπηρεσίας. Στον κατάλογο αυτό είναι ορθό, να περιλαμβάνονται όλες οι υπηρεσίες μαζί με τις εφαρμογές, την υποδομή αλλά και τις λειτουργίες άλλων επιχειρήσεων χρειάζονται για την ορθή λειτουργία. Στα πλαίσια της IP

όπως η διαχείριση και η συντήρηση των domain name servers, των dynamic host configuration protocol servers, κλπ. Το σωστό περιεχόμενο είναι το πρώτο βήμα για τη δημιουργία ενός SLA, άλλα δεν πρέπει να δίνεται λιγότερη έμφαση στον τρόπο με τον οποίο είναι γραμμένο, τη γλώσσα που χρησιμοποιείται. Είναι καλό να αποφεύγονται οι πολλοί νομικοί όροι και το SLA να μην ξεφεύγει σε μέγεθος, ώστε οι πελάτες να μπορούν αν το διαβάσουν, ώστε να τηρούν και τους όρους που αυτό περιλαμβάνει.

- **Το αναμενόμενο επίπεδο απόδοσης της υπηρεσίας καθώς και συγκεκριμένη αναφορά στην αξιοπιστία και στον χρόνο απόκρισης.** Ξεκινώντας από την αξιοπιστία θα πρέπει να αναφέρουμε ότι περιλαμβάνει τις απαιτήσεις διαθεσιμότητας, δηλαδή πότε είναι διαθέσιμη η υπηρεσία για χρήση καθώς και ποιά είναι τα χρονικά όρια μη λειτουργίας της που πρέπει να αναμένονται. Αναφερόμενοι στον όρο αξιοπιστία εννοούμε την σύγκριση (το ποσοστό) ανάμεσα στην πραγματική διαθεσιμότητα και στην προβλεπόμενη. Ίσως, για την αποφυγή παρερμηνεύσεων να είναι απαραίτητο να αναφερθεί μέσα στο SLA πότε ένα σύστημα είναι πλήρως διαθέσιμο και πότε όχι. Αν γίνει αυτό τότε ως αξιοπιστία θα πρέπει να ορίσουμε το ποσοστό του χρόνου που το σύστημα είναι πλήρως διαθέσιμο (σε πραγματικό

χρόνο) όπως αυτό περιγράφεται ως προς τον προβλεπόμενο χρόνο που αυτό θα έπρεπε να είναι. Συνήθως για τα όρια της αξιοπιστίας υπάρχει ένα ανεκτό παράθυρο στο οποίο μπορεί να κινείται το χρονικό διάστημα.

Όσον αφορά τώρα την απόκριση του συστήματος που αναφέραμε πιο πάνω ορίζεται ως το πόσο σύντομα θα εκτελεστεί η φυσιολογική πορεία των διαδικασιών (σε φυσιολογικές συνθήκες ή συνθήκες φόρτου). Τέλος έχουμε ένα από τα σημαντικότερα μεγέθη που περιγράφονται μέσα σε ένα SLA και αυτό είναι η απόδοση της συγκεκριμένης υπηρεσίας αλλά και οι στόχοι απόδοσης που έχουν τεθεί από τον πάροχο. Οι στόχοι απόδοσης έχουν τη δυνατότητα να εκτροχιάσουν ένα κατά τα άλλα επιτυχές SLA τόσο από την πλευρά του πελάτη όσο και από την πλευρά του παρόχου. Το SLA θα πρέπει να γράφεται με στόχο τον πελάτη και να περιέχει στόχους που να είναι αρκετά συγκεκριμένοι. Οι στόχοι αυτοί θα μπορούσαν να περιλαμβάνουν το ελάχιστο αλλά και το μέγιστο της αποδιδόμενης υπηρεσίας.

- **Διάρκεια.** Απαραίτητο τμήμα ενός συμφώνου είναι η διευκρίνηση για την έναρξη αλλά και την λήξη μιας συνεργασίας. Οι πληροφορίες για την διάρκεια μιας συμφωνίας συνήθως περιγράφονται ανάμεσα στους όρους συμβάσεων. Η μέρα έναρξης του SLA επιτρέπει στον πελάτη να αρχίσει να μετρά την απόδοση των συμφωνηθέντων υπηρεσιών από αυτή την ημερομηνία (με την επιφύλαξη τυχόν διαφορετικής ρύθμισης). Εάν παρέχεται μια νέα υπηρεσία ή αναθεωρούνται απλά οι υπηρεσίες που προσφέρονται στους πελάτες, απαιτείται αρκετός χρόνος για να διαβιβαστούν οι λεπτομέρειες της συμφωνίας σε αυτούς. Είναι κάλο να συνάπτονται μακροχρόνιες συμβάσεις, σύμφωνα βέβαια με τις ανάγκες του εκάστοτε πελάτη με τους φορείς παροχής υπηρεσιών ώστε να μην τίθεται συνεχώς θέμα ανανέωσης της συμφωνίας.

- **Αντιπρόσωπος πελατών.** Τυπικά ένα άτομο αντιπροσωπεύει όλους τους πελάτες μιας επιχείρησης για τους σκοπούς της διαπραγμάτευσης και παράδοσης των υπηρεσιών. Επιπλέον είναι υπεύθυνος για την ενημέρωση των πελατών που αντιπροσωπεύει για τους όρους αλλά και τις πληροφορίες του SLA.

- **Διαδικασία αναφοράς προβλημάτων με την υπηρεσία.** Στο πεδίο αυτό συμπεριλαμβάνονται όλες οι πληροφορίες για την επίλυση ενός προβλήματος, τη μορφή με την οποία θα πρέπει να αρχειοθετηθούν οι καταγγελίες και τα παράπονα καθώς και τα βήματα, τα οποία θα πρέπει να γίνουν για την γρήγορη επίλυση του προβλήματος. Στο συμφωνητικό επίσης θα περιγράφεται και μια προθεσμία στην οποία θα είχαμε απόκριση επίλυσης του προβλήματος (δηλαδή θα ξεκινούσαν οι εργασίες για την επίλυση του προβλήματος) καθώς επίσης και το χρονικό διάστημα στο οποίο το πρόβλημα θα έπρεπε να έχει επιλυθεί.

- **Το χρονικό πλαίσιο για την απόκριση και τη λύση του προβλήματος.** Αυτό το πεδίο διευκρινίζει μια προθεσμία στην οποία κάποιος θα άρχιζε να ερευνά ένα πρόβλημα που αναφέρθηκε. Η έναρξη της έρευνας χαρακτηρίζεται από έναν αντιπρόσωπο του προμηθευτή που έρχεται σε επαφή με τον πελάτη που εξέθεσε το πρόβλημα αρχικά. Μπορεί επίσης να υπάρξει μια προθεσμία μέχρι την οποία το πρόβλημα θα επιλυόταν. Ένα SLA μπορεί να διευκρινίσει ότι μια πεσμένη σύνδεση θα επανερχόταν μέσα σε 24 ώρες.

- **Διαδικασία για την επίβλεψη και την αναφορά του επιπέδου εξυπηρέτησης.** Στο πεδίο αυτό γίνεται περιγραφή για το πώς ελέγχονται και αναφέρονται τα επίπεδα απόδοσης, για παράδειγμα, ποιος θα κάνει τον έλεγχο, ποιοι τύποι στατιστικών θα συλλεχτούν, πόσο συχνά θα συλλέγονται, και πώς μπορούν να έχουν πρόσβαση σε παρελθοντικές/τρέχουσες στατιστικές. Ίσως κάποιοι προμηθευτές δικτύων, να επέτρεπαν σε πελάτες να έχουν

πρόσβαση άμεσα στα αποτελέσματα των στατιστικών καθώς και στο μέρος του δικτύου που τους αναλογεί, μέσω ενός διαχειριστικού εργαλείου. Παρ όλα αυτά και ενώ ο πελάτης θα είχε πρόσβαση στα αποτελέσματα των μετρήσεων, δεν θα είχε την άδεια να τροποποιήσει είτε να διαμορφώσει την λειτουργία οποιουδήποτε τμήματος του δικτύου ακόμη και αν αυτό το τμήμα αφορούσα καθαρά και μόνο αυτόν.

- **Κριτήρια αξιολόγησης.** Χωρίς κριτήρια αξιολόγησης δεν θα υπήρχε κανένα αντικειμενικό μέσο να καθορίσει πόσο καλά εκτελείται η οργάνωση των Τεχνολογιών Πληροφορικής. Ένα από τα αμοιβαία οφέλη ενός SLA είναι πως η εταιρία προμηθευτής και ο πελάτης καθορίζουν πως θα κρίνεται η υπηρεσία που λαμβάνει δεύτερος. Με την καθιέρωση των αντικειμενικών μετρήσεων γίνεται εκτός από την συστηματική παρακολούθηση και την πλήρη ανάπτυξη στατιστικών ιστορικών και αφαίρεση της αβεβαιότητας για την ορθή ή όχι λειτουργία.

- **Συνέπειες για τον φορέα παροχής υπηρεσιών που δεν εκπληρώνει τις υποχρεώσεις του.** Το SLA εν γένει δημιουργήθηκε για την διασφάλιση του επιπέδου εξυπηρέτησης που θα πρέπει να αναμένει ο πελάτης από έναν πάροχο. Έτσι συχνά, αν όχι πάντα, αναφέρεται και το πεδίο της ρήτρας ή της ποινής που θα πρέπει να υποστεί ο πάροχος για μη τήρηση των συμφωνηθέντων όρων. Είναι εξαιρετικά σύνηθες, να δοθούν ορισμένες πιστώσεις στους πελάτες, όταν δεν ικανοποιούνται οι προσδοκίες υπηρεσιών. Επιπλέον ο πάροχος αντιμετωπίζει συνέπειες, όπως ή δυνατότητα του πελάτη να τερματίσει την συνεργασία αλλά και να ζητήσει αποζημίωση ενός μέρους των εισοδημάτων που χάνει λόγω της απώλειας υπηρεσίας. Οι συνέπειες ποικίλουν βέβαια ανάλογα με τη φύση της σχέσης που έχει ο πελάτης με τον προμηθευτή.

- **Περιπτώσεις Διαφυγής και περιορισμοί.** Οι περιπτώσεις διαφυγής είναι όροι κάτω από τους οποίους δεν υφίσταται το επίπεδο εξυπηρέτησης, και αντίστοιχα κάτω από τους οποίους δεν γίνεται να εκπληρωθεί το απαραίτητο SLA. Παραδείγματα τέτοιων περιπτώσεων θα μπορούσαν να θεωρηθούν οι καταστροφές εξοπλισμού του παρόχου είτε από κάποιο φυσικό φαινόμενο είτε από πυρκαγιά, πλημμύρα κτλ. Είναι επιπλέον συχνό φαινόμενο να επιβάλλονται μερικοί περιορισμοί στην συμπεριφορά του πελάτη, τόσο για την χρήση του δικτύου όσο και για την σχέση ανάμεσα στα δύο μέρη. Έτσι ένας διαχειριστής ενός δικτύου μπορεί να αποβάλει έναν πελάτη αν αυτός προσπαθήσει να παραβιάσει την ασφάλεια του δικτύου, να τροποποιήσει κάποιες ρυθμίσεις αλλά και να κάνει χρήση του δικτύου εκτός των συμφωνηθέντων.

Εξαιρέσεις

Όπως είναι συνηθισμένο σε κάθε συμφωνία ή ασφάλεια υπάρχουν κάποιες εξαιρέσεις στους όρους που συμφωνούνται. Έτσι και το SLA για μια υπηρεσία ιστού πρέπει να περιλάβει με λεπτομέρειες τις εξαιρέσεις αυτές. Προσπαθώντας να βρούμε μια ταξινόμηση για τις εξαιρέσεις καταλήξαμε στο άρθρο της Judith Mayerson - Use SLAs in a Web services context, Part 1: Guarantee your Web service with a SLA ,το 2004, η οποία έκανε μια αυθαίρετη διαίρεση σε τέσσερις περιοχές εξαιρέσεων.

- Αποτυχίες
- Ζητήματα Δικτύων όχι μέσα στον άμεσο έλεγχο του φορέα παροχής υπηρεσιών
- Άρνηση της Υπηρεσίας
- Σχεδιασμένη Συντήρηση

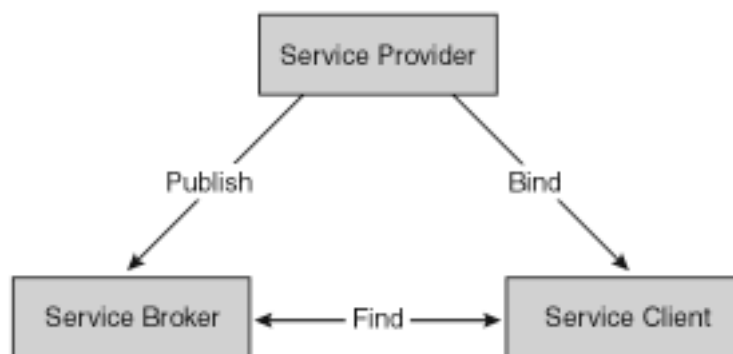
Στον πίνακα που ακολουθεί υπάρχουν εξηγήσεις για συγκεκριμένες περιπτώσεις που μπορούν να αποδοθούν στις τέσσερις αυτές κατηγορίες. Άλλες εξαιρέσεις μπορούν να προστεθούν για να ταιριάξουν με την κατάσταση του παρόχου, εφ' όσον μπορούν να πάρουν οι επιχειρήσεις πελατών το λογικό αντάλλαγμα για το χρόνο διακοπής. Με τη συμπερίληψη των εξαιρέσεων σε ένα SLA, ένας προμηθευτής μπορεί να προστατευτεί από την ευθύνη σε περίπτωση προβλημάτων ή διακοπών λειτουργίας δικτύων. Αφ' εταίρου, εάν οι ανταγωνιστικές υπηρεσίες προσφέρουν SLAs με λιγότερες εξαιρέσεις, ο πελάτης έχει την επιλογή να διαλέξει εκείνες τις συμφωνίες που προσφέρουν περισσότερο «uptime» στις επιχειρηματικές λειτουργίες και με καλύτερες εγγυήσεις υπηρεσιών. Ακόμη και διαφορές μεταξύ 99.5%, 99.9% και 99.999% εγγυήσεων «uptime» διαθεσιμότητας μπορούν να επηρεάσουν τους θύνοντες στην επιλογή ενός SLA.

Τύπος	Παραδείγματα
Αποτυχίες	Αποτυχία υλικού (είναι σπάνιες οι περιπτώσεις ελαττωματικού υλικού) Αποτυχία τηλεπικοινωνιών (τυχαία κοπή γραμμής οπτικών ινών) Ρωγμές Λογισμικού Διακοπή συστήματος ελέγχου/μέτρησης
Ζητήματα Δικτύων όχι μέσα στον άμεσο έλεγχο του φορέα παροχής υπηρεσιών	DNS ζητήματα όχι στον άμεσο έλεγχο του φορέα παροχής υπηρεσιών
Άρνηση της Υπηρεσίας	Αμέλεια Πελατών/σκόπιμη ή κακή χρήση Πλημμύρες Δικτύων, hacks, attacks Πράξεις ανωτέρας βίας, πολεμικές απεργίες μη διαθεσιμότητα των τηλεπικοινωνιών, ανικανότητα να αποκτηθούν οι προμήθειες ή ο απαιτούμενος εξοπλισμός για την παροχή του SLA
Σχεδιασμένη συντήρηση	Βελτιώσεις υλικού Βελτιώσεις Λογισμικού Backups

Πίνακας 15.

4.4.2. Αρχιτεκτονική Υπηρεσιών Ιστού

Η αρχιτεκτονική όπως παρουσιάζεται και στην εικόνα που ακολουθεί (Εικόνα 30), απαιτεί τρεις ρόλους υπηρεσιών. Έναν φορέα υπηρεσιών (service provider), έναν πελάτη (service client) και έναν μεσίτη υπηρεσιών (service broker). Ο φορέας παροχής υπηρεσιών δημοσιεύει (publish) μια SLA-καλυμμένη υπηρεσία Ιστού, με τη δημιουργία μιας υπηρεσίας Ιστού σε μια κατάλληλη πλατφόρμα, και την παραγωγή ενός έγγραφου WSDL (Web Service Definition Language) και ενός βασικού SLA για την υπηρεσία.. Στη συνέχεια στέλνει τις λεπτομέρειες της υπηρεσίας αυτής στον μεσίτη υπηρεσιών για την αποθήκευση της σε μια αποθήκη. Ο πελάτης υπηρεσιών εγγράφεται στον μεσίτη. Έτσι αποκτά τη δυνατότητα να δει και να αγοράσει οποιαδήποτε από τις υπηρεσίες που έχει αποθηκευμένες ο μεσίτης. Όταν βρει την επιθυμητή υπηρεσία ιστού τότε ανακτά το WSDL και το SLA για την υπηρεσία. Διαπραγματεύεται με τον προμηθευτή για να τροποποιήσει και να οριστικοποιήσει το SLA και να δεσμεύσει την συγκεκριμένη υπηρεσία Ιστού.



Εικόνα 30: Αρχιτεκτονική μιας Διαδικτυακής υπηρεσίας που καλύπτεται από SLA

Πηγή: Judith Myerson, Use SLAs in a Web services context, Part 1: Guarantee your Web service with a SLA, 2004

Οι συμφωνίες επιπέδων εξυπηρέτησης, ή " SLA' s" είναι δυσνόητοι αλλά χρήσιμοι μηχανισμοί για τη διαχείριση του κίνδυνου μιας τρέχουσας σχέσης με τους φορείς παροχής υπηρεσιών ΤΠ. Δυστυχώς, τα περισσότερα SLA' s που παρουσιάζονται στις συμβάσεις υπηρεσιών είναι άνευ αξίας, διακοσμητικές προσθήκες . Ένα SLA μπορεί να είναι εξαιρετικά ισχυρό εργαλείο για να βοηθήσει τον πελάτη και το φορέα παροχής υπηρεσιών να αποκομίσουν όσο περισσότερα γίνεται από μια σχέση.

4.5. Προδιαγραφές Επιπέδων Εξυπηρέτησης

Οι προδιαγραφές των επιπέδων εξυπηρέτησης (SLS - Service Level Specifications) είναι οι τεχνικές προδιαγραφές που προκύπτουν από το SLA. Το SLS μπορεί να είναι μια προδιαγραφή άμεσα σχετική με το SLA, αλλά επίσης μπορεί να είναι και μια ερμηνεία του SLA, μια προσαρμογή ανάλογα με τον προμηθευτή ή τον χρήστη.

Για να προταθεί μια end-to-end λύση για την διαχείριση SLA/QoS, θα πρέπει πρώτα απ' όλα να καθοριστούν οι υπηρεσίες, οι παράμετροι SLS, καθώς και μια ταξινόμηση των υπηρεσιών

ανάλογα με τις παραμέτρους. Το γεγονός ότι γίνεται εστίαση στην ποιότητα της εξυπηρέτησης και όχι στο επίπεδο της τεχνολογίας δικτύων, καθιστά πιο εύκολο τον καθορισμό service/SLA/QoS σε ανεξάρτητο βαθμό από το τεχνολογικό υπόβαθρο στο οποίο αυτά βασίζονται για να υλοποιηθούν.

4.5.1. Υπηρεσίες και Παράμετροι

Στον παρακάτω πίνακα (Πίνακα 4) μπορούμε να δούμε έναν κατάλογο υπηρεσιών με το μεγαλύτερο δυνατό εύρος (ήχος, βίντεο, δεδομένα, tunnel) [32].

Κωδικός	Περιγραφή
S1	Φωνή
S2	Βίντεο + Φωνή
S3	Τηλεφωνικές Υπηρεσίες
S4	Πολυμέσα
S5	Video on Demand
S6	White Board
S7	VPNx(IP VPN, optical VPN, tel VPN = PMR)
S8	Δεδομένα Πραγματικού χρόνου
S9	Διαδραστικά δεδομένα (διαδίκτυο, συναλλαγές, ηλεκτρονικό εμπόριο, διακομιστής ηλεκτρονικού ταχυδρομείου)
S10	Ροή Δεδομένων (FTP, bulk data)

Πίνακας 15.

Πηγή: Emmanuel Marilly, Olivier Martinot, H el ene Papini, Danny Goderis, SERVICE LEVEL AGREEMENTS: A MAIN CHALLENGE for Next Generation Networks

Οι παράμετροι οι οποίες εξετάζονται και περιλαμβάνονται σε ένα SLS είναι πρακτικά τα μετρούμενα μεγέθη για τον καθορισμό μιας QoS. Στον παρακάτω πίνακα φαίνονται συγκεντρωτικά αυτές οι παράμετροι.

Κωδικός	Όνομα Παραμέτρου
P1	Απώλεια
P2	Καθυστέρηση
P3	Jitter (Διασπορά Σήματος)
P4	Εύρος Ζώνης

Κωδικός	Όνομα Παραμέτρου
P5	Καταναλωτής/Πάροχος
P6	Έλεγχος(Monitoring)
P7	Έλεγχος Πρόσβασης
P8	Τοπολογία
P9	Αξιοπιστία/Προστασία (MDT, MTBF, MTTR)
P10	Μονής κατεύθυνσης/συμμετρικό/ασύμμετρο

Πίνακας 14

Πηγή: Emmanuel Marilly, Olivier Martinot, H el ene Papini, Danny Goderis, SERVICE LEVEL AGREEMENTS: A MAIN CHALLENGE for Next Generation Networks

Όπως είναι λογικό δεν είναι  λες οι παράμετροι το ίδιο σημαντικές. Τουλάχιστον όχι σε  λες τις περιπτώσεις. Το ενδιαφέρον γι αυτές αλλάζει όχι μόνο ανάμεσα σε διαφορετικές υπηρεσίες, αλλά και ανάμεσα και σε διαφορετικά μέρη που κάνουν την συμφωνία. Έτσι  πως μπορούμε να δούμε και από τον πίνακα οι παράμετροι P1, P2, P3 θα έχουν ιδιαίτερη βαρύτητα για ένα χρήστη, αλλά οι P4 και P9 είναι περισσότερο προσανατολισμένες για μια επιχείρηση. Αν υποθέσουμε ότι στην πλευρά του πελάτη έχουμε ένα χρήστη, τότε θα οι παράμετροι που γίνονται περισσότερο αντιληπτές θα είναι η απώλεια, η καθυστέρηση καθώς και το Jitter. Αν  μως στη θέση του τελικού χρήστη τοποθετήσουμε μια επιχείρηση, τότε οι καθολικές παράμετροι που αντιστοιχούν σε ένα επίπεδο τιμολόγησης είναι το  ρος ζώνης και η αξιοπιστία (MDT, MTBF, MTTR). Όταν η αξιοπιστία ή το παρεχόμενο  ρος ζώνης δεν είναι ικανοποιητικά, μπορεί επίσης να είναι ένας καθοριστικός λόγος να κινηθεί προς έναν  λλο προμηθευτή [33].

Για καλύτερη οπτική των παραπάνω λεχθέντων θα ενώσουμε σε ένα πίνακα τα περιεχόμενα των 2 πινάκων που αναφέραμε παραπάνω. Επίσης θα δούμε πως γίνεται η αλληλεπίδραση των SLS παραμέτρων με τις υπηρεσίες. Χρησιμοποιείται συμβολισμός τεσσάρων στοιχείων τα οποία δείχνουν το είδος της αλληλεπίδρασης και αυτά είναι:

- ++ : πολύ υψηλή απόδοση
- + : υψηλή απόδοση
- += : μέση (εξ ορισμού απόδοση)
- +0 : αδιάφορο

		P1	P2	P3	P4	P5	P6	P7	P8	P9	P10
		Απώλεια	Καθυστερήση	Jitter	Εύρος ζώνης	Καταναλωτής / πάροχος	Έλεγχος (monitoring)	Έλεγχος πρόσβασης	Τοπολογία	Αξιοπιστία	κατεύθυνση
S1	Φωνή	=	++	++	+	++	=	++	=	+	2
S2	Βιντεόφωνο	=	++	+	++	++	=	++	=	+	2
S3	Τηλεφωνικές υπηρεσίες	=	=	0	0	++	=	++	=	+	1
S4	Πολυμεσα	+	++	0	++	++	=	++	=	+	1.5
S5	Video on demand	=	++	0	++	++	=	++	=	+	1
S6	White board	++	+	0	0	++	=	++	=	+	2
S7	VPN	Εξαρτάται από τον τύπο δεδομένων ροής					=	++	++	++	2
S8	Δεδομένα πραγματικού χρόνου	++	++	0	+	++	=	++	=	+	2
S9	Διαδραστικά δεδομένα	++	+	0	0	++	=	++	=	+	1
S10	Ροή δεδομένων	++	=	0	0	++	=	++	=	+	1

Πίνακας 15.

4.6. Σύμφωνα Επιπέδου Εξυπηρέτησης σε Ασύρματα Δίκτυα

Αφού καλύψαμε τα γενικά χαρακτηριστικά των SLAs, τα οποία βρίσκουμε στην πλειοψηφία των δικτύων θα προσπαθήσουμε να δούμε τα συμφωνητικά αυτά από την πλευρά των ασύρματων δικτύων. Μέχρι σήμερα οι διαχειριστές δικτύων είχαν να ανησυχούν μόνο για την φυσική ασφάλεια των δικτύων τους καθώς και των σταθερών μηχανημάτων τεχνολογίας που χρησιμοποιούσαν. Σε αυτά τα μηχανήματα περιλαμβάνονται ηλεκτρονικοί υπολογιστές/διακομιστές, δρομολογητές και firewalls. Όμως με την πτώση του κόστους του εξοπλισμού ασύρματων δικτύων αλλά και λόγω της ανάγκης για φορητότητα των τερματικών και των δικτυακών υπηρεσιών που αυτά χρησιμοποιούν δημιουργήθηκε ένα νέο πεδίο στον οποίο εμφανίζονται προβλήματα ασφάλειας.

Αυτή η νέα τεχνολογία έχει βοηθήσει να χαμηλώσει το κόστος των δικτύων, να παρέχει πρόσβαση σε μέρη που δεν υπήρχε πριν, και κατέστησε τον όρο «κινητή υπολογιστική» μια πραγματικότητα. Επίσης, έχει αλλάξει δραστικά την περίμετρο ασφαλείας δικτύων για τις επιχειρήσεις όλων των μεγεθών. Παραδοσιακά, τα εταιρικά δίκτυα συνδέονταν με τον εξωτερικό κόσμο μόλις σε λίγα σημεία. Αυτό επέτρεπε στους διευθυντές δικτύων να επικεντρωθούν στην προστασία αυτών των περιορισμένων σημείων πρόσβασης. Θα μπορούσαν να μπουν firewalls και άλλες άμυνες σε αυτά τα κρίσιμα σημεία. Το εσωτερικό του δικτύου αντιμετωπιζόταν ως έμπιστο επειδή δεν υπήρξε κανένας τρόπος να μπει κανείς εκεί πέρα από τα προστατευμένα σημεία. (Πληροφορίες για τον τρόπο που χειρίζονται ανάλογα θέματα λάβαμε από ενδεικτικά σύμφωνα επιπέδων εξυπηρέτησης εταιρειών διαχείρισης ασύρματων δικτύων όπως είναι η Covad, η Lightning Bolt αλλά και η NEC.)

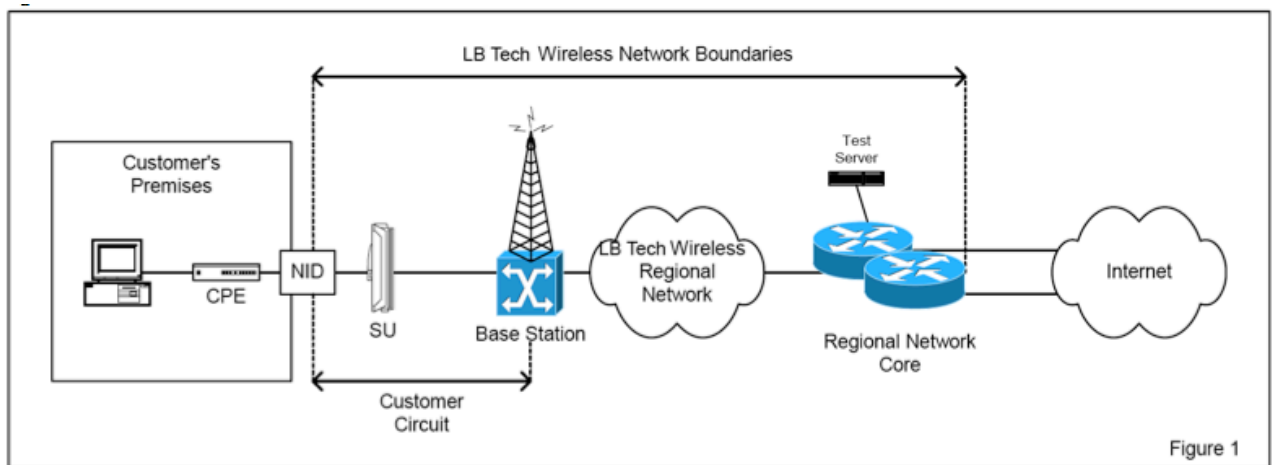
Λόγω της μεγάλης δυσκολίας για την επίτευξη των στόχων μια συμφωνίας ασύρματου δικτύου, αλλά και λόγω της πρόσφατης εκρηκτικής ανάπτυξης των ασύρματων δικτύων τα SLA που τα αφορούν δεν έχουν μεγάλη ιστορία και σίγουρα όχι και τόσο μεγάλη εμφάνιση όσο έχουν αυτά για τα σταθερά δίκτυα. Παρ' όλα αυτά η συνεχής αύξηση των απαιτήσεων για ορατότητα καθώς και για ασύρματα δίκτυα, αλλάζουν συχνά το τοπίο με όλο και περισσότερες εταιρείες να δημιουργούν SLAs που να αφορούν τις υπηρεσίες που παρέχουν σε ένα ασύρματο δίκτυο.

Χαρακτηριστικό παράδειγμα είναι η εταιρεία NEC η οποία έχει αναπτύξει ένα σύστημα για την διαχείριση επιπέδων εξυπηρέτησης, επιτρέποντας την παροχή υψηλής ποιότητας κινητών υπηρεσιών σε επιχειρήσεις. Το αναπτυγμένο αυτό σύστημα είναι μια πλατφόρμα υπηρεσιών που υποστηρίζει τους κινητούς φορείς εκμετάλλευσης δικτύου, επιτρέποντας τους να τους παρέχονται περιεκτικές υπηρεσίες βασισμένες στις συμφωνίες επιπέδων εξυπηρέτησης (SLA) με τις επιχειρήσεις. Υποστηρίζει τη διοικητική SLA ροή της δουλειάς, που αποτελείται από την αξιολόγηση, τον έλεγχο και τη συντήρηση της ποιότητας δικτύων σε μια περιορισμένη περιοχή SLA.

Η υλοποίηση της πλατφόρμας αυτής ήταν το αποτέλεσμα της ανάπτυξης των ακόλουθων τεχνολογιών:

- Τεχνολογία ράδιο-ποιοτικής αξιολόγησης, η οποία αξιολογεί την κατάσταση των ραδιοκυμάτων σε κάθε πάτωμα των κτηρίων της επιχείρησης, η οποία και καλείται ως περιοχή στόχος για το SLA.
- «Anormal» τεχνολογία μεταλλείας κυττάρων που ανιχνεύει αποτελεσματικά και προβλέπει τις ελαττωματικές ράδιο-περιοχές κυττάρων (περιοχές υπηρεσιών), οι οποίες έχουν τη δυνατότητα να παραβιάσουν το SLA. Αυτή τεχνολογία αναλύει στατιστικά τα στοιχεία απόδοσης που ελέγχονται από τα επιχειρησιακά κινητά τερματικά και το ράδιο δίκτυο πρόσβασης μέσω της ανίχνευσης ανωμαλίας και της ανύψωσης δεδομένων χρονικής σειράς.
- Τεχνολογία έλεγχου συντήρησης SLA, η οποία αποκαθιστά τα επίπεδα εξυπηρέτησης στα ανιχνευμένα/προβλεφθέντα ελαττωματικά κύτταρα σε ένα επίπεδο ισοδύναμο με την ποιότητα μετάδοσης που εξασφαλίζεται στην επιχείρηση (χρήστης στόχων SLA). Αύτη η τεχνολογία ρυθμίζει δυναμικά τις παραμέτρους του έλεγχου αποδοχής και του έλεγχου παράδοσης στον ελεγκτή ραδιοφωνικών δικτύων (RNC).

Άλλο χαρακτηριστικό παράδειγμα είναι και η εταιρεία Lightning Bolt της οποίας παρουσιάζεται ένα σχεδιάγραμμα των SLA, που αφορά το ασύρματο το οποίο και παρέχει.



Εικόνα 31. Σχέδιο Τοπολογίας δικτύου Εταιρείας LB

Για την εταιρεία LB το σύμφωνο επιπέδων εξυπηρέτησης έχει ως στόχους:

Διαθεσιμότητα Δικτύου: 99.9%

Information Rate: Όπως αυτό καθορίζεται αν προϊόν

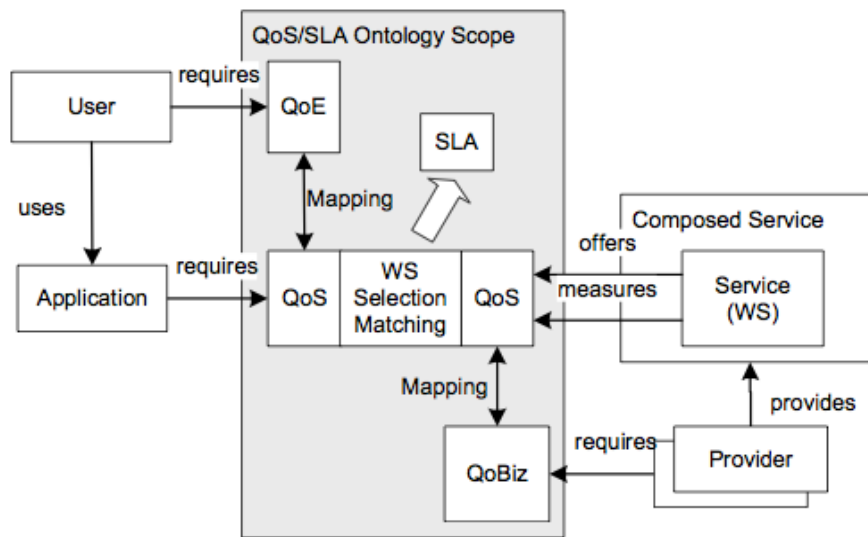
Καθυστέρηση Δικτύου: 50ms

Απώλεια Πακέτων: <1%

Οι στόχοι που βλέπουμε παραπάνω είναι χαρακτηριστικοί για εταιρείες ασύρματων δικτύων για το λόγο αυτό θα προσπαθήσουμε να αναλύσουμε το συγκεκριμένο SLA και μέσω αυτής της οδο ύ να γίνει κατανοητή γενικά η διαδικασία και η μορφή των συμφωνιών που υφίστανται στα ασύρματα δίκτυα.

4.7. Σύμφωνα Επιπέδου Εξυπηρέτησης και Ποιότητα Υπηρεσίας

Όπως είναι προφανές από την περιγραφή των του όρου SLA καθώς και QoS μπορούμε να καταλάβουμε πως αυτά τα δύο “μεγέθη” είναι αλληλένδετα αφού το ένα στηρίζει το άλλο. Όπως είδαμε τόσο τα σύμφωνα επιπέδων εξυπηρέτησης όσο και η ποιότητα εξυπηρέτησης έχουν ορισμένες παραμέτρους, οι οποίες λειτουργούν σαν χαρακτηριστικά γνωρίσματα. Παρ’ όλα αυτά δεν έχει αναφερθεί ο συνδετικός κρίκος ή ο τρόπος με τον οποίο γίνεται απόλυτα η σύνδεση αυτών των δύο. Όπως φαίνεται και στην παρακάτω εικόνα, η διαδικασία για την επιλογή μιας υπηρεσίας από ένα πελάτη μπορεί να περιγράφει με μία εικόνα όπως αυτή που ακολουθεί. (αλγόριθμο – Εικόνα 32).



Εικόνα 32. Αλγόριθμος Επιλογής Κατάλληλου SLA για την υποστήριξη μιας καθορισμένης ποιότητας υπηρεσίας.

Όπως παρατηρούμε ο χρήστης δεν απαιτεί QoS αλλά QoE (Quality of Experience) . Το QoS απαιτείται από την εφαρμογή την οποία ο χρήστης θέλει να εκτελέσει. Παρ όλα αυτά το QoS της εφαρμογής και το QoE του χρήστη θα πρέπει να συμπίπτουν σε έναν και μόνο πάροχο. Μόλις γίνει αυτή η διασταύρωση ο πελάτης θα κάνει προσφορά στον φορέα είτε άμεσα είτε μέσα από έναν μεσολαβητή, όπως αναφέραμε προηγουμένως. Παρ όλα αυτά θα πρέπει να αναφέρουμε ότι αυτή είναι μια γενική θεώρηση.

Μια αρχική προσέγγιση στο συγκεκριμένο θέμα δεν είναι απλά η σύγκριση των απαιτήσεων του χρήστη με τις πληροφορίες που έχει δημοσιεύσει ο πάροχος, αλλά και την διασταύρωση των μετρήσεων που έχουν γίνει για την υπηρεσία αυτή. Η επιλογή του κατάλληλου SLA όπως αυτό φαίνεται και στην Εικόνα 31, λαμβάνει υπ όψιν της την QoS από τεχνικής άποψης, την αντίληψη του χρήστη για την ποιότητα με την Quality of Experience αλλά και την Quality of Business. Όλα αυτά φαίνονται στο παραπάνω σχήμα με την προσθήκη επιπλέον σχημάτων στον χάρτη απόφασης και μεταφράζονται στα επιμέρους επίπεδα ποιότητας.

Σε μια δεύτερη προσέγγιση είναι η παρακολούθηση της ποιότητας εξυπηρέτησης καθώς και ο εντοπισμός διάφορων παραβιάσεων του SLA. Για τον λόγο αυτό γίνονται μετρήσεις για να ανιχνευτούν οι πραγματικές συμπεριφορές της υπηρεσίας. Έτσι δημιουργείται μια περιγραφή των απαιτήσεων η οποία και συγκρίνεται με τα αντίστοιχα στοιχεία και τις τιμές που παίρνουν στο SLA. Στην περίπτωση αυτή οι τιμές αυτές δεν συμφωνούν με αυτές που έχουν μετρηθεί, τότε αν μπορεί να γίνει τροποποίηση του SLA με την πρακτική υποστήριξη των νέων αριθμών, υπάρχει συμφωνία ανάμεσα στα δύο μέρη.

Θα πρέπει να γίνει κατανοητό ότι για εικονικούς λόγους κυρίως, οι επιχειρήσεις αλλά και οι τελικοί χρήστες, δεν είναι διατεθειμένοι τις περισσότερες φορές να αγοράσουν υπηρεσία πλεονάζουσα σε δυνατότητες από αυτή που θέλουν. Οι QoS παρακολουθείται σε πραγματικό χρόνο και τα χαρακτηριστικά της καταγράφονται. Μερικές ελεγχόμενες παράμετροι μπορούν

να ρυθμιστούν έτσι ώστε να αντιδράσω σε μια πιθανή μεταβολή της ποιότητας, για παράδειγμα μειώνοντας τον αριθμό των συνδέσεων.

Για να μπορούμε να έχουμε την ιδανική επιλογή ενός παρόχου μιας υπηρεσίας εκτός από φυσικά την γνώση των απαιτήσεων μας θα πρέπει να ξέρουμε τα εξής:

- Θα πρέπει να γίνει εντοπισμός των καλύτερων QoS που είναι απαραίτητες για την λειτουργία της πραγματικής εφαρμογής. Θα πρέπει να έχουμε υπ' όψιν μας ότι μια εφαρμογή μπορεί να χρησιμοποιεί πολλές υπηρεσίες, οι οποίες μπορεί να παρέχονται από διαφορετικό φορέα. Επίσης λόγω της ποικιλότητας είναι δυνατό να υπάρχουν διαφορετικά επίπεδα για την κάθε υπηρεσία.
- Θα πρέπει να γίνει διαλογή επίσης και ανάλογα με το πεδίο στο οποίο θα εκτελεστεί η εφαρμογή αυτή. Θα έχουμε διαφορετικές απαιτήσεις για δύο διαφορετικές υπηρεσίες ιστού.
- Είναι βασικό επίσης να γίνεται ορθή αντιστοίχιση της διαφημιζόμενης υπηρεσίας και των απαιτήσεων. Αυτό προϋποθέτει πως ο πάροχος της εν λόγω υπηρεσίας έχει δώσει με λεπτομερή περιγραφή τα χαρακτηριστικά της υπηρεσίας του.
- Όταν έχουν γίνει τα απαιτούμενα βήματα για τον προσδιορισμό της ποιότητας υπηρεσίας την οποία απαιτεί η εφαρμογή μας αλλά και αυτών που δίνουν οι πάροχοι, τότε θα πρέπει να γίνει η επιλογή ανάμεσα στις προτεινόμενες εταιρείες ανάλογα με το SLA που αυτές προσφέρουν.
- Η ποιότητα εξυπηρέτησης δεν είναι το μοναδικό που θα πρέπει να κοιτάξει ο χρήστης σε ένα SLA.
- Επιλογή με βάση ρεαλιστικά στοιχεία για το αν οι στόχοι που υποστηρίζει μια εταιρεία είναι υλοποιήσιμοι.
- Επιλογή με βάση το κέντρο βάρους ευθύνης που δίνει ένα SLA. Όπως έχουμε αναφέρει και προηγουμένως υπάρχει η περίπτωση ένα SLA να δεσμεύει τον πάροχο αλλά να δεσμεύει και τον πελάτη. Ο πελάτης θα πρέπει να αποφασίσει για το ποσό της δέσμευσης που είναι διατεθειμένος να δεχτεί αλλά και για το ποσό της δέσμευσης που θέλει να έχει ο πάροχος του.
- Επιλογή με βάση την κοστολόγηση, διάρκεια.
- Επιλογή με βάση τις ρήτρες που θα πρέπει να υπάρχουν για τα δύο μέρη σε περίπτωση αθέτησης της συμφωνίας.

5. Εργαλεία Μέτρησης Ποιότητας Υπηρεσίας

5.1. Εισαγωγή

Η ποιότητα υπηρεσίας είναι απαραίτητο να ελέγχεται τόσο από την πλευρά του πελάτη όσο και από την πλευρά του παρόχου. Επιπλέον είναι απαραίτητο ο πάροχος των υπηρεσιών να μπορεί να ελέγξει και να δοκιμάσει το δίκτυο του για να έχει τη δυνατότητα να προβλέψει τη συμπεριφορά του σε πιθανά προβλήματα. Για τον λόγο αυτό έχουν δημιουργηθεί συστήματα αλλά και σουίτες προγραμμάτων τα οποία παρακολουθούν, καταγράφουν και ελέγχουν το δίκτυο αλλά και τις υπηρεσίες που δέχεται/παρέχει ένας πελάτης/φορέας. Στο κεφάλαιο αυτό θα προσπαθήσουμε να αναφέρουμε κάποια από αυτά και να περιγράψουμε τη λειτουργία τους καθώς και τα χαρακτηριστικά τους γνωρίσματα.

5.2. QVoice Symphony

Η σουίτα προγραμμάτων TEMS Symphony περιλαμβάνει λύσεις για δοκιμές που αφορούν την συγκριτική μέτρηση επιδόσεων του δικτύου [40]. Περιλαμβάνει εργαλεία όπως είναι το QVoice Symphony καθώς και το QVoice Presentation. Για να μπορέσουν να προσελκύσουν και να κρατήσουν ικανοποιημένους πελάτες, οι διαχειριστές θα πρέπει να μάχονται με την υπ αριθμόν ένα αιτία δυσαρέσκειας των πελατών τους που είναι η κακή ποιότητα υπηρεσιών που προκαλείται είτε από ανεπαρκείς υπηρεσίες και εφαρμογές είτε από αδύναμη απόδοση του δικτύου. Η σουίτα TEMS δίνει μια ενδοσκοπική αντίληψη της ποιότητας υπηρεσιών που αντιλαμβάνεται ένας πελάτης τόσο για υπηρεσίες ομιλίας και αποστολής μηνυμάτων μέσω IP όσο και για υπηρεσίες βίντεο από τον διαχειριστή. Οι πληροφορίες αυτές είναι κρίσιμες για την βελτίωση των δικτύων, για την αλλαγή τους αλλά και για επενδύσεις που με τη σειρά τους θα βελτιώσουν την ικανοποίηση των πελατών.

Το QVoice Symphony είναι μια συγκριτική λύση για την μέτρηση απόδοσης της ποιότητας υπηρεσίας για διαχειριστές κυψελωτών δικτύων. Παρέχει δοκιμές αλλά και ελέγχους τόσο για υπηρεσίες ομιλίας όσο και για υπηρεσίες δεδομένων και βίντεο, που όπως είναι γνωστό απευθύνονται σε μια μεγάλη μερίδα κινητών τηλεφώνων αλλά και τερματικών. Συνήθως το QVoice Symphony συνοδεύεται και από το QVoice Presentation για να έχει ο χρήστης του μια εικονική απεικόνιση τόσο του δικτύου όσο και των μετρήσεων αλλά και στατιστικών. Για να μπορούν να διεξαχθούν οι δοκιμές αλλά και οι έλεγχοι στο δίκτυο χρειάζεται εξοπλισμός ειδικών χαρακτηριστικών όπως είναι ο εξοπλισμός που φαίνεται στις παρακάτω εικόνες (Εικόνα 33).



a) QVoice Symphony

b) QVoice Companion

Εικόνα 33. Απεικονίζονται τα QVoice Symphony και QVoice Presentation

Βασικά Χαρακτηριστικά

- Μεγάλο εύρος από πλήρως αυτοματοποιημένες, προγραμματισμένες εκ των προτέρων και κοινές δοκιμές για τον έλεγχο υπηρεσιών ομιλίας, δεδομένων, μηνυμάτων καθώς και βίντεο.
- Υποστήριξη πολλών τερματικών τηλεφώνων και καρτών αποθήκευσης δεδομένων
- Αυτόματος προγραμματισμός επίγειων μονάδων
- Υποστήριξη Υποδοχέων Σάρωσης
- Απεικόνιση δεδομένων πραγματικού χρόνου καθορισμένη από το χρήστη
- Προαιρετική επιλογή απομακρυσμένου GUI
- Αντικατάσταση εντοπισμού (Προχωρημένη Λειτουργία) – βασικός έλεγχος υπηρεσιών ομιλίας και δεδομένων με οποιοδήποτε κινητό τερματικό και κάρτα αποθήκευσης.
- Απλή διαμόρφωση και εγκατάσταση με 5 επίπεδα προστασίας με χρήση κωδικού.
- Αυτόματος εντοπισμός αποτυχίας ο οποίος περιλαμβάνει αυτόματο τερματισμό της ενέργειας καθώς και κυκλική εναλλαγή των προτύπων κλήσης.
- Αυτόματος εντοπισμός των συνδεδεμένων κινητών τερματικών και των καρτών αποθήκευσης δεδομένων
- Υποστήριξη σάρωσης για LTE

Οι μετρήσεις του προγράμματος QVoice Symphony είναι δυνατό να γίνουν σε όλες τις γενιές κυψελωτών δικτύων (2G,2.5G,3G & 4G). Κατηγοριοποιούνται σε δύο ομάδες.

Απεικόνιση Συνδρομητή

Το QVoice Symphony συλλέγει μετρήσεις ποιότητας υπηρεσίας συμπεριλαμβανομένου και ποιότητας πραγματικής ομιλίας και βίντεο, ρυθμούς απόρριψης κλήσεων και ρυθμούς χαμένων κλήσεων, SMS/MMS/Email ρυθμούς επιτυχίας και χρόνους παράδοσης, FTP/HTTP/UDP ρυθμούς διέλευσης και άλλες παραμέτρους για δεδομένα IP.

Απεικόνιση Μηχανικού

Απεικονίζει λεπτομερείς πληροφορίες για τον διάλογο όπως είναι η δύναμη του σήματος, η ποιότητα του σήματος, οι παρεμβολές, οι γειτονικές κυψέλες, μηνύματα επιπέδου 3 κα. Αυτές οι πληροφορίες μπορούν να δώσουν σε έναν μηχανικό την απάντηση για την χαμηλή ποιότητα υπηρεσιών σε ένα σημείο του δικτύου.

Μετρήσεις Ποιότητας Υπηρεσίας.

Οι παράμετροι για τον έλεγχο και την δοκιμή υπηρεσιών ομιλίας καθορίζονται από τον χρήστη και μπορούν να είναι MOC,MTC,MOC/MTC ή mobile – mobile. Η κατεύθυνση του σήματος ομιλίας μπορεί να είναι uplink, downlink ή half-duplex. Η διάρκεια κλήσης, το χρονικό διάστημα που μεσολαβεί για την επανάκληση, οι παύσεις καθώς και άλλα κριτήρια μπορούν να προγραμματιστούν εκ των προτέρων.

Σημαντικά δεδομένα για την μέτρηση της ποιότητας υπηρεσίας φωνής είναι:

- **Προσβασιμότητα Υπηρεσίας (Service Accessibility)** - Το ποσοστό των προσπαθειών κλήσεων που αποτυγχάνουν.
- **Χρόνος Εγκατάστασης Κλήσης (Call Setup Time)** – Είναι ο χρόνος που μεσολαβεί μέχρι να αρχίσει να χτυπάει το τηλέφωνο
- **Retainability** – Η συχνότητα που δεν διακόπτονται οι χρήστες από αποτυχημένες κλήσεις
- **Ποιότητα Φωνής (Voice Quality)** – Η τιμή που πετυχαίνει η ποιότητα φωνής σαν στατιστικό στοιχείο πάνω σε μια μεγάλη ομάδα χρηστών. (MOS)

Οι δοκιμές δεδομένων κινητών ελέγχουν την απόδοση του δικτύου με βασικό κριτήριο τη μεταφορά δεδομένων και μηνυμάτων. Υπάρχουν σουίτες και πρωτόκολλα δοκιμών διαθέσιμα για HTTP,FTP,UDP,WAP,SMS/MMS,PING και email.

Τα βασικά χαρακτηριστικά εδώ είναι:

- Προσβασιμότητα Υπηρεσίας
- Χρόνος Εγκατάστασης
- Διάρκεια Συνόδου
- Μέσος Ρυθμός Δεδομένων
- Ποσοστό Χαμένων Δεδομένων

Τεχνικά Χαρακτηριστικά για την Συλλογή Δεδομένων

- Βέλτιστος αλγόριθμος για δοκιμή ομιλίας, βίντεο και ήχου βασισμένο σε βιομηχανικά κριτήρια και κανόνες της ITU.
- Οι τιμές MOS Ομιλίας απεικονίζονται σε πραγματικό χρόνο κατά τη διάρκεια των δοκιμών.
- Απεικόνιση της ποιότητας υπηρεσίας που βιώνει ο συνδρομητής τόσο για το uplink όσο και για το downlink
- Προσομοίωση της συμπεριφοράς του συνδρομητή
- Μετρήσεις για διαφορετικού είδους διεπαφές: GSM,GPRS,EDGE,WCDMA,CDMA-1X
- Υποστήριξη τελευταίων τεχνολογιών HSDPA/HSUPA, WiMAX(4G),CDMA,EV-DO

Το QVoice Symphony μπορεί να υποστηρίξει μια μεγάλη γκάμα κινητών τερματικών για όλες τις δοκιμές. Οι συσκευές αυτές μπορούν να καλύψουν ζώνες πολλών συχνοτήτων για GSM, GPRS, EDGE, WCDMA, CDMA-1X και EV-DO, LTE. Επίσης υποστηρίζει ένα μεγάλο αριθμό από κάρτες δεδομένων και USB modems για την δοκιμή IP υπηρεσιών όπως είναι οι σαρωτές λήψης για κάλυψη RF και ανάλυση παρεμβολών σε multi-band και συχνοτήτες για τις διάφορες γεωγραφικές περιοχές.

QVoice Presentation

Αποτελεί το εργαλείο που αναλύει και επεξεργάζεται τα δεδομένα ενώ έχει τη δυνατότητα για την παραγωγή αναφορών πάνω στα μετρούμενα ζητήματα. Τα δεδομένα των μετρήσεων μπορούν να συλλεχθούν με ευέλικτους τρόπους, όπως τα δεδομένα τα οποία συλλέγονται από διαφορετικά μέρη και τα οποία ενώνονται και συγχρονίζονται για να μελετηθούν ομαδοποιημένα. Με την χρήση του QVoice Presentation υπάρχει αυτοματοποιημένη ταξινόμηση και ανάλυση των προβληματικών κλήσεων αλλά μπορούν να εφαρμοστούν και διεργασίες δεδομένων χωρίς την παρουσία εξειδικευμένου προσωπικού.

Παρέχει πληροφορίες ακριβείς και λεπτομερείς σε μεγάλο βαθμό που όμως είναι εύκολα κατανοητές λόγω του τρόπου με τον οποίο αυτές απεικονίζονται. Ο χρήστης μπορεί να επιλέξει τον τρόπο απεικόνισης όπως αναφέραμε ανάμεσα σε απεικόνιση συνδρομητή ή απεικόνιση μηχανικού.

5.3. NEMO Analyze

Πρόκειται για ένα παντοδύναμο και ευπροσάρμοστο εργαλείο τελευταίας τεχνολογίας για την συγκριτική μέτρηση της απόδοσης, την επίλυση προβλημάτων αλλά και την στατιστική αναφορά δεδομένων. Το σύστημα αποτελείται από ένα επίπεδο στο οποίο και βασίζεται η επικοινωνία client/server καθώς και η υποστήριξη μιας βάσης δεδομένων η οποία και έχει σχεδιαστεί ειδικά για υψηλές αποδόσεις και μετά-επεξεργασία δεδομένων καθώς και δοκιμαστικών ελέγχων [41].

Ο αναλυτής NEMO προσφέρει ένα ολοκληρωμένο σετ από ειδικούς τεχνολογικούς δείκτες απόδοσης για τις τελευταίες ασύρματες τεχνολογίες αλλά και παρουσίαση ήδη γνωστών δεδομένων με τέτοιο τρόπο ώστε να επιτυγχάνεται η καλύτερη απεικόνιση των πληροφοριών.

Παρέχει υποστήριξη για το μοντέλο CSV(Character Separated Value) το οποίο δίνει τη δυνατότητα εισαγωγής δεδομένων ASCII στην βάση δεδομένων. Αυτό έχει ως συνέπεια την ικανότητα συνεργασίας προγραμμάτων όπως είναι το Wireshark/Etheral καθώς και μετρητών δικτύου με το αναλυτή NEMO και στην συνέχεια την ολοκληρωμένη επεξεργασία των πακέτων και των δεδομένων που έχει καταγράψει το καθένα.

Οι εκδόσεις τον NEMO είναι τέσσερις και είναι οι εξής:

- Nemo Analyze Lite – Παρέχει λύσεις περιορισμένες στην στατιστική αναφορά των δεδομένων δοκιμών.
- Nemo Analyze Standard – Παρέχει λύσεις για τις περισσότερες κοινές εργασίας επεξεργασίας και στατιστικών αναφορών.
- Nemo Analyze Professional – Είναι μια έκδοση για προχωρημένους με συγκεκριμένο KPI και εξελιγμένες ιδιότητες αναφοράς και ανάλυσης.

- Nemo Enterprise

Τα βασικά χαρακτηριστικά του Nemo Analyze είναι τα εξής:

- Ολοκληρωμένη λειτουργία πάνω σε ένα και μόνο φορητό υπολογιστή
- Ακριβής και πλήρως μορφοποιούμενη απεικόνιση, με χρονικό συγχρονισμό και ενσωματωμένα δεδομένα εργασιών τα οποία σε συνδυασμό με τα πολλά διαφορετικά είδη απεικόνισης (διαφορετικά χρώματα, γραμμές κτλ) χαρτογραφούν και απεικονίζουν πλήρως ένα δίκτυο.
- Υποστήριξη όλων των κυρίαρχων τεχνολογιών δικτύου όπως είναι GSM, GPRS, EDGE, WCDMA, CDMA ONE, CDMA 2000, TDMA, DVB-H, UMA, HSPA+, TD-SDMA, WiMAX, LTE.
- Πλήρης αυτοματισμός καθ' όλη τη διάρκεια της επεξεργασίας από το upload των αποτελεσμάτων των δοκιμαστικών ελέγχων μέχρι την εξαγωγή αναφορών πάνω σε αυτά.

5.4. Network Analysis NITRO (NAN)

Είναι ένα γραφικό εργαλείο για την καταγραφή και την ανάλυση της δικτυακής κίνησης από και προς τον προσωπικό υπολογιστή του χρήστη. Το NAN απεικονίζει τα στατιστικά των πακέτων σε δενδρική απεικόνιση, απεικόνιση πίνακα και σε απεικόνιση διαγραμμάτων. Στα πακέτα μπορούν να εφαρμοστούν φίλτρα δενδρικής απεικόνισης με όρους network interface, local address, remote address.

Η απεικόνιση πίνακα δείχνει πακέτα απ' όλους τους κόμβους της τοπολογίας. Η απεικόνιση διαγραμμάτων δείχνει την δράση των πρωτοκόλλων για ένα κόμβο της τοπολογίας σε σχέση με τον χρόνο. Το NAN μπορεί να εγκατασταθεί τόσο σε μηχανήματα που έχουν λειτουργικό πρόγραμμα Windows όσο και σε μηχανήματα με λειτουργικό πρόγραμμα βασισμένο σε UNIX όπως είναι το Linux.

Οι λειτουργίες και τα χαρακτηριστικά που περιέχει και υποστηρίζει το NAN είναι:

- ARP/UDP/TCP/ICMP πακέτα
- Υποστήριξη όλων των πρωτοκόλλων επιπέδων εφαρμογής
- Δενδρική Ιεράρχηση των φίλτρων πακέτων
- Ανάλυση τόσο με χρήση πινάκων όσο και με χρήση διαγραμμάτων
- Δυνατότητα Παύσης της καταγραφής ή της ανάλυσης
- Δυνατότητα Αντιγραφής και Επικόλλησης των δεδομένων από την απεικόνιση του πίνακα σε άλλα εργαλεία.

5.5. TEMS Investigation

Είναι ένα εργαλείο για την επίλυση προβλημάτων, την αναγνώριση και την βέλτιστη λειτουργία ασύρματων δικτύων. Αποτελεί την βασική λύση για την συντήρηση αλλά και την εξέλιξη ασύρματων δικτύων. Έχει σχεδιαστεί για χρήση εντός οχήματος, εντός κτηρίων και για πεζή χρήση. Με την χρήση του TEMS Investigation οι διαχειριστές μπορούν να επιτύχουν βελτιωμένη ποιότητα ομιλίας, αυξημένη προσβασιμότητα, περισσότερες επιτυχημένες προσπάθειες κλήσεων και καλύτερη απόδοση υπηρεσιών [40].

Υποστηρίζει τύπους δικτύων όπως GSM,GPRD,EDGE,WCDMA και HSDPA. Η απεικόνιση των μετρήσεων γίνεται σε έναν χάρτη πραγματικού χρόνου. Παράλληλα ένας γεννήτορας αναφορών κάνει δυνατή την εξακρίβωση της τοποθεσίας και της κάλυψης της κυψέλης. Η χρηστικότητα του χάρτη στο TEMS Investigation παρέχει βασικά στατιστικά για την κατανομή του σήματος στην μετρούμενη περιοχή. Το εργαλείο ανάλυσης μονοπατιών επιτρέπει στον χρήστη την εργασία με πολλαπλά αρχεία καταγραφών όταν οι εργασίες που διενεργούνται είναι πλήρως βασισμένες στην στατιστική συμπεριφορά.

Αποτελεί ένα πλήρες εργαλείο για τους διαχειριστές αφού δίνει την δυνατότητα για συλλεχθούν, να αναλυθούν και να επανα-επεξεργαστούν δεδομένα δικτύου σε καθημερινή βάση. Επιτρέπει την συνεχή καταγραφή του δικτύου καθώς και την επίλυση προβλημάτων εκτός από τον εντοπισμό τους.

Η δυνατότητα του να μπορεί να λειτουργήσει με διαφορετικό τρόπο το κάνει ιδανικό για δίκτυα που χρησιμοποιούν τεχνολογίες WCDMA/HSPA, CDMA και GPS/GPRS/EDGE. Η πολύ-λειτουργικότητα του δίνει στον χειριστή την δυνατότητα να:

- Αναγνωρίζει και να βελτιώσει την εσωτερική στο σύστημα παράδοση αλλά και την επανεπιλογή κυψέλης.
- Λειτουργήσει με Συμπιεσμένη Συμπεριφορά
- Αναγνωρίζει και να συγκρίνει τόσο κάλυψη όσο και απόδοση ανάμεσα σε δίκτυα
- Αναγνωρίζει WCDMA/HSPA και GSM προσβασιμότητα.

Εν συντομία και ενδεικτικά αναφέρουμε κάποια βασικά χαρακτηριστικά του:

- Υποστήριξη GSM,GPRD,EDGE,WCDMA,HSUPA και HSDPA
- Υποστήριξη τερματικών Sony Ericsson, Nokia, Motorola, Qualcomm και Qualcomm chipset based terminals
- Υποστήριξη σαρωτών και σάρωσης
- Μετρήσεις Ποιότητας Ήχου
- Δυνατότητες για επεξεργασία και ενισχυμένη λειτουργία αναφοράς
- Δυνατότητα για λήψη μετρήσεων μέσα σε κτήρια
- Χάρτης πραγματικού χρόνου
- Δοκιμή ποιότητας Video Streaming
- Δυνατότητες ανίχνευσης πρωτοκόλλων IP
- Σηματοδοσία



Εικόνα 34. Στιγμιότυπο από καταγραφή του οργάνου TEMS Investigation

5.6. XCAL

Παρέχει και αυτό δυνατότητες για δοκιμαστικά σενάρια καθώς και για επεξεργασία των καταγεγραμμένων δεδομένων. Έχει ειδικά υποπρογράμματα για την επίλυση σφαλμάτων, την καταγραφή και την συντήρηση του δικτύου. Το XCAL δίνει μια καλύτερη αντίληψη της QoS/QoE και εγγυάται μια συνεχή υποστήριξη για δίκτυα που βασίζονται σε τεχνολογίες είναι GSM, GPRS, EDGE, WCDMA, CDMA ONE, CDMA 2000, TDMA, DVB-H, UMA, HSPA+, TD-SDMA, WiMAX, LTE..Είναι ενημερωμένο και υποστηρίζει πλήρως όλα τα διεθνή κριτήρια [14].

Αναλόγως με το επίπεδο δικτύου το XCAL δίνει διαφορετικές πληροφορίες με αποτέλεσμα να γίνεται πιο ολοκληρωμένη ανάλυση και παρακολούθηση του δικτύου.

- Πληροφορίες για το Level 1.
 - i) Downlink/Uplink παράμετροι σχετιζόμενες με την κάλυψη (RSSI, UE/CPE ενέργεια μετάδοσης, CINR για το downlink κτλ) για κάθε τεχνολογία ασύρματης πρόσβασης (RAT-Radio Access Technology)
 - ii) Πληροφορίες κυψελών που εξυπηρετούν το κάθε RAT
- Πληροφορίες για το Level 2
 - i) BLER (FER), ARQ (HARQ) για το κάθε RAT
 - ii) Διαχείριση πόρων και χρονοπρογραμματισμός πληροφοριών για κάθε RAT

- iii) Σηματοδοσία φυσικού στρώματος για κάθε RAT
- Πληροφορίες για το Level 3
 - i) Σηματοδοσία επιπέδου 3 για το κάθε RAT
- NAS Layer Πληροφορίες
 - i) Πληροφορίες ανώτερων επιπέδων πρωτοκόλλων
- Πληροφορίες Εφαρμογών TCP/IP
 - i) Καταγραφή και αποκρυπτογράφηση πακέτων TCP/IP
 - ii) KPI επιπέδου εφαρμογής (Διέλευση, QoS δικτύου, Ποιότητα Ομιλίας, Ποιότητα βίντεο κτλ)

5.7. LTE Base Station Emulator - LTE BSE (AWT 700A)

Το AWT 700A (Advanced Wireless Tester) [24] μπορεί να αποδώσει ένα προσομοιωμένο αλλά παρ' όλα αυτά πραγματικού χρόνου περιβάλλον σταθμού βάσης για δοκιμές LTE τερματικών και διαφόρων άλλου δοκιμών όπως είναι και η eNB emulation. Η LTE BSE σουίτα εργαλείων μπορεί να χρησιμοποιηθεί καθ' όλη τη διάρκεια σχεδιασμού ενός LTE από τα αρχικά στάδια του R&D δια μέσω της ολοκλήρωσης, της εξακρίβωσης, της διαμόρφωσης και της διαλειτουργικότητας μέχρι την εκκίνηση δοκιμών του διαχειριστή. Είναι αρκετά εύκολο στην χρήση και περιλαμβάνει αδιάκοπη κάλυψη δοκιμών, έτσι είναι ιδανικό εργαλείο είτε ο χρήστης θέλει να κάνει ανάπτυξη πρωτοκόλλων είτε θέλει να κάνει δοκιμές εφαρμογών [14]. Τα βασικά χαρακτηριστικά του εργαλείου αυτού φαίνονται παρακάτω:

5.7.1. Χαρακτηριστικά

- Προσομοίωση πραγματικού χρόνου
Ραδιοφωνικής Πρόσβασης δικτύου LTE.
- Ενσωματωμένος παραγωγός σήματος και αναλυτής σήματος.
- Δοκιμή L1(PHY) σε L1(PHY).
- Δοκιμή Πρωτοκόλλου L2/L3 σε L2/L3.
- Δοκιμή εφαρμογής E2E με ενσωματωμένο είτε EPC είτε EPCE.
- Ασφάλεια της ολοκλήρωσης, επαλήθευσης, διαλειτουργίας, συμμόρφωσης του συστήματος LTE μέσα από ποικίλες λειτουργικές δοκιμές ανάπτυξης.
- Βάση δεδομένων για σύστημα Δοκιμής Πρωτοκόλλου Συμμόρφωσης και σύστημα Δοκιμής Ραδιοφωνικής Συμμόρφωσης.



5.7.2. Σύνθεση Δοκιμής

Βασική σύνθεση

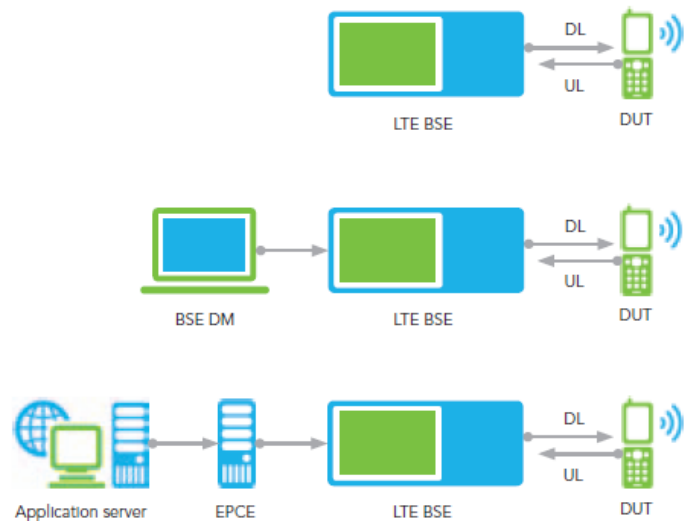
- Αυτόνομη δοκιμή Πρωτοκόλλου/ RF

Αυτονομία

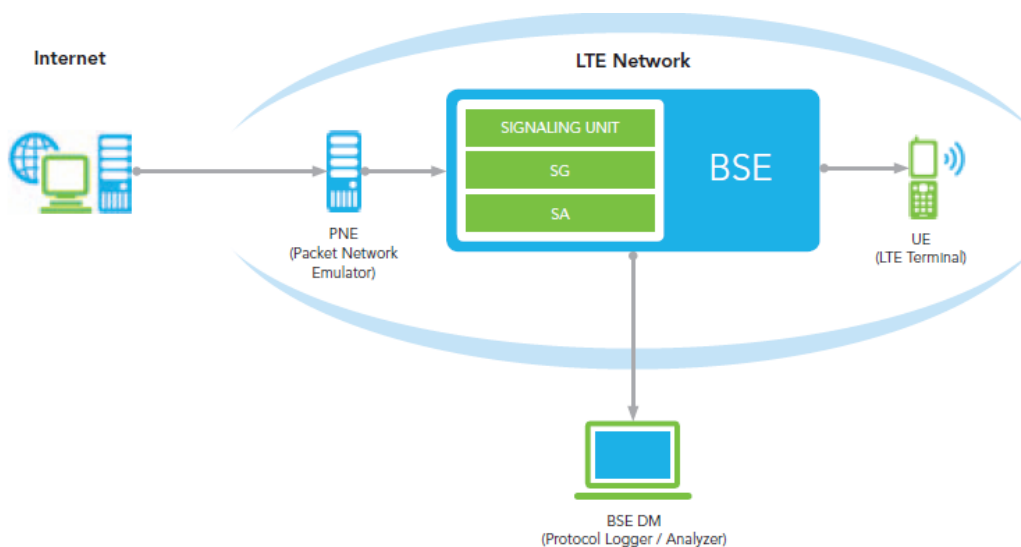
- Αποθηκεύει και αναλύει το μήνυμα σηματοδότησης και τα δεδομένα που φορτώνονται από το λογισμικό (Διαγνωστικού ελεγκτή εκπομπών) BSE DM

Δοκιμή E2E

- Παρουσιάζει τη δοκιμή E2E από το UE στο διακομιστή της εφαρμογής



Εικόνα 35. Σχεδιάγραμμα διάταξης για κάθε μια από τις δοκιμές.



Εικόνα 36. Αρχιτεκτονική BSE

5.8. LTE Air Interface Protocol Analyzer – LTE Air Sniffer

Ο LTE Air Sniffer είναι ένας αναλυτής πρωτοκόλλου over-the-air UMTS-LTE [24]. Αποτελεί απαραίτητο εργαλείο για την ανάπτυξη και επαλήθευση air interface πρωτοκόλλων και της εκτίμησης της επίδοσής της [14]. Λόγω της ευέλικτης αρχιτεκτονικής του μπορεί να προσφέρει εύρωστη παρακολούθηση σήματος LTE RF, παρακολούθηση μηνυμάτων πρωτοκόλλου L1/L2/L3/IP, εφαρμογές μετέπειτα ανάλυσης του ληφθέντος σήματος αλλά και των πακέτων που ανήκουν στα πρωτόκολλα που ήδη αναφέραμε. Έχει τη δυνατότητα να λειτουργήσει τόσο με τρόπο πραγματικού χρόνου όσο και με τρόπο μετέπειτα ανάλυσης. Κάποια βασικά χαρακτηριστικά του εν λόγω εργαλείου φαίνονται ονομαστικά στην συνέχεια.

5.8.1. Χαρακτηριστικά και Πλεονεκτήματα

- Ενσωματωμένο χαρακτηριστικό μη συνδεδεμένης ανάλυσης
- Τα μηνύματα πρωτοκόλλου καταγραφής ιστορικού σε sub-frame χρόνο ενώνονται στο μέγεθος για το σκληρό δίσκο για χαρακτηριστικά ανάλυσης πρωτοκόλλου και φτάνουν μέχρι 4GB, 16GB ή 6T για διανυσματική ανάλυση του σήματος
- Επίπεδα L1 / L2 / L3 / NAS για πλήρη αποκωδικοποίηση μηνυμάτων
- Σύλληψη και ανάλυση σημάτων LTE DL/ UL RF και μηνυμάτων IP L1/ L2/ L3 μέσω κεραίας ή καλωδίου RF
- Σύστημα ανεξαρτήτου πώλησης UE και eNB
- Αποκωδικοποίηση πραγματικού χρόνου και έκθεση των LTE L1/ L2/ L3 και πληροφοριών RF, όπως RSSI, CINR, EVM κλπ
- Διασταυρωμένη ανάλυση σε πραγματικό χρόνο πολλαπλών επιπέδων LTE RF και L1/ L2/ L3/ IP και ποικίλων χαρακτηριστικών σύλληψης όπως συνθήκες RF, L1/ L2/ L3 συμβάντα μηνυμάτων κλπ
- Επεξεργασία με διασταυρωμένη ανάλυση δεδομένων, που προέρχονται από την ανάλυση πρωτοκόλλου αλλά και από τα χαρακτηριστικά ανάλυσης σήματος. Αυτό μας επιτρέπει τη συγχρονισμένη και ολοκληρωμένη ανάλυση σε περίπλοκα θέματα και προβλήματα πάνω σε πολλαπλά επίπεδα των LTE RF και L1/ L2/ L3/ IP
- Φιλική προς το χρήστη εγκατάσταση της οθόνης και της εικόνας μηνυμάτων
- Εύκολο για χρήση και ρύθμιση.



5.9. RF Parametric Tester – Versatile Subscriber Station Tester (VST)

Ιδανικό εργαλείο για τη βαθμονόμηση του RF και την επαλήθευση κινητών συσκευών και μονάδων μέτρησης, παρέχει ένα γρήγορο αναλυτή σήματος (SA) και παραγωγό σήματος (SG) και υποστηρίζει ένα ευρύ φάσμα συχνοτήτων (90MHz -6GHz) [14].

5.9.1. Η Χρήση του VST

- Εκτελεί παραμετρικές δοκιμές για κινητά Tx και εφαρμογές RF
- Πραγματοποιεί γρήγορη και εύκολη βαθμονόμηση. Επίσης αυτοματοποιεί τις δοκιμές, συμπεριλαμβανομένου και βαθμονόμηση του συνόλου του κυκλώματος
- Εκτελεί λειτουργίες SA και SG
- Βελτιώνει την κατασκευαστική δυνατότητα μέσω αποτελεσματικών δοκιμών και ταχύτερης βαθμονόμησης
- Επεξεργάζεται δοκιμές εξορθολογισμού και μειώνει το OPEX, μέσα από την ταυτόχρονη μέτρηση του DL και του UL

5.9.2. Βασικά Χαρακτηριστικά

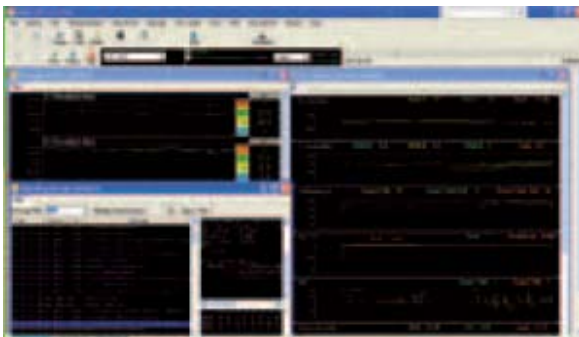
- Πολλαπλές τεχνολογίες (Wi-Fi, WiMAX, LTE)
- Ένα μεγάλο φάσμα εύρους ζώνης (μέχρι 6GHz)
- Λειτουργία μετάδοσης και λήψης RF/PHY



5.10. LTE Test Tool

5.10.1. Βασικά Χαρακτηριστικά

- Εύκολο στη γνώση και στη ρύθμιση
- Υποστηρίζει ποικίλες συσκευές χρόνου LTE- αυθεντικά UE's και σκάνερ
- Υποστηρίζει ποικίλα σενάρια δοκιμών LTE - FTP, Ping, UDP, VoIP, VOD, κλπ
- Υποστηρίζει μετρητές LTE - PHY, HARQ, Scheduling, RRM, MAC, RRC, NAS
- Υποστηρίζει εκτελέσεις KPI δικτύου LTE



Αναλόγως με το επίπεδο δικτύου το XCAL δίνει διαφορετικές πληροφορίες .

- Πληροφορίες για το Layer 1

Πληροφορίες κυψέλης, καναλιού, CQI, κυψέλη υπηρεσιών RSPQ, UE Tx δύναμη, πληροφορίες RACH, τάξη ανίχνευσης PDSCH BLER, PBCH BLER, PDCCH, παροχή UL/DL, θέση DownLink - UpLink Ack/Nack, throughput PDSCH/ PUSCH PHY

- Πληροφορίες για τα Layer 2,3

Πληροφορίες PDCP (pdu, throughput, διαμόρφωση, ασφάλεια), πληροφορίες RLC (κατάσταση, throughput) , πληροφορίες MAC(pdu, throughput, πληροφορίες HARQ Ack/Nack, BLER, TA) , μηνύματα RRC, πληροφορίες γειτονικής κυψέλης.

5.11. Handheld XCAL

Το XCAL – Mobile είναι ένα εργαλείο που επιτρέπει τη δοκιμή των QoS και QoE στις τεχνολογίες GSM/UMTS. Υποστηρίζει εκτενής εφαρμογή και παρέχει μέτρηση δικτύου σε πραγματικό χρόνο και δυνατότητα οπτικοποίησης σε απλά κινητά τηλέφωνα (Nokia N95, N96, N97 κλπ). Όλα τα χαρακτηριστικά ελέγχονται χρησιμοποιώντας τα τυπικά χειροκίνητα κουμπιά, έτσι ώστε να μπορεί να τα χρησιμοποιήσει ο καθένας.



Εικόνα 37. Διαφορετικά στιγμιότυπα από την εκτέλεση του Handheld XCAL σε τερματικό.

5.11.1. Βασικά Χαρακτηριστικά

Γενικά

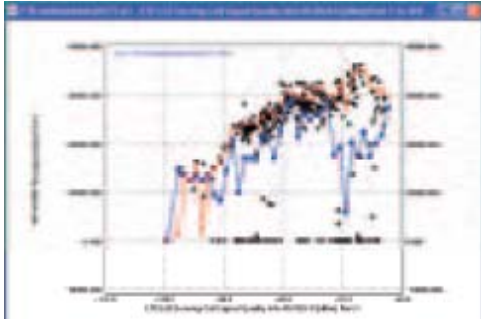
- Κατανοητή αυτόματη κλήση
- Δοκιμή διαμορφώσιμη από το χρήστη
- Κλήση scripting
- Φόρτωση δεδομένων FTP
- Εσωτερική τροποποίηση
- Χαρτογράφηση πραγματικού χρόνου
- Ενεργή και παθητική καταγραφή

5.11.2. Βασικές Πληροφορίες

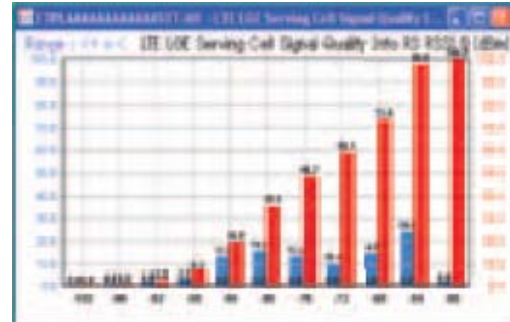
- Ισχυρές πληροφορίες(όσον αφορά το σύστημα, το δίκτυο, τον τομέα κλπ)
- Κατάσταση μνήμης
- CI, LAC, MCC, MNC
- Πληροφορίες θέσης και GPS
- Ρύθμιση αυτόματης κλήσης
- Διαχείριση αρχείων
- Φόρτωση FTP στο διακομιστή
- Μέτρηση σε εσωτερικό χώρο
- Κατάσταση αρχείου καταγραφής ιστορικού

5.12. XCAP

Αποτελεί μια ισχυρή ρυθμιζόμενη πλατφόρμα ανάλυσης που επιτρέπει την απόκτηση, ανάλυση και αναφορά των δεδομένων που συγκεντρώνονται σε XCAL [19]. Υποστηρίζει όλα τα ασύρματα πρότυπα και μεγάλα πρότυπα δεδομένων τρίτης ομάδας.



Εικόνα 38. Στιγμιότυπο γράφου CDF



Στιγμιότυπο γράφου PDF

5.12.1. Βασικά Χαρακτηριστικά

- Λειτουργία εισαγωγής/ εξαγωγής δεδομένων – Δυνατότητα εισαγωγής δεδομένων από τύπο αρχείου CSV για την μετεπεξεργασία αποκτηθέντων δεδομένων, εκτός από τη σειρά των XCAL. Επίσης κάθε δεδομένο μπορεί να εξαχθεί σε Excel, CSV, MIF, Google Map ή σε μορφή κειμένου.
- Λειτουργία επανάληψης – Επαναλαμβάνει τα δεδομένα σε γράφους, χάρτες, πίνακες και παράθυρα μηνυμάτων με τον ίδιο τρόπο όπως και η ίδια καταγραφή ιστορικού.
- Λειτουργία συγχρονισμού πολλαπλών παραθύρων – Στην περίπτωση που υπάρχουν πολλά παράθυρα ανοιχτά και μια ομάδα δεδομένων εκτελείται σε ένα από αυτά, τα υπόλοιπα εκθέτουν την ίδια χρονική ζώνη, που διευκολύνει την πολυδιάστατη μετεπεξεργασία δεδομένων
- Λειτουργία γειτονικής λίστας μετεπεξεργασίας(CDMA2000, WCDMA, WiMAX, LTE) – Αυτή η λειτουργία δίνει τη δυνατότητα της μετεπεξεργασίας ελλιπών και ολοκληρωμένων για Γειτονική λίστα καταγραμμένων συστημάτων, συγκρίνοντας τα δεδομένα μεταξύ scanner και ακουστικού
- Φιλτράρισμα δεδομένων – Αποσπά δεδομένα που ικανοποιούν τις προδιαγεγραμμένες συνθήκες
- Λειτουργία μετεπεξεργασίας της μόλυνσης του πλοηγού (CDMA2000, WCDMA, WiMAX, LTE) – Απομακρύνει τις περιοχές με ιό.



Εικόνα 39.Στιγμιότυπο έκθεσης δεδομένων καταγραφής ιστορικού.



Εικόνα 40.Στιγμιότυπο δεδομένων καταγραφής ιστορικού που εκθέτονται σε χάρτη.

5.13. XCAL - MO

Αποτελεί μια πλατφόρμα λειτουργικού που βελτιώνει σημαντικά τη λειτουργική αποτελεσματικότητα. Οι χρήστες μπορούν να συνδέσουν έως και 20 handset, τα οποία στη συνέχεια καλούν αυτόματα το δίκτυο. Τα επακόλουθα δεδομένα μπορούν να καταγραφούν, να εκτεθούν και να χρησιμοποιηθούν για τη βαθμονόμηση και βελτιστοποίηση της συνολικής εκτέλεσης του δικτύου [14].

5.13.1. Βασικά Χαρακτηριστικά

- Αυτόματη κλήση (αυτοματοποιημένα script δοκιμών και δρομολόγησης)
- Χαρτογράφηση πραγματικού χρόνου για ισχυρή γεω-χωρική οπτικοποίηση
- Ταυτόχρονες καταγραφές όλων των πληροφοριών RAN και εκτιμήσεις παραμέτρων από συνδεδεμένες συσκευές

Τα σημεία μέτρησης περιλαμβάνουν :

- Δεδομένα RAN (CINR, Ec/Io/Tx - Rx power κλπ)
- Πληροφορίες επιπέδου MAC
- Επίπεδο 3 μηνυμάτων (και μηνυμάτων διαχείρισης MAC) και μηνυμάτων πρωτοκόλλων υψηλού επιπέδου που περιλαμβάνουν μηνύματα TCP / IP

5.13.2. Βελτιστοποίηση Δικτύου LTE απ' άκρο σ' άκρο.

Βασικά χαρακτηριστικά

- Μεγάλη γεω-χωρική οπτικοποίηση, αναφορά και ανάλυση.
- Συνεχής καταγραφή δικτύου για κάθε κόμβο, με λειτουργίες προειδοποίησης για διορθωτικές κινήσεις ενάντια στον κίνδυνο.
- Εκλεπτυσμένη στατιστική αναφορά.
- Πλούσια χαρακτηριστικά ανάλυσης π.χ. ανίχνευση κλήσης από άκρη σε άκρη και ανάλυση, απώλεια πακέτων και root cause, ανάλυση εφαρμογής (FTP, HTTP κλπ) ανάλυσης πρωτοκόλλων root cause- Διαγράμματα Ροής Ίχνους Χρήστη σε πραγματικό χρόνο.

5.14. Measurement Lab

Το Measurement Lab (M-Lab) είναι ένα εργαλείο, το οποίο κυρίως αναφέρεται για ερευνητές πρόθυμους να εξελίξουν και να αναβαθμίσουν τα δικτυακά τους εργαλεία [19]. Πρόκειται

για μια ανοιχτή και κατανεμημένη πλατφόρμα server η οποία θέτει ως βασικό στόχο λειτουργίας την απόκτηση πληροφοριών για τις ευρυζωνικές συνδέσεις με σκοπό την βελτίωση της ποιότητας σύνδεσης του κοινού. Κατά τη διάρκεια δυσλειτουργίας μιας εφαρμογής, είναι αβέβαιη η πηγή προέλευσης του προβλήματος. Υπάρχει η πιθανότητα να είναι υπαιτιότητα της ευρυζωνικής σύνδεσης, της εφαρμογής είτε κάποιου άλλου παράγοντα. Ίσως να είναι και ένα από τα πιο βασικά προβλήματα που αντιμετωπίζουν οι διαχειριστές ενός δικτύου σε καθημερινή βάση. Είναι βασικό για την ανάπτυξη και την εξέλιξη του διαδικτύου, τόσο να δοθεί προτεραιότητα στην διαφάνεια του Internet όσο και να υπάρξει σωστή πληροφόρηση των πελατών για τα δικαιώματα και τις υποχρεώσεις τους πάνω στις ευρυζωνικές τους συνδέσεις.

Για το λόγο αυτό πολλοί application developers ασχολούνται με το αντικείμενο της ανάπτυξης εφαρμογών μέτρησης της απόδοσης αλλά και παρακολούθησης των ευρυζωνικών συνδέσεων. Οι εφαρμογές αυτές αναπτύσσονται για χρήση από απλούς πελάτες και είναι ζωτικής σημασίας να είναι απλές και εύχρηστες. Τα δεδομένα που συλλέγονται μέσω του M-Lab κοινοποιούνται και είναι διαθέσιμα προς χρήση απ' όλους τους χρήστες, γεγονός που βοηθά κατά πολύ την ανάπτυξη των εφαρμογών που προαναφέραμε αλλά σίγουρα και το ερευνητικό έργο που βασίζεται στην συνολική κίνηση του δικτύου.

Για τους λόγους που προαναφέραμε το M-Lab χρησιμοποιεί κάποια εργαλεία που αναφέρουμε παρακάτω για να καταγράψει την κίνηση και γενικότερα τη συμπεριφορά των δικτύων στον κόσμο:

- **Network Diagnostic Tool:** Παρέχει περίπλοκη δοκιμή για την ταχύτητα της σύνδεσης και οι αναφορές τους δεν περιορίζονται στην καταγραφή της ταχύτητας upload και download αλλά γίνεται και η προσπάθεια για τον εντοπισμό πιθανών λαθών ή σφαλμάτων που με τη σειρά τους μπορεί να περιορίζουν κάποια από τις δύο ταχύτητες. Οι αιτίες (σφάλματα) χωρίζονται σε αιτίες δικτύου και αιτίες εφαρμογής ή υπολογιστή.

Ειδικότερα, [12] το εργαλείο αυτό μετράει την TCP διέλευση ανάμεσα σε ένα client ο οποίος τρέχει σε έναν host του χρήστη και σε έναν M-Lab server [26]. Δεδομένα για την δοκιμή αποστέλλονται και στις δύο κατευθύνσεις. Ενώ η δοκιμή συνεχίζει να εκτελείται, ο server συλλέγει επιπλέον πακέτα και δεδομένα όπως tcpdump και web100 αρχεία καταγραφών. Τα metadata περιέχουν πληροφορίες για τον client, περιλαμβάνοντας το είδος του λειτουργικού συστήματος και την έκδοσή του. Η δοκιμή παρέχει λεπτομερείς πληροφορίες για επίπεδο πακέτου μαζί με στατιστικά σε επίπεδο πυρήνα για το πώς απέδωσε η TCP σύνδεση στο συγκεκριμένο μονοπάτι.

Συλλογή Δεδομένων:

Ο χρήστης με την εκτέλεση του συγκεκριμένου εργαλείου θα καταγράψει την IP διεύθυνση του χρήστη, την ταχύτητα upload και download επικεφαλίδες TCP καθώς και παραμέτρους δικτύου.

- **Glasnost Test:** Το εργαλείο αυτό προσπαθεί να ανιχνεύσει εάν ο Internet Service Provider του χρήστη χρησιμοποιεί κάποια πολιτική για την ρύθμιση της κίνησης. Επιπλέον ο χρήστης μπορεί να εξακριβώσει εάν ο πάροχος της σύνδεσής του επιταχύνει, επιβραδύνει ή μπλοκάρει πακέτα email, http κτλ.

Συλλογή Δεδομένων:

Ο server μέτρησης καταγράφει την IP διεύθυνση του χρήστη, και όλα τα πακέτα δεδομένων που δέχτηκε ο server από τον υπολογιστή του χρήστη καθώς και αυτά που έστειλε ο server στον υπολογιστή του χρήστη. Επιπλέον έχει τη δυνατότητα να καταγράφει σφάλματα στην επικοινωνία καθώς και τον ρυθμό διέλευσης της κίνησης για τις συγκεκριμένες επικοινωνιακές ροές. Τα δεδομένα αυτά στέλνονται στον server.

- **Network path & application diagnostics:** Προσπαθεί να εντοπίσει τις αιτίες που προκαλούν μείωση της ταχύτητας στο μονοπάτι που ακολουθούν τα πακέτα για το τελευταίο μέρος της σύνδεσης.

Πιο αναλυτικά χρησιμοποιεί το TCP για να μπορέσει να μετρήσει τον απ άκρο σ' άκρο ρυθμό διέλευσης αλλά και πληροφορίες για τις ουρές των μεταγωγών και των δρομολογητών σε αυτή τη διαδρομή. Χρησιμοποιείται το Web100 για να περιορίσει την έκταση της συμφόρησης που καταγράφεται αλλά προκαλείται από την ίδια την μέτρηση.

Συλλογή Δεδομένων:

Με την μεταφορά δεδομένων ανάμεσα στον server και στον υπολογιστή του χρήστη το NPAD συλλέγει λεπτομερή στατιστικά στοιχεία για τους μηχανισμούς που διαμορφώνουν αλλά και χειρίζονται την κίνηση. Αποτέλεσμα όλων αυτών είναι ο χρήστης να λαμβάνει μια αναφορά πλούσια σε καταγραφές από IP διευθύνσεις, upload/download speed καθώς και από επικεφαλίδες πακέτων TCP.

- **Pathload2 Bandwidth Test:** Το εργαλείο αυτό μετράει το διαθέσιμο εύρος ζώνης της σύνδεσης του χρήστη. Το διαθέσιμο εύρος ζώνης ορίζεται ως η μέγιστη ταχύτητα μετάδοσης πληροφορίας από το χρήστη προς το δίκτυο χωρίς το δίκτυο να υποστεί συμφόρηση. Για να μπορέσει να επιτύχει μια τέτοια μέτρηση χρησιμοποιεί ακολουθίες από ειδικά σχεδιασμένα UDP πακέτα. Η μέτρηση γίνεται και προς τις δύο κατευθύνσεις και τα αποτελέσματα αποθηκεύονται μαζί με στατιστικά για την καθυστέρηση των πακέτων και τις απώλειες σε ένα αρχείο.

Συλλογή Δεδομένων:

Για κάθε μέτρηση, το Pathload2 συλλέγει την διεύθυνση IP του χρήστη, την διάρκεια της μέτρησης, το μετρούμενο διαθέσιμο εύρος ζώνης καθώς και την downstream και την upstream εξάγοντας τα απαραίτητα στατιστικά.

- **ShaperProbe:** Έχει τη δυνατότητα να εντοπίσει εάν ο ISP χρησιμοποιεί κάποιο εργαλείο διαμόρφωσης της κίνησης. Η διαμόρφωση της κίνησης σημαίνει πως ο ISP αυτόματα μειώνει τον ρυθμό πρόσβασης του χρήστη όταν αυτός ξεπεράσει κάποιο όριο δεδομένων στο download ή στο upload. Το συγκεκριμένο πρόγραμμα εντοπίζει αν χρησιμοποιείται κάτι τέτοιο στην σύνδεση του χρήστη τόσο στο upload μέρος όσο και στο download μέρος της σύνδεσης. Δίνει αναφορά για τον ρυθμό διαμόρφωσης καθώς και για το μέγιστο πλήθος bytes που μπορεί να λάβει ή να στείλει ο χρήστης μέχρι να εκκινήσει ο μηχανισμός διαμόρφωσης.

Συλλογή Δεδομένων:

Για κάθε μέτρηση το ShaperProbe καταγράφει την IP διεύθυνση του χρήστη καθώς και την καθυστέρηση αλλά και την απόρριψη πακέτων που συμβαίνουν από τον χρήστη στον server αλλά και ανάποδα.

- **WindRider:** Το εργαλείο αυτό προσπαθεί να ανιχνεύσει εάν ο πάροχος κινητής ευρυζωνικότητας έχει διαφοροποιήσει την συμπεριφορά του και την παροχή υπηρεσιών σε ορισμένες εφαρμογές είτε άλλες υπηρεσίες. Για παράδειγμα ένας πάροχος θα μπορούσε να θέσει προτεραιότητα για την εξυπηρέτηση ορισμένων ιστοσελίδων αλλά και συγκεκριμένων εφαρμογών.

Συλλογή Δεδομένων:

Ο server που εκτελεί τη μέτρηση καταγράφει την IP διεύθυνση της κινητής συσκευής και μετράει την καθυστέρηση της κίνησης που στέλνεται από και προς την κινητή συσκευή. Επιπλέον καταγράφει την καθυστέρηση που βιώνεται σε διαφορετικές ιστοσελίδες.

Η κινητή εφαρμογή εκκινεί μια σειρά από μεταφορές τόσο downstream όσο και upstream με τον Measurement Lab Server να καταγράφει τα στατιστικά που αφορούν την παρατηρούμενη απόδοση. Οι διαφορετικές μεταφορές είναι καταναμημένες σε διαφορετικές ports (θύρες) έτσι ώστε να μπορούμε να καταλάβουμε εάν ο πάροχος διαφοροποιεί την κίνηση βασισμένος στο είδος της εφαρμογής.

- **Bismark test:** Το Bismark είναι ένα πρόγραμμα εκτίμησης απόδοσης της ευρυζωνικότητας στα σπίτια. Αναπτύχθηκε από την Georgia Tech για την ανάπτυξη μιας Open WRT- βασισμένης πλατφόρμας για την εκτέλεση μετρήσεων της απόδοσης των ISP, όσο και της κίνησης στο εσωτερικό δίκτυο του σπιτιού. Ο χρήστης μπορεί να χρησιμοποιήσει το BisMark για να μετρήσει την επίδοση του ISP του, να απεικονίσει και να παρακολουθήσει πρότυπα κίνησης από τους χρήστες και τις συσκευές στο εσωτερικό δίκτυο τους σπιτιού και να διαχειριστεί τις παραμέτρους χρήσης.

Συλλογή Δεδομένων:

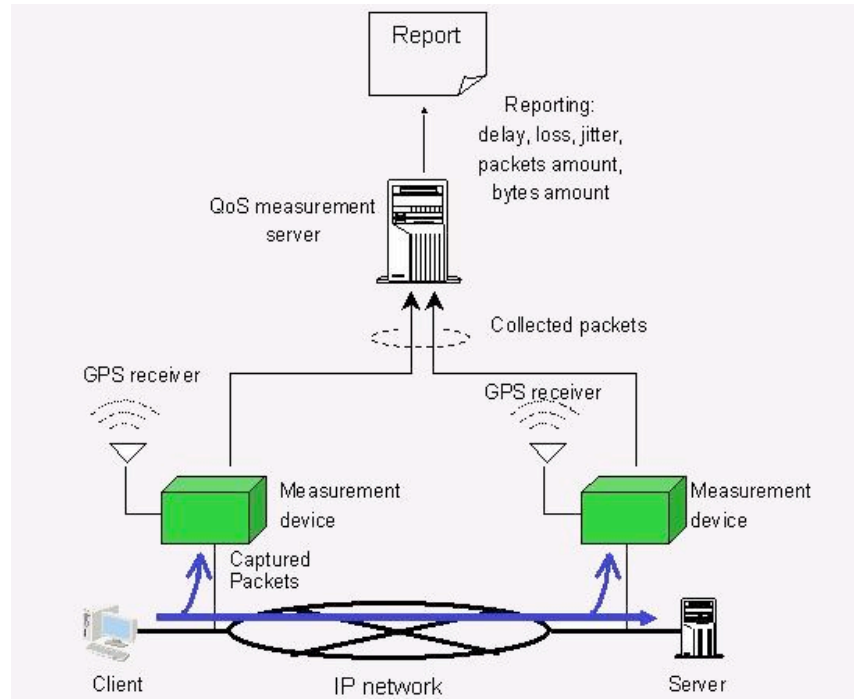
Οι συσκευές συνδέονται στον server κάθε 5-10 λεπτά και εκτελείται μια σειρά από δοκιμές για την μέτρηση μεγεθών όπως η καθυστέρηση (latency), η απώλεια πακέτων (packet loss) καθώς και η διασπορά (jitter). Οι δοκιμές για την μέτρηση του ρυθμού διέλευσης εκτελούνται κάθε 30 λεπτά ενώ δοκιμές για την χωρητικότητα εκτελούνται κάθε 6 ώρες.

- **NANO user test:** Το εργαλείο αυτό είναι ακόμη στο στάδιο της ανάπτυξης και θέτει ως στόχο της ανίχνευση για το αν ένας πάροχος Internet υποβαθμίζει την απόδοση μιας συγκεκριμένης ομάδας χρηστών, εφαρμογών ή προορισμών.

5.15. 6QM

Το πρόγραμμα αυτό είναι αφιερωμένο στην έρευνα και ανάπτυξη των τεχνολογιών μέτρησης της ποιότητας υπηρεσίας σε δίκτυα IPv6 [13]. Θα δημιουργήσει ένα περιεκτικό σύστημα το οποίο θα ενοποιεί τις διαφορετικές αναγκαίες συναρτήσεις για την μέτρηση της ποιότητας υπηρεσίας όπως είναι η συλλογή πακέτων, η ακριβής χρονική σφράγιση, η συλλογή δεδομένων, η παραγωγή αποτελεσμάτων για τα μετρικά μεγέθη της QoS καθώς και μια παρουσίαση των συνολικών συμπερασμάτων.

Για να μπορέσουμε να επιτύχουμε την ανάπτυξη εφαρμογών διαχείρισης αλλά και καταγραφής των δικτύων IPv6 στο μέλλον, είναι ζωτικής σημασίας να διευκολυνθεί αλλά και να αναπτυχθεί ο τομέας της καταγραφής και της μέτρησης των δικτύων αυτών στην έκταση που είναι εφικτό. Έτσι το πρόγραμμα αυτό συλλέγει και αναλύει δεδομένα από συστατικά στοιχεία του server ενώ τα αποθηκεύει για μετέπειτα χρήση και επεξεργασία.



Εικόνα 41. Διάγραμμα για τον τρόπο λειτουργίας του 6QM

Το εν λόγω πρόγραμμα όπως αναφέραμε έχει θέσει ως στόχο την περιεκτική προσέγγιση ως προς την ποιότητα υπηρεσίας σε δίκτυα IPv6. Για το λόγο αυτό δημιουργούνται τα αντίστοιχα εργαλεία:

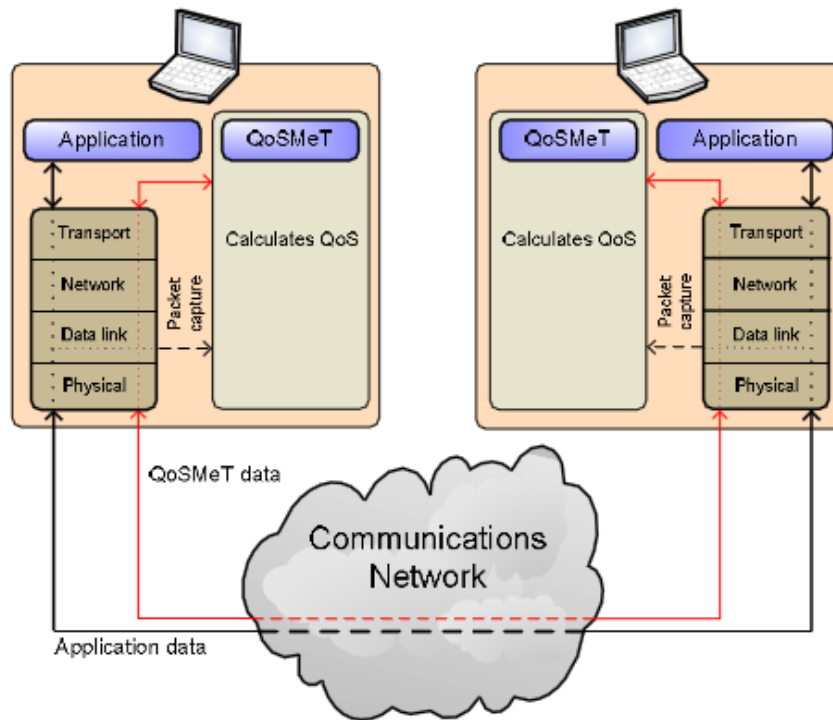
- Ανάπτυξη μιας συσκευής μέτρησης. Η συσκευή μέτρησης εισάγει μια χρονική σφραγίδα στα συλληφθέντα πακέτα με υψηλή ακρίβεια (επιπέδου micro second). Κάθε συσκευή έχει λειτουργία χρονικού συγχρονισμού με την χρήση GPS ή κάποιου εφάμιλλου μηχανισμού.
- Ανάπτυξη ενός server μέτρησης. Ο server μέτρησης συλλέγει τα συλληφθέντα πακέτα. Τότε παρέχει όχι μόνο δεδομένα χρήσης της υπηρεσίας αλλά επιπλέον και μετρικά στοιχεία για την ποιότητα υπηρεσίας (καθυστέρηση, απώλεια πακέτων, διασπορά κτλ) για κίνηση IPv6 αναλύοντας τις πληροφορίες που έλαβε μέσω της συσκευής καταγραφής.
- Ενοποίηση και τοπικές δοκιμές για τις συσκευές που περιγράψαμε.
- Δημιουργία ενός συνόλου κατευθυντήριων γραμμών για πιθανές εφαρμογές και περαιτέρω έρευνα της μέτρησης IPv6 ποιότητας υπηρεσίας σε διαφορετικά σενάρια.

5.16. QoSMeT

Πρόκειται για ένα εργαλείο παθητικής μέτρησης [5]. Το QoSMeT έχει τη δυνατότητα να παρακολουθεί και να καταγράφει την απόδοση της ποιότητας υπηρεσίας που βιώνει μια συγκεκριμένη εφαρμογή στο δίκτυο. Επιτρέπει την απ' άκρο σ' άκρο μέτρηση πραγματικού χρόνου για οποιαδήποτε δικτυακή εφαρμογή σε αμφότερες τις τηλεπικοινωνιακές κατευθύνσεις πάνω σε ετερογενή δίκτυα. Το QoSMeT δεν είναι μόνο ανεξάρτητο των στρωμάτων πρωτοκόλλων μεταφοράς και εφαρμογής αλλά επίσης και των πρωτοκόλλων δικτύου και φυσικού στρώματος όσο αυτά υποστηρίζουν IP. Επίσης είναι ικανό να μετρήσει την μονόδρομη καθυστέρηση.

Στην εικόνα που ακολουθεί (Εικόνα 32) φαίνεται η κατασκευή του QoSMeT. Πρόκειται για ένα «ελαφρύ» λογισμικό το οποίο τρέχει στην ίδια συσκευή με την μετρούμενη εφαρμογή. Το λογισμικό είναι εγκατεστημένο και στα δύο επικοινωνιακά τερματικά για να είναι σε θέση να επιτελέσει απ' άκρο σ' άκρο αμφίδρομες μετρήσεις. Το QoSMeT βασίζεται στις μετρήσεις του στρώματος δικτύου, όπου για παράδειγμα θα συλλέξει και θα καταγράψει πακέτα απ' ευθείας από την δικτυακή διεπαφή της συσκευής. Με τον τρόπο αυτό ερχόμαστε πιο κοντά στο δίκτυο όπως αυτό είναι και είμαστε ικανοί να μετρήσουμε την επίδραση του δικτύου στην QoS της εφαρμογής. Το QoSMeT καταγράφει βασικές πληροφορίες για τα πακέτα όπως για παράδειγμα τους ακριβείς χρόνους άφιξης και αναχώρησης των πακέτων.

Οι ετικέτες χρόνου προκύπτουν από ένα τοπικό ρολόι στο τερματικό αλλά δεν προσθέτονται στα πακέτα δεδομένων για να κρατηθεί ανεπηρέαστη η ροή δεδομένων της εφαρμογής. Παρ' όλα αυτά για να είμαστε σε θέση να κάνουμε υπολογισμούς QoS σε πραγματικό χρόνο, οι ακριβείς πληροφορίες πακέτου θα πρέπει με κάποιο τρόπο να ανταλλάσσονται μεταξύ των peers κατά τη διάρκεια της μέτρησης. Μια ξεχωριστή σύνδεση TCP χρησιμοποιείται για την μεταφορά αυτών των πληροφοριών ελέγχου. Φυσικά, αυτό θα προσθέσει μια επιβάρυνση στο δίκτυο αλλά στις περισσότερες περιπτώσεις η επιβάρυνση είναι αμελητέα. Ο χρήστης μπορεί να ορίσει πόσο συχνά θέλει να αποστέλλονται οι πληροφορίες ελέγχου, ώστε σε δίκτυα με περιορισμένη χωρητικότητα όπως για παράδειγμα το GPRS να μπορεί να επιλέξει να αποστέλλονται πιο σπάνια. Προφανώς υπάρχει μια ανταπόδοση ανάμεσα στις στην συχνότητα ανταλλαγής πληροφοριών ελέγχου και στην αυστηρή χρονική πραγματικότητα της καταγραφής.



Εικόνα 42. Αρχιτεκτονική QoSMeT

Εκτός από την παρακολούθηση και καταγραφή σε πραγματικό χρόνο το QoSMeT υποστηρίζει και ακριβή μετέπειτα ανάλυση των δεδομένων. Εκτός από καταγραφή σε πραγματικό χρόνο το QoSMeT μπορεί να μπει και σε λειτουργία μη πραγματικού χρόνου, όπου δεν υπάρχει μοίρασμα πληροφοριών κατά τη διάρκεια της μέτρησης. Αυτός ο τρόπος λειτουργίας του είναι βολικός εάν είναι θεμιτή η πλήρης ακρίβεια στην βάση δεδομένων και εάν το μετρούμενο δίκτυο πρέπει να έχει το ελάχιστο φορτίο.

5.17. Network Access Neutrality Observatory

Το NANO [25] αναγνωρίζει την κλιμάκωση της απόδοσης που συνάγεται από την παραβίαση της ουδετερότητας του δικτύου από τον Internet Service Provider (ISP), όπως είναι η διαφοροποίηση της παροχής υπηρεσιών για συγκεκριμένους χρήστες ή εφαρμογές. Τα ήδη υπάρχοντα συστήματα για την ανίχνευση της διαφορετικής επεξεργασίας και συμπεριφοράς είναι τις περισσότερες φορές ειδικά κατασκευασμένα για μια συγκεκριμένη εφαρμογή ή για μια συγκεκριμένη κατεύθυνση. Επειδή ο ISP μπορεί να αλλάξει τις πολιτικές διαφοροποίησης και του μηχανισμού, ο χρήστης χρειάζεται να ανιχνεύει τις διαφοροποιήσεις αυτές ανεξάρτητα από τις εφαρμογές που μπορεί να είναι το αντικείμενο εξέτασης καθώς και ανεξάρτητα από τους μηχανισμούς που το επιτυγχάνουν αυτό. Για να αποκλειστεί η διαφοροποίηση από άλλες αιτίες μείωσης της απόδοσης όπως είναι η υπερφόρτωση του δικτύου καθώς και κάποιες περιπτώσεις σφαλμάτων το NANO χρησιμοποιεί μια στατιστική μέθοδο για να εγκαταστήσει μια αιτιατή σχέση ανάμεσα στον ISP και στην απόδοση της επιθυμητής υπηρεσίας.

Οι «πράκτορες» του NANO που έχουν καταμεριστεί σε διαφορετικούς συμμετέχοντες πελάτες σε όλο το διαδίκτυο συλλέγουν δεδομένα απόδοσης για συγκεκριμένες υπηρεσίες

και αναφέρουν τις πληροφορίες αυτές σε κεντρικούς servers, οι οποίοι αναλύουν τις μετρήσεις και δίνουν τα αποτελέσματα για την κλιμάκωση της απόδοσης μιας υπηρεσίας.

Παθητική παρακολούθηση από «πράκτορα» του NANO:

Ο πράκτορας του NANO καταγράφει τον αριθμό των πακέτων που μεταφέρονται για κάθε ενεργή ροή δεδομένων ανά μονάδα χρόνου, ενώ ταυτόχρονα καταγράφει όλα τα μονοπάτια για απροσδόκητα συμβάντα όπως είναι η απώλεια πακέτων. Ο «πράκτορας» αντιστοιχίζει κάθε ροή με την υπηρεσία ή την εφαρμογή στην οποία ανήκει. Επιπλέον καταγράφεται το φορτίο που φέρει ο υπολογιστής του «πράκτορα».

Ιδιωτικότητα:

Το NANO θα καταγράψει όλη την κίνηση του Internet που γίνεται από τον υπολογιστή και θα στείλει περιορισμένα στατιστικά, αλλά δεν συλλέγει στοιχεία από το τι στέλνεται. Για παράδειγμα αν προσπαθήσει ο χρήστης να επισκεφτεί την σελίδα cnn.com θα καταγράψει την IP διεύθυνση του χρήστη, την IP διεύθυνση της σελίδας, και τον αριθμό των πακέτων που μεταδίδονται, αλλά δεν θα καταγράψει το περιεχόμενο που ο browser του χρήστη στέλνει στον cnn.com. Επιπλέον ένας χρήστης «πράκτορα» NANO μπορεί να αποφασίσει να εξαιρέσει μια ομάδα από ιστοσελίδες ή υπηρεσίες από την καταγραφή, για παράδειγμα ένας χρήστης θα μπορούσε να εξαιρέσει mail.google.com εάν δεν θέλει να καταγράψει την υπηρεσία ηλεκτρονικού ταχυδρομείου. Αυτό δίνει στο NANO την εντολή να προσπεράσει τα πακέτα τα οποία σχετίζονται με την συγκεκριμένη διεύθυνση. Μπορεί επίσης να γίνει προσωρινή παύση της καταγραφής.

6. Android – Περιγραφή Εφαρμογής

6.1. Εισαγωγή

Καθώς τα smart phones και τα tablets γίνονται ολοένα και πιο δημοφιλή, το ενδιαφέρον έχει στραφεί στην ανάπτυξη ενός λειτουργικού συστήματος κατάλληλου για τις συσκευές αυτές. Ένα από τα λειτουργικά συστήματα τα οποία αναπτύχθηκαν είναι και το Android. Αποτελεί ένα σύστημα ιδανικό για συσκευές που λειτουργούν με τη χρήση μπαταρίας και δομούνται κυρίως από υλικοτεχνικό εξοπλισμό. Τέτοιες συσκευές είναι τα κινητά τερματικά (smart phones, tablets) όπως αναφέραμε, αλλά και συσκευές εντοπισμού θέσης (GPS). Όπως όλα τα λειτουργικά συστήματα, το Android δίνει τη δυνατότητα στις διάφορες εφαρμογές που φιλοξενεί να επικοινωνούν και να χειρίζονται τα χαρακτηριστικά της εκάστοτε συσκευής.

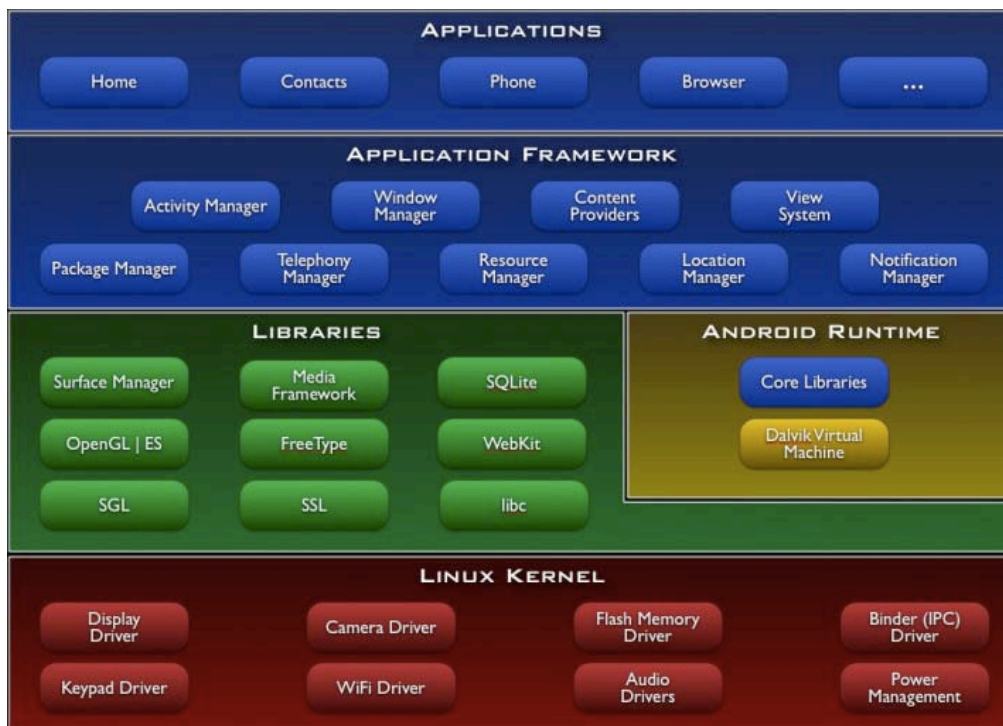
Η εταιρεία Android Inc. ιδρύθηκε στο Palo Alto της California, τον Οκτώβριο του 2003. Συνιδρυτές ήταν οι Andy Rubin, Rich Miner, Nick Sears και Chris White. Αρχικός στόχος υπήρξε η ανάπτυξη εξυπνότερων κινητών συσκευών που γνώριζαν την τοποθεσία του χρήστη τους και τις προτιμήσεις του. Η Google απέκτησε την Android Inc τον Αύγουστο του 2005 κάνοντας τη θυγατρική της Google Inc, με σχεδόν όλους τους ιδρυτές της να παραμένουν στην εταιρεία μετά την ένωση αυτή.

Η βασική διαφορά του, από τα άλλα λειτουργικά συστήματα που ανέπτυξαν διάφορες εταιρείες για παρόμοιους σκοπούς, είναι πως το Android βασίζεται εξ' ολοκλήρου στη χρήση Java. Έτσι, για την ανάπτυξη μιας Android εφαρμογής γίνεται, εκτός από τη χρήση Java, και η χρήση ενός εικονικού προσομοιωτή. Για το λόγο αυτό, για την ανάπτυξη του παρέχεται μια εικονική μηχανή (Dalvik) η οποία εκτελεί τον δικό της κώδικα byte, καθώς και μια βιβλιοθήκη συναρτήσεων και εργαλείων που ονομάζεται Android SDK. Τα βασικά χαρακτηριστικά του Android είναι τα εξής:

- Ένα πλαίσιο εφαρμογών που επιτρέπει την επαναχρησιμοποίηση και αντικατάσταση των περιεχομένων.
- Εικονική μηχανή Dalvik που προορίζεται για κινητές συσκευές.
- Ενσωματωμένο φυλλομετρητή που βασίζεται στο open source της μηχανής WebKit.
- Γραφικά που παρέχονται από μια βιβλιοθήκη δισδιάστατων γραφικών - τα τρισδιάστατα γραφικά είναι βασισμένα πάνω στην OpenGL ES 1.0.
- SQLite για τη δομή αποθήκευσης δεδομένων.
- Υποστήριξη μέσων για ήχο, video, και πρότυπα εικόνων (MPEG4, H.264, MP3, AAC, AMR, JPG, PNG, GIF).
- GSM Telephony.
- Bluetooth, EDGE, 3G, and WiFi.
- Camera, GPS, πυξίδα και επιταχυνσιόμετρο.
- Ευρύ περιβάλλον ανάπτυξης που περιλαμβάνει έναν προσομοιωτή συσκευής, εργαλεία για απομάκρυνση ιών (debugging), μνήμη και προφίλ επίδοσης και ένα πρόσθετο για το Eclipse IDE.

6.2. Η Αρχιτεκτονική του Android.

Βασικό προτέρημα του λειτουργικού συστήματος Android, το οποίο κατατάσσει το εν λόγω σύστημα σε ένα από τα κυρίαρχα αυτή τη στιγμή στην αγορά, αλλά ταυτόχρονα και στα πιο δημοφιλή ανάμεσα στους developers είναι η αρχιτεκτονική του. Στο ακόλουθο διάγραμμα φαίνονται τα βασικά συστατικά του λειτουργικού συστήματος του Android.



Εικόνα. Αρχιτεκτονική Λειτουργικού Συστήματος Android

Αναλύοντας τη δομή του, είναι δυνατός ο διαχωρισμός τεσσάρων επιπέδων.

- Εφαρμογών
- Πλαισίου Εφαρμογών
- Βιβλιοθηκών
- Χρόνου Εκτέλεσης
- Πυρήνας Linux

6.2.1. Εφαρμογές

Το Android συμβαδίζει με ένα σύνολο εφαρμογών, στις οποίες συμπεριλαμβάνονται η ηλεκτρονική αλληλογραφία, η αποστολή και λήψη SMS, η χρήση ημερολογίου, η απεικόνιση χαρτών και ο ταυτόχρονος εντοπισμός θέσης, εφαρμογές διαχείρισης επαφών, φυλλομετρητή για περιήγηση στο διαδίκτυο κ.α.

6.2.2. Πλαίσιο Εφαρμογών

Το Android παρέχει στους προγραμματιστές ένα θελκτικό περιβάλλον για να δημιουργήσουν εξαιρετικά καινοτόμες εφαρμογές, καθώς είναι ελεύθεροι να χρησιμοποιήσουν τα πλεονεκτήματα του

λειτουργικού της συσκευής, να έχουν πρόσβαση στις πληροφορίες τοποθεσίας, να τρέξουν υπηρεσίες παρασκηνίου, να προσθέσουν κοινοποιήσεις στη μπάρα κατάστασης κλπ.

Η Αρχιτεκτονική των εφαρμογών έχει σχεδιαστεί για να απλοποιήσει την επαναχρησιμοποίηση των περιεχομένων, με την έννοια ότι κάθε εφαρμογή μπορεί να εκδώσει τις δυνατότητες της και κάθε άλλη εφαρμογή να τις χρησιμοποιήσει (υπόκειται βέβαια στους περιορισμούς ασφάλειας που επιβάλλει το πλαίσιο). Αυτός ο ίδιος μηχανισμός επιτρέπει στα περιεχόμενα να αντικαθίστανται από το χρήστη. Σε όλες τις εφαρμογές βρίσκεται ένα σύνολο υπηρεσιών και συστημάτων, που περιλαμβάνει :

- Ένα εκτεταμένο σύνολο των Views, που μπορεί να χρησιμοποιηθεί για να φτιάξει μια εφαρμογή, με λίστες, πλαίσια κειμένου, κουμπιά, ακόμη κι έναν ενσωματωμένο περιηγητή δικτύου.
- Παρόχους περιεχομένου, που επιτρέπουν στις εφαρμογές να έχουν πρόσβαση σε δεδομένα από άλλες εφαρμογές (όπως στις Επαφές), ή για να κοινοποιήσουν τα δικά τους δεδομένα.
- Ένα χειριστή πόρων, που παρέχει πρόσβαση σε μη κωδικοποιημένες πηγές, όπως σε εντοπισμένα γραφικά και αρχεία layout.
- Ένα χειριστή κοινοποιήσεων, που δίνει τη δυνατότητα σε όλες τις εφαρμογές για έκθεση συνηθισμένων προειδοποιήσεων στη μπάρα κατάστασης.
- Ένα διαχειριστή δραστηριοτήτων, που χειρίζεται την κατάσταση των εφαρμογών και παρέχει μια κοινή πλοήγηση back stack.

6.2.3. Βιβλιοθήκες

Το Android περιλαμβάνει ένα σύνολο βιβλιοθηκών των C/C++ που χρησιμοποιούνται από διάφορα συστατικά του συστήματος. Αυτές οι δυνατότητες εκτίθενται στους προγραμματιστές μέσω του πλαισίου εφαρμογής. Κάποιες από τις βιβλιοθήκες αναφέρονται παρακάτω :

- System C library
- Media Libraries
- Surface Manager
- LibWebCore
- SGL
- 3D libraries
- FreeType
- SQLite

6.2.4. Χρόνος Εκτέλεσης του Android

Κάθε εφαρμογή του Android εκτελείται στη δική της διαδικασία, με το παράδειγμα της εικονικής μηχανής Dalvik. Η μηχανή αυτή έχει γραφτεί έτσι ώστε μια συσκευή να μπορεί να τρέξει πολλαπλά Virtual Machines αποτελεσματικά. Η Dalvik VM εκτελεί αρχεία στο πρότυπο Dalvik Executable (.dex), που προορίζεται για ελάχιστο ίχνος μνήμης, Η VM είναι βασισμένη στην καταγραφή και τρέχει κλάσεις που συμπιέζονται από ένα συμπίεστη της γλώσσας Java, που έχει μετατραπεί σε ένα αρχείο .dex από το συμπεριλαμβανόμενο εργαλείο «dex». Η παραπάνω μηχανή στηρίζεται στον πυρήνα του Linux για λειτουργικότητα, όπως τη διαστρωμάτωση και τη διαχείριση μνήμης χαμηλού επιπέδου.

6.2.5. Πυρήνας Linux

Το Android στηρίζεται στην έκδοση 2.6 του Linux για υπηρεσίες συστήματος όπως η ασφάλεια, η διαχείριση επεξεργασίας και λειτουργίες δικτύου. Ο πυρήνας επίσης δρα σαν ένα στρώμα επικοινωνίας ανάμεσα στο hardware και το software.

6.3. Ιστορική Αναδρομή των Εκδόσεων

Στον παρακάτω πίνακα φαίνονται οι εκδόσεις του Android, από το ξεκίνημά τους, με το κωδικό όνομα της καθεμίας.

Platform Version	API Level	Version Code
Android 1.0	1	BASE
Android 1.1	2	BASE 1.1
Android 1.5	3	CUPCAKE
Android 1.6	4	DONUT
Android 2.0	5	ECLAIR
Android 2.0.1	6	ECLAIR 0.1
Android 2.1.x	7	ECLAIR MR1
Android 2.2.x	8	FROYO
Android 2.3 Android 2.3.1 Android 2.3.2	9	GINGERBREAD
Android 2.3.3 Android 2.3.4	10	GINGERBREAD MR1

Πρέπει να κατανοήσουμε ότι η αναγνώριση του API Level παίζει σημαντικό ρόλο στο να διασφαλιστεί η συμβατότητα της εφαρμογής με τις συσκευές στις οποίες θα εγκατασταθεί. Κάθε διαδοχική έκδοση της πλατφόρμας Android μπορεί να περιλαμβάνει ενημερώσεις στο πλαίσιο εφαρμογών API της Android. Οι ενημερώσεις αυτές είναι έτσι σχεδιασμένες ώστε το νέο επίπεδο API να παραμένει συμβατό με τις παλαιότερες εκδόσεις. Έτσι, οι περισσότερες αλλαγές είναι προσθετικές και εισάγουν μια νέα ή αντικαθιστούν μια λειτουργικότητα.

Καθώς τα μέρη της API ενημερώνονται, τα παλαιότερα που αντικαθίστανται, αν και δε χρησιμοποιούνται πλέον, δεν απομακρύνονται τελείως. Το πλαίσιο API, που παρέχει μια πλατφόρμα Android, εξειδικεύεται χρησιμοποιώντας έναν ακέραιο αριθμό αναγνώρισης, γνωστό και ως API Level. Κάθε έκδοση της πλατφόρμας υποστηρίζει ακριβώς ένα τέτοιο Level, αν και η υποστήριξη για τα προηγούμενα levels είναι δεδομένη.

- Η έκδοση Eclair 2.0 περιλάμβανε ένα νέο περιηγητή δικτύου, με ένα νέο user interface και την υποστήριξη της HTML5 και W3C Geolocation API, καθώς επίσης και μια εξελιγμένη εφαρμογή camera με χαρακτηριστικά όπως ψηφιακό zoom, flash, εφέ χρωμάτων κτλ.
- Η έκδοση Eclair 2.1 περιλάμβανε υποστήριξη ελέγχου ομιλίας μέσω ολόκληρου του OS. Επίσης περιλάμβανε νέο launcher, με πέντε οθόνες αντί για τρεις, κινούμενο φόντο και ένα πλήκτρο για την έναρξη του menu (αντί για slider), καθώς και μια νέα weather app και βελτιωμένη λειτουργικότητα στις εφαρμογές των Email και Phonebook.

- Η έκδοση Frodo εισήγαγε τις βελτιώσεις ταχύτητας με βελτιστοποίηση του JIT και τη μηχανή V8 JavaScript του Chrome, πρόσθεσε τη δέσμευση του Wi-Fi και υποστήριξη του Adobe Flash.
- Η έκδοση Gingerbread καθόριζε την user interface, βελτίωσε το πληκτρολόγιο και τις επιλογές της αντιγραφής/επικόλλησης, την υποστήριξη SIP (κλήσεις VoIP) και πρόσθεσε την υποστήριξη της Επικοινωνίας κοντινού πεδίου.

6.4. Δομή Εφαρμογής

Η εκτέλεση εφαρμογών είναι ίσως ο μεγαλύτερος στόχος για ένα λειτουργικό σύστημα και το Android παρέχει ποικίλους τρόπους για τη σύνδεση, εκτέλεση αλλά και διαχείριση εφαρμογών. Για το λόγο αυτό, το android διαφοροποιεί τους όρους application, process, task, thread.

6.4.1. Processes και Threads

Διαχωρίζονται πέντε τύποι διαδικασιών (process) στο Android για τον έλεγχο της λειτουργίας του συστήματος, καθώς και των προγραμμάτων που εκτελούνται. Οι διάφοροι αυτοί τύποι έχουν και ξεχωριστά επίπεδα σημασίας. Έτσι η ιεραρχία που προκύπτει, αναλόγως με την σημασία της κάθε διαδικασίας, είναι η ακόλουθη:

Foreground: Είναι μια διαδικασία, η οποία εκτελεί μια Activity, μια υπηρεσία η οποία παρέχει την Activity, μια υπηρεσία εκκίνησης ή τερματισμού, ή ένας Broadcast Receiver.

Visible: Εάν μια διαδικασία διατηρεί μια Activity σε κατάσταση παύσης, αλλά ορατή ή μια υπηρεσία είναι δεσμευμένη σε μία ορατή Activity χωρίς καμία συνιστώσα στο προσκήνιο, τότε αυτή η διαδικασία καλείται ορατή διαδικασία (visible process).

Background: Μια Activity η οποία δεν είναι πλέον ορατή διατηρείται από μια background διαδικασία.

Empty: Οι διαδικασίες αυτές δεν διαθέτουν κανένα ενεργό συστατικό και υπάρχουν μόνο για λόγους μνήμης.

Εάν το σύστημα μας οδηγηθεί σε κατάσταση έλλειψης πόρων μνήμης, τότε η ιεραρχία της σημασίας των διαδικασιών αποτελεί βασικό κριτήριο της απόφασης του συστήματος, για τον τερματισμό κάποιας διαδικασίας και την απελευθέρωση πόρων μνήμης. Για τον λόγο αυτό, οι Empty Processes (Κενές Διαδικασίες) είναι πιο πιθανό να τερματιστούν πρώτες, ενώ στη συνέχεια αναμένεται να ακολουθήσουν οι Background. Συνήθως μόνο οι διαδικασίες empty και background τερματίζονται.

Οι διαδικασίες μπορούν να περιέχουν πολλά threads, όπως αυτό είναι σύνηθες για τα συστήματα που βασίζονται στο Linux. Οι περισσότερες Android εφαρμογές αποτελούνται από πολλαπλά thread για το διαχωρισμό του UI από την είσοδο των δεδομένων, καθώς και μακροχρόνιους υπολογισμούς.

6.4.2. Applications and Tasks

Οι Android εφαρμογές εκτελούνται από διαδικασίες και από τα συμπεριλαμβανόμενα thread. Οι δύο όροι task και application είναι στενά αλληλένδετοι, δεδομένου πως ένα task μπορεί να γίνει αντιληπτό σαν εφαρμογή από το χρήστη. Στην πραγματικότητα τα tasks είναι μια σειρά από activities. Αποτελούν πρακτικά ένα τύπο λογικού ιστορικού των ενεργειών του χρήστη.

Για παράδειγμα, η εκκίνηση μιας εφαρμογής email και στη συνέχεια το άνοιγμα ενός link, το οποίο περιέχεται μέσα σε ένα από τα ληφθέντα mail, ενώ συμπεριλαμβάνει δύο εφαρμογές καθώς και τουλάχιστον δύο activities. Το βασικό πλεονέκτημα του task είναι η δυνατότητα που δίνει στο χρήστη να μπορεί να επιστρέψει σε προηγούμενα στάδια, βήμα προς βήμα όπως κατά το διάβασμα μιας στοίβας.

6.5. Σύστημα Δυναμικής Δειγματοληψίας WhereAmI

Το σύστημα αυτό αναπτύχθηκε για τους σκοπούς εκπλήρωσης της παρούσας διπλωματικής εργασίας και αποτελεί ένα εργαλείο που προσφέρει στο χειριστή του τη δυνατότητα για χρονική αλλά και χωρική δειγματοληψία του σήματος μιας εταιρείας κινητής τηλεφωνίας. Για την υλοποίηση του χρησιμοποιήθηκε η δυνατότητα του κινητού για μέτρηση της στάθμης και της ισχύος του σήματος, συνδυασμένη με την παροχή πληροφοριών εντοπισμού θέσης που δίνει το Android αλλά και η Google. Το σύστημα έχει σαν βασικό σκοπό τη συνεχή μέτρηση της ισχύος του σήματος που λαμβάνει το κινητό, τον προσδιορισμό της θέσης του χρήστη, την οποία και προβάλλει σε χρήστη αλλά και την αποθήκευση των μετρήσεων για μετ' έπειτα επεξεργασία. Βασικό στοιχείο για την υλοποίησή του ήταν η ανάπτυξη της ιδιότητας δυναμικής δειγματοληψίας της ποιότητας του δικτύου κινητής τηλεφωνίας.

Για την επίτευξη της δυναμικότητας του τρόπου δειγματοληψίας υιοθετήθηκε η αντίληψη της ανάπτυξης τόσο χρονικής όσο και χωρικής δειγματοληψίας. Η υλοποίηση τόσο της χρονικής όσο και της χωρικής δειγματοληψίας έγκειται στην χρήση ενός βρόχου με μετρητή, ο οποίος μόλις ολοκληρώσει το πέρας του διαστήματος (χωρικού ή χρονικού) επιλέγει να αποθηκεύσει την παρούσα μέτρηση. Με τον τρόπο αυτό κρατάμε μετρήσεις ανάλογα με το χρονικό διάστημα ή χωρικό διάστημα που θέλει ο χρήστης, κάνοντας έτσι τη χρονική ή χωρική δειγματοληψία του σήματος για μια διαδρομή την οποία αποφασίζει ο χρήστης.

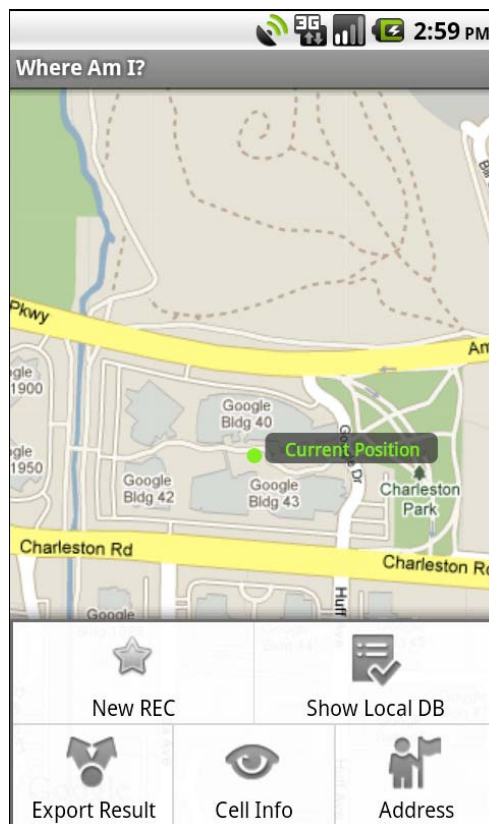
Οι επιθυμητές μετρήσεις αποθηκεύονται σε δύο διαφορετικά μέρη του κινητού, με δύο διαφορετικές μορφές. Η πρώτη μορφή είναι η προσθήκη των μετρήσεων σε μια βάση δεδομένων που βρίσκεται εσωτερικά του κινητού και είναι προσβάσιμη μόνο από την συγκεκριμένη εφαρμογή. Η δεύτερη μορφή είναι η εγγραφή των δεδομένων σε ένα αρχείο XML με συγκεκριμένη μορφή και η αποθήκευση του αρχείου αυτού, στην κάρτα μνήμης sd του κινητού τηλεφώνου.

Οι πληροφορίες που αποθηκεύονται σε κάθε καταγραφή είναι το όνομα της διαδρομής, το οποίο θέτει ο χρήστης, το γεωγραφικό μήκος και πλάτος, την ισχύ του σήματος (τόσο σε dBm όσο και ο λόγος CINR), το είδος της καταγραφής (καταγραφή χρόνου ή απόστασης ή ταχύτητας), τον τύπο του δικτύου (UMTS, CDMA κτλ), βασικά στοιχεία της κυψέλης που βρίσκεται το τερματικό όπως είναι ο Mobile Network code και ο Mobile Country Code, η id της κυψέλης που βρισκόμαστε καθώς και στοιχεία των γειτονικών κυψελών. Ο κώδικας για τις βασικές λειτουργίες του συστήματος φαίνεται παρακάτω.

6.5.1. Η εφαρμογή WhereAmI

Στην ενότητα αυτή θα αναλύσουμε το σύστημα WhereAmI αναφέροντας τις επιμέρους δυνατότητες, αλλά και επιλογές που έχει ο χρήστης χρησιμοποιώντας το. Ξεκινώντας από την αρχική οθόνη της εφαρμογής, ο χρήστης μπορεί να δει έναν χάρτη στο οποίο φαίνεται η

θέση του με την χρήση μιας πράσινης κοιλίδας και αντίστοιχου κειμένου. Όταν ο χρήστης πατήσει το πλήκτρο μενού εμφανίζεται το κεντρικό μενού της εφαρμογής, στο οποίο ο χρήστης έχει τις παρακάτω επιλογές, όπως φαίνεται και στην εικόνα που ακολουθεί.



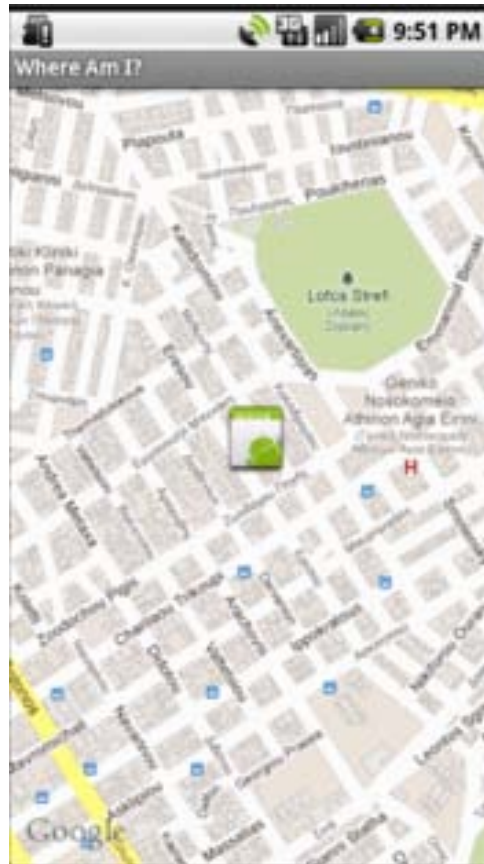
Τα εργαλεία που δίνονται στο χρήστη στο κεντρικό μενού είναι τα παρακάτω:

1. **New REC:** Με την επιλογή του ο χρήστης εισέρχεται στην κλάση ονοματοδοσίας στην οποία θα κληθεί να εισάγει το όνομα που θέλει να έχει η παρούσα καταγραφή αλλά και να ορίσει το είδος της καταγραφής ως εσωτερικού ή εξωτερικού χώρου.
2. **Show Local DB:** Η επιλογή αυτή μεταφέρει τον χρήστη στην κλάση προβολής των περιεχομένων της βάσης δεδομένων, όπως αυτά έχουν μέχρι την δεδομένη στιγμή. Η βάση δεδομένων που αναφερόμαστε είναι η εσωτερική βάση δεδομένων του τερματικού.
3. **Export Result:** Ο χρήστης μεταφέρεται σε μια κλάση που του δίνει την δυνατότητα της εξαγωγής των δεδομένων της βάσης, σε αρχεία διαφορετικής μορφής.
4. **Cell Info:** Με την επιλογή του ο χρήστης μπορεί να δει τα χαρακτηριστικά των γειτονικών κυψελών, τις οποίες μπορεί να εντοπίσει το τερματικό.
5. **Address:** Με την επιλογή του γίνεται δυνατή η αναζήτηση και η προβολή σε χάρτη μιας προκαθορισμένης τοποθεσίας, την οποία εισάγει ο χρήστης.

Για λόγους καλύτερης συνοχής στην περιγραφή του συστήματος, θα αναφερθούμε αντίστροφα στην ανάλυση των επιμέρους προγραμμάτων και επιλογών που έχει ο χρήστης.

Με την επιλογή του κουμπιού **Address** παρουσιάζεται στην οθόνη του τερματικού ένας χάρτης που απεικονίζει ένα προκαθορισμένο σημείο. Ο χρήστης έχει τη δυνατότητα, πατώντας το κουμπί μενού, να εισάγει μια διεύθυνση και εν συνεχεία αφού επιβεβαιώσει την επιλογή του να του απεικονίσει η εφαρμογή την συγκεκριμένη τοποθεσία στον χάρτη. Για

την αποφυγή συγκρούσεων με συνωνυμίες οδών σε διαφορετικές πόλεις, ο χρήστης προτρέπεται να εισάγει χώρα, πόλη, καθώς και οδό και αριθμό. Το αποτέλεσμα της εκτέλεσης αυτής της εφαρμογής φαίνεται στη διπλανή εικόνα.



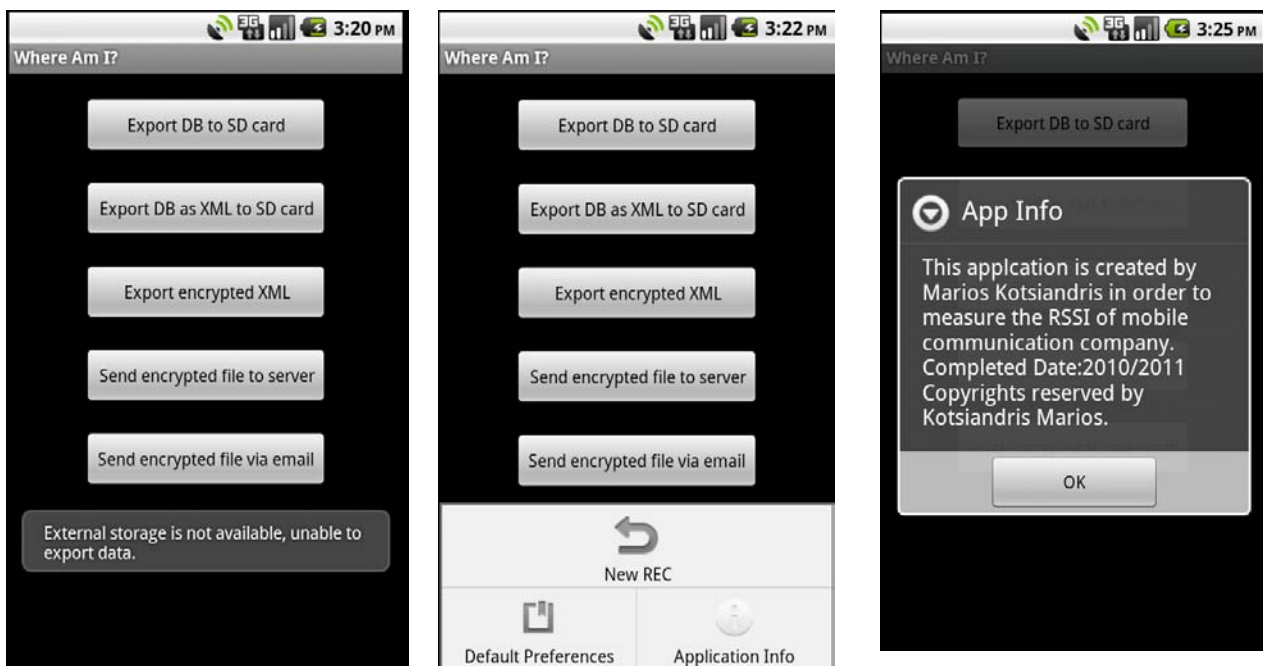
Με την επιλογή **Cell Info**, ο χρήστης έχει την δυνατότητα να δει τα χαρακτηριστικά όλων των γειτονικών κυψελών, με τις οποίες μπορεί να συνδεθεί το τερματικό του. Τα χαρακτηριστικά αυτά περιλαμβάνουν το σήμα που δέχεται το τερματικό από τη συγκεκριμένη κυψέλη στην παρούσα στιγμή, την ταυτότητα της κυψέλης (γνωστή και ως cell id), την περιοχή στην οποία βρίσκεται η κυψέλη αυτή, η οποία και προβάλλεται με έναν κωδικό που αντιστοιχίζεται στην περιοχή (Location Area Code), αλλά επίσης και στοιχεία όπως το είδος του δικτύου (CDMA, HSPA κτλ), τον Mobile Network Code αλλά και τον Mobile Country Code.

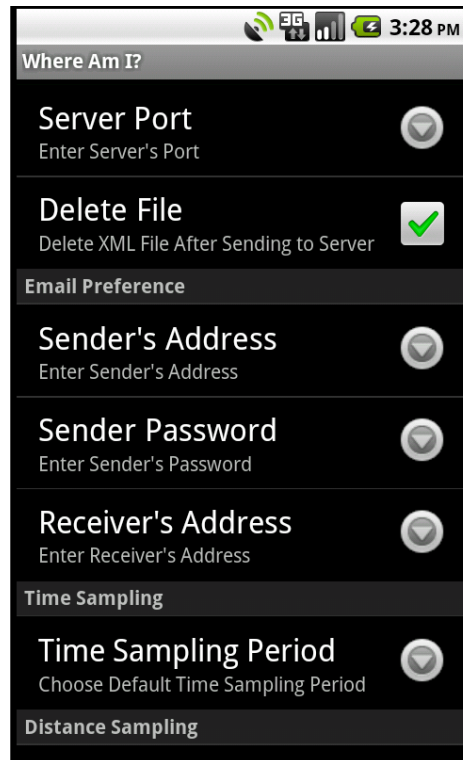
Με το κουμπί **Export Result** δίνεται η δυνατότητα στο χρήστη να μεταβεί κατευθείαν στο μενού εξαγωγής των δεδομένων της βάσης, χωρίς να χρειαστεί να περάσει από τα ενδιάμεσα στάδια του συστήματος καταγραφής. Η κλάση αυτή με τη σειρά της έχει τα εξής χαρακτηριστικά:

- Εξαγωγή της βάσης δεδομένων στην sd card σαν αρχείο βάσης δεδομένων (example.db). Το αρχείο αυτό θα βρίσκεται στο φάκελο /sd/exampledata/example.db
- Εξαγωγή των περιεχομένων της βάσης στην sd card σαν ένα αρχείο xml. Το αρχείο θα βρίσκεται στην κάρτα μνήμης με path: sdcard/exampledata/samples.xml

- Η επόμενη επιλογή είναι να γίνει εξαγωγή του αρχείου σε κρυπτογραφημένη μορφή, η οποία και δίνει στο αρχείο μια κατάληξη την οποία έχουμε δώσει εμείς σαν προεπιλογή στο πρόγραμμα και η οποία είναι μοναδική. Αυτό το κάναμε για να μην έχουμε σύμπτωση κατάληξης με άλλα αρχεία, γεγονός που οδηγεί το χρήστη να χρησιμοποιήσει τον αποκρυπτογράφο που παρέχεται μαζί με την εφαρμογή, για να το ανοίξει και να το διαβάσει.
- Ο χρήστης μπορεί να επιλέξει την αποστολή του κρυπτογραφημένου αρχείου σε έναν socket server.
- Μπορεί να επισυνάψει το κρυπτογραφημένο αρχείο σε ένα email και να το ταχυδρομήσει στο gmail λογαριασμό της εφαρμογής.

Για κάθε μία από τις επιλογές που κάνει ο χρήστης, το σύστημα ελέγχει αν η κάρτα μνήμης είναι παρούσα έτσι ώστε να μπορεί να γίνει η αποθήκευση των δεδομένων κατά την εξαγωγή τους. Σε περίπτωση που δεν υπάρχει κάρτα μνήμης στο τερματικό, έχουμε την ειδοποίηση του χρήστη για έλλειψη αποθηκευτικού χώρου, καθώς και την απόρριψη της ενέργειας εξαγωγής. Αυτό φαίνεται και στην εικόνα που ακολουθεί. Αντιθέτως, σε περίπτωση που η εξωτερική κάρτα μνήμης είναι συνδεδεμένη στο κινητό, το σύστημα ειδοποιεί το χρήστη για την ομαλή και σωστή αποθήκευση των δεδομένων του. Επιπλέον, θα πρέπει να αναφέρουμε ότι ο χρήστης έχει την δυνατότητα με το πάτημα του μενού να επιστρέψει στην New REC Class, δηλαδή να κάνει μια νέα καταγραφή, να ανοίξει την κλάση αποθήκευσης των «αγαπημένων επιλογών» (Default Preferences) του δηλαδή των επιλογών που θέλει να ορίζονται αυτόματα με την έναρξη της εφαρμογής, αλλά και να δει γενικές πληροφορίες για την εφαρμογή με το πάτημα του Application Info.

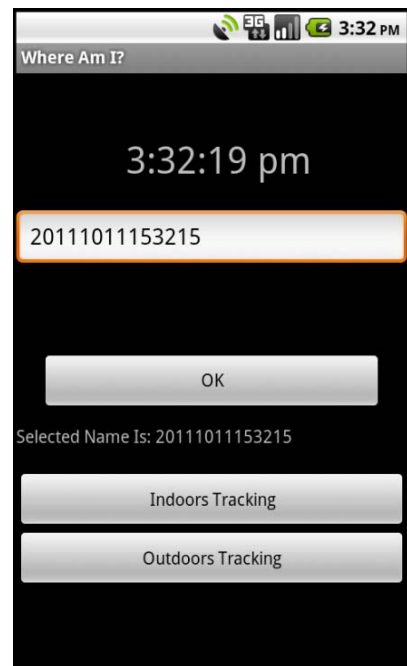
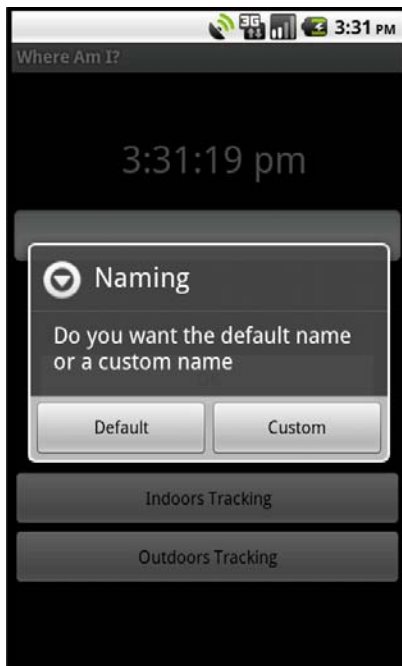




Τέλος, σειρά έχει η επιλογή του **New REC**. Με την επιλογή αυτή, ο χρήστης εκκινεί μια νέα καταγραφή. Αρχικά, στην οθόνη εμφανίζεται ένα dialog interface το οποίο και δίνει στο χρήστη την δυνατότητα της επιλογής του ονόματος που θα χρησιμοποιήσει για την καταγραφή του. Οι επιλογές του χρήστη είναι Default και Custom. Με την Default επιλογή το όνομα της καταγραφής θα πάρει τη μορφή ΕΤΟΣ/ΜΗΝΑΣ/ΗΜΕΡΑ/ΩΡΑ/ΛΕΠΤΑ/ΔΕΥΤΕΡΟΛΕΠΤΑ.

Με τον τρόπο αυτό, γίνεται εφικτή η μοναδικότητα ανάμεσα στα ονόματα των καταγραφών και αυτό γιατί ο χρήστης δεν μπορεί να εκκινήσει 2 καταγραφές στο χρονικό διάστημα του ενός δευτερολέπτου. Παρ' όλα αυτά, σε περίπτωση που είναι επιθυμητή η ονοματοδοσία της συγκεκριμένης καταγραφής, με ένα όνομα διαφορετικό από το προκαθορισμένο, ο χρήστης μπορεί να το δώσει είτε επιλέγοντας Custom είτε ακόμη και αφού έχει επιλέξει το προκαθορισμένο να το αλλάξει στο πλαίσιο κειμένου που φαίνεται στην οθόνη. Αν ο χρήστης πατήσει OK θα προβληθεί στην οθόνη του το όνομα το οποίο έχει δώσει.

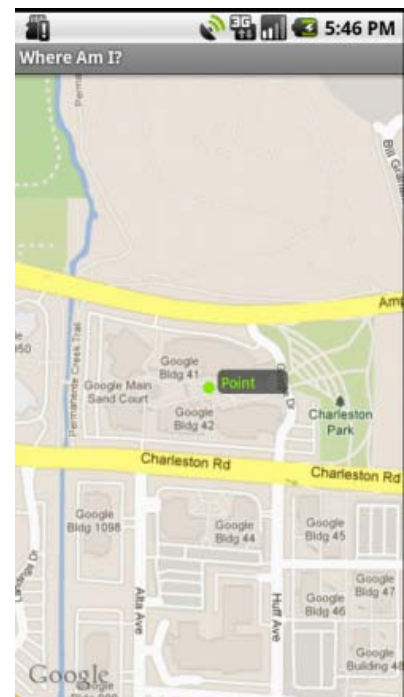
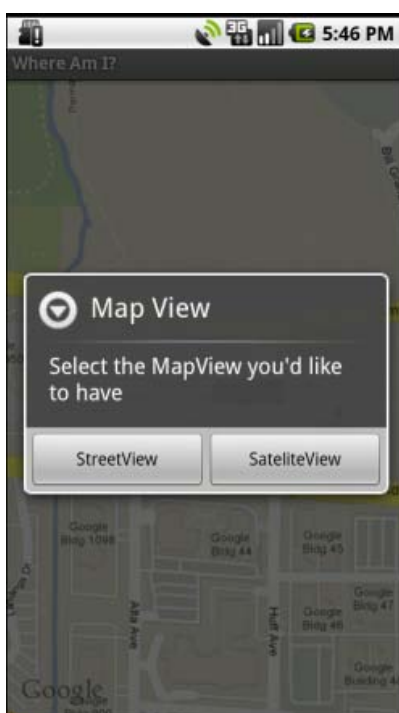
Στη συνέχεια, έρχεται η επιλογή ανάμεσα στο είδος της καταγραφής. Οι καταγραφές μπορούν να γίνουν τόσο σε εσωτερικά κτιρίων όσο και σε εξωτερικούς χώρους. Για να περάσουμε αυτήν την πληροφορία στα δεδομένα μας, χωρίς όμως να επιβαρύνουμε επιπλέον την εφαρμογή, επιλέξαμε να περάσουμε τον τύπο INDOORS, OUTDOORS στο όνομα της καταγραφής. Έτσι, ο χρήστης με την πίεση των πλήκτρων Indoors και Outdoors, εκτός από το να μεταβεί στην επόμενη κλάση, διευκρινίζει αυτόματα και το είδος της καταγραφής στο όνομα της. Έτσι, το τελικό όνομα της καταγραφής θα είναι <CUSTOM/DEFAULT><INDOORS/OUTDOORS>.

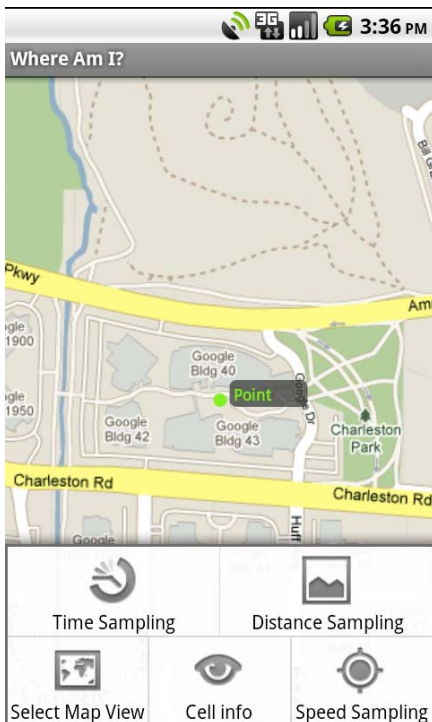


Με την επιλογή είτε Indoors είτε Outdoors, ο χρήστης μεταφέρεται σε μια άλλη activity για την επιμέρους ρύθμιση και επιλογή των τρόπων καταγραφής του σήματος. Με την εκκίνηση της εφαρμογής αυτής, ο χρήστης καλείται να επιλέξει τον τρόπο με τον οποίο θέλει να γίνεται η προβολή του χάρτη. Δίνονται δύο επιλογές, οι οποίες και είναι:

- Προβολή Φωτογραφιών από δορυφόρο.
- Προβολή με τη Μορφή Οδικού Χάρτη.

Τα αποτελέσματα της κάθε μιας επιλογής φαίνονται παρακάτω:





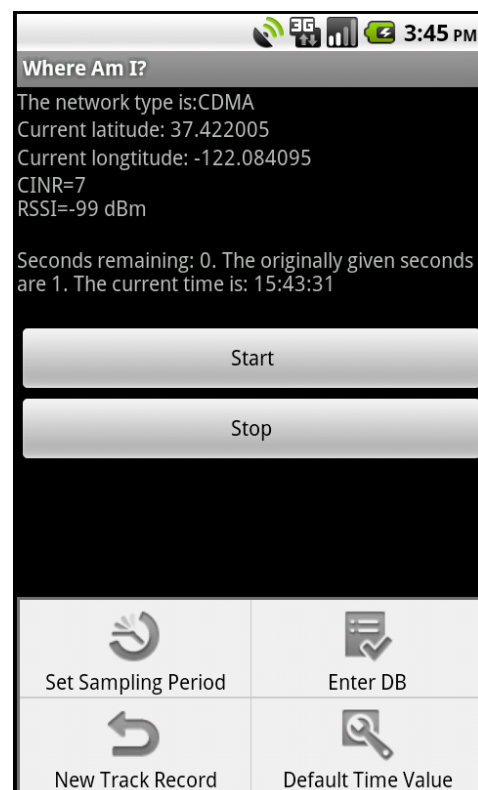
Η activity αυτή φιλοξενεί ένα menu, στο οποίο ο χρήστης μπορεί να επιλέξει ένα από τα παρακάτω:

- Αλλαγή του τρόπου προβολής του χάρτη είτε με φωτογραφίες δορυφόρου, είτε με προβολή οδικού χάρτη.
- Εκκίνηση χρονικής δειγματοληψίας του δικτύου. (Time Sampling)
- Εκκίνηση χωρικής δειγματοληψίας του δικτύου. (Distance Sampling)
- Πληροφορίες γειτονικών κυψελών (Cell info - περιγράφηκε στην επιλογή του αρχικού μενού)
- Εκκίνηση καταγραφής με βάση την ταχύτητα του κινητού. (Speed Sampling)

Θα πρέπει να σημειώσουμε ότι σε περίπτωση που στην New REC Class ο χρήστης έχει επιλέξει καταγραφή στο εσωτερικό κτιρίων (Indoors Button), τότε το πρόγραμμα τον ενημερώνει ότι σε περίπτωση που θέλει να κάνει χωρική καταγραφή ίσως αυτό να προκαλέσει ανακρίβεια στις μετρήσεις του αφού το στίγμα του θα θεωρείται σταθερό και η θέση του δεν θα μεταβάλλεται.

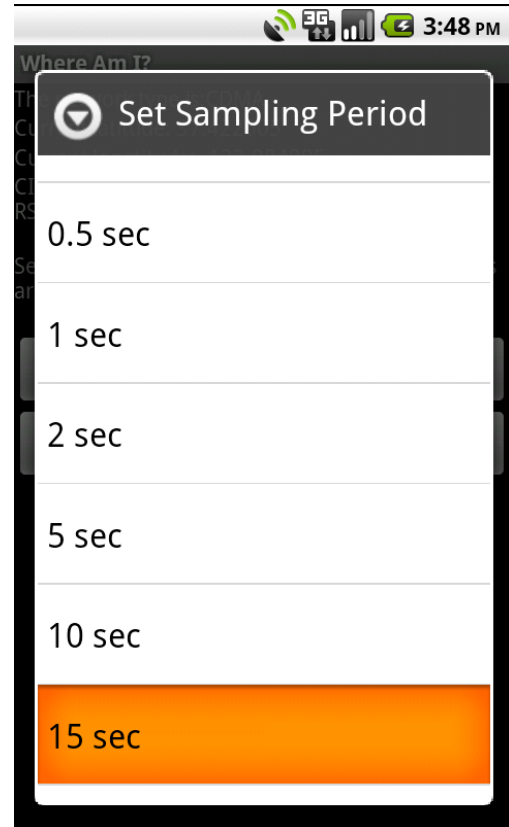
6.5.1.1. Χρονική Δειγματοληψία Σήματος

Με την εκκίνηση αυτής της activity, ο χρήστης μπορεί να κάνει χρονική δειγματοληψία σήματος για το δίκτυο κινητής τηλεφωνίας. Στην οθόνη, όπως βλέπουμε και στη διπλανή εικόνα φαίνονται το γεωγραφικό μήκος και πλάτος, ο λόγος Carrier to Interference Noise Ratio καθώς και η RSSI τιμή του σήματος. Επιπλέον φαίνεται ο αντίστροφος μετρητής των δευτερολέπτων, όπου στον μηδενισμό του θα έχουμε κάθε φορά τη λήψη μιας καταγραφής.



Στο menu της activity αυτή ο χρήστης έχει τις παρακάτω επιλογές:

- Επιλογή χρονικού διαστήματος που θα μεσολαβεί ανάμεσα σε δύο καταγραφές (φαίνεται στη διπλανή εικόνα).
- Είσοδο και προβολή της Βάσης Δεδομένων.
- Επιλογή για την εκκίνηση μιας νέας καταγραφής.
- Επαναφορά της χρονικής επιλογής στα αρχικά second.

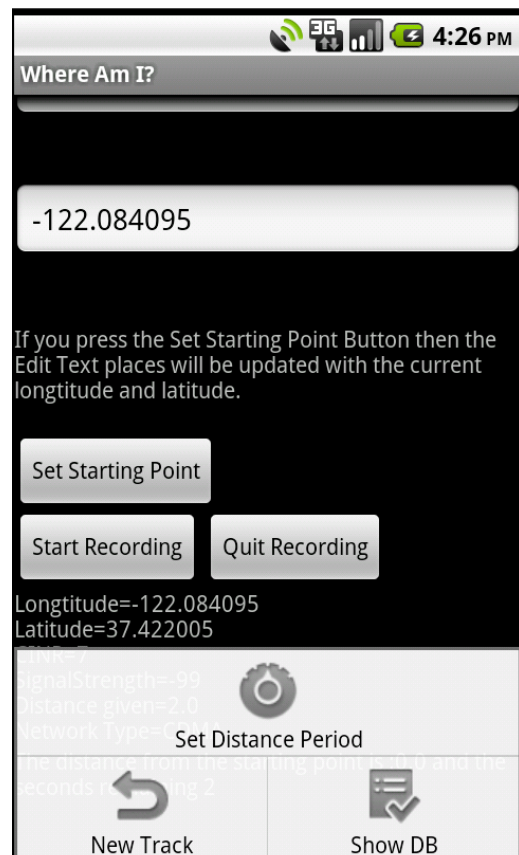


6.5.1.2. Χωρική Δειγματοληψία Σήματος

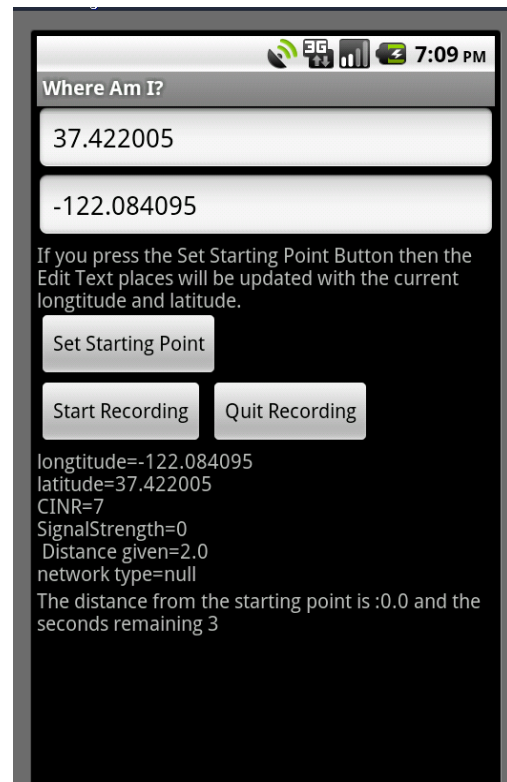
Με την εκκίνηση της activity distance ο χρήστης έχει την δυνατότητα να κάνει μια χωρική δειγματοληψία του σήματος. Αρχικά, όπως βλέπουμε και στο διπλανό σήμα, πατώντας το κουμπί Set Starting Point, ο χρήστης ορίζει την τρέχουσα θέση του ως σημείο εκκίνησης της καταγραφής. Στη συνέχεια μπορεί να επιλέξει την απόσταση που θέλει να παρεμβάλλεται από μια λίστα προτεινόμενων αποστάσεων.

Με την επιλογή Start Recording ξεκινά η καταγραφή, ενώ στο χρήστη γίνονται εμφανή τα τρέχοντα στοιχεία της καταγραφής όπως

- Γεωγραφικό Μήκος
- Γεωγραφικό Πλάτος
- CINR
- RSSI
- Επιλεγμένη απόσταση
- Τύπος Δικτύου



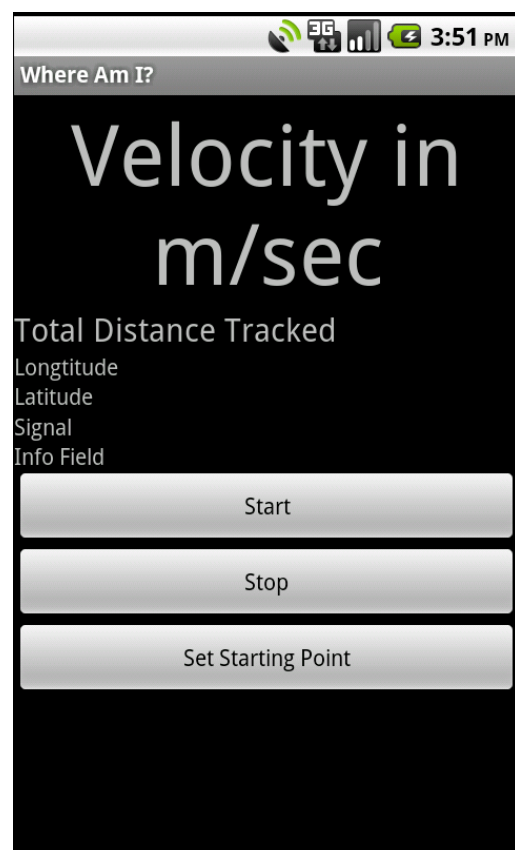
Επίσης δίνεται και η τρέχουσα απόσταση από το προηγούμενο σημείο καταγραφής. Για να εξαλείψουμε την περίπτωση όπου ο χρήστης παραμένει ακίνητος για μεγάλο χρονικό διάστημα, αλλά και την περίπτωση της εξαιρετικά μειωμένης ακρίβειας, προσθέσαμε την ασφαλιστική δικλείδα ενός παράλληλου χρονικού βρόχου, ο οποίος εξασφαλίζει τη λήψη καταγραφής μετά από εύλογο χρονικό διάστημα.



6.5.1.3. Δειγματοληψία με βάση την Ταχύτητα

Στην activity Speed η δειγματοληψία του σήματος γίνεται ανάλογα με την ταχύτητα του κινητού τερματικού τη δεδομένη στιγμή. Έτσι σε περίπτωση που το τερματικό κινείται με ταχύτητα 30 m/s, το χρονικό διάστημα ανάμεσα σε μία καταγραφή θα είναι μικρότερο από αυτό σε περίπτωση κίνησης με 2 m/s. Ο χρήστης καλείται ξανά να δηλώσει το σημείο εκκίνησης του, ενώ γνωστοποιούνται σε αυτόν δεδομένα για την καταγραφή όπως:

- Τρέχουσα ταχύτητα.
- Συνολική απόσταση από το σημείο εκκίνησης.
- Γεωγραφικό Μήκος
- Γεωγραφικό Πλάτος
- RSSI



6.5.1.4. Προβολή Βάσης Δεδομένων

Οι μετρήσεις που καταγράφονται, με την εκτέλεση οποιασδήποτε από τις παραπάνω activities, αποθηκεύονται σε μία εσωτερική βάση δεδομένων του τηλεφώνου, στην οποία ο χρήστης έχει πρόσβαση μόνο μέσω της εκτέλεσης του συστήματος. Όπως φαίνεται και στην παρακάτω εικόνα, ο χρήστης μπορεί να διαχειριστεί τα περιεχόμενα της βάσης με δύο τρόπους:

- Να προβάλει τα συνολικά δεδομένα που είναι αποθηκευμένα στην Β.Δ.
- Να διαγράψει όλα τα δεδομένα που περιέχονται στην Β.Δ.
- Να διαγράψει συγκεκριμένη καταγραφή από τη Β.Δ.

Η εσωτερική αυτή βάση δεδομένων φιλοξενεί τα εξής στοιχεία για την κάθε καταγραφή με την σειρά που αναφέρονται:

- Όνομα καταγραφής.
- Γεωγραφικό Πλάτος
- Γεωγραφικό Μήκος
- RSSI
- CINR
- TimeStamp
- MCC
- MNC
- LAC
- Τύπος καταγραφής (πχ. Χρονικός, Χωρικός, Ταχύτητα)
- Τύπος δικτύου (πχ. CDMA, GPRS κτλ.)
- Cell id
- Current Speed (Σε περίπτωση που η καταγραφή έχει γίνει με την activity speed)
- Neighbor Cell #
- Neighbor Cell # cell id
- Neighbor Cell # rssi

Στα πεδία των γειτονικών κυψελων αποθηκεύεται αρχικά ο αριθμός της κυψέλης (κυψέλη 1,2,3 κτλ) καθώς και αντίστοιχα ο κωδικός cell id της κάθε κυψέλης και το σήμα σε dBm που λαμβάνει το τερματικό από την κυψέλη αυτή.

Τα στοιχεία αυτά, με την ακριβή σειρά που αναφέρθηκαν, φαίνονται στις παρακάτω εικόνες. Το μενού που δώσαμε στην κλάση αυτή δίνει ακριβώς τις ίδιες δυνατότητες με αυτό της εναρκτήριας κλάσης.

5.5.2. Υλοποίηση του Συστήματος

Ο χρήστης/client έχει τη δυνατότητα να ορίσει τον τρόπο με τον οποίο επιθυμεί να γίνει η δειγματοληψία του σήματος. Για τον λόγο αυτό δημιουργήθηκαν βρόχοι επανάληψης οι οποίοι εκτελούνται ανάλογα με το χρονικό ή χωρικό διάστημα που επιλέγει ο χρήστης από μια λίστα. Τα δεδομένα ανάμεσα στις κλάσεις μεταφέρονται με την χρήση getters και setters ενώ αντίστοιχα γίνεται και χρήση intents για την μετάβαση από τη μία κλάση στην επόμενη. Τα μενο ύ επιλογών υλοποιήθηκαν με την χρήση γλώσσας XML ενώ συνδέθηκαν διαδραστικά με εργαλεία Java τα οποία είναι διαθέσιμα στους Android Developers. Η επιλογή της έκδοσης των καταγραφών σε αρχείο XML έγινε για λόγους μη περαιτέρω επιβάρυνσης του συστήματος, αφού τα αρχεία XML κρατούν μια αποθηκευμένη δομή υψηλού περιεχομένου με μικρό μέγεθος αλλά και με μεγάλη ακρίβεια. Ο τρόπος κρυπτογράφησης επιλέχθηκε να υλοποιείται με την χρήση Ιδιωτικού Κλειδιού (password) αφού θεωρείται ένας από τους καλύτερους τρόπους ασφάλειας ενός αρχείου.

Στις υποενότητες που ακολουθούν γίνεται περιγραφή των βασικών τμημάτων της εφαρμογής καθώς και των κλάσεων που απασχολούν.

5.5.2.1. Χρονική Δειγματοληψία Σήματος

Στον κώδικα που ακολουθεί, βλέπουμε τον τρόπο με τον οποίο πετύχαμε το σύστημα να κάνει χρονική δειγματοληψία της ποιότητας του σήματος που λαμβάνει η συσκευή μας. Θα πρέπει να λάβουμε υπ' όψιν μας πως οι καταγραφές που λαμβάνει το τερματικό μας έχουν τόση ακρίβεια όση μπορεί να αποδώσει το τερματικό μας.

Αρχικά θέλουμε να καταγράψουμε τη θέση στην οποία βρισκόμαστε (γεωγραφικό μήκος και πλάτος). Για να το κάνουμε αυτό, χρειαζόμαστε έναν Listener ο οποίος θα επιλέγει κάθε φορά από ποιόν provider θα αναζητήσει πληροφορίες για τη θέση. Το android χρησιμοποιεί δύο ειδών providers για τον εντοπισμό της θέσης, έναν GPS ή έναν Network provider. Στο τμήμα κώδικα που ακολουθεί βλέπουμε τόσο τον ορισμό του Location Provider όσο και τα κριτήρια που ζητάμε να ικανοποιεί.

```

/*////Update the listener, and start it ////*/
MyListener = new MyPhoneStateListener();
Tel = ( TelephonyManager
) getSystemService(Context.TELEPHONY_SERVICE);
Tel.listen(MyListener ,PhoneStateListener.LISTEN_SIGNAL_STRENGTHS);

//my location listener input
LocationManager locationManager;
String context = Context.LOCATION_SERVICE;
locationManager = (LocationManager) getSystemService(context);

//location provider criteria

Criteria criteria = new Criteria();
criteria.setAccuracy(Criteria.ACCURACY_FINE);
criteria.setAltitudeRequired(false);
criteria.setBearingRequired(false);
criteria.setCostAllowed(true);
criteria.setPowerRequirement(Criteria.POWER_LOW);
String provider = locationManager.getBestProvider(criteria, true);

```

```

Location location = locationManager.getLastKnownLocation(provider);

updateWithNewLocation(location);

locationManager.requestLocationUpdates(provider, 0, 0,
locationListener);

```

Στη συνέχεια για να έχουμε την δυνατότητα να κάνουμε χρονική δειγματοληψία επιλέγουμε να αποθηκεύσουμε μέτρηση μέσω ενός επαναλαμβανόμενου βρόχου. Ο χρήστης επιλέγει το χρονικό διάστημα το οποίο μεσολαβεί ανάμεσα στις καταγραφές. Αυτό γίνεται με τη επιλογή των δευτερολέπτων που παρεμβάλλονται ανάμεσα σε δύο καταγραφές από το μενού. Τη στιγμή που λήξει το διάστημα αυτό ο βρόχος στέλνει την καταγραφή στην βάση δεδομένων.

```

public Runnable mUpdateTimeTask = new Runnable() {

    public void run() {

        TextView writesec=(TextView)findViewById(R.id.writesec);
        TextView write=(TextView)findViewById(R.id.write);
        if ((Okbutton==true)&&(seconds>=0)) {
            neighbor();
            //hour=cal.get(Calendar.HOUR_OF_DAY);
            //min = cal.get(Calendar.MINUTE);
            //sec = cal.get(Calendar.SECOND);
            //it may be over-reacting but it takes all cases.
Must be tested in real phone

            write.setText("CINR="+ mStrength+"\n"+"RSSI="+
rssi_signal +" dBm "); //signalnew
            writesec.setText("Seconds remaining: "+seconds+".
The originally given seconds are "+ backupseconds+". The current time is:
"+ time.now());
            seconds--;
        } else {
            if((Stopbutton==true)&&(Okbutton==false)) {
                writesec.setText("termatise");
                seconds=backupseconds;
            }
            else{

                seconds=backupseconds;
                writesec.setText("The record is saved ");

                dh.insert(trackname.valueOf(TrackingName),latitudestr.valueOf(lati),l
ongtitudestr.valueOf(longti),signalstrengthstr.valueOf(rssi_signal),
cinrstr.valueOf(mStrength), time.now(), mccstr, mncstr,
String.valueOf(lac), "T",
nettype.valueOf(nettype2),cellid.valueOf(cell_idstr),"-
",namesmtx[0],cellidmtx[0],rssimtx[0],namesmtx[1],cellidmtx[1],rssimtx[1],n
amesmtx[2],cellidmtx[2],rssimtx[2],namesmtx[3],cellidmtx[3],rssimtx[3],name
smtx[4],cellidmtx[4],rssimtx[4],namesmtx[5],cellidmtx[5],rssimtx[5],namesmt
x[6],cellidmtx[6],rssimtx[6], namesmtx[7],cellidmtx[7],rssimtx[7] );

            }

        }

    }
}

```

```

        myHandler.postDelayed(this, 1000);
    }
};

```

Αναλυτικότερα, για να δούμε τον τρόπο με τον οποίο το τερματικό αναγνωρίζει και μετρά τα στοιχεία που του αναθέτουμε να αποθηκεύσει θα παραθέσουμε το αντίστοιχο τμήμα του κώδικα μαζί με μια μικρή περιγραφή στην συνέχεια του κεφαλαίου.

5.5.2.2. Μέτρηση Ισχύος Σήματος

Η ισχύς του σήματος μετράται σε dbm και για να μπορέσουμε να την μετρήσουμε θα πρέπει να φτιάξουμε έναν PhoneStateListener. Στο σύστημα βέβαια επιλέγουμε να μετρήσουμε τόσο την απλή ισχύ του σήματος όσο και τον λόγο σήματος προς παρεμβολές (CINR – Carrier to Interference Noise Ratio). Με τον ίδιο PhoneStateListener έχουμε τη δυνατότητα να εξακριβώσουμε και τον τύπο του δικτύου. Όπως αναφέραμε προηγουμένως, το δίκτυο στο οποίο βρισκόμαστε μπορεί να είναι 2G, αλλά και 3G. Όπως φαίνεται και στον κώδικα που ακολουθεί, δε γίνεται τόσο ο διαχωρισμός και η αποθήκευση του δικτύου σαν 2G ή σαν 3G όσο καταγράφεται η τεχνολογία που χρησιμοποιεί το εν λόγω δίκτυο.

Εν συνεχεία χρειαζόμαστε να αποθηκεύσουμε και τον αριθμό αναγνώρισης της κυψέλης. Ο συγκεκριμένος αριθμός είναι βασικό να αποθηκεύεται αλλά και να καταγράφεται, αφού μας δίνει μια επιπλέον οπτική για την γεωγραφική κάλυψη του δικτύου όσον αφορά τη λειτουργία των κυψελών. Αυτό το πετυχαίνουμε με τη χρήση της εντολής `loc.getCellid()`.

```

private class MyPhoneStateListener extends PhoneStateListener
{
    int networktype;
    String networkTypeString;
    GsmCellLocation loc;

    //Get the Signal strength from the provider, each time there is an update
    @Override
    public void onSignalStrengthsChanged(SignalStrength signalStrength)
    {
        networktype = Tel.getNetworkType();
        //cell_id = loc.getCid();
        //cell_idstr= new Integer(cell_id).toString();
        switch ( networktype ){
        case TelephonyManager.NETWORK_TYPE_UNKNOWN: networkTypeString = "UNKNOWN";
        break;
        case TelephonyManager.NETWORK_TYPE_GPRS:      networkTypeString = "GPRS";
        break;
        case TelephonyManager.NETWORK_TYPE_EDGE:      networkTypeString = "EDGE";
        break;
        case TelephonyManager.NETWORK_TYPE_UMTS:      networkTypeString = "UMTS";
        case TelephonyManager.NETWORK_TYPE_CDMA:      networkTypeString = "CDMA";
        break;
        case TelephonyManager.NETWORK_TYPE_EVDO_0:   networkTypeString = "EVDO_0";
        break;

```

```

case TelephonyManager.NETWORK_TYPE_EVDO_A:    networkTypeString = "EVDO_A";
break;
case TelephonyManager.NETWORK_TYPE_1xRTT:    networkTypeString = "1xRTT";
break;
case TelephonyManager.NETWORK_TYPE_HSDPA:    networkTypeString = "HSDPA";
break;
case TelephonyManager.NETWORK_TYPE_HSUPA:    networkTypeString = "HSUPA";
break;
case TelephonyManager.NETWORK_TYPE_HSPA:    networkTypeString = "HSPA";
break;
}
if (signalStrength.isGsm())
mStrength = signalStrength.getGsmSignalStrength();
else{
strength = -1;
if (signalStrength.getEvdoDbm() < 0){
strength = signalStrength.getEvdoDbm();
}
else if (signalStrength.getCdmaDbm() < 0){
strength = signalStrength.getCdmaDbm();
}
if (strength < 0){
//convert to asu
rssi_signal = Math.round((strength + 113f));
}

super.onSignalStrengthsChanged(signalStrength);
+ String.valueOf(signalStrength.getGsmSignalStrength()),
      String                                mysignal                                =
String.valueOf(signalStrength.getGsmSignalStrength());

signalnew=mysignal;
}
}

```

5.5.2.3. Χωρική Δειγματοληψία Σήματος

Όσον αφορά τη χωρική δειγματοληψία σήματος τα στοιχεία που θέλουμε να καταγράψουμε παραμένουν τα ίδια, οπότε χάριν οικονομίας δε θα αναφερθούμε στον τρόπο που υλοποιούνται. Η βασική διαφορά ανάμεσα στον κώδικα υλοποίησης της χωρικής δειγματοληψίας από τη χρονική είναι ο τρόπος με τον οποίο λαμβάνουμε τις καταγραφές.

Αρχικά ο χρήστης θέτει ένα σημείο, ως σημείο εκκίνησης. Το σημείο αυτό ορίζεται από τις συντεταγμένες γεωγραφικού μήκους και πλάτους. Στη συνέχεια, ο χρήστης ορίζει την απόσταση η οποία θέλει να παρεμβάλλεται ανάμεσα σε δύο διαδοχικές μετρήσεις του. Ο βρόχος μετράει την απόσταση από το σημείο εκκίνησης, κάθε ένα δευτερόλεπτο. Σε περίπτωση που η απόσταση είναι ίδια με αυτή που έθεσε ο χρήστης, τότε ο βρόχος λαμβάνει μια καταγραφή και ενημερώνει το αρχικό σημείο με την παρούσα θέση. Η εφαρμογή, εκτός από το χωρικό βρόχο, εκτελεί παράλληλα και ένα χρονικό βρόχο. Έτσι στην περίπτωση που ο χρήστης παραμένει στάσιμος, για συγκεκριμένο χρονικό διάστημα σε ένα σημείο ή για άλλους λόγους, και η συνθήκη απόστασης δεν ικανοποιηθεί τότε ο χωρικός βρόχος εναλλάσσεται με το χρονικό και η καταγραφή λαμβάνεται με το τέλος του χρονικού διαστήματος.

Για να μην έχουμε εκφυλισμό της χωρικής δειγματοληψίας σε χρονική, οι δύο αυτοί βρόχοι εκτελούνται παράλληλα και για την αποφυγή δυσλειτουργιών που οφείλονται στην ακρίβεια, ο βρόχος χωρικής δειγματοληψίας λαμβάνει καταγραφή σε περίπτωση που η απόσταση είναι μέσα σε ένα διάστημα τιμών σχετικής ακρίβειας με το δοθέν από τον χρήστη.

Στο τμήμα του κώδικα που ακολουθεί φαίνονται εμφανώς οι λειτουργίες που εξηγήσαμε ανωτέρω.

```
private Runnable mUpdateDistanceTask = new Runnable() {

    public void run() {

        Location aLoc = new Location("aLoc");
        aLoc.setLatitude(lati);
        aLoc.setLongitude(longti);

        Location bLoc=new Location("bLoc");
        bLoc.setLatitude(Starting_Point_Latitude);
        bLoc.setLongitude(Starting_Point_Longitude);

        temp_dist=bLoc.distanceTo(aLoc);

        TextView
        writedist3=(TextView)findViewById(R.id.write_dist);
        TextView
        writedist2=(TextView)findViewById(R.id.write_dist2);

        if ((Okbutton=true)&&(seconds>=0)) {
            neighbor();

            writedist3.setText("Longitude="+longti+"\n"+"Latitude="+lati+"\n"+"C
            INR="+mStrength+"\n"+"SignalStrength="+rssi_signal+"\n"+"Distance
            given="+distance+"\n"+"Network Type="+ nettype2);
            writedist2.setText("The distance from the starting
            point is :"+temp_dist+" and the seconds remaining "+seconds);

            if (((temp_dist>distance-
            1)&&(temp_dist<distance+1))) {
                Starting_Point_Latitude=lati;
                Starting_Point_Longitude=longti;
                //record

                dbh.insert(trackname.valueOf(TrackingName1),latitudestr.valueOf(lati)
                ,longitudestr.valueOf(longti),signalstrengthstr.valueOf(rssi_signal),cinrs
                tr.valueOf(signalnew),time.now(), mccstr, mncstr,
                String.valueOf(lac),"D",nettype.valueOf(nettype2),cellid.valueOf(cell_idstr
                ),"-",
                namesmtx[0],cellidmtx[0],rssimtx[0],namesmtx[1],cellidmtx[1],rssimtx[1],nam
                esmtx[2],cellidmtx[2],rssimtx[2],namesmtx[3],cellidmtx[3],rssimtx[3],namesm
                tx[4],cellidmtx[4],rssimtx[4],namesmtx[5],cellidmtx[5],rssimtx[5],namesmtx[
                6],cellidmtx[6],rssimtx[6], namesmtx[7],cellidmtx[7],rssimtx[7]);

            }
            else if ((seconds<=0)){
                neighbor();
                seconds=backupseconds;
                writedist2.setText("To zero and back again");
            }
        }
    }
}
```

```

        dbh.insert(trackname.valueOf(TrackingName1),latitudestr.valueOf(lati)
,longitudestr.valueOf(longti),signalstrengthstr.valueOf(rssi_signal),cinrs
tr.valueOf(signalnew),time.now(),mccstr,mncstr,
String.valueOf(lac),"D",nettype.valueOf(nettype2),cellid.valueOf(cell_idstr
),"-",
namesmtx[0],cellidmtx[0],rssimtx[0],namesmtx[1],cellidmtx[1],rssimtx[1],nam
esmtx[2],cellidmtx[2],rssimtx[2],namesmtx[3],cellidmtx[3],rssimtx[3],namesm
tx[4],cellidmtx[4],rssimtx[4],namesmtx[5],cellidmtx[5],rssimtx[5],namesmtx[
6],cellidmtx[6],rssimtx[6],namesmtx[7],cellidmtx[7],rssimtx[7]);

    }

    seconds--;
}

else {
    if (Stopbutton)
        writedist2.setText("termatise");
}

myHandler.postDelayed(this,1000);
}

};

```

5.5.2.4. Δειγματοληψία με Βάση την Ταχύτητα

Με αντίστοιχο τρόπο υλοποιείται και η εφαρμογή για την εναλλαγή του χρονικού διαστήματος της δειγματοληψίας αναλόγως με την ταχύτητα. Η εφαρμογή επικοινωνεί και ζητά ταυτόχρονα με την ανανέωση της θέσης και πληροφορίες ταχύτητας. Ανάλογα με τις τιμές που λαμβάνει ο χρήστης, επιλέγεται χρονικό διάστημα τέτοιο ώστε η απόσταση που θα έχει διανύσει το κινητό να είναι σε λογικά πλαίσια για την εν λόγω μέτρηση. Για παράδειγμα, σε περίπτωση που το τερματικό κινείται με χαμηλή ταχύτητα δεν είναι αναγκαίο να έχουμε πυκνή δειγματοληψία και αυτό γιατί δεν έχουμε δραστική αλλαγή της θέσης του κινητού. Αντιθέτως, σε περιπτώσεις υψηλών ταχυτήτων το κινητό τερματικό μεταβάλλει την θέση του με ραγδαίο ρυθμό και για το λόγο αυτό θα ήταν αναγκαία η μεσολάβηση μικρότερου διαστήματος ανάμεσα σε διαδοχικές μετρήσεις.

Ο τρόπος λειτουργίας της εφαρμογής αυτής ακολουθεί τα εξής βήματα. Αρχικά στον χρήστη εμφανίζεται η οθόνη όπου απεικονίζονται οι τρέχουσες βασικές πληροφορίες. Έτσι ο χρήστης μπορεί να ενημερωθεί για την ταχύτητά του, για την απόσταση που έχει καλύψει από το σημείο εκκίνησης μέχρι το παρόν σημείο αλλά και για το γεωγραφικό μήκος και πλάτος του. Ο χρήστης έχει τη δυνατότητα επιλογής σημείου εκκίνησης με το πάτημα του κουμπιού Set Starting Point το οποίο καθορίζει ως σημείο εκκίνησης την τρέχουσα θέση του χρήστη. Επίσης το μενού που φιλοξενεί η εφαρμογή αυτή δίνει τη δυνατότητα στον χρήστη να εκκινήσει μια καινούρια καταγραφή αλλά και να εισέλθει στην βάση δεδομένων για να παρακολουθήσει τις μετρήσεις που έγιναν.

Η δομή της εφαρμογής αυτής ακολουθεί τον τρόπο ανάπτυξης των δύο προηγούμενων κλάσεων (Χρονική και Χωρική δειγματοληψία) για το λόγο αυτό δεν θα αναφερθούμε περαιτέρω στην ανάλυση των μερών αυτών. Επιγραμματικά θα αναφέρουμε πως η εφαρμογή μπορεί να χωριστεί στα εξής μέρη:

- Προσδιορισμός Θέσης
- Μέτρηση Επιθυμητών Παραμέτρων με την Χρήση PhoneStateListener
- Χρήση Βρόχου Απόφασης για την Επιλογή του Χρονικού Μεσοδιαστήματος
- Χρήση Επαναληπτικού Βρόχου για την συνεχή καταγραφή του δικτύου.

5.5.2.5. Εξαγωγή και αποστολή Δεδομένων

Η λειτουργία και η ανάπτυξη ενός συστήματος για τους σκοπούς που υλοποιήθηκε η παρούσα διπλωματική προϋποθέτει την επιθυμία του χρήστη για αποθήκευση των μετρήσεων, αλλά και για αποστολή τους σε ανώτερη μονάδα επεξεργασίας. Για το λόγο αυτό υλοποιήσαμε μια ξεχωριστή κλάση για την διαχείριση των δεδομένων, την οποία και ονομάσαμε **ManageData.java**. Στην κλάση αυτή, ο χρήστης έχει την δυνατότητα να εξάγει τις μετρήσεις που έχει λάβει μέσω της συσκευής σε διαφορετικές μορφές αρχείων, αλλά και να αποστείλει αυτά τα αρχεία σε επιθυμητούς προορισμούς είτε χρησιμοποιώντας κρυπτογράφηση είτε όχι. Κάθε επιλογή που δίνεται στο χρήστη αντιστοιχίζεται και από το αντίστοιχο κουμπί.

Αναλυτικότερα η λειτουργίες της κλάσης αυτής είναι καταμερισμένες σε μεθόδους διαφορετικών κλάσεων. Η activity αυτή λειτουργεί ως διαχειριστής και συντονιστής των επιμέρους κλάσεων. Είναι κομβικής σημασίας αφού αποτελεί βασικό εργαλείο εξοικονόμησης πόρων συστήματος αλλά και βασικό στοιχείο για την εύκολη χρήση του συστήματος καταγραφής από τον χρήστη. Στις υποενότητες που ακολουθούν αναλύονται διαδοχικά οι επιμέρους επιλογές που έχει ο χρήστης.

5.5.2.6. Εξαγωγή Δεδομένων σε Database File.

Όπως αναφέραμε και προηγουμένως, οι μετρήσεις που λαμβάνει η συσκευή αποθηκεύονται σε μια εσωτερική βάση δεδομένων στο κινητό, η οποία είναι προσβάσιμη μόνο από το σύστημά μας και ξένη προς το υπόλοιπο τερματικό και τις επιμέρους εφαρμογές του. Για τον λόγο αυτό, σε περίπτωση που ο χρήστης θέλει να εξάγει τα δεδομένα του σε αρχείο database (.db) και να τα προσπελάσει αργότερα με εικονικό φυλλομετρητή, ειδικό για συγκεκριμένα αρχεία, αρκεί να πατήσει το πλήκτρο «Export DB to SD card». Η κλάση που εκτελείται αρχικά ελέγχει για την ύπαρξη εξωτερικής κάρτας sd στο κινητό τερματικό και ενημερώνει αντίστοιχα τον χρήστη με ανάλογο μήνυμα. Η λειτουργία της κλάσης αυτής προϋποθέτει την κλήση μεθόδου αντιγραφής της κλάσης DataHelper.java και στη συνέχεια την αποθήκευση της σε αντίστοιχο αρχείο. Έτσι την στιγμή που ο χρήστης πατήσει το εν λόγω κουμπί, θα εκτελείται το παρακάτω τμήμα κώδικα το οποίο αντιγράφει την βάση δεδομένων σε ένα αρχείο και το αποθηκεύει στην αφαιρούμενη κάρτα μνήμης.

```
private class ExportDatabaseFileTask extends
AsyncTask<String, Void, Boolean> {
private final ProgressDialog dialog = new ProgressDialog(
ManageData.this);

// can use UI thread here
protected void onPreExecute() {
this.dialog.setMessage("Exporting database...");
this.dialog.show();
}

protected Boolean doInBackground(final String... args) {

File dbFile = new File(Environment.getDataDirectory()
```

```

+ "/data/com.paad.whereami/databases/example.db");

File exportDir = new File(
Environment.getExternalStorageDirectory(), "exampledata");
if (!exportDir.exists()) {
exportDir.mkdirs();
}
File file = new File(exportDir, dbFile.getName());

try {
file.createNewFile();
this.copyFile(dbFile, file);
return true;
} catch (IOException e) {
Log.e(MyApplication.APP_NAME, e.getMessage(), e);
return false;
}
}

// can use UI thread here
protected void onPostExecute(final Boolean success) {
if (this.dialog.isShowing()) {
this.dialog.dismiss();
}
if (success) {
Toast.makeText(ManageData.this, "Export successful!",
Toast.LENGTH_SHORT).show();
} else {
Toast.makeText(ManageData.this, "Export failed",
Toast.LENGTH_SHORT).show();
}
}

void copyFile(File src, File dst) throws IOException {
FileChannel inChannel = new FileInputStream(src).getChannel();
FileChannel outChannel = new FileOutputStream(dst).getChannel();
try {
inChannel.transferTo(0, inChannel.size(), outChannel);
} finally {
if (inChannel != null)
inChannel.close();
if (outChannel != null)
outChannel.close();
}
}
}

```

5.5.2.7. Εξαγωγή των μετρήσεων σε αρχείο XML

Όπως αναφέραμε προηγουμένως, για την ευκολότερη μεταφορά των δεδομένων καθώς και για την αποθηκευσή τους δημιουργείται ένα αρχείο XML στο οποίο προστίθενται κάθε φορά οι μετρήσεις που λαμβάνει το κινητό τερματικό. Η κλάση ManageData.java που αναφέραμε προηγουμένως δίνει στον χρήστη τη δυνατότητα να κάνει μια ανάλογη εξαγωγή με το πάτημα του αντίστοιχου κουμπιού. Το κουμπί αυτό διαβάζει τη βάση δεδομένων, δημιουργεί το αρχείο XML και στέλνει το ήδη δημιουργημένο XML αρχείο σε συγκεκριμένο φάκελο της εξωτερικής κάρτας. Η δημιουργία του αρχείου XML είναι βασική όπως προαναφέραμε

αφού είναι ένας εύκολος, ορθά δομημένος και προγραμματιστικά διαχειρίσιμος τρόπος μεταφοράς δεδομένων ανάμεσα σε συστήματα. Στην παρούσα φάση ανάμεσα στον client/χρήστη και στον server.

```

public void writeheader(String path){

    try {
        File file= new File(path);
        BufferedWriter output = new BufferedWriter(new
FileWriter(file,true));
        output.write("<?xml version=\"1.0\" encoding=\"utf-
8\"?>");

        output.newLine();
        output.write("<root>");
        output.newLine();
        output.flush();
    }

    catch(Exception e){}

}

public void writing(String path, String trackname, String latitude,
String longtitude, String signalstrength, String cinr, String time, String
MCC, String MNC, String LAC, String type, String networktype, String
cellid, String speed, String Neighborcell1, String N1_cellid, String
N1_rssi, String Neighborcell2, String N2_cellid, String N2_rssi, String
Neighborcell3, String N3_cellid, String N3_rssi, String Neighborcell4,
String N4_cellid, String N4_rssi, String Neighborcell5, String N5_cellid,
String N5_rssi, String Neighborcell6, String N6_cellid, String N6_rssi,
String Neighborcell7, String N7_cellid, String N7_rssi, String
Neighborcell8, String N8_cellid, String N8_rssi ){

    try {
        File file= new File(path);
        BufferedWriter output = new BufferedWriter(new
FileWriter(file,true));
        /*output.write("<?xml version=\"1.0\"
encoding=\"utf-8\"?>");
        output.newLine();
        output.write("<root>");
        output.newLine();
        output.write("<trackname>");
        output.newLine();
        output.write("<row>");
        output.newLine();*/
        output.write("<trackname>");
        output.newLine();
        output.write("<row>");
        output.newLine();

        output.write("<title>"+trackname.valueOf(trackname)+"</title>");
        output.newLine();
        output.write("
<latitude>"+latitudestr.valueOf(latitude)+"</latitude>");
        output.newLine();
        output.write("
<longtitude>"+longtitudestr.valueOf(longtitude)+"</longtitude>");
        output.newLine();
    }
}

```

```

        output.write("
        <signalstrength>"+signalstrengthstr.valueOf(signalstrength)+"</signal
strength>");
        output.newLine();

output.write("<Cinr>"+cinrstr.valueOf(cinr)+"</Cinr>");
output.newLine();
output.write("<TimeStamp>"+time+"</TimeStamp>");
output.newLine();
output.write("<MCC>"+MCC+"</MCC>");
output.newLine();
output.write("<MNC>"+MNC+"</MNC>");
output.newLine();
output.write("<LAC>"+LAC+"</LAC>");
output.newLine();
output.write("<type>"+type+"</type>");
output.newLine();

output.write("<networktype>"+nettype.valueOf(networktype)+"</networkt
ype>");
output.newLine();

output.write("<Cell_id>"+cellid.valueOf(cellid)+"</Cell_id>");
output.newLine();
output.write("<speed>"+ speed + "</speed>");
output.newLine();

output.write("<Neighbor1>"+Neighborcell1+"</Neighbor1>");
output.newLine();
output.write("<N1cellid>"+ N1_cellid
+ "</N1cellid>");
output.newLine();
output.write("<N1rssi>"+ N1_rssi + "</N1rssi>");
output.newLine();
output.write("<Neighbor2>"+ Neighborcell2
+ "</Neighbor2>");
output.newLine();
output.write("<N2cellid>"+ N2_cellid
+ "</N2cellid>");
output.newLine();
output.write("<N2rssi>"+ N2_rssi + "</N2rssi>");
output.newLine();
output.write("<Neighbor3>"+ Neighborcell3
+ "</Neighbor3>");
output.newLine();
output.write("<N3cellid>"+
N3_cellid+"</N3cellid>");
output.newLine();
output.write("<N3rssi>"+ N3_rssi+"</N3rssi>");
output.newLine();

output.write("<Neighbor4>"+Neighborcell4+"</Neighbor4>");
output.newLine();
output.write("<N4cellid>"+N4_cellid+"</N4cellid>");
output.newLine();
output.write("<N4rssi>"+ N4_rssi + "</N4rssi>");
output.newLine();

output.write("<Neighbor5>"+Neighborcell5+"</Neighbor5>");
output.newLine();
output.write("<N5cellid>"+N5_cellid+"</N5cellid>");

```

```

        output.newLine();
        output.write("<N5rssi>"+N5_rssi+"</N5rssi>");
        output.newLine();

        output.write("<Neighbor6>"+Neighborcell6+"</Neighbor6>");
        output.newLine();
        output.write("<N6cellid>"+
N6_cellid+"</N6cellid>");
        output.newLine();
        output.write("<N6rssi>"+N6_rssi+"</N6rssi>");
        output.newLine();

        output.write("<Neighbor7>"+Neighborcell7+"</Neighbor7>");
        output.newLine();
        output.write("<N7cellid>"+N7_cellid+"</N7cellid>");
        output.newLine();
        output.write("<N7rssi>"+N7_rssi+"</N7rssi>");
        output.newLine();

        output.write("<Neighbor8>"+Neighborcell8+"</Neighbor8>");
        output.newLine();
        output.write("<N8cellid>"+N8_cellid+"</N8cellid>");
        output.newLine();
        output.write("<N8rssi>"+N8_rssi+"</N8rssi>");
        output.newLine();

        output.write("</row>");
        output.newLine();
        output.write("</trackname>");
        output.newLine();
        output.flush();
        /*output.write("</root>");
        output.newLine();
        output.flush();
*/
    } catch (Exception e) {}
}

public void writelast(String path){
    File file= new File(path);
    try {
        BufferedWriter output = new BufferedWriter(new
FileWriter(file,true));

        output.write("</root>");
        output.newLine();
        output.flush();

    } catch (IOException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    }
}
}

```

5.5.2.8. Εξαγωγή Κρυπτογραφημένου Αρχείου XML

Δίνει την επιλογή στον χρήστη να κάνει εξαγωγή ενός κρυπτογραφημένου αρχείου XML, το οποίο και επιλέξαμε να εξάγεται με βάση ένα κωδικό τον οποίο έχουμε ήδη εισάγει και ο οποίος χρησιμοποιείται για να αποκρυπτογραφήσει και το αρχείο σε επόμενη φάση. Ως πρώτο βήμα, για να αποφύγουμε τυχόν ανυπαρξία του αρχείου μέσα στην κάρτα, η εφαρμογή δημιουργεί ξανά το αρχείο και στη συνέχεια το κρυπτογραφεί. Το κρυπτογραφημένο αρχείο, για να αποφύγουμε την περίπτωση ανάγνωσης από κάποιο άλλο πρόγραμμα, του δώσαμε την κατάληξη .marios. Με την κατάληξη αυτή το αρχείο είναι ξένο προς όλα τα ήδη γνωστά προγράμματα και έτσι δεν υπάρχει πιθανότητα σύγχυσης για το χρήστη. Η κρυπτογράφηση των περιεχομένων του αρχείου βασίζεται στην ανάγνωση του αρχείου byte προς byte και η αλλαγή των τιμών τους αναλόγως με τον κωδικό που έχει τεθεί. Με τον τρόπο αυτό καθίσταται σχεδόν αδύνατο για όποιον δεν έχει το κλειδί να διαβάσει το συγκεκριμένο αρχείο. Η κρυπτογράφηση γίνεται με τον κώδικα που ακολουθεί.

```
public void encryptfile(String destpath){
    try {
        FileInputStream infile=new FileInputStream(READYTOENCRYPT);
        FileOutputStream outfile = new FileOutputStream(destpath);
        PBEKeySpec keySpec = new PBEKeySpec(PASSWORD.toCharArray());
        SecretKeyFactory keyFactory =
        SecretKeyFactory.getInstance("PBEWithMD5AndDES");
        SecretKey passwordKey = keyFactory.generateSecret(keySpec);

        byte[] salt = new byte[8];
        Random rnd = new Random();
        rnd.nextBytes(salt);
        int iterations = 100;
        PBEPParameterSpec parameterSpec = new PBEPParameterSpec(salt, iterations);

        Cipher cipher = Cipher.getInstance("PBEWithMD5AndDES");
        cipher.init(Cipher.ENCRYPT_MODE, passwordKey, parameterSpec);

        outfile.write(salt);

        byte[] input = new byte[64];
        int bytesRead;
        while ((bytesRead = infile.read(input)) != -1)
        {
            byte[] output = cipher.update(input, 0, bytesRead);
            if (output != null) outfile.write(output);
        }

        byte[] output = cipher.doFinal();
        if (output != null) outfile.write(output);

        infile.close();
        outfile.flush();
        outfile.close();
        File file= new File(READYTOENCRYPT);
        file.delete();

    } catch (Exception e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    }
}
```

5.5.2.9. Αποστολή Εξαγόμενου Αρχείου σε Server

Ένας από τους τρόπους που έχει ο χρήστης για να μοιράσει το αρχείο είναι η αποστολή σε έναν socket server τον οποίο υλοποιήσαμε και θα λειτουργεί σε ένα σταθερό τερματικό (υπολογιστή). Ο χρήστης θα πρέπει να εισάγει την IP address του server καθώς και την port στην οποία θέλει να στείλει το συγκεκριμένο αρχείο. Για λόγους ασφάλειας το συγκεκριμένο εργαλείο επιλέγει αυτόματα να στείλει το κρυπτογραφημένο αρχείο με αποτέλεσμα να απαλοίφεται η πιθανότητα επιλογής λάθους αρχείου από τον χρήστη. Για να επιτύχουμε τη λειτουργία αυτή, ενσωματώνουμε στο κινητό τερματικό μας μια κλάση client, η οποία και κάνει την αποστολή του αρχείου. Ο κώδικας της κλάσης αυτής φαίνεται παρακάτω.

```
public class TCPclient {
public static void clientsender(String IP, int Port) throws Exception {

Socket clientSocket = null;
BufferedOutputStream outToServer = null;
//here is the connection part
try {
clientSocket = new Socket(IP, Port);
outToServer = new BufferedOutputStream(clientSocket.getOutputStream());
} catch (IOException ex) {
// Do exception handling
}

if (outToServer != null) {
//pairnw to file
File myFile = new File("/sdcard/exampledata/samples_encrypted.marios");
//ftiaxnw pinaka byte iso me to mege8os tou arxeiou
byte[] mybytearray = new byte[(int) myFile.length()];
FileInputStream fis = null;

try {
fis = new FileInputStream(myFile);
} catch (FileNotFoundException ex) {
// Do exception handling
}

BufferedInputStream bis = new BufferedInputStream(fis);
try {
//diabazw ton pinaka kai stelnw to perilexomeno ston server
bis.read(mybytearray, 0, mybytearray.length);
outToServer.write(mybytearray, 0, mybytearray.length);
outToServer.flush();
outToServer.close();
clientSocket.close();

// File sent, exit the main method
return;
} catch (IOException ex) {
// Do exception handling
}
}
}
}
```

5.5.2.10. Αποστολή Εξαγόμενου Αρχείου μέσω Ηλεκτρονικού Ταχυδρομείου

Με την επιλογή αυτή, ο χειριστής της εφαρμογής μπορεί να αποστείλει με ηλεκτρονικό μήνυμα (email) το αρχείο με τις καταγραφές σε ένα λογαριασμό, ο οποίος δημιουργήθηκε με μοναδικό σκοπό τη χρήση του από τη συγκεκριμένη εφαρμογή. Αρχικά παρατηρούμε πώς ορίζονται τα απαραίτητα στοιχεία για την αποστολή του email (sender address, receiver address, sender password). Στη συνέχεια δημιουργούμε το κυρίως σώμα του μηνύματος, προσθέτοντας θέμα (subject) στο email, γράφοντας ένα ενδεικτικό κείμενο και τέλος επισυνάπτοντας το κρυπτογραφημένο αρχείο με τις μετρήσεις. Η δυνατότητα που δίνει η επιλογή αυτή στον χρήστη να μπορεί να έχει αποθηκευμένα σε ένα λογαριασμό ηλεκτρονικής αλληλογραφίας αρχεία καταγραφών ισχυροποιεί ακόμη περισσότερο το τμήμα της ασφαλούς μεταφοράς. Το τμήμα του κώδικα που υλοποιεί την εν λόγω υπηρεσία είναι:

```

public static void sendmail() {

    String host = "smtp.gmail.com";
    String from = "whereami.application@gmail.com" ;//senderstr;
    String pass = "marios.241865";
    //String filename = "/Users/marioskotsiandres/Desktop/marios.txt";
    String filename = "/sdcard/exampledata/file.xml";
    //get system properties
    Properties props = System.getProperties();
    props.put("mail.smtp.starttls.enable", "true"); // added this line
    props.put("mail.smtp.host", host);
    props.put("mail.smtp.user", from);
    props.put("mail.smtp.password", pass);
    props.put("mail.smtp.port", "587");
    props.put("mail.smtp.auth", "true");
    String[] to = {"whereami.application@gmail.com"}; // added this
line
//create a new session
    Session session = Session.getDefaultInstance(props, null);

    MimeMessage message = new MimeMessage(session);
    try {
        message.setFrom(new InternetAddress(from));
    } catch (Exception e) {
    }

    InternetAddress[] toAddress = new InternetAddress[to.length];

    // To get the array of addresses
    for (int i = 0; i < to.length; i++) { // changed from a while loop
        try {
            toAddress[i] = new InternetAddress(to[i]);
        } catch (Exception e) {
        }
    }
    for (int i = 0; i < toAddress.length; i++) { // changed from a
while loop
        try {
            message.addRecipient(Message.RecipientType.TO,
toAddress[i]);
        } catch (Exception e) {
        }
    }
    try {
        message.setSubject("sending in a group");
        //message.setText("Sent from whereami application!!!!");

```

```

//newly added code
    BodyPart messageBodyPart = new MimeBodyPart();
    messageBodyPart.setText("Here's the file");
//begin the file part
    Multipart multipart = new MimeMultipart();
    multipart.addBodyPart(messageBodyPart);

    messageBodyPart = new MimeBodyPart();

    DataSource source = new FileDataSource(filename);

    messageBodyPart.setDataHandler(new DataHandler(source));

    messageBodyPart.setFileName(filename);

    multipart.addBodyPart(messageBodyPart);

    message.setContent(multipart);

    Transport transport = session.getTransport("smtp");
    transport.connect(host, from, pass);
    transport.sendMessage(message, message.getAllRecipients());
    transport.close();
    //System.out.println("Mail Sent Successfully");

    } catch (Exception e) {
    }
}

```

5.5.2.11. Αποκρυπτογράφηση αρχείου.

Όπως αναφέραμε και στις προηγούμενες ενότητες, για λόγους ασφαλείας επιλέγουμε το αρχείο που θα μεταφερθεί είτε στον server είτε μέσω ηλεκτρονικού ταχυδρομείου να είναι κρυπτογραφημένο. Για το λόγο αυτό υλοποιήσαμε μια desktop application, στην οποία ενσωματώσαμε τόσο τον thread socket server για την μεταφορά του αρχείου όσο και έναν decryptor, ο οποίος είναι υπεύθυνος για την αποκρυπτογράφηση του. Παράλληλα στην εφαρμογή αυτή, υλοποιήσαμε και έναν parser, ο οποίος διαβάζει το αποκρυπτογραφημένο αρχείο και αποθηκεύει τις καταγραφές σε μια βάση δεδομένων. Η αποκρυπτογράφηση του αρχείου γίνεται με τον παρακάτω κώδικα.

```

import javax.crypto.*;
import javax.crypto.spec.*;
import java.io.*;

public class Decrypt {

    private static FileInputStream inFile;
    private static FileOutputStream outFile;
    private String password = "strong_pass";

    public void execute(String encryptedfilepath, String outputpath) {

        try {
            inFile = new FileInputStream(encryptedfilepath);
            outFile = new FileOutputStream(outputpath);

```

```

PBEKeySpec keySpec = new PBEKeySpec(password.toCharArray());
SecretKeyFactory keyFactory =
    SecretKeyFactory.getInstance("PBEWithMD5AndDES");
SecretKey passwordKey = keyFactory.generateSecret(keySpec);

byte[] salt = new byte[8];
inFile.read(salt);
int iterations = 100;

PBEPParameterSpec parameterSpec = new PBEPParameterSpec(salt,
iterations);

Cipher cipher = Cipher.getInstance("PBEWithMD5AndDES");
cipher.init(Cipher.DECRYPT_MODE, passwordKey, parameterSpec);

byte[] input = new byte[64];
int bytesRead;
while ((bytesRead = inFile.read(input)) != -1) {
    byte[] output = cipher.update(input, 0, bytesRead);
    if (output != null) {
        outFile.write(output);
    }
}

byte[] output = cipher.doFinal();
if (output != null) {
    outFile.write(output);
}

inFile.close();
outFile.flush();
outFile.close();

    } catch (Exception e) {
    }
}
}

```

5.5.2.12. Διάβασμα αρχείου XML.

Για την ανάγνωση του αρχείου φτιάξαμε έναν parser ο οποίος διαβάζει το xml file και αντιστοιχεί το κάθε στοιχείο (element) του xml με ένα πεδίο της βάσης δεδομένων. Η ανάγνωση του αρχείου γίνεται σειριακά από το Root Element και κατόπιν στα «παιδιά» του. Η μορφή του αρχείου xml που καλείται να διαβάσει ο παραπάνω κώδικας φαίνεται στη συνέχεια. Μέσα στην ετικέτα <row> ενσωματώνονται οι πληροφορίες που καταγράφονται από το κινητό τηλέφωνο. Για το λόγο αυτό ο parser αρκεί να ταυτίσει τις ετικέτες με τις τιμές και να τις εισάγει στην βάση δεδομένων.

```

<?xml version="1.0" encoding="utf-8" ?>
<root>
<trackname>
<row>
<title>20111011140043OUTDOORS</title>
<latitude>37.422005</latitude>
<longtitude>-122.084095</longtitude>
<signalstrength>-99</signalstrength>

```



```

<Cinr>7</Cinr>
<TimeStamp>14:01:16</TimeStamp>
<MCC>310</MCC>
<MNC>260</MNC>
<LAC>0</LAC>
<type>T</type>
<networktype>CDMA</networktype>
<Cell_id>null</Cell_id>
<speed>-</speed>
<Neighbor1>NeighgorCell_1</Neighbor1>
<N1cellid>empty</N1cellid>
<N1rssi>empty</N1rssi>
<Neighbor2>NeighgorCell_2</Neighbor2>
<N2cellid>empty</N2cellid>
<N2rssi>empty</N2rssi>
<Neighbor3>NeighgorCell_3</Neighbor3>
<N3cellid>empty</N3cellid>
<N3rssi>empty</N3rssi>
<Neighbor4>NeighgorCell_4</Neighbor4>
<N4cellid>empty</N4cellid>
<N4rssi>empty</N4rssi>
<Neighbor5>NeighgorCell_5</Neighbor5>
<N5cellid>empty</N5cellid>
<N5rssi>empty</N5rssi>
<Neighbor6>NeighgorCell_6</Neighbor6>
<N6cellid>empty</N6cellid>
<N6rssi>empty</N6rssi>
<Neighbor7>NeighgorCell_7</Neighbor7>
<N7cellid>empty</N7cellid>
<N7rssi>empty</N7rssi>
<Neighbor8>NeighgorCell_8</Neighbor8>
<N8cellid>empty</N8cellid>
<N8rssi>empty</N8rssi>
  </row>
</trackname>
<trackname>
<row>
<title>20111011140043OUTDOORS</title>
<latitude>37.422005</latitude>
<longtitude>-122.084095</longtitude>
<signalstrength>-99</signalstrength>
<Cinr>7</Cinr>
<TimeStamp>14:01:12</TimeStamp>
<MCC>310</MCC>
<MNC>260</MNC>
<LAC>0</LAC>
<type>T</type>
<networktype>CDMA</networktype>
<Cell_id>null</Cell_id>
<speed>-</speed>
<Neighbor1>NeighgorCell_1</Neighbor1>
<N1cellid>empty</N1cellid>
<N1rssi>empty</N1rssi>

```

```

<Neighbor2>NeighgorCell_2</Neighbor2>
<N2cellid>empty</N2cellid>
<N2rssi>empty</N2rssi>
<Neighbor3>NeighgorCell_3</Neighbor3>
<N3cellid>empty</N3cellid>
<N3rssi>empty</N3rssi>
<Neighbor4>NeighgorCell_4</Neighbor4>
<N4cellid>empty</N4cellid>
<N4rssi>empty</N4rssi>
<Neighbor5>NeighgorCell_5</Neighbor5>
<N5cellid>empty</N5cellid>
<N5rssi>empty</N5rssi>
<Neighbor6>NeighgorCell_6</Neighbor6>
<N6cellid>empty</N6cellid>
<N6rssi>empty</N6rssi>
<Neighbor7>NeighgorCell_7</Neighbor7>
<N7cellid>empty</N7cellid>
<N7rssi>empty</N7rssi>
<Neighbor8>NeighgorCell_8</Neighbor8>
<N8cellid>empty</N8cellid>
<N8rssi>empty</N8rssi>
  </row>
</trackname>
</root>

```

```

public class ReadXML {

    public void insert_into_db(String filePath) {
        Connection conn = new DBHelper().connect();
        try {
            File fXmlFile = new File(filePath);
            DocumentBuilderFactory dbFactory =
DocumentBuilderFactory.newInstance();
            DocumentBuilder dBuilder = dbFactory.newDocumentBuilder();
            Document doc = dBuilder.parse(fXmlFile);
            doc.getDocumentElement().normalize();
            //System.out.println("Root element : " +
doc.getDocumentElement().getNodeName());
            NodeList nList = doc.getElementsByTagName("row");
            for (int temp = 0; temp < nList.getLength(); temp++) {
                Node nNode = nList.item(temp);
                if (nNode.getNodeType() == Node.ELEMENT_NODE) {
                    Element eElement = (Element) nNode;
                    String query = "INSERT INTO samples VALUES (?, ?, ?, ?,
?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?,
?, ?, ?, ?, ?, ?, ?, ?, ?)";
                    PreparedStatement ps = conn.prepareStatement(query);
                    ps.setString(1, getTagValue("title", eElement));
                    ps.setString(2, getTagValue("latitude", eElement));
                    ps.setString(3, getTagValue("longtitude", eElement));
                    ps.setString(4, getTagValue("signalstrength",
eElement));
                    ps.setString(5, getTagValue("Cinr", eElement));
                    ps.setString(6, getTagValue("TimeStamp", eElement));
                    ps.setString(7, getTagValue("MCC", eElement));

```

```

        ps.setString(8, getTagValue("MNC", eElement));
        ps.setString(9, getTagValue("LAC", eElement));
        ps.setString(10, getTagValue("type", eElement));
        ps.setString(11, getTagValue("networktype", eElement));
        ps.setString(12, getTagValue("Cell_id", eElement));
        ps.setString(13, getTagValue("speed", eElement));
        ps.setString(14, getTagValue("Neighbor1", eElement));
        ps.setString(15, getTagValue("N1cellid", eElement));
        ps.setString(16, getTagValue("N1rssi", eElement));
        ps.setString(17, getTagValue("Neighbor2", eElement));
        ps.setString(18, getTagValue("N2cellid", eElement));
        ps.setString(19, getTagValue("N2rssi", eElement));
        ps.setString(20, getTagValue("Neighbor3", eElement));
        ps.setString(21, getTagValue("N3cellid", eElement));
        ps.setString(22, getTagValue("N3rssi", eElement));
        ps.setString(23, getTagValue("Neighbor4", eElement));
        ps.setString(24, getTagValue("N4cellid", eElement));
        ps.setString(25, getTagValue("N4rssi", eElement));
        ps.setString(26, getTagValue("Neighbor5", eElement));
        ps.setString(27, getTagValue("N5cellid", eElement));
        ps.setString(28, getTagValue("N5rssi", eElement));
        ps.setString(29, getTagValue("Neighbor6", eElement));
        ps.setString(30, getTagValue("N6cellid", eElement));
        ps.setString(31, getTagValue("N6rssi", eElement));
        ps.setString(32, getTagValue("Neighbor7", eElement));
        ps.setString(33, getTagValue("N7cellid", eElement));
        ps.setString(34, getTagValue("N7rssi", eElement));
        ps.setString(35, getTagValue("Neighbor8", eElement));
        ps.setString(36, getTagValue("N8cellid", eElement));
        ps.setString(37, getTagValue("N8rssi", eElement));

        ps.executeUpdate();
        ps.close();

        // System.out.println("Inserted: "+getTagValue("title", eElement)+",
        "+getTagValue("latitude", eElement)+", "+getTagValue("longitude",
        eElement)+", "+getTagValue("signalstrength", eElement)+",
        "+getTagValue("type", eElement)+", "+getTagValue("networktype",
        eElement)+", "+getTagValue("Cell_id", eElement)+", "+getTagValue("speed",
        eElement));
    }
    }
    conn.close();
} catch (Exception e) {
    e.printStackTrace();
}
}

private static String getTagValue(String sTag, Element eElement) {
    NodeList nList =
eElement.getElementsByTagName(sTag).item(0).getChildNodes();
    Node nValue = (Node) nList.item(0);
    return nValue.getNodeValue();
}
}
}

```

Το διάγραμμα ροής των δεδομένων μεταξύ του client (κινητού τερματικού), του server και της βάσης δεδομένων φαίνεται παρακάτω:



Ο χρήστης λαμβάνει καταγραφές με τη χρήση κάποιων από τους τρόπους που περιγράψαμε παραπάνω. Στη συνέχεια κρυπτογραφεί το αρχείο με τη χρήση κωδικού και το εξάγει σε μορφή xml. Τέλος, έχει την δυνατότητα να το στείλει είτε στον λογαριασμό αλληλογραφίας είτε σε ένα server.

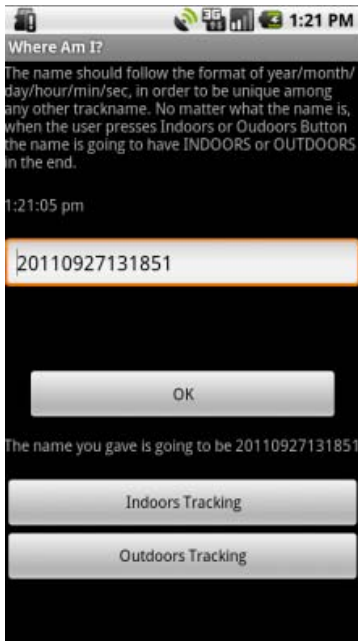


Ο sever έχει υλοποιηθεί σε μία desktop application η οποία μπορεί να εκτελεστεί σε οποιονδήποτε υπολογιστή. Για το λόγο αυτό δίνεται στο χρήστη η δυνατότητα να μπορεί να επιλέξει την διεύθυνση IP καθώς και την port στην οποία θα στείλει το αρχείο του. Παράλληλα με τον server, μπορεί να εκτελεστεί και ο αποκωδικοποιητής, ο οποίος διαβάζει το κρυπτογραφημένο αρχείο και το αποκρυπτογραφεί αποθηκεύοντας το στην επιφάνεια εργασίας.

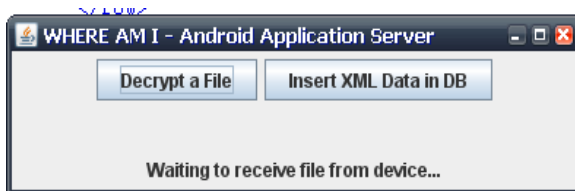


Στη συνέχεια γίνεται εισαγωγή των περιεχομένων του αποκρυπτογραφημένου αρχείου στη βάση δεδομένων. Η βάση είναι εγκατεστημένη σε υπολογιστή και είναι προσβάσιμη μέσω του terminal του υπολογιστή. Οποιαδήποτε επεξεργασία στις καταγραφές μπορεί να γίνει εκεί με την χρήση κατάλληλων εντολών SQL.

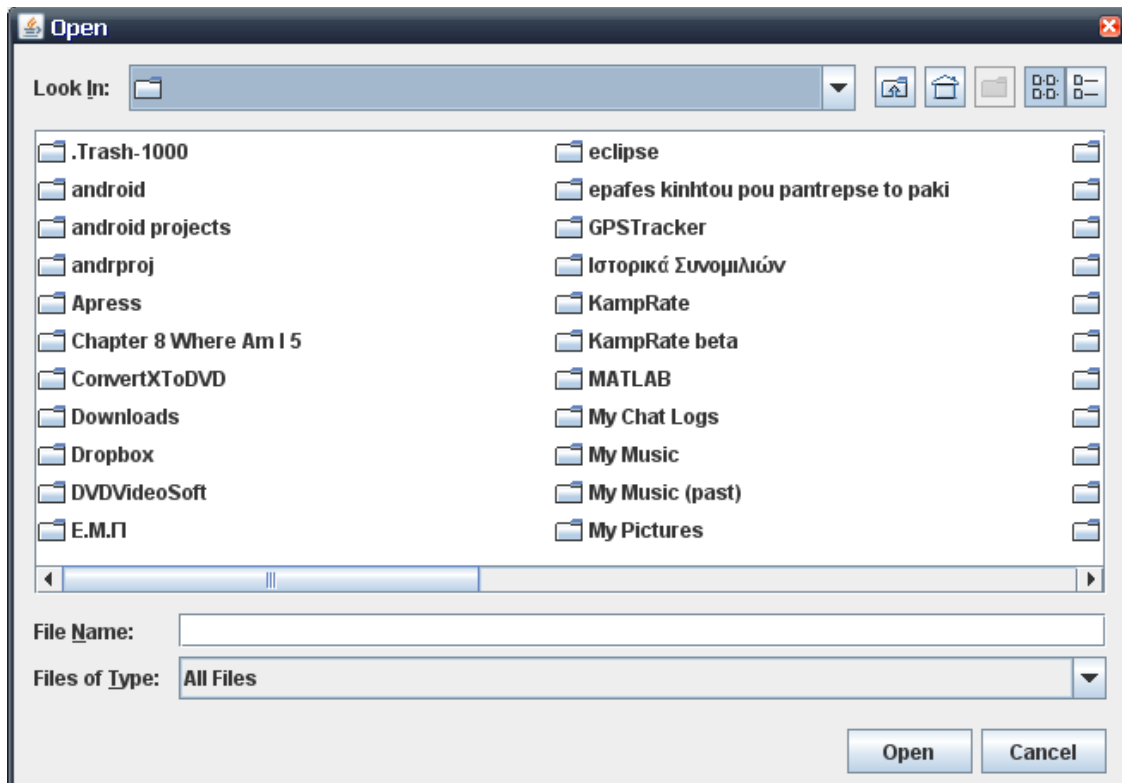
Στην περίπτωση που ο χρήστης εκκινήσει μια νέα καταγραφή, όπως αυτή που φαίνεται στην παρακάτω εικόνα, δημιουργείται ένα αρχείο xml με τις καταγραφές. Το αρχείο αυτό κρυπτογραφείται και αποστέλλεται στον server. Η μορφή του αρχείου φαίνεται στην ανάλυση της κλάσης XMLReader.java. Ο χρήστης εκκινεί την Desktop application την οποία δημιουργήσαμε και εκτελεί με τη σειρά τα επόμενα βήματα.



Το όνομα που δόθηκε είναι 20110927131851. Μόλις ο χρήστης πατήσει να μεταβεί στην επόμενη κλάση, είτε μέσω του Indoors είτε μέσω του Outdoors, το όνομα της καταγραφής γίνεται 20110927131851OUTDOORS ή INDOORS. Στο συγκεκριμένο παράδειγμα το όνομα της καταγραφής είναι 20110927131851OUTDOORS. Με τον τρόπο αυτό γίνεται αποστολή της πληροφορίας για τον τύπο καταγραφής, χωρίς επιπλέον επιβάρυνση της βάσης και τις εφαρμογής μας. Στη συνέχεια και ενώ ο χρήστης δειγματοληπτει το δίκτυο κινητής τηλεφωνίας για αόριστο χρονικό διάστημα, τερματίζει την εφαρμογή και την αποστέλλει το αρχείο στον server.



Μόλις το αρχείο φτάσει στον server, ο χρήστης με την επιλογή του κουμπιού Decrypt a File μπορεί να επιλέξει το επιθυμητό αρχείο και να το αποκρυπτογραφήσει. Η επιλογή θα γίνει από ένα παράθυρο περιήγησης των φακέλων του συστήματος. Στη συνέχεια με αντίστοιχο τρόπο επιλέγει το αποκρυπτογραφημένο αρχείο και το εισάγει στην βάση δεδομένων.



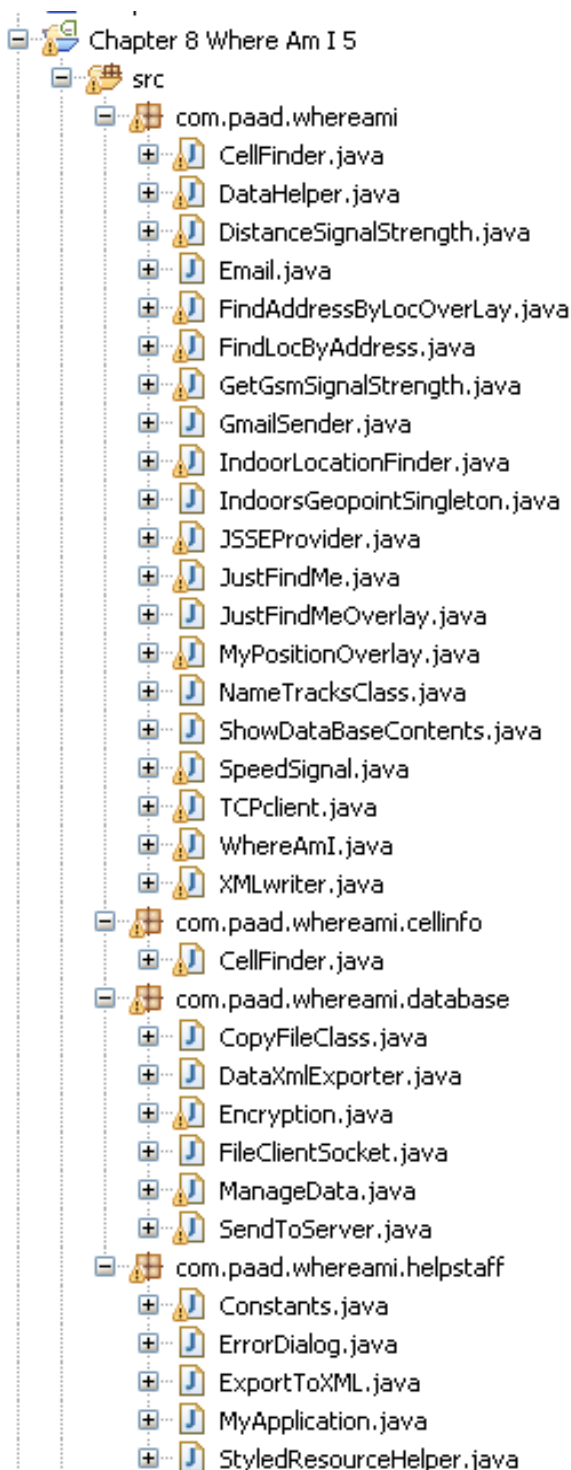
Έτσι αν κοιτάξουμε, ελέγξουμε τα περιεχόμενα της βάσης δεδομένων θα παρατηρήσουμε πως υπάρχουν περιεχόμενα με τις εν λόγω καταγραφές. Όπως φαίνεται και στην παρακάτω εικόνα. (Λόγω μεγάλου μεγέθους της βάσης δεδομένων επιλέξαμε να δείξουμε μια συμπτυγμένη εκδοχή της).

trackname	latitude	longitude	signal_strength	cinr	type	network_type	cell_id	speed
201109271318510UTD00RS	21.49749	67.59746	68.7	8	T	CDMA	0X731	-
201109271318510UTD00RS	21.49759	67.59756	68.7	8	T	CDMA	0X731	-
201109271318510UTD00RS	21.49769	67.59766	68.7	8	T	CDMA	0X731	-
201109271318510UTD00RS	21.49779	67.59776	68.7	8	T	CDMA	0X731	-
201109271318510UTD00RS	21.49789	67.59786	68.7	8	T	CDMA	0X731	-
201109271318510UTD00RS	21.49789	67.59786	68.7	8	T	CDMA	0X731	-
20110927132637IND00RS	21.49749	67.59746	68.7	9	T	GPRS	0X737	-
20110927132637IND00RS	21.49749	67.59746	68.7	9	T	GPRS	0X737	-
20110927132637IND00RS	21.49749	67.59746	68.7	9	T	GPRS	0X737	-
20110927132637IND00RS	21.49749	67.59746	68.7	9	T	GPRS	0X737	-
20110927132637IND00RS	21.49749	67.59746	68.7	9	T	GPRS	0X737	-
20110927132637IND00RS	21.49749	67.59746	68.7	9	T	GPRS	0X737	-
20110927132637IND00RS	21.49749	67.59746	68.7	9	T	GPRS	0X737	-
20110927132637IND00RS	21.49749	67.59746	68.7	9	T	GPRS	0X737	-
20110927132637IND00RS	21.49749	67.59746	68.7	9	T	GPRS	0X737	-
20110927132637IND00RS	21.49749	67.59746	68.7	9	T	GPRS	0X737	-
20110927132637IND00RS	21.49749	67.59746	68.7	9	T	GPRS	0X737	-
20110927132637IND00RS	21.49749	67.59746	68.7	9	T	GPRS	0X737	-
20110927132637IND00RS	21.49749	67.59746	68.7	9	T	GPRS	0X737	-
20110927132637IND00RS	21.49749	67.59746	68.7	9	T	GPRS	0X737	-
20110927132637IND00RS	21.49749	67.59746	68.7	9	T	GPRS	0X737	-
20110927132637IND00RS	21.49749	67.59746	68.7	9	T	GPRS	0X737	-

20 rows in set (0.00 sec)

5.6. Διάρθρωση Συστήματος

Στην ενότητα αυτή γίνεται ανάλυση των κυριότερων κλάσεων που απαρτίζουν το σύστημα καταγραφής, καθώς και περιγραφή του τρόπου με τον οποίο συνεργάζονται για την ολοκλήρωση της διαδικασίας δειγματοληψίας και την αποθήκευσης των μετρήσεων στην βάση δεδομένων.



Η κλάση **JustFindMe.java** υλοποιεί την αρχική activity του συστήματος. Εμφανίζει την τρέχουσα θέση του χρήστη, ενώ με την χρήση μεθόδων της κλάσης **JustFindMeOverlay.java** γίνεται εφικτή η γραφική απεικόνιση του σημείου που βρίσκεται ο χρήστης, με την χρήση μιας πράσινης κοιλίδας. Η κλάση **JustFindMe** φιλοξενεί και το μενού επιλογών, από το οποίο ο χρήστης μπορεί να εκκινήσει είτε μια νέα καταγραφή, να δει δεδομένα για τις γειτονικές κυψέλες, να εκκινήσει την εφαρμογή **Address**, καθώς και να μεταβεί στην κλάση εξαγωγής δεδομένων. Επίσης, ο χρήστης μπορεί να επιλέξει την εκκίνηση εφαρμογής εντοπισμού θέσης, η οποία υλοποιείται με την κλάση **FindLocByAddress.java**.

Η κλάση **NameTracksClass.java**, εκτελείται μόλις ο χρήστης επιλέξει την εκκίνηση μιας νέας καταγραφής. Η εκκίνηση της προβάλλει ένα παράθυρο διαλόγου στον χρήστη, ο οποίος επιλέγει τον τρόπο με τον οποίο θέλει να ονομάσει τη νέα καταγραφή. Η επιλογή αυτή όπως αναλύθηκε και προηγουμένως περιλαμβάνει δύο εκδοχές. Η πρώτη εκδοχή είναι η ονοματοδοσία με αυτόματα δημιουργούμενο όνομα, ενώ η δεύτερη είναι η ονοματοδοσία με όνομα το οποίο εισάγει ο χρήστης. Όπως είναι φανερό, λόγω του μεγάλου πλήθους καταγραφών είναι υπαρκτή η πιθανότητα για σύγχυση ανάμεσα στα ονόματα των καταγραφών. Για το λόγο αυτό, είναι προτιμότερη η χρήση του αυτόματου ονόματος. Ένα επιπλέον χαρακτηριστικό είναι η δυνατότητα του χρήστη για επιλογή του περιβάλλοντος καταγραφής. Ο χρήστης μπορεί να χαρακτηρίσει τις καταγραφές ως εσωτερικού χώρου ή ως εξωτερικού. Με την επιλογή των πλήκτρων **Indoors** ή **Outdoors**, η πληροφορία αυτή εντάσσεται στο όνομα της καταγραφής, προσθέτοντας την κατάλληλη **INDOORS** ή **OUTDOORS** αντίστοιχα στο όνομα.

Η κλάση **WhereAmI.java** εμφανίζεται αμέσως μετά την **NameTracksClass.java**, με την επιλογή **Indoors** ή **Outdoors** από τον χρήστη. Εμφανίζει ένα χάρτη όπου ο χρήστης μπορεί να ενημερωθεί για την τρέχουσα τοποθεσία του, όπως γίνεται και στην **JustFindMe.java**, με διαφορά όμως την προσθήκη μιας μικρής πυξίδας στην πάνω αριστερή γωνία της οθόνης, η οποία λειτουργεί χρησιμοποιώντας τους αισθητήρες του κινητού τηλεφώνου. Ο χρήστης μπορεί να αλλάξει τον τρόπο εμφάνισης του χάρτη είτε με την απεικόνιση της περιοχής με χρήση αεροφωτογραφιών από δορυφόρο, είτε με την εμφάνιση της ως οδικού χάρτη. Επίσης η **activity** φιλοξενεί και ένα μενού επιλογών, όπου ο χρήστης καλείται να επιλέξει τον τύπο δειγματοληψίας με τον οποίο θέλει να κάνει μετρήσεις. Οι δυνατοί τρόποι δειγματοληψίας είναι :

- Χρονική Δειγματοληψία.
- Χωρική Δειγματοληψία.
- Δειγματοληψία ανάλογη της ταχύτητας.

Η κάθε μία από αυτές τις επιλογές, υλοποιείται με αντίστοιχες **activities** οι οποίες αναλύονται στη συνέχεια. Εκτός από την επιλογή του τρόπου δειγματοληψίας, ο χρήστης μπορεί να ενημερωθεί ξανά για τα δεδομένα των γειτονικών κυψελών όπως αυτά αναλύονται στην κλάση **CellFinder.java**.

Η κλάση **GetGsmSignalStrength.java** χρησιμοποιείται για τη χρονική δειγματοληψία του σήματος και την αποθήκευση των καταγραφών στη βάση δεδομένων. Αναλυτικότερα, η κλάση λαμβάνει τις μετρήσεις με την ολοκλήρωση ενός χρονικού διαστήματος, το οποίο επιλέγει ο χρήστης από μια λίστα προτεινόμενων επιλογών. Η κλάση αυτή φιλοξενεί έναν **location provider**, ο οποίος ενημερώνει την κλάση για την τρέχουσα θέση του χρήστη ανα πάσα στιγμή. Με τον τρόπο αυτό γίνεται εφικτός ο έγκαιρος αλλά και το δυνατόν ακριβής εντοπισμός της θέσης του χρήστη με την απεικόνιση του γεωγραφικού μήκους και πλάτους του. Εκτός όμως από τον προσδιορισμό της θέσης του χρήστη, η κλάση αυτή έχει τα απαραίτητα εργαλεία για τον καθορισμό και των απαραίτητων παραμέτρων. Για τον καθορισμό κάθε μίας από τις παραμέτρους χρησιμοποιήθηκαν μέθοδοι που υλοποιήθηκαν και εντάχθηκαν στον κυρίως σώμα της **activity**. Με τον τρόπο αυτό, γίνεται εφικτή η παράλληλη καταγραφή στοιχείων σήματος αλλά και γενικότερων τεχνολογικών πληροφοριών.

Πιο αναλυτικά, όσον αφορά τα χαρακτηριστικά του σήματος γίνεται καταγραφή τόσο της ισχύος σήματος που λαμβάνει το κινητό, η οποία μετράται σε **dbm**, όσο και του λόγου **CINR** (**Carrier to Interference Noise Ratio**), ο οποίος είναι εξαιρετικά σημαντικός για το δίκτυο αφού δηλώνει τον λόγο φέροντος σήματος προς τον θόρυβο που προκαλείται από παρεμβολές. Όσον αφορά τις υπόλοιπες παραμέτρους καταγραφής γίνεται εντοπισμός της ταυτότητας της κυψέλης, **Cellid**, αφού είναι βασική η αντιστοίχιση της καταγραφής με την κυψέλη στην οποία έγινε. Γίνεται επίσης αποθήκευση του τύπου της καταγραφής, δηλαδή του είδους της δειγματοληψίας. Όπως αναφέραμε κάθε καταγραφή μπορεί να γίνει με έναν από τους τρεις τρόπους δυναμικής δειγματοληψίας. Για το λόγο αυτό κρίθηκε σκόπιμο να υπάρχει αποθηκευμένη η πληροφορία αυτή για τον χρήστη με σκοπό την καλύτερη επεξεργασία των καταγραφών. Έτσι η χρονική δειγματοληψία απεικονίζεται με την εισαγωγή στο πεδίο **type** της **database** ενός χαρακτήρα «T».

Η κλάση **DistanceSignalStrength.java** εκτελείται για τη χωρική δειγματοληψία του σήματος. Λειτουργεί με τρόπο ανάλογο με αυτόν της **GetGsmSignalStrength.java** και αποθηκεύει τα περιεχόμενα των καταγραφών αυτόματα στην βάση δεδομένων. Οι

παράμετροι που καλείται η κλάση αυτή να καταγράψει και να αποθηκεύσει είναι οι ίδιες, με μόνη διαφορά πως υπάρχει διαφορετικός τρόπος για τη λήψη τους. Η καταγραφές πλέον γίνονται με την ολοκλήρωση μια καθορισμένη απόστασης την οποία επιλέγει ο χρήστης στην εκκίνηση της συγκεκριμένης activity. Ανά πάσα στιγμή ο χρήστης μπορεί να δει στην οθόνη του κινητού του, όλες τις παραμέτρους όπως αυτές αναφέρθηκαν στην κλάση χρονικής δειγματοληψίας. Η διαφορά ανάμεσα στα εισερχόμενα στοιχεία στις καταγραφές έγκειται στην είσοδο της τιμής «D» στο πεδίο type.

Βέβαια για λόγους ορθή λειτουργίας του συστήματος, χρησιμοποιήθηκε ως ασφαλιστική δικλείδα η χρήση χρονικού μετρητή ο οποίος εξαναγκάζει το σύστημα στην λήψη καταγραφής μετά από την πάροδο ενός χρονικού διαστήματος όπου δεν θα δικαιολογεί την αδράνεια του χρήστη. Έτσι αναλυτικότερα τη στιγμή που ο χρήστης επιλέγει το χωρικό διάστημα, το οποίο θέλει να παρεμβάλεται ανάμεσα σε δύο καταγραφές, η κλάση αυτόματα ενημερώνει το σύστημα για τον μέγιστο επιτρεπτό χρόνο, που έχει ο χρήστης για να διανύσει το συγκεκριμένο διάστημα.

Έτσι για παράδειγμα, σε περίπτωση επιλογής 10m από τον χρήστη υπάρχει η πρόληψη για την πάροδο 15 δευτερολέπτων μέχρι την ολοκλήρωση του διαστήματος. Σε περίπτωση τερματισμού του χρονικού μετρητή αλλά μη ολοκλήρωσης του χωρικού διαστήματος, το σύστημα εξαναγκάζεται σε λήψη μέτρησης και ταυτόχρονα σε ενημέρωση των παραμέτρων αλλά και της θέσης του χρήστη. Ο λόγος ύπαρξης αυτού του χρονικού περιορισμού έγκειται στην προσπάθεια για διαφύλαξη της συνέχειας των μετρήσεων. Έτσι σε περίπτωση όπου ο χρήστης παραμένει στάσιμος για μεγάλο χρονικό διάστημα το τερματικό δεν σταματά να λαμβάνει μετρήσεις αλλά παρ' όλα αυτά συνεχίζει δίνοντας έτσι κατά την επεξεργασία των μετρήσεων μια ολοκληρωμένη εικόνα για την συγκεκριμένη καταγραφή.

Η κλάση **SpeedSignal.java** χρησιμοποιείται στην περίπτωση δειγματοληψίας με βάση την ταχύτητα του κινητού τερματικού. Εκτελείται όταν ο χρήστης πατήσει το κουμπί Speed στο menu της κλάσης WhereAmI.java. Ειδικότερα η κλάση SpeedSignal.java προέκυψε από την προσπάθεια καλύτερης δειγματοληψίας της ποιότητας του δικτύου που αντιλαμβάνεται ένας χρήστης κατά τη διάρκεια κίνησης του με μεγάλη ταχύτητα. Λόγω της συνεχούς χρήσης κινητού κατά τη διάρκεια της οδήγησης, κρίθηκε αναγκαία η υλοποίηση ενός συστήματος όπου η δειγματοληψία του σήματος θα γίνεται με απόφαση που θα λαμβάνεται από το σύστημα με βάση της τρέχουσα ταχύτητα του κινητού.

Αναλυτικότερα, σε περιπτώσεις όπου χρήστης και τερματικό κινούνται με μεγάλες ταχύτητες είναι βασικό η απόσταση δύο διαδοχικών καταγραφών να είναι χρονικά μικρότερη από αυτή που θα υπάρχει σε περίπτωση κίνησης με μικρή ταχύτητα. Έτσι έχει προβλεφθεί η αυτόματη επιλογή του χρονικού διαστήματος που θα παρεμβάλεται αναλόγως με την ταχύτητα του κινητού. Έτσι σε περίπτωση που το τερματικό κινείται με ταχύτητα 23m/sec οι καταγραφές λαμβάνονται ανα δευτερόλεπτο. Σε περίπτωση αλλαγής της ταχύτητας το χρονικό διάστημα αυξομειώνεται ανάλογα. Οι καταγραφές που προκύπτουν από την κλάση αυτή δεικτοδοτούνται με την εισαγωγή της λέξης «speed» στο πεδίο type της βάσης δεδομένων. Οι υπόλοιπες καταγραφόμενες παράμετροι είναι οι ίδιες με αυτές που έχουν περιγραφεί και στις δύο προηγούμενες κλάσεις και για λόγους οικονομίας δεν θα επαναληφθούν.

Η κλάση **ShowDataBaseContents.java** χρησιμοποιείται για την εμφάνιση του περιεχομένου της εσωτερικής βάσης δεδομένων του κινητού. Τα δεδομένα εμφανίζονται σε μορφή λίστας, ενώ ο χρήστης ενημερώνεται για την αντιστοιχία των τιμών που βλέπει με τα πεδία που αντιστοιχούν. Αναλυτικότερα, η κλάση αυτή εκτελεί έναν βρόχο παρουσίασης των ήδη καταγεγραμμένων παραμέτρων με μορφή λίστας. Χρησιμοποιώντας τις μεθόδους που

βρίσκονται στην κλάση **DataHelper.java** ο βρόχος αυτός «ανοίγει» την βάση δεδομένων του κινητού τηλεφώνου και διαβάζοντας τα περιεχόμενα τα απεικονίζει σε μορφή λίστας. Ο χρήστης έχει τη δυνατότητα να δει ολόκληρη τη λίστα αλλά και να διαγράψει τη λίστα σε περίπτωση που θέλει να αδειάσει την εσωτερική βάση δεδομένων του συστήματος καταγραφής από παλιές καταγραφές. Επίσης ο χρήστης με το μενού της κλάσης αυτής μπορεί να μεταβεί στην επόμενη κλάση όπου γίνεται η εξαγωγή και η διαχείριση των δεδομένων. Επιπλέον ο χρήστης έχει τη δυνατότητα να εκκινήσει μια νέα καταγραφή αλλά και να εκμεταλλευτεί όλες τις δυνατότητες που του δόθηκαν και στην αρχική κλάση. Ο λόγος που επιλέχθηκε αυτή η επανάληψη ήταν κυρίως για λόγους καλύτερης χρήσης της εφαρμογής και για ευκολία του χρήστη.

Η κλάση **CellFinder.java** εμφανίζει σε μορφή λίστας όλες τις γειτονικές κυψέλες με τις οποίες επικοινωνεί το κινητό τερματικό καθώς και βασικά στοιχεία για κάθε μία από αυτές, όπως είναι η ταυτότητα τους αλλά και η ισχύς του σήματος που λαμβάνει το κινητό από αυτές. Αναλυτικότερα τα στοιχεία που παρέχει αυτή η κλάση στον χρήστη είναι:

- Μία λίστα με τις γειτονικές κυψέλες κινητής τηλεφωνίας της κυψέλης που είναι συνδεδεμένος ο χρήστης.
- Την τιμή της ισχύος του σήματος που λαμβάνει το τερματικό από τις κυψέλες αυτές μετρούμενη σε dbm.
- Καταγραφή του Location Area Code για κάθε μία από τις γειτονικές κυψέλες αλλά και για την παρούσα.

Με την κλάση αυτή ο χρήστης μπορεί να παρακολουθήσει το περιβάλλον στο οποίο γίνεται η μέτρηση συγκρίνοντας την ποιότητα σήματος που δέχεται από κάθε σταθμό βάσης.

Η κλάση **DataHelper.java** χρησιμοποιείται για την εισαγωγή των μετρήσεων στην εσωτερική βάση δεδομένων του κινητού. Η βάση δεδομένων έχει υλοποιηθεί με την χρήση SQLite γλώσσας προγραμματισμού. Ειδικότερα η κλάση αυτή είναι υπεύθυνη για όλες τις διαδικασίες διαχείρισης της εσωτερικής βάσης δεδομένων του τερματικού. Με μεθόδους που αναλύσαμε στην ενότητα περιγραφής του κώδικα γίνεται αντιληπτή η σημασία ύπαρξης μιας κλάσης με ρόλο διαχειριστή για την σύνδεση επεξεργασία και διαχείριση της βάσης.

Η κλάση **FindLocByAddress.java** μαζί με την κλάση **FindLocByAddressOverlay.java** χρησιμοποιούνται για την εύρεση του στίγματος μιας τοποθεσίας στο χάρτη με την είσοδο από το χρήστη της διεύθυνσης. Αναλυτικότερα η κλάση αυτή υλοποιήθηκε έτσι ώστε ο χρήστης να μπορεί να βρει το στίγμα μιας τοποθεσίας στον χάρτη περίπου όπως γίνεται και με την υπηρεσία GoogleMaps. Για τον εντοπισμό της θέσης ο χρήστης καλείται να εισάγει Χώρα, Πόλη, Οδό και αριθμό σε ένα παράθυρο διαλόγου και μετά να επιβεβαιώσει τα δεδομένα πατώντας OK. Η αμέσως επόμενη ενέργεια που εκτελείται είναι ο εντόπισμός και η απεικόνιση της τοποθεσίας στον χάρτη με την χρήση εικονιδίου Android.

Η κλάση **ManageData.java** χρησιμοποιείται για την διαχείριση των δεδομένων που έχει καταγράψει ο χρήστης. Αναλυτικότερα χρησιμοποιεί (καλεί) μεθόδους και άλλες κλάσεις για την διαχείριση των δεδομένων. Κάποιες από αυτές τις κλάσεις που χρησιμοποιούνται είναι οι **ExportToXML.java**, η **SendToServer.java**, καθώς και η **TCPClient.java**. Το User Interface της κλάσης αυτής περιλαμβάνει τα κουμπιά επιλογής όπου ο χρήστης με την χρήση τους μπορεί να επεξεργαστεί τις καταγεγραμμένες μετρήσεις με τον τρόπο που θέλει. Οι επιλογές που του δίνονται είναι:

- Να αντιγράψει τη βάση δεδομένων του κινητού και να την εξάγει στην sd card με τη μορφή database file (.db).

- Να εξάγει τα δεδομένα με την μορφή XML αρχείου (.xml) με τη χρήση της κλάσης **ExportToXML.java**.
- Να εξάγει τα δεδομένα με μορφή encrypted XML αρχείου (.marios) με τη χρήση της κλάσης **Encryption.java**.
- Να κάνει αποστολή του κρυπτογραφημένου αρχείου στον server με τη χρήση της κλάσης **TCPClient.java**.
- Να την αποστείλει μέσω ηλεκτρονικού ταχυδρομείου με τη χρήση **Email.java**.

Η κλάση **Encryption.java** χρησιμοποιείται για την κρυπτογράφηση του αρχείου XML πριν την αποστολή του. Βασίζεται στην χρήση κωδικού ασφαλείας ο οποίος αλλάζει τις τιμές των byte του αρχείου αλλά και την επέκτασή του, κάνοντας ανέφικτο για κάποιον μη εξουσιοδοτημένο χρήστη να το ανοίξει αλλά και να το διαβάσει. Η κλάση αυτή σε συνεργασία με την κλάση **TCPClient.java** είναι ο ακρογωνιαίος λίθος της ασφάλειας του συστήματός μας. Η συνεργασία των μεθόδων των δύο κλάσεων αυτών μέσω του διαχειριστικού ρόλου της κλάσης **ManageData.java** δίνει τη δυνατότητα της επικοινωνίας και της συνεργασίας του τερματικού με μια απομακρυσμένη μονάδα επεξεργασίας.

Τέλος, για να αναφερθούμε στην πλευρά του Server, το σύστημα περιλαμβάνει μια **Java Desktop Application**. Στην εφαρμογή αυτή ενθυλακώνεται ένας thread socket server ο οποίος από τη στιγμή εκκίνησης της Application αναμένει τη λήψη αρχείου από τον client. Επελέγη η δημιουργία thread server και όχι απλού server για λόγους εξέλιξης του συστήματός μας. Έτσι σε περίπτωση που έχουμε πολλούς clients (συστήματα καταγραφής) δεν θα χρειαστεί εξέλιξη του συστήματος υποδοχής των μηνυμάτων. Επίσης η ενσωμάτωση του αποκρυπτογράφου στην ίδια εφαρμογή με τον server δίνει ακαιρεότητα στο σύστημα και ευκολία χρήσης. Η βάση δεδομένων που υλοποιήθηκε στην πλευρά του χρήστη πραγματοποιήθηκε με την χρήση της γλώσσας προγραμματισμού SQL. Η επικοινωνία με την βάση γίνεται μέσω του terminal του υπολογιστή, μέσω του οποίου ο χρήστης μπορεί να επεξεργαστεί τις καταγραφές με χρήση queries της γλώσσας SQL.

6. Παράρτημα - Πηγαίος Κώδικας Εφαρμογής

JustFindMe.java

```

package com.paad.whereami;

import java.io.IOException;
import java.util.List;
import java.util.Locale;

import android.app.AlertDialog;
import android.content.Context;
import android.content.DialogInterface;
import android.content.Intent;
import android.location.Address;
import android.location.Criteria;
import android.location.Geocoder;
import android.location.Location;
import android.location.LocationListener;
import android.location.LocationManager;
import android.os.Bundle;
import android.view.Menu;
import android.view.MenuInflater;
import android.view.MenuItem;
import android.view.View;
import android.view.ViewGroup.LayoutParams;
import android.widget.LinearLayout;
import android.widget.TextView;

import com.google.android.maps.GeoPoint;
import com.google.android.maps.MapActivity;
import com.google.android.maps.MapController;
import com.google.android.maps.MapView;
import com.google.android.maps.Overlay;

public class JustFindMe extends MapActivity{

    MapView mapView;
    MapController mapController;
    JustFindMeOverlay positionOverlay;

    public boolean onCreateOptionsMenu(Menu export_menu){
        super.onCreateOptionsMenu((android.view.Menu) export_menu);
        MenuInflater m_signal = getMenuInflater();
        m_signal.inflate(R.menu.export_menu, (android.view.Menu)
export_menu);
        return true;
    }

    public boolean onOptionsItemSelected(MenuItem item){
        switch(item.getItemId()){
            case R.id.Export_button:
                startActivity(new
Intent("com.paad.whereami.database.MANAGEDATA"));

```

```

        return true;
    case R.id.CellFinder_button:
        startActivity(new
Intent("com.paad.whereami.cellinfo.CELLFINDER"));
        return true;
    case R.id.Move_to_Show_data:
        startActivity(new
Intent("com.paad.whereami.SHOWDATABASECONTENTS"));
        return true;
    case R.id.about0721:
        AlertDialog          alert          =          new
AlertDialog.Builder(this).create();
        alert.setTitle("App Info");
        alert.setMessage("This applcation is created by Marios
Kotsiandris in order to measure the RSSI of mobile communication company.
Completed Date:2010/2011 Copyrights reserved by Kotsiandris Marios.");
        alert.setButton("OK",          new
DialogInterface.OnClickListener() {
            public void onClick(DialogInterface dialog, int
which) {
                // here you can add functions
            }
        });
        alert.show();
    case R.id.whereami0721:
        startActivity(new
Intent("com.paad.whereami.FINDLOCBYADDRESS"));
        return true;
    case R.id.NameClassIntent:
        startActivity(new
Intent("com.paad.whereami.NAMETRACKSCLASS"));
    }
    return(super.onOptionsItemSelected(item));
}

@Override
protected boolean isRouteDisplayed() {
    // TODO Auto-generated method stub
    return false;
}

@Override
protected void onPause() {
    // TODO Auto-generated method stub
    super.onPause();
}

@Override
protected void onResume() {
    // TODO Auto-generated method stub
    super.onResume();
}

/** Called when the activity is first created. */
@Override
public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.findmeindex);

    mapView = (MapView) findViewById(R.id.mymMapView);
    mapController = mapView.getController();

```

```

        LinearLayout zoomLayout =
        (LinearLayout)findViewById(R.id.zoom);
        View zoomView = mapView.getZoomControls();
        mapView.setStreetView(true);

        zoomLayout.addView(zoomView,
            new LinearLayout.LayoutParams(
                LayoutParams.WRAP_CONTENT,
                LayoutParams.WRAP_CONTENT));
        mapView.displayZoomControls(true);

        mapController.setZoom(17);

        // Add the MyPositionOverlay
        positionOverlay = new JustFindMeOverlay();
        List<Overlay> overlays = mapView.getOverlays();
        overlays.add(positionOverlay);

        LocationManager locationManager;
        String context = Context.LOCATION_SERVICE;
        locationManager = (LocationManager) getSystemService(context);

        //eisagw tis proypo8eseis pou 8elw na exei o gps provider
        Criteria criteria = new Criteria();
        criteria.setAccuracy(Criteria.ACCURACY_FINE);
        criteria.setAltitudeRequired(false);
        criteria.setBearingRequired(false);
        criteria.setCostAllowed(true);
        criteria.setPowerRequirement(Criteria.POWER_LOW);
        String provider = locationManager.getBestProvider(criteria,
true);

        Location location =
        locationManager.getLastKnownLocation(provider);

        updateWithNewLocation(location);

        locationManager.requestLocationUpdates(provider, 10, 0,
locationListener);
    }

    private final LocationListener locationListener = new
LocationListener() {
        public void onLocationChanged(Location location) {
            updateWithNewLocation(location);
        }

        public void onProviderDisabled(String provider){
            updateWithNewLocation(null);
        }

        public void onProviderEnabled(String provider){ }
        public void onStatusChanged(String provider, int status,
Bundle extras){ }
    };

    private void updateWithNewLocation(Location location) {
        String latLongString;
        String addressString = "No address found";

```

```

        if (location != null) {
            // Update my location marker
            positionOverlay.setLocation(location);

            // Update the map location.
            Double geoLat = location.getLatitude()*1E6;
            Double geoLng = location.getLongitude()*1E6;
            GeoPoint point = new GeoPoint(geoLat.intValue(),
                geoLng.intValue());

            mapController.animateTo(point);

            double lat = location.getLatitude();
            double lng = location.getLongitude();
            latLongString = "Lat:" + lat + "\nLong:" + lng;

            double latitude = location.getLatitude();
            double longitude = location.getLongitude();

        } else {
            latLongString = "No location found";
        }
    }
}

```

GetGsmSignalStrength.java

```

package com.paad.whereami;

import android.app.Activity;
import android.content.Context;
import android.content.Intent;
import android.content.SharedPreferences;
import android.location.Criteria;
import android.location.Location;
import android.location.LocationListener;
import android.location.LocationManager;
import android.os.Bundle;
import android.os.Handler;
import android.preference.PreferenceManager;
import android.telephony.CellLocation;
import android.telephony.NeighboringCellInfo;
import android.telephony.PhoneStateListener;
import android.telephony.SignalStrength;
import android.telephony.TelephonyManager;
import android.telephony.gsm.GsmCellLocation;
import android.util.Log;
import android.view.Menu;
import android.view.MenuInflater;
import android.view.MenuItem;
import android.view.View;
import android.widget.Button;
import android.widget.TextView;
import android.widget.Toast;

import java.io.BufferedWriter;
import java.io.File;

```

```

import java.io.FileWriter;
import java.io.IOException;
import java.text.SimpleDateFormat;
import java.util.Calendar;
import java.util.Date;
import java.util.GregorianCalendar;
import java.util.List;

public class GetGsmSignalStrength extends Activity
{
    /* This variables need to be global, so we can used them onResume and
onPause method to
    stop the listener */

    TelephonyManager      Tel;
    MyPhoneStateListener  MyListener;
    int seconds;
    int backupseconds;
    int distance;
    boolean Okbutton;
    boolean Stopbutton;
    double longti;
    double lati;
    String signalnew;
    String trackname;
    Button Starting_button;
    Button Stop_button;
    Button Delete_button;
    Button Show_database_button;
    DataHelper dh;
    Handler myHandler= new Handler();
    int mDbm=-1;
    String dbmStr;
    int rssi_signal;
    int cell_id;
    String cell_idstr;
    Calendar cal = new GregorianCalendar();
    int hour,min,sec;
    Date dt = new Date();
    DateUtils time = new DateUtils();
    String mccstr;
    String mncstr;
    String lacstr;
    int lac;
    static String[]
namesmtx={"NeighgorCell_1","NeighgorCell_2","NeighgorCell_3","NeighgorCell_
4","NeighgorCell_5","NeighgorCell_6","NeighgorCell_7","NeighgorCell_8"};
    static String[] rssimtx =
{"empty","empty","empty","empty","empty","empty","empty","empty"};
    static String[] cellidmtx =
{"empty","empty","empty","empty","empty","empty","empty","empty"};

    String TrackingName=NameTracksClass.getTrack_Name_Id_Finder_Global();
    String latitudestr;
    String longtitudestr;
    String signalstrengthstr;
    String cinrstr;
    String type;
    int mStrength;
    int strength, dBm;
    String nettype, cellid;

```



```

String nettype2;
String GENERATEDFILE="/sdcard/temp/samples.xml";
String ENCRYPTEDFILE = "/sdcard/exampledata/samples.xml";

////////////////////////////////////
////////////////////////////////////
//add buttons to the activity
public boolean onCreateOptionsMenu(Menu signal_menu){
    super.onCreateOptionsMenu((android.view.Menu) signal_menu);
    MenuInflater m_signal = getMenuInflater();
    m_signal.inflate(R.menu.signal_menu,(android.view.Menu)
signal_menu);
    return true;
}

public boolean onOptionsItemSelected(MenuItem item){
    switch(item.getItemId()){
        case R.id.Show_Button:
            startActivity(new
Intent("com.paad.whereami.SHOWDATABASECONTENTS"));
            return true;

        case R.id.onesec:
            backupseconds=1;
            return true;

        case R.id.twosec:
            backupseconds=2;
            return true;
        case R.id.fivesec:
            backupseconds=5;
            return true;

        case R.id.tensec:
            backupseconds=10;
            return true;

        case R.id.fifteensec:
            backupseconds=15;
            break;

        case R.id.twentysec:
            backupseconds=20;
            break;
        case R.id.exit_to_first:
            startActivity(new
Intent("com.paad.whereami.NAMETRACKSCLASS"));
            return true;
        case R.id.default_settings_time:
            backupseconds=1;
            return true;

    }
    return(super.onOptionsItemSelected(item));
}
}

```

```

////////////////////////////////////
////////////////////////////////////
//end putting buttons

/** Called when the activity is first created. */
@Override

//opening layout
public void onCreate(Bundle savedInstanceState){
    super.onCreate(savedInstanceState);
    setContentView(R.layout.signal);

    SharedPreferences                getPrefs                =
PreferenceManager.getDefaultSharedPreferences(getApplicationContext());
    String tempTime = getPrefs.getString("Samplinsec", "2");
    int temptime = Integer.parseInt(String.valueOf(tempTime));
    backupseconds = temptime;

    /**////Update the listener, and start it /***/
    MyListener = new MyPhoneStateListener();
    Tel = (TelephonyManager
)getSystemService(Context.TELEPHONY_SERVICE);
    Tel.listen(MyListener
,PhoneStateListener.LISTEN_SIGNAL_STRENGTHS);

    //my location listener input
    LocationManager locationManager;
    String context = Context.LOCATION_SERVICE;
    locationManager = (LocationManager)getSystemService(context);

    /**////eisagw tis proypo8eseis pou 8elw
na exei o gps provider////////////////////////////////////
Criteria criteria = new Criteria();
criteria.setAccuracy(Criteria.ACCURACY_FINE);
criteria.setAltitudeRequired(false);
criteria.setBearingRequired(false);
criteria.setCostAllowed(true);
criteria.setPowerRequirement(Criteria.POWER_LOW);
String provider = locationManager.getBestProvider(criteria,
true);

Location location =
locationManager.getLastKnownLocation(provider);

updateWithNewLocation(location);

locationManager.requestLocationUpdates(provider, 0, 0,
locationListener);

////////////////////////////////////
////////////////////////////////////

this.Starting_button=(Button)
findViewById(R.id.Starting_button);
this.Stop_button=(Button)findViewById(R.id.Stop_button);

```

```

        this.Starting_button.setOnClickListener(new
View.OnClickListener() {

        @Override
        public void onClick(View v) {
            // TODO Auto-generated method stub
            Okbutton=true;
            Stopbutton=false;
            seconds=backupseconds;
            //xmlcheckcreator();

            if ( Okbutton){
                myHandler.removeCallbacks(mUpdateTimeTask);
                myHandler.postDelayed(mUpdateTimeTask, 1000);

            }

        }
    });

    this.Stop_button.setOnClickListener(new View.OnClickListener()

    {

        @Override
        public void onClick(View v) {
            // TODO Auto-generated method stub
            Okbutton=false;
            Stopbutton=true;
            seconds=backupseconds;
            myHandler.removeCallbacks(mUpdateTimeTask);

        }
    });

    this.dh = new DataHelper(this);

    String nametrack= "marios";
    String latitudestr = Double.toString(lati);
    String longitudestr = Double.toString(longti);
    String signalstrengthstr=signalnew;
    String types="T";
    String nettypestr;

    ////////////////////////////////////////
    ////////////////////////////////////////
    }

    ////////////////////////////////////////LOCATION            LISTENER
METHOD//////////////////////////////////////
    private final LocationListener locationListener = new
LocationListener() {
        public void onLocationChanged(Location location) {

```

```

        updateWithNewLocation(location);
    }

    public void onProviderDisabled(String provider){
        updateWithNewLocation(null);
    }

    public void onProviderEnabled(String provider){ }
    public void onStatusChanged(String provider, int status,
        Bundle extras){ }
};
///////////////////////////////////////////////////END                OF                LOCATION
LISTENER//////////////////////////////////////

///////////////////////////////////////////////////LOCATION                UPDATE
METHOD//////////////////////////////////////
    public void updateWithNewLocation(Location location) {
        String latLongString;
        TextView myLocationText;
        myLocationText = (TextView)findViewById(R.id.myLocationText);

        if (location != null) {

            // Update the map location.

            double lat = location.getLatitude();
            double lng = location.getLongitude();
            latLongString = "Lat:" + lat + "\nLong:" + lng;

            double latitude = location.getLatitude();
            double longtitude = location.getLongitude();
            TextView dest=(TextView) findViewById(R.id.destin);
            TextView dest2 = (TextView) findViewById(R.id.destin2);
            dest.setText("Current latitude: " + latitude);
            dest2.setText("Current longtitude: "+ longtitude);
            lati=latitude;
            longti=longtitude;

        }

    }

///////////////////////////////////////////////////END                OF                LOCATION                UPDATE
//////////////////////////////////////

    /* Called when the application is minimized */
    @Override
    protected void onPause()
    {
        super.onPause();
        Tel.listen(MyListener, PhoneStateListener.LISTEN_NONE);
    }

    /* Called when the application resumes */
    @Override
    protected void onResume()
    {
        super.onResume();

```

```

Tel.listen(MyListener,PhoneStateListener.LISTEN_SIGNAL_STRENGTHS);
}

/* P P P P P P P P P P */
/* Start the PhoneState listener */
/* P P P P P P P P P P */

private class MyPhoneStateListener extends PhoneStateListener
{
    int networktype;
    String networkTypeString;
    GsmCellLocation loc;

    // Get the Signal strength from the provider, each time there
is an update
    @Override
    public void onSignalStrengthsChanged(SignalStrength
signalStrength)
    {
        networktype = Tel.getNetworkType();
        String networkOperator = Tel.getNetworkOperator();
        String mcc = networkOperator.substring(0, 3);
        String mnc = networkOperator.substring(3);

        mccstr = String.valueOf(mcc);
        mncstr = String.valueOf(mnc);

        lacstr=Integer.toString(lac);

        // cell_id = loc.getCid();
        // cell_idstr= new Integer(cell_id).toString();
        switch ( networktype ){
        case TelephonyManager.NETWORK_TYPE_UNKNOWN:
networkTypeString = "UNKNOWN"; break;
        case TelephonyManager.NETWORK_TYPE_GPRS:
networkTypeString = "GPRS"; break;
        case TelephonyManager.NETWORK_TYPE_EDGE:
networkTypeString = "EDGE"; break;
        case TelephonyManager.NETWORK_TYPE_UMTS:
networkTypeString = "UMTS";
        case TelephonyManager.NETWORK_TYPE_CDMA:
networkTypeString = "CDMA"; break;
        case TelephonyManager.NETWORK_TYPE_EVDO_0:
networkTypeString = "EVDO_0"; break;
        case TelephonyManager.NETWORK_TYPE_EVDO_A:
networkTypeString = "EVDO_A"; break;
        case TelephonyManager.NETWORK_TYPE_1xRTT:
networkTypeString = "1xRTT"; break;
        case TelephonyManager.NETWORK_TYPE_HSDPA:
networkTypeString = "HSDPA"; break;
        case TelephonyManager.NETWORK_TYPE_HSUPA:
networkTypeString = "HSUPA"; break;
        case TelephonyManager.NETWORK_TYPE_HSPA:
networkTypeString = "HSPA"; break;
        }
        TextView tv1=(TextView) findViewById(R.id.mysignalview);
        tv1.setText("The network type is:" + networkTypeString +
" ");
        nettype2=networkTypeString;

```

```

        if (signalStrength.isGsm()){
            mStrength = signalStrength.getGsmSignalStrength();
            rssi_signal = 2*mStrength-113;
        }
        else{

            strength = -1;
            if (signalStrength.getEvdoDbm() < 0){
                strength = signalStrength.getEvdoDbm();
            }
            else if (signalStrength.getCdmaDbm() < 0){
                strength = signalStrength.getCdmaDbm();
            }
            if (strength < 0){
                // convert to asu
                rssi_signal = Math.round((strength + 113f));
            }

            super.onSignalStrengthsChanged(signalStrength);
            Toast.makeText(getApplicationContext(), "Measuring

Cinr = "
                                +
String.valueOf(signalStrength.getGsmSignalStrength()),
Toast.LENGTH_SHORT).show();
                                String                                mysignal                                =
String.valueOf(signalStrength.getGsmSignalStrength());

                                signalnew=mysignal;

        }
    }

    @Override
    public void onCellLocationChanged(CellLocation location) {
        // TODO Auto-generated method stub
        super.onCellLocationChanged(location);
        loc = (GsmCellLocation) location;
        lac = loc.getLac();
        cell_id = loc.getCid();
        networktype = Tel.getNetworkType();
        switch ( networktype ){
            case                                TelephonyManager.NETWORK_TYPE_UNKNOWN:
networkTypeString = "UNKNOWN"; break;
            case                                TelephonyManager.NETWORK_TYPE_GPRS:
networkTypeString = "GPRS"; break;
            case                                TelephonyManager.NETWORK_TYPE_EDGE:
networkTypeString = "EDGE"; break;
            case                                TelephonyManager.NETWORK_TYPE_UMTS:
networkTypeString = "UMTS";
            case                                TelephonyManager.NETWORK_TYPE_CDMA:
networkTypeString = "CDMA"; break;
            case                                TelephonyManager.NETWORK_TYPE_EVDO_0:
networkTypeString = "EVDO_0"; break;
            case                                TelephonyManager.NETWORK_TYPE_EVDO_A:
networkTypeString = "EVDO_A"; break;

```

```

        case TelephonyManager.NETWORK_TYPE_1xRTT:
networkTypeString = "1xRTT"; break;
        case TelephonyManager.NETWORK_TYPE_HSDPA:
networkTypeString = "HSDPA"; break;
        case TelephonyManager.NETWORK_TYPE_HSUPA:
networkTypeString = "HSUPA"; break;
        case TelephonyManager.NETWORK_TYPE_HSPA:
networkTypeString = "HSPA"; break;
    }
    cell_idstr= new Integer(cell_id).toString();

}
}
}

```

```

//////////cut code here
public Runnable mUpdateTimeTask = new Runnable() {

    public void run() {

        TextView writesec=(TextView)findViewById(R.id.writesec);
        TextView write=(TextView)findViewById(R.id.write);
        if ((Okbutton==true)&&(seconds>=0)) {
            neighbor();
            //hour=cal.get(Calendar.HOUR_OF_DAY);
            //min = cal.get(Calendar.MINUTE);
            //sec = cal.get(Calendar.SECOND);
            //it may be over-reacting but it takes all cases.
Must be tested in real phone

            write.setText("CINR="+        mStrength+"\n"+"RSSI="+
rssi_signal +" dBm "); //signalnew
            writesec.setText("Seconds remaining: "+seconds+".
The originally given seconds are "+ backupseconds+". The current time is:
"+ time.now());
            seconds--;
        } else {
            if((Stopbutton==true)&&(Okbutton==false)){
                writesec.setText("termatise");
                seconds=backupseconds;
            }
            else{

                seconds=backupseconds;
                writesec.setText("The record is saved ");

                dh.insert(trackname.valueOf(TrackingName),latitudestr.valueOf(lati),l
ongtitudestr.valueOf(longti),signalstrengthstr.valueOf(rssi_signal),
cinrstr.valueOf(mStrength),        time.now(),        mccstr,        mncstr,
String.valueOf(lac),        "T",
nettype.valueOf(nettype2),cellid.valueOf(cell_idstr),"-
",namesmtx[0],cellidmtx[0],rssimtx[0],namesmtx[1],cellidmtx[1],rssimtx[1],n
amesmtx[2],cellidmtx[2],rssimtx[2],namesmtx[3],cellidmtx[3],rssimtx[3],name

```

```

smtx[4],cellidmtx[4],rssimtx[4],namesmtx[5],cellidmtx[5],rssimtx[5],namesmt
x[6],cellidmtx[6],rssimtx[6], namesmtx[7],cellidmtx[7],rssimtx[7] );

        }

    }

    myHandler.postDelayed(this,1000);
}

};

```

```

public void neighbor(){
    List<NeighboringCellInfo>          NeighboringList          =
Tel.getNeighboringCellInfo();
    String stringNeighboring = "Neighboring List- Lac : Cid : RSSI\n";

    for (int i=0; i < NeighboringList.size(); i++){

        String dBm;
        int rssi = NeighboringList.get(i).getRssi();

        if(rssi == NeighboringCellInfo.UNKNOWN_RSSI){
            dBm = "Unknown RSSI";
        }else{

            dBm = String.valueOf(-113 + 2 * rssi) + " dBm";
            rssimtx[i]=String.valueOf(dBm);
        }

        cellidmtx[i]=String.valueOf(NeighboringList.get(i).getCid());

        stringNeighboring = stringNeighboring
            + String.valueOf(NeighboringList.get(i).getLac()) + " : "
            + String.valueOf(NeighboringList.get(i).getCid()) + " : "
            + dBm + "\n";
    }
}
}
}

```

```
/* GetGsmSignalStrength */
```

DistanceSignalStrength.java

```

package com.paad.whereami;

import java.io.BufferedWriter;
import java.io.File;
import java.io.FileWriter;
import java.io.IOException;
import java.lang.Object;
import java.text.DecimalFormat;

```



```
import java.text.NumberFormat;
import java.util.Calendar;
import java.util.Date;
import java.util.GregorianCalendar;
import java.util.List;

import android.app.Activity;
import android.app.Application;
import android.content.Context;
import android.content.Intent;
import android.content.SharedPreferences;
import android.location.Criteria;
import android.location.Location;
import android.location.LocationListener;
import android.location.LocationManager;
import android.os.Bundle;
import android.os.Handler;
import android.os.SystemClock;
import android.preference.PreferenceManager;
import android.telephony.CellLocation;
import android.telephony.NeighboringCellInfo;
import android.telephony.PhoneStateListener;
import android.telephony.SignalStrength;
import android.telephony.TelephonyManager;
import android.telephony.gsm.GsmCellLocation;
import android.util.Log;
import android.view.Menu;
import android.view.MenuInflater;
import android.view.MenuItem;
import android.view.SubMenu;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.TextView;
import android.widget.Toast;

public class DistanceSignalStrength extends Activity
{
    /* This variables need to be global, so we can used them onResume and
onPause method to
stop the listener */

    TelephonyManager Tel;
    MyPhoneStateListener MyListener;
    int seconds;
    int backupseconds=3;
    float temp_dist;

    boolean Okbutton;
    boolean Stopbutton;
    double longti;
    double lati;
    String signalnew;
    String trackna;
    Button Starting_button;
    Button Stop_button;
    DataHelper dbh;
    private Button Starting_Point_Button;
    private Button Start_Recording_Button;
```

```

private Button Quit_Button;
double Starting_Point_Latitude;
double Starting_Point_Longtitude;
private Handler myHandler= new Handler();
EditText latitudeEditText;
EditText longtitudeEditText;
private TextView output1;
LocationManager locationManager;
float tempdist;
private static final NumberFormat nf = new
DecimalFormat("##.#####");
String
TrackingName1=NameTracksClass.getTrack_Name_Id_Finder_Global();
String GENERATEDFILE="/sdcard/temp/samples.xml";
int mStrength;
int strength;
int rssi_signal;
int cell_id;
String cell_idstr;
String trackname;
String latitudestr;
String longtitudestr;
String type;
String signalstrengthstr;
String nettype, cellid;
String nettype2;
String cinrstr;
Calendar cal = new GregorianCalendar();
int hour,min,sec;
Date dt = new Date();
DateUtils time = new DateUtils();
String mccstr;
String mncstr;
String lacstr;
int lac;
static String[]
namesmtx={"NeighgorCell_1","NeighgorCell_2","NeighgorCell_3","NeighgorCell_
4","NeighgorCell_5","NeighgorCell_6","NeighgorCell_7","NeighgorCell_8"};
static String[] rssimtx =
{"empty","empty","empty","empty","empty","empty","empty","empty"};
static String[] cellidmtx =
{"empty","empty","empty","empty","empty","empty","empty","empty"};

String disTemp;
float distance;//Float.parseFloat(String.valueOf(disTemp));

/*****
*****/
//add buttons to the activity
public boolean onCreateOptionsMenu(Menu signal_menu_distance){
super.onCreateOptionsMenu((android.view.Menu)
signal_menu_distance);
MenuInflater my_signal_distance = getMenuInflater();

my_signal_distance.inflate(R.menu.signal_menu_distance,(android.view.
Menu) signal_menu_distance);
return true;
}

public boolean onOptionsItemSelected(MenuItem item){
switch(item.getItemId()){

```

```

        case R.id.twometers:
            distance=2;
            backupseconds=4;
            break;

        case R.id.fivemeters:
            distance=5;
            backupseconds = 8;
            break;

        case R.id.tenmeters:
            distance=10;
            backupseconds = 15;
            break;

        case R.id.twentymeters:
            distance=20;
            backupseconds = 30;
            break;
        case R.id.thirtymeters:
            distance=30;
            backupseconds = 45;
        case R.id.fiftymeters:
            distance=50;
            backupseconds=65;

        case R.id.Exit_to_first1:
            startActivity(new
Intent("com.paad.whereami.NAMETRACKSCLASS"));
            return true;

        case R.id.ShowDataBase2:
            startActivity(new
Intent("com.paad.whereami.SHOWDATABASECONTENTS"));
            break;

    }
    return(super.onOptionsItemSelected(item));
}
//end putting buttons

/** Called when the activity is first created. */
@Override
//opening layout
public void onCreate(Bundle savedInstanceState){

    super.onCreate(savedInstanceState);
    setContentView(R.layout.signal_distance);

    SharedPreferences getPrefs =
PreferenceManager.getDefaultSharedPreferences(getBaseContext());
    disTemp = getPrefs.getString("Samplindist", "2");
    String disTemp2 = String.valueOf(disTemp);
    distance = Float.parseFloat(disTemp2);
    if (distance == 2){
        backupseconds = 4;
    }else if (distance==5){
        backupseconds = 8;
    }else if (distance==10){

```

```

        backupseconds=15;
    }else if(distance==20){
        backupseconds=30;
    }else if (distance==30){
        backupseconds=45;
    }else if (distance==50){
        backupseconds=65;
    }

    output1 = (TextView) findViewById(R.id.output1);
    latitudeEditText=(EditText)findViewById(R.id.point_latitude);

    longitudeEditText=(EditText)findViewById(R.id.point_longitude);

    /* Update the listener, and start it */
    MyListener = new MyPhoneStateListener();
    Tel = (TelephonyManager
) getSystemService(Context.TELEPHONY_SERVICE);
    Tel.listen(MyListener
,PhoneStateListener.LISTEN_SIGNAL_STRENGTHS);

    //my location listener input
    LocationManager locationManager;
    String context = Context.LOCATION_SERVICE;
    locationManager = (LocationManager) getSystemService(context);

    //entering the providers criteria
    Criteria criteria = new Criteria();
    criteria.setAccuracy(Criteria.ACCURACY_FINE);
    criteria.setAltitudeRequired(false);
    criteria.setBearingRequired(false);
    criteria.setCostAllowed(true);
    criteria.setPowerRequirement(Criteria.POWER_LOW);
    String provider = locationManager.getBestProvider(criteria,
true);

    Location location =
locationManager.getLastKnownLocation(provider);

    locationManager.updateWithNewLocation(location);

    locationManager.requestLocationUpdates(provider, 0,
0,locationListener);

    //setting up the buttons of the activity
    Starting_Point_Button=(Button) findViewById(R.id.Start_point);
    Start_Recording_Button=(Button) findViewById(R.id.Start_rec);
    Quit_Button=(Button) findViewById(R.id.quit_rec);

    Start_Recording_Button.setOnClickListener(new
View.OnClickListener() {
        @Override
        public void onClick(View v) {
            // TODO Auto-generated method stub
            Okbutton=true;
            seconds=backupseconds;
            //xmlcheckcreator();

```

```

        if ( Okbutton){

            myHandler.removeCallbacks(mUpdateDistanceTask);
            myHandler.postDelayed(mUpdateDistanceTask,
1000);
        }
    });

    Starting_Point_Button.setOnClickListener(new
View.OnClickListener() {
        @Override
        public void onClick(View v) {
            // TODO Auto-generated method stub
            Starting_Point_Longtitude=longti;
            Starting_Point_Latitude=lati;
            latitudeEditText.setText(nf.format(lati));
            longitudeEditText.setText(nf.format(longti));
        }
    });

    Quit_Button.setOnClickListener(new View.OnClickListener() {

        @Override
        public void onClick(View v) {
            // TODO Auto-generated method stub
            Okbutton=false;
            Stopbutton=true;
            myHandler.removeCallbacks(mUpdateDistanceTask);
            seconds = backupseconds;
            /*
            try{
                File file= new File(GENERATEDFILE);
                BufferedWriter output = new
BufferedWriter(new FileWriter(file,true));
                output.write("</trackname>");
                //output.write("</"+
trackname.valueOf(TrackingName1) +">");
                output.newLine();
                output.flush();
            }catch (IOException e){};
            */
        }
    });

    this.dbh = new DataHelper(this);
}

//THAT'S THE END OF THE ON CREATE METHOD

/*****
*****/

//LOCATION LISTENER METHOD
private final LocationListener locationListener = new
LocationListener() {
    public void onLocationChanged(Location location) {

```

```

        updateWithNewLocation(location);
    }

    public void onProviderDisabled(String provider){
        updateWithNewLocation(null);
    }

    public void onProviderEnabled(String provider){ }
    public void onStatusChanged(String provider, int status,
        Bundle extras){ }
};
//END OF LOCATION LISTENER

//LOCATION UPDATE METHOD
public void updateWithNewLocation(Location location) {
    String latLongString;
    TextView myLocationText;
    myLocationText = (TextView)findViewById(R.id.myLocationText);

    if (location != null) {

        // Update the map location.

        double lat = location.getLatitude();
        double lng = location.getLongitude();
        latLongString = "Lat:" + lat + "\nLong:" + lng;

        double latitude = location.getLatitude();
        double longtitude = location.getLongitude();
        TextView signal_dist=(TextView)
findViewById(R.id.write_dist);
        signal_dist.setText("latitude: " + latitude + " " +
"longtitude: "+ longtitude);
        lati=latitude;
        longti=longtitude;

    }

}

//END OF LOCATION UPDATE

/* Called when the application is minimized */
//@Override
protected void onPause()
{
    super.onPause();
    Tel.listen(MyListener, PhoneStateListener.LISTEN_NONE);
}

//Called when the application resumes
@Override
protected void onResume()
{
    super.onResume();

    Tel.listen(MyListener,PhoneStateListener.LISTEN_SIGNAL_STRENGTHS);
}

```

```

/* P P P P P P P P P P */
/* Start the PhoneState listener */
/* P P P P P P P P P P */

private class MyPhoneStateListener extends PhoneStateListener
{
    int networktype;
    String networkTypeString;
    GsmCellLocation loc;

    @Override
    public void onSignalStrengthsChanged(SignalStrength
signalStrength)
    {
        networktype = Tel.getNetworkType();
        String networkOperator = Tel.getNetworkOperator();

        String mcc = networkOperator.substring(0, 3);
        String mnc = networkOperator.substring(3);

        mccstr = String.valueOf(mcc);
        mncstr = String.valueOf(mnc);
        switch ( networktype ) {
networkTypeString = "UNKNOWN"; break;
networkTypeString = "GPRS"; break;
networkTypeString = "EDGE"; break;
networkTypeString = "UMTS";
networkTypeString = "CDMA"; break;
networkTypeString = "EVDO_0"; break;
networkTypeString = "EVDO_A"; break;
networkTypeString = "1xRTT"; break;
networkTypeString = "HSDPA"; break;
networkTypeString = "HSUPA"; break;
networkTypeString = "HSPA"; break;
        }
        nettype2=String.valueOf(networkTypeString);

        if (signalStrength.isGsm()){
            mStrength = signalStrength.getGsmSignalStrength();
            rssi_signal = 2*mStrength-113;
        }
        else{
            strength = -1;
            if (signalStrength.getEvdoDbm() < 0){
                strength = signalStrength.getEvdoDbm();
            }
            else if (signalStrength.getCdmaDbm() < 0){
                strength = signalStrength.getCdmaDbm();
            }
        }
    }
}

```

```

    }
    if (strength < 0){
        // convert to asu
        rssi_signal = Math.round((strength + 113f));
    }

    super.onSignalStrengthsChanged(signalStrength);
    Toast.makeText(getApplicationContext(), "Recording
Cinr = "
                                +
String.valueOf(signalStrength.getGsmSignalStrength()),
Toast.LENGTH_SHORT).show();
                                String                mysignal                =
String.valueOf(signalStrength.getGsmSignalStrength());

                                signalnew=mysignal;

    }

}

@Override
public void onCellLocationChanged(CellLocation location) {
    // TODO Auto-generated method stub
    super.onCellLocationChanged(location);

    loc = (GsmCellLocation) location;
    cell_id = loc.getCid();
    networktype = Tel.getNetworkType();
    switch ( networktype ){
    case TelephonyManager.NETWORK_TYPE_UNKNOWN:
networkTypeString = "UNKNOWN"; break;
    case TelephonyManager.NETWORK_TYPE_GPRS:
networkTypeString = "GPRS"; break;
    case TelephonyManager.NETWORK_TYPE_EDGE:
networkTypeString = "EDGE"; break;
    case TelephonyManager.NETWORK_TYPE_UMTS:
networkTypeString = "UMTS";
    case TelephonyManager.NETWORK_TYPE_CDMA:
networkTypeString = "CDMA"; break;
    case TelephonyManager.NETWORK_TYPE_EVDO_0:
networkTypeString = "EVDO_0"; break;
    case TelephonyManager.NETWORK_TYPE_EVDO_A:
networkTypeString = "EVDO_A"; break;
    case TelephonyManager.NETWORK_TYPE_1xRTT:
networkTypeString = "1xRTT"; break;
    case TelephonyManager.NETWORK_TYPE_HSDPA:
networkTypeString = "HSDPA"; break;
    case TelephonyManager.NETWORK_TYPE_HSUPA:
networkTypeString = "HSUPA"; break;
    case TelephonyManager.NETWORK_TYPE_HSPA:
networkTypeString = "HSPA"; break;

    }
    cell_idstr= new Integer(cell_id).toString();

    List<NeighboringCellInfo> NeighboringList =
Tel.getNeighboringCellInfo();

```



```

String stringNeighboring = "Neighboring List- Lac : Cid :
RSSI\n";

for(int i=0; i < NeighboringList.size(); i++){

String dBm;
int rssi = NeighboringList.get(i).getRssi();

if(rssi == NeighboringCellInfo.UNKNOWN_RSSI){
dBm = "Unknown RSSI";
}else{

dBm = String.valueOf(-113 + 2 * rssi) + " dBm";
rssimtx[i]=dBm;
}

cellidmtx[i]=String.valueOf(NeighboringList.get(i).getCid());

stringNeighboring = stringNeighboring
+ String.valueOf(NeighboringList.get(i).getLac()) + " :
"
+ String.valueOf(NeighboringList.get(i).getCid()) + " :
"
+ dBm +"\n";
}
}

private Runnable mUpdateDistanceTask = new Runnable() {

public void run() {

Location aLoc = new Location("aLoc");
aLoc.setLatitude(lati);
aLoc.setLongitude(longti);

Location bLoc=new Location("bLoc");
bLoc.setLatitude(Starting_Point_Latitude);
bLoc.setLongitude(Starting_Point_Longitude);

temp_dist=bLoc.distanceTo(aLoc);

TextView
writedist3=(TextView)findViewById(R.id.write_dist);
TextView
writedist2=(TextView)findViewById(R.id.write_dist2);

if ((Okbutton=true)&&(seconds>=0)) {
neighbor();

writedist3.setText("Longitude="+longti+"\n"+"Latitude="+lati+"\n"+"C
INR="+mStrength
+"\n"+"SignalStrength="+rssi_signal+"\n"+"Distance
given="+distance + "\n"+"Network Type="+ nettype2);
writedist2.setText("The distance from the starting
point is :"+temp_dist+" and the seconds remaining "+seconds);

```

```

        if (((temp_dist>distance-
1)&&(temp_dist<distance+1))) {
            Starting_Point_Latitude=lati;
            Starting_Point_Longitude=longti;
            //record

            dbh.insert(trackname.valueOf(TrackingName1),latitudestr.valueOf(lati)
,longitudestr.valueOf(longti),signalstrengthstr.valueOf(rssi_signal),cinrs
tr.valueOf(signalnew),time.now(),          mccstr,          mncstr,
String.valueOf(lac),"D",nettype.valueOf(nettype2),cellid.valueOf(cell_idstr
),"-",
namesmtx[0],cellidmtx[0],rssimtx[0],namesmtx[1],cellidmtx[1],rssimtx[1],nam
esmtx[2],cellidmtx[2],rssimtx[2],namesmtx[3],cellidmtx[3],rssimtx[3],namesm
tx[4],cellidmtx[4],rssimtx[4],namesmtx[5],cellidmtx[5],rssimtx[5],namesmtx[
6],cellidmtx[6],rssimtx[6], namesmtx[7],cellidmtx[7],rssimtx[7]);

        }
        else if ((seconds<=0)){
            neighbor();
            seconds=backupseconds;
            writedist2.setText("To zero and back again");

            dbh.insert(trackname.valueOf(TrackingName1),latitudestr.valueOf(lati)
,longitudestr.valueOf(longti),signalstrengthstr.valueOf(rssi_signal),cinrs
tr.valueOf(signalnew),time.now(),          mccstr,          mncstr,
String.valueOf(lac),"D",nettype.valueOf(nettype2),cellid.valueOf(cell_idstr
),"-",
namesmtx[0],cellidmtx[0],rssimtx[0],namesmtx[1],cellidmtx[1],rssimtx[1],nam
esmtx[2],cellidmtx[2],rssimtx[2],namesmtx[3],cellidmtx[3],rssimtx[3],namesm
tx[4],cellidmtx[4],rssimtx[4],namesmtx[5],cellidmtx[5],rssimtx[5],namesmtx[
6],cellidmtx[6],rssimtx[6], namesmtx[7],cellidmtx[7],rssimtx[7]);

        }

        seconds--;
    }
    else {
        if (Stopbutton)
            writedist2.setText("termatise");
    }
    myHandler.postDelayed(this,1000);
}

};

/*
public boolean xmlcheckcreator(){
    //File file= new File("/sdcard/exampledata/samples.xml");
    File file= new File(GENERATEDFILE);
    if (file.exists()==false){
        try {
            file.createNewFile();
            BufferedWriter output = new BufferedWriter(new
FileWriter(file,true));
            output.write("<?xml version=\"1.0\" encoding=\"utf-
8\"?>");
            output.newLine();

```

```

        output.write("<root>");
        output.newLine();
        output.write("<trackname>");
        //output.write("<" +
trackname.valueOf(TrackingName1) + ">");
        output.newLine();
        output.flush();

        } catch (IOException e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        }
    }
    else{
        try {
            BufferedWriter output = new BufferedWriter(new
FileWriter(file,true));
            output.write("<trackname>");
            //output.write("<" +
trackname.valueOf(TrackingName1) + ">");
            output.newLine();
            output.flush();
        } catch(IOException e){};
    }

    }
    return true;
}

*/
public void neighbor(){
    List<NeighboringCellInfo>           NeighboringList           =
Tel.getNeighboringCellInfo();
    String stringNeighboring = "Neighboring List- Lac : Cid :
RSSI\n";

    for (int i=0; i < NeighboringList.size(); i++){

        String dBm;
        int rssi = NeighboringList.get(i).getRssi();

        if(rssi == NeighboringCellInfo.UNKNOWN_RSSI){
            dBm = "Unknown RSSI";
        }else{

            dBm = String.valueOf(-113 + 2 * rssi) + " dBm";
            rssimtx[i]=String.valueOf(dBm);
        }

        cellidmtx[i]=String.valueOf(NeighboringList.get(i).getCid());

        stringNeighboring = stringNeighboring
+ String.valueOf(NeighboringList.get(i).getLac()) + " : "
+ String.valueOf(NeighboringList.get(i).getCid()) + " : "
+ dBm + "\n";
    }
}
}
}

```

```
/* DistanceSignalStrength */
```

SpeedSignal.java

```
package com.paad.whereami;

import java.io.BufferedWriter;
import java.io.File;
import java.io.FileWriter;
import java.io.IOException;
import java.util.Calendar;
import java.util.Date;
import java.util.GregorianCalendar;
import java.util.List;

import android.app.Activity;
import android.content.Context;
import android.content.Intent;
import android.location.Criteria;
import android.location.Location;
import android.location.LocationListener;
import android.location.LocationManager;
import android.os.Bundle;
import android.os.Handler;
import android.telephony.CellLocation;
import android.telephony.NeighboringCellInfo;
import android.telephony.PhoneStateListener;
import android.telephony.SignalStrength;
import android.telephony.TelephonyManager;
import android.telephony.gsm.GsmCellLocation;
import android.view.Menu;
import android.view.MenuInflater;
import android.view.MenuItem;
import android.view.View;
import android.widget.Button;
import android.widget.TextView;
import android.widget.Toast;

public class SpeedSignal extends Activity {

    Button Start_Button;
    Button Stop_Button;
    Button Starting_Point;
    TextView Velocitytxt, latitudetxt, longitudetxt, distancetxt,
    signaltxt, infotxt;
    double lati, longti;
    double startLatitude, startLongitude;
    double tempPointLat, tempPointLng;
    String latitudestr, longitudestr;
    Handler myHandler = new Handler();
    Boolean OkButton, StopButton;
    int seconds=5;
    float distance;
    int backupseconds=2;
```

```

float temp_dist;
float speed;
TelephonyManager      Tel;
MyPhoneStateListener  MyListener;
String nettype2;
int mStrength;
int strength;
int rssi_signal;
int cell_id;
String cell_idstr;
String signalnew;
String TrackingName=NameTracksClass.getTrack_Name_Id_Finder_Global();
String signalstrengthstr;
String type;
DataHelper dh;
String trackname;
String nettype;
String cellid;
String speedstr;
String cinrstr;
Calendar cal = new GregorianCalendar();
int hour,min,sec;
Date dt = new Date();
DateUtils time = new DateUtils();
String mccstr;
String mncstr;
String lacstr;
int lac;
static                                     String[]
namesmtx={"NeighgorCell_1","NeighgorCell_2","NeighgorCell_3","NeighgorCell_
4","NeighgorCell_5","NeighgorCell_6","NeighgorCell_7","NeighgorCell_8"};
static                                     String[]          rssimtx          =
{"empty","empty","empty","empty","empty","empty","empty","empty"};
static                                     String[]          cellidmtx         =
{"empty","empty","empty","empty","empty","empty","empty","empty"};

String GENERATEDFILE="/sdcard/temp/samples.xml";
String ENCRYPTEDFILE = "/sdcard/exampledata/samples.xml";

public boolean onCreateOptionsMenu(Menu speed_menu){
    super.onCreateOptionsMenu((android.view.Menu) speed_menu);
    MenuInflater m_signal = getMenuInflater();
    m_signal.inflate(R.menu.speed_menu,(android.view.Menu)
speed_menu);
    return true;
}

public boolean onOptionsItemSelected(MenuItem item){
    switch(item.getItemId()){
        case R.id.Show_Button_speed:
            startActivity(new
Intent("com.paad.whereami.SHOWDATABASECONTENTS"));
            return true;

            case R.id.exit_to_first_speed:
                startActivity(new
Intent("com.paad.whereami.NAMETRACKSCLASS"));
                return true;

    }
}

```

```

        return(super.onOptionsItemSelected(item));
    }

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        // TODO Auto-generated method stub
        super.onCreate(savedInstanceState);
        setContentView(R.layout.speedsignal);

        /*////Update the listener, and start it ////*/
        MyListener = new MyPhoneStateListener();
        Tel = (TelephonyManager
) getSystemService(Context.TELEPHONY_SERVICE);
        Tel.listen(MyListener
, PhoneStateListener.LISTEN_SIGNAL_STRENGTHS);

        Start_Button=(Button) findViewById(R.id.widget30);
        Stop_Button=(Button) findViewById(R.id.widget31);
        Velocitytxt=(TextView) findViewById(R.id.widget32);
        latitudetxt=(TextView) findViewById(R.id.widget35);
        longitudetxt=(TextView) findViewById(R.id.widget34);
        distancetxt=(TextView) findViewById(R.id.widget33);
        Starting_Point=(Button) findViewById(R.id.widget37);
        signaltxt=(TextView) findViewById(R.id.widget36);

        infotxt=(TextView) findViewById(R.id.widget38);

        Start_Button.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Velocitytxt.setText("REC Started");
                latitudetxt.setText("Latitude: "+latitudestr);
                longitudetxt.setText("Longitude: "
+longitudestr);

                OkButton=true;
                StopButton=false;
                //xmlcheckcreator();
                if ( OkButton){
                    myHandler.removeCallbacks(mUpdateTimeTask);
                    myHandler.postDelayed(mUpdateTimeTask, 500);
                }
            }
        });

        Stop_Button.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Velocitytxt.setText("REC Stopped");
                StopButton=true;
                OkButton=false;
                myHandler.removeCallbacks(mUpdateTimeTask);

                /*
                try{
                    File file= new File(GENERATEDFILE);
                    BufferedWriter output = new
BufferedWriter(new FileWriter(file,true));

```

```

        output.write("</trackname>");
        //output.write("</"+
trackname.valueOf(TrackingName) +">");
        output.newLine();
        output.flush();
    }catch (IOException e){};*/

    }
});

Starting_Point.setOnClickListener(new View.OnClickListener() {

    @Override
    public void onClick(View v) {
        // TODO Auto-generated method stub
        startLatitude=lati;
        startLongitude=longti;

        tempPointLat=lati;
        tempPointLng=longti;
    }
});

this.dh = new DataHelper(this);

//Location Code
LocationManager locationManager;
String context = Context.LOCATION_SERVICE;
locationManager = (LocationManager) getSystemService(context);

//Import Location Provider Criteria
Criteria criteria = new Criteria();
criteria.setAccuracy(Criteria.ACCURACY_FINE);
criteria.setAltitudeRequired(false);
criteria.setBearingRequired(false);
criteria.setCostAllowed(true);
criteria.setPowerRequirement(Criteria.POWER_LOW);
String provider = locationManager.getBestProvider(criteria,
true);

        Location                location                =
locationManager.getLastKnownLocation(provider);

        updateWithNewLocation(location);

        locationManager.requestLocationUpdates(provider,                0,
0,locationListener);

//Location Code End

}//////OnCreate Code End\\\\\\

//Location Listener Code
private final LocationListener locationListener = new
LocationListener() {
    public void onLocationChanged(Location location) {
        updateWithNewLocation(location);
        speed=location.getSpeed();
    }
}

```

```

        public void onProviderDisabled(String provider){
            updateWithNewLocation(null);
        }

        public void onProviderEnabled(String provider){ }
        public void onStatusChanged(String provider, int status,
            Bundle extras){ }
};
//Location Listener Code End

//Location Update Code
public void updateWithNewLocation(Location location) {
    String latLongString;
    TextView myLocationText;
    myLocationText = (TextView)findViewById(R.id.myLocationText);
    if (location != null) {
        double lat = location.getLatitude();
        double lng = location.getLongitude();
        latLongString = "Lat:" + lat + "\nLong:" + lng;

        double latitude = location.getLatitude();
        double longitude = location.getLongitude();
        lati=latitude;
        longti=longitude;
        latitudestr=Double.toString(lati);
        longitudestr=Double.toString(longti);

    }

}
//Location Update Code End

/* Called when the application is minimized */
@Override
protected void onPause()
{
    super.onPause();
    Tel.listen(MyListener, PhoneStateListener.LISTEN_NONE);
}

/* Called when the application resumes */
@Override
protected void onResume()
{
    super.onResume();

    Tel.listen(MyListener,PhoneStateListener.LISTEN_SIGNAL_STRENGTHS);
}

//PhoneStateListener Code
private class MyPhoneStateListener extends PhoneStateListener
{
    int networktype;
    String networkTypeString;
    GsmCellLocation loc;

    // Get the Signal strength from the provider, each time there
    is an update

```



```

@Override
public void onSignalStrengthsChanged(SignalStrength
signalStrength)
{
    networktype = Tel.getNetworkType();
    String networkOperator = Tel.getNetworkOperator();
    String mcc = networkOperator.substring(0, 3);
    String mnc = networkOperator.substring(3);

    mccstr = String.valueOf(mcc);
    mncstr = String.valueOf(mnc);

    lacstr=Integer.toString(lac);
    // cell_id = loc.getCid();
    // cell_idstr= new Integer(cell_id).toString();
    switch ( networktype ){
    case TelephonyManager.NETWORK_TYPE_UNKNOWN:
networkTypeString = "UNKNOWN"; break;
    case TelephonyManager.NETWORK_TYPE_GPRS:
networkTypeString = "GPRS"; break;
    case TelephonyManager.NETWORK_TYPE_EDGE:
networkTypeString = "EDGE"; break;
    case TelephonyManager.NETWORK_TYPE_UMTS:
networkTypeString = "UMTS";
    case TelephonyManager.NETWORK_TYPE_CDMA:
networkTypeString = "CDMA"; break;
    case TelephonyManager.NETWORK_TYPE_EVDO_0:
networkTypeString = "EVDO_0"; break;
    case TelephonyManager.NETWORK_TYPE_EVDO_A:
networkTypeString = "EVDO_A"; break;
    case TelephonyManager.NETWORK_TYPE_1xRTT:
networkTypeString = "1xRTT"; break;
    case TelephonyManager.NETWORK_TYPE_HSDPA:
networkTypeString = "HSDPA"; break;
    case TelephonyManager.NETWORK_TYPE_HSUPA:
networkTypeString = "HSUPA"; break;
    case TelephonyManager.NETWORK_TYPE_HSPA:
networkTypeString = "HSPA"; break;
    }
    nettype2=networkTypeString;

    if (signalStrength.isGsm())
        mStrength = signalStrength.getGsmSignalStrength();
    else{
        strength = -1;
        if (signalStrength.getEvdoDbm() < 0){
            strength = signalStrength.getEvdoDbm();
        }
        else if (signalStrength.getCdmaDbm() < 0){
            strength = signalStrength.getCdmaDbm();
        }
        if (strength < 0){
            // convert to asu
            rssi_signal = Math.round((strength + 113f));
        }

        super.onSignalStrengthsChanged(signalStrength);
        Toast.makeText(getApplicationContext(), "Metraw
shma Cinr = "

```

```

+
String.valueOf(signalStrength.getGsmSignalStrength()),
Toast.LENGTH_SHORT).show();
        String          mysignal          =
String.valueOf(signalStrength.getGsmSignalStrength());

        signalnew=mysignal;

    }

}

@Override
public void onCellLocationChanged(CellLocation location) {
    // TODO Auto-generated method stub
    super.onCellLocationChanged(location);

    loc = (GsmCellLocation) location;
    lac = loc.getLac();
    cell_id = loc.getCid();
    networktype = Tel.getNetworkType();
    switch ( networktype ){
networkTypeString = "UNKNOWN"; break;          TelephonyManager.NETWORK_TYPE_UNKNOWN:
networkTypeString = "GPRS"; break;            TelephonyManager.NETWORK_TYPE_GPRS:
networkTypeString = "EDGE"; break;            TelephonyManager.NETWORK_TYPE_EDGE:
networkTypeString = "UMTS";                   TelephonyManager.NETWORK_TYPE_UMTS:
networkTypeString = "CDMA"; break;            TelephonyManager.NETWORK_TYPE_CDMA:
networkTypeString = "EVDO_0"; break;          TelephonyManager.NETWORK_TYPE_EVDO_0:
networkTypeString = "EVDO_A"; break;          TelephonyManager.NETWORK_TYPE_EVDO_A:
networkTypeString = "1xRTT"; break;           TelephonyManager.NETWORK_TYPE_1xRTT:
networkTypeString = "HSDPA"; break;           TelephonyManager.NETWORK_TYPE_HSDPA:
networkTypeString = "HSUPA"; break;           TelephonyManager.NETWORK_TYPE_HSUPA:
networkTypeString = "HSPA"; break;            TelephonyManager.NETWORK_TYPE_HSPA:

    }
    cell_idstr= new Integer(cell_id).toString();

}

}

//PhoneStateListener Code Ends

public Runnable mUpdateTimeTask = new Runnable() {

```

```

public void run() {
    Velocitytxt.setText(speed + " m/s");
    if ((speed>=0)&&(speed<=6)){
        backupseconds=5;
    }
    if ((speed>6)&&(speed<23)){
        backupseconds=2;
    }
    if ((speed>23)){
        backupseconds=1;
    }

    Location aLoc = new Location("aLoc");
    aLoc.setLatitude(lati);
    aLoc.setLongitude(longti);

    Location bLoc=new Location("bLoc");
    bLoc.setLatitude(startLatitude);
    bLoc.setLongitude(startLongitude);

    Location cLoc = new Location ("cLoc");
    cLoc.setLatitude(tempPointLat);
    cLoc.setLongitude(tempPointLng);

    temp_dist=bLoc.distanceTo(aLoc);

    if ((OkButton==true)&&(seconds>0)){
        infotxt.setText("Recording in progress. Seconds
remaining "+ seconds);
        distancetxt.setText("Total Distance Travelled: " +
temp_dist);
        signaltxt.setText("Signal: "+rssi_signal+" dBm");
        neighbor();
        seconds--;
    }else{
        if (seconds==0){
            neighbor();
            seconds=backupseconds;
            infotxt.setText("A recording added in the
database");
            dh.insert(trackname.valueOf(TrackingName),latitudestr.valueOf(lati),l
ongtitudestr.valueOf(longti),signalstrengthstr.valueOf(rssi_signal),
cinrstr.valueOf(mStrength), time.now(), mccstr, mncstr,
String.valueOf(lac), "T",
nettype.valueOf(nettype2),cellid.valueOf(cell_idstr),"-
",namesmtx[0],cellidmtx[0],rssimtx[0],namesmtx[1],cellidmtx[1],rssimtx[1],n
amesmtx[2],cellidmtx[2],rssimtx[2],namesmtx[3],cellidmtx[3],rssimtx[3],name
smtx[4],cellidmtx[4],rssimtx[4],namesmtx[5],cellidmtx[5],rssimtx[5],namesmt
x[6],cellidmtx[6],rssimtx[6], namesmtx[7],cellidmtx[7],rssimtx[7] );
        }
    }
    if(StopButton==true){
        infotxt.setText("The recording of the path
stopped");
        seconds=backupseconds;
    }
}

```

```

        myHandler.postDelayed(this,1000);
    }

};

    public void neighbor(){
        List<NeighboringCellInfo>      NeighboringList      =
Tel.getNeighboringCellInfo();
        String stringNeighboring = "Neighboring List- Lac : Cid :
RSSI\n";

        for (int i=0; i < NeighboringList.size(); i++){

            String dBm;
            int rssi = NeighboringList.get(i).getRssi();

            if(rssi == NeighboringCellInfo.UNKNOWN_RSSI){
                dBm = "Unknown RSSI";
            }else{

                dBm = String.valueOf(-113 + 2 * rssi) + " dBm";
                rssimtx[i]=String.valueOf(dBm);
            }

            cellidmtx[i]=String.valueOf(NeighboringList.get(i).getCid());

            stringNeighboring = stringNeighboring
+ String.valueOf(NeighboringList.get(i).getLac() + " : "
+ String.valueOf(NeighboringList.get(i).getCid() + " : "
+ dBm + "\n";
        }
    }
}

```

WhereAmI.java

```

package com.paad.whereami;

import java.io.IOException;
import java.util.List;
import java.util.Locale;

import com.google.android.maps.GeoPoint;
import com.google.android.maps.MapActivity;
import com.google.android.maps.MapController;
import com.google.android.maps.MapView;
import com.google.android.maps.MyLocationOverlay;
import com.google.android.maps.Overlay;

import android.app.AlertDialog;
import android.content.Context;
import android.content.DialogInterface;
import android.content.Intent;
import android.location.Address;
import android.location.Criteria;
import android.location.Geocoder;

```

```

import android.location.Location;
import android.location.LocationListener;
import android.location.LocationManager;
import android.os.Bundle;
import android.widget.TextView;
import android.view.Menu;
import android.view.MenuInflater;
import android.view.MenuItem;
import android.view.View;
import android.widget.LinearLayout;
import com.google.android.maps.MapView.LayoutParams;
import android.graphics.Bitmap;
import android.graphics.BitmapFactory;
import android.graphics.Canvas;
import android.graphics.Point;
import com.google.android.maps.GeoPoint;

public class WhereAmI extends MapActivity {

@Override
protected boolean isRouteDisplayed() {
return false;
}

MapController mapController;
MyPositionOverlay positionOverlay;
GeoPoint p;
MapView myMapView;
Boolean satview=false;
Boolean strview=true;
MyLocationOverlay compass;
static Boolean indoors=NameTracksClass.getIndoorSelection();

//add buttons to the activity
public boolean onCreateOptionsMenu(Menu menu){
super.onCreateOptionsMenu((android.view.Menu) menu);
MenuInflater m_signal = getMenuInflater();
m_signal.inflate(R.menu.menu,(android.view.Menu) menu);
return true;
}

public boolean onOptionsItemSelected(MenuItem item){
switch(item.getItemId()){
case R.id.Time_Button:
startActivity(new
Intent("com.paad.whereami.GETGSM SIGNALSTRENGTH"));
return true;

case R.id.Distance_Button:
//startActivity(new
Intent("com.paad.whereami.DISTANCE SIGNALSTRENGTH"));

Indoors();
//startActivity(new
Intent("com.paad.whereami.DISTANCE SIGNALSTRENGTH"));

return true;
}
}
}

```

```

        case R.id.SateliteView:
            //satview=true;
            //strview=false;
            myMapView.setSatellite(true);

            return true;
        case R.id.TrafficView:
            //satview=false;
            //strview=true;
            myMapView.setSatellite(false);
            myMapView.setTraffic(true);
            return true;

        case R.id.cellfinder1:
            startActivity(new
Intent("com.paad.whereami.cellinfo.CELLFINDER"));
            return true;

        case R.id.speedsignal:
            startActivity(new
Intent("com.paad.whereami.SPEEDSIGNAL"));
            }
            return false;
    }

//end putting buttons

@Override
public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.main);

//sthn prokeimenh koitazei to main kai paei kai xtypaei sto mymapview
myMapView= (MapView)findViewById(R.id.myMapView);

    mapController = myMapView.getController();

//pros8etw thn mpara gia na exw zoom
    LinearLayout zoomLayout = (LinearLayout)findViewById(R.id.zoom);
    View zoomView = myMapView.getZoomControls();

    zoomLayout.addView(zoomView,
        new LinearLayout.LayoutParams(
            LayoutParams.WRAP_CONTENT,
            LayoutParams.WRAP_CONTENT));

//8a oristei o tropos me ton opoio 8a ginetai h probolh toy xarth
    myMapView.setSatellite(false);
    myMapView.setStreetView(true);
    myMapView.setTraffic(false);

myMapView.displayZoomControls(false);
// einai to default zoom pou 8etw ston xarth
mapController.setZoom(17);

// Add the MyPositionOverlay
positionOverlay = new MyPositionOverlay();
List<Overlay> overlays = myMapView.getOverlays();
overlays.add(positionOverlay);

```

```

//add a working compass
compass =new MyLocationOverlay(this,myMapView);
myMapView.getOverlays().add(compass);

LocationManager locationManager;
String context = Context.LOCATION_SERVICE;
locationManager = (LocationManager)getSystemService(context);

//eisagw tis proypo8eseis pou 8elw na exei o gps provider
Criteria criteria = new Criteria();
criteria.setAccuracy(Criteria.ACCURACY_FINE);
criteria.setAltitudeRequired(false);
criteria.setBearingRequired(false);
criteria.setCostAllowed(true);
criteria.setPowerRequirement(Criteria.POWER_LOW);
String provider = locationManager.getBestProvider(criteria, true);

Location location = locationManager.getLastKnownLocation(provider);

updateWithNewLocation(location);

locationManager.requestLocationUpdates(provider, 0, 0,
                                     locationManager);

//import alert dialog into my application

AlertDialog alert = new AlertDialog.Builder(this).create();
alert.setTitle("Map View ");
alert.setMessage("Select the MapView you'd like to have");
alert.setButton("StreetView", new DialogInterface.OnClickListener() {
    public void onClick(DialogInterface dialog, int which) {
        // here you can add functions
        myMapView.setSatellite(false);
        strview = true;
        satview = false;
    }
});
alert.setButton2("SateliteView", new DialogInterface.OnClickListener() {
    @Override
    public void onClick(DialogInterface dialog, int which) {
        // TODO Auto-generated method stub
        myMapView.setSatellite(true);
        strview = true;
        satview = true;
    }
});

alert.show();

/*
*/
}

```

```

private final LocationListener locationListener = new LocationListener()
{
    public void onLocationChanged(Location location) {
        updateWithNewLocation(location);
    }

    public void onProviderDisabled(String provider){
        updateWithNewLocation(null);
    }

    public void onProviderEnabled(String provider){ }
    public void onStatusChanged(String provider, int status,
        Bundle extras){ }
};

private void updateWithNewLocation(Location location) {
    String latLongString;
    TextView myLocationText;
    myLocationText = (TextView)findViewById(R.id.myLocationText);
    String addressString = "No address found";

    if (location != null) {
        // Update my location marker
        positionOverlay.setLocation(location);

        // Update the map location.
        Double geoLat = location.getLatitude()*1E6;
        Double geoLng = location.getLongitude()*1E6;
        GeoPoint point = new GeoPoint(geoLat.intValue(),
            geoLng.intValue());

        mapController.animateTo(point);

        double lat = location.getLatitude();
        double lng = location.getLongitude();
        latLongString = "Lat:" + lat + "\nLong:" + lng;

        double latitude = location.getLatitude();
        double longitude = location.getLongitude();

        Geocoder gc = new Geocoder(this, Locale.getDefault());
        try {
            List<Address> addresses = gc.getFromLocation(latitude,
                longitude, 1);

            StringBuilder sb = new StringBuilder();
            if (addresses.size() > 0) {
                Address address = addresses.get(0);

                for (int i = 0; i < address.getMaxAddressLineIndex(); i++)
                    sb.append(address.getAddressLine(i)).append("\n");

                sb.append(address.getLocality()).append("\n");
                sb.append(address.getPostalCode()).append("\n");
                sb.append(address.getCountryName());
            }
            addressString = sb.toString();
        } catch (IOException e) {}
    } else {
        latLongString = "No location found";
    }
    myLocationText.setText("Your Current Position is:\n" +

```



```

        latLongString + "\n" + addressString);
    }

    @Override
    protected void onPause() {
        // TODO Auto-generated method stub
        super.onPause();
        compass.disableCompass();
    }

    @Override
    protected void onResume() {
        // TODO Auto-generated method stub
        super.onResume();
        compass.enableCompass();
    }

    public void Indoors(){
        AlertDialog alert2 = new AlertDialog.Builder(this).create();
        alert2.setTitle("BE CAREFULL ");
        alert2.setMessage("You have selected an indoors measurement
task so the distance activity is not going to " +
            "be so accurate and the loop is going to work with
time. Are you sure that you want to continue? "+
            "In case you are not please select the Time
Option.");
        alert2.setButton("Yes", new DialogInterface.OnClickListener() {
            public void onClick(DialogInterface dialog, int which) {
                // here you can add functions
                startActivity(new
Intent("com.paad.whereami.DISTANCESIGNALSTRENGTH"));
            }
        });
        alert2.setButton2("No", new DialogInterface.OnClickListener() {

            @Override
            public void onClick(DialogInterface dialog, int which) {
                // TODO Auto-generated method stub

            }
        });
        if (indoors==true){
            alert2.show();
            //startActivity(new
Intent("com.paad.whereami.DISTANCESIGNALSTRENGTH"));
        }else if (indoors==false){
            startActivity(new
Intent("com.paad.whereami.DISTANCESIGNALSTRENGTH"));
        }
    }

}

```

TCPClient.java

```

package com.paad.whereami;

import java.io.BufferedReader;
import java.io.BufferedOutputStream;

```

```

import java.io.ByteArrayOutputStream;
import java.io.File;
import java.io.FileInputStream;
import java.io.FileNotFoundException;
import java.io.FileOutputStream;
import java.io.IOException;
import java.net.Socket;

public class TCPclient {
    public static void clientsender(String IP, int Port) throws Exception {

        Socket clientSocket = null;
        BufferedOutputStream outToServer = null;
        //here is the connection part
        try {
            clientSocket = new Socket(IP, Port);
            outToServer = new
BufferedOutputStream(clientSocket.getOutputStream());
        } catch (IOException ex) {
            // Do exception handling
        }

        if (outToServer != null) {
            //pairnw to file
            File myFile = new
File("/sdcard/EXAMPLEDATA/samples_encrypted.marios");
            //ftiaxnw pinaka byte iso me to mege8os tou arxeiou
            byte[] mybytearray = new byte[(int) myFile.length()];
            FileInputStream fis = null;

            try {
                fis = new FileInputStream(myFile);
            } catch (FileNotFoundException ex) {
                // Do exception handling
            }

            BufferedInputStream bis = new BufferedInputStream(fis);

            try {
                //diabazw ton pinaka kai stelnw to periexomeno ston server
                bis.read(mybytearray, 0, mybytearray.length);
                outToServer.write(mybytearray, 0, mybytearray.length);
                outToServer.flush();
                outToServer.close();
                clientSocket.close();

                // File sent, exit the main method
                return;
            } catch (IOException ex) {
                // Do exception handling
            }
        }
    }
}

```

Encryption.java

```

package com.paad.whereami.database;

import javax.crypto.*;
import java.io.*;
import java.security.*;
import javax.crypto.spec.*;
import java.util.*;

public class Encryption {

String READYTOENCRYPT="/sdcard/temp/samplesforencryption.xml";

String GENERATEDFILE="/sdcard/temp/samples.xml";
String EXPORTEDFILE="/sdcard/EXAMPLEDATA/samples.xml";
String PASSWORD="strong_pass";

public void preparefile(){
    CopyFileClass.copy(GENERATEDFILE, READYTOENCRYPT);
    try{
        File file= new File(READYTOENCRYPT);
        BufferedWriter output = new BufferedWriter(new
FileWriter(file,true));
        output.write("</root>");
        output.newLine();
        output.flush();
    }
    catch(IOException e){}
}

public void encryptfile(String destpath){
    try {
        FileInputStream infile=new FileInputStream(READYTOENCRYPT);
        FileOutputStream outfile = new FileOutputStream(destpath);
        PBEKeySpec keySpec = new PBEKeySpec(PASSWORD.toCharArray());
        SecretKeyFactory keyFactory =
        SecretKeyFactory.getInstance("PBEWithMD5AndDES");
        SecretKey passwordKey = keyFactory.generateSecret(keySpec);

        byte[] salt = new byte[8];
        Random rnd = new Random();
        rnd.nextBytes(salt);
        int iterations = 100;
        PBEPParameterSpec parameterSpec = new PBEPParameterSpec(salt,
iterations);

        Cipher cipher = Cipher.getInstance("PBEWithMD5AndDES");
        cipher.init(Cipher.ENCRYPT_MODE, passwordKey, parameterSpec);

        outfile.write(salt);

        byte[] input = new byte[64];
        int bytesRead;
        while ((bytesRead = infile.read(input)) != -1)
        {
            byte[] output = cipher.update(input, 0, bytesRead);
            if (output != null) outfile.write(output);
        }
    }
}
}

```

```

        byte[] output = cipher.doFinal();
        if (output != null) outfile.write(output);

        infile.close();
        outfile.flush();
        outfile.close();
        File file= new File(READYTOENCRYPT);
        file.delete();

    } catch (Exception e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    }
}
}

```

ShowDataBaseContents.java

```

package com.paad.whereami;

import java.util.Arrays;
import java.util.List;

import android.app.Activity;
import android.app.AlertDialog;
import android.content.DialogInterface;
import android.content.Intent;
import android.os.Bundle;
import android.util.Log;
import android.view.Menu;
import android.view.MenuInflater;
import android.view.MenuItem;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.TextView;
import android.widget.Toast;

public class ShowDataBaseContents extends Activity{

    Button Delete_button;
    Button ShowDataBase_button;
    Button Export_button;
    Button Delete_Single_track;
    DataHelper datahelper;
    TextView output0721;
    AlertDialog alert;

```

```

String tracknameToDelete;
List<String> dblist_glob;
static CharSequence[] items;

public boolean onCreateOptionsMenu(Menu export_menu){
    super.onCreateOptionsMenu((android.view.Menu) export_menu);
    MenuInflater m_signal = getMenuInflater();
    m_signal.inflate(R.menu.export_menu, (android.view.Menu)
export_menu);
    return true;
}

public boolean onOptionsItemSelected(MenuItem item){
    switch(item.getItemId()){
        case R.id.Export_button:
            startActivity(new
Intent("com.paad.whereami.database.MANAGEDATA"));
            return true;
        case R.id.CellFinder_button:
            startActivity(new
Intent("com.paad.whereami.cellinfo.CELLFINDER"));
            return true;
        case R.id.Move_to_Show_data:
            startActivity(new
Intent("com.paad.whereami.SHOWDATABASECONTENTS"));
            return true;

        case R.id.whereami0721:
            startActivity(new
Intent("com.paad.whereami.FINDLOCBYADDRESS"));
            return true;
        case R.id.NameClassIntent:
            startActivity(new
Intent("com.paad.whereami.NAMETRACKSCLASS"));
            }
        return(super.onOptionsItemSelected(item));
    }
}

@Override
protected void onCreate(Bundle savedInstanceState) {
    // TODO Auto-generated method stub
    super.onCreate(savedInstanceState);
    setContentView(R.layout.show_data);

    this.datahelper = new DataHelper(this);
    Delete_button=(Button)findViewById(R.id.delete_button0721);

    ShowDataBase_button=(Button)findViewById(R.id.ShowDataBase_button0721
);
};

```

```

        Delete_Single_track =
(Button)findViewById(R.id.Delete_Via_trackname);

        output0721=(TextView)findViewById(R.id.out_text0721);
        //Export_button=(Button) findViewById(R.id.export_button0721);
        refresh();

        this.Delete_button.setOnClickListener(new
View.OnClickListener() {

            @Override
            public void onClick(View v) {
                // TODO Auto-generated method stub
                datahelper.deleteAll();
            }
        });

        this.ShowDataBase_button.setOnClickListener(new
View.OnClickListener() {

            @Override
            public void onClick(View v) {
                // TODO Auto-generated method stub
                List <String> names = datahelper.selectAll();

                StringBuilder sb = new StringBuilder();
                sb.append("\n"+"");
                for (String name : names) {
                    sb.append(name + "\n");
                }

                // Log.d("EXAMPLE", "names size - " +
names.size());

                output0721.setText(sb.toString());
            }
        });

        this.Delete_Single_track.setOnClickListener(new
View.OnClickListener() {

            @Override
            public void onClick(View v) {

                alert.show();
                // TODO Auto-generated method stub

                //datahelper.deleteTrack("2020111010005012OUTDOORS");
                //datahelper.XmlFile();
            }
        });

```

```

        /*
        alert = new AlertDialog.Builder(this).create();
        alert.setTitle("Delete Track ");
        alert.setMessage("Type the trackname of the data you want
to delete: ");
        final EditText input = new EditText(this);
        input.setText("Trackname");
        alert.setView(input);

        alert.setButton("Delete", new
DialogInterface.OnClickListener() {
            public void onClick(DialogInterface dialog, int which)
{
                // here you can add functions
                String trackname = input.getText().toString();

                datahelper.deleteTrack(trackname);

            }
        });
        alert.setButton2("Cancel", new
DialogInterface.OnClickListener() {

            @Override
            public void onClick(DialogInterface dialog, int
which) {

                // TODO Auto-generated method stub

            }
        });
        */

        AlertDialog.Builder builder = new AlertDialog.Builder(this);
        builder.setTitle("Pick the track you want to delete");
        builder.setSingleChoiceItems(items, -1, new
DialogInterface.OnClickListener() {
            public void onClick(DialogInterface dialog, int item) {
                Toast.makeText(getApplicationContext(),
items[item], Toast.LENGTH_SHORT).show();
                tracknameToDelete = String.valueOf(items[item]);

            }
        });
        builder.setPositiveButton("Delete",
        new DialogInterface.OnClickListener() {
            public void onClick(DialogInterface dialog, int id) {
                Toast.makeText(getApplicationContext(),
tracknameToDelete, Toast.LENGTH_SHORT).show();
                datahelper.deleteTrack(tracknameToDelete);
            }
        });
    }
}

```

```

        startActivity(getIntent()); finish();
    }
});
builder.setNegativeButton("Cancel",
    new DialogInterface.OnClickListener() {
        public void onClick(DialogInterface dialog, int id) {
        }
    });
alert = builder.create();
}

public void refresh(){
    dblist_glob = datahelper.selectTrack();
    items = dblist_glob.toArray(new
CharSequence[dblist_glob.size()]);
}
}

```

MyPositionOvelay.java

```

package com.paad.whereami;

import android.graphics.Canvas;
import android.graphics.Paint;
import android.graphics.Point;
import android.graphics.RectF;
import android.location.Location;
import android.view.MotionEvent;
import android.widget.Toast;

import com.google.android.maps.GeoPoint;
import com.google.android.maps.MapView;
import com.google.android.maps.Overlay;
import com.google.android.maps.Projection;

public class MyPositionOverlay extends Overlay {

    private final int mRadius = 5;

    Location location;

    public Location getLocation() {
        return location;
    }
    public void setLocation(Location location) {
        this.location = location;
    }
}

```



```
@Override
public boolean onTap(GeoPoint point, MapView mapView) {
    return false;
}

@Override
public void draw(Canvas canvas, MapView mapView, boolean shadow) {
    Projection projection = mapView.getProjection();

    if (shadow == false) {
        // Get the current location
        Double latitude = location.getLatitude()*1E6;
        Double longitude = location.getLongitude()*1E6;
        GeoPoint geoPoint;
        geoPoint = new
            GeoPoint(latitude.intValue(),longitude.intValue());

        // Convert the location to screen pixels
        Point point = new Point();
        projection.toPixels(geoPoint, point);

        RectF oval = new RectF(point.x - mRadius, point.y - mRadius,
                               point.x + mRadius, point.y + mRadius);

        // Setup the paint
        Paint paint = new Paint();
        paint.setARGB(250, 132, 240, 32);
        paint.setAntiAlias(true);
        paint.setFakeBoldText(true);
        // setup the color of the bullet and the size
        Paint backPaint = new Paint();
        backPaint.setARGB(175, 50, 50, 50);
        backPaint.setAntiAlias(true);

        RectF backRect = new RectF(point.x + 2 + mRadius,
                                   point.y - 3*mRadius,
                                   point.x + 65, point.y + mRadius);

        // Draw the marker
        canvas.drawOval(oval, paint);
        canvas.drawRoundRect(backRect, 5, 5, backPaint);
        canvas.drawText("Point",
                        point.x + 2*mRadius, point.y,
                        paint);
    }
    super.draw(canvas, mapView, shadow);
}

}
```

CellFinder.java

```
package com.paad.whereami.cellinfo;

import java.util.List;

import org.apache.http.HttpResponse;
import org.apache.http.client.HttpClient;
import org.apache.http.client.methods.HttpGet;
import org.apache.http.impl.client.DefaultHttpClient;
import org.apache.http.util.EntityUtils;

import android.app.Activity;
import android.content.Context;
import android.content.Intent;
import android.net.Uri;
import android.os.Bundle;
import android.telephony.CellLocation;
import android.telephony.NeighboringCellInfo;
import android.telephony.PhoneStateListener;
import android.telephony.SignalStrength;
import android.telephony.TelephonyManager;
import android.telephony.gsm.GsmCellLocation;
import android.view.View;
import android.widget.Button;
import android.widget.SeekBar;
import android.widget.TextView;
import android.widget.Toast;
import com.paad.whereami.R;
import com.paad.whereami.MyPhoneStateListener;

public class CellFinder extends Activity {

    String cid;
    String cell_id;
    String mccstr;
    String mncstr;
    String cinrstr;
    String rssi;
    String nettypestr;
    String lacstr;
    String signalnew;
    TelephonyManager telephonyManager;
    GsmCellLocation cellLocation;
    String networkTypeString;
    MyPhoneStateListener MyListener = new MyPhoneStateListener();
    boolean execloop = true;
    int lac;
    int cellid;
```

```

/** Called when the activity is first created. */
@Override
public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.cellfindermain);

    //retrieve a reference to an instance of TelephonyManager
    telephonyManager =
(TelephonyManager) getSystemService(Context.TELEPHONY_SERVICE);
    cellLocation =
(GsmCellLocation) telephonyManager.getCellLocation();
    telephonyManager.listen(MyListener,
PhoneStateListener.LISTEN_SIGNAL_STRENGTHS);

    int cid = cellLocation.getCid();
    int lac = cellLocation.getLac();

}

public class MyPhoneStateListener extends PhoneStateListener{

    int networktype;
    GsmCellLocation loc;
    String networkTypeString;
    String mySignal;

    // Get the Signal strength from the provider, each time there
is an update
    @Override
    public void onSignalStrengthsChanged(SignalStrength
signalStrength)
    {
        TextView textNettype = (TextView)
findViewById(R.id.cell_nettype);
        TextView textMCC = (TextView) findViewById(R.id.mcc);
        TextView textMNC = (TextView) findViewById(R.id.mnc);

        networktype = telephonyManager.getNetworkType();
        String networkOperator =
telephonyManager.getNetworkOperator();
        String mcc = networkOperator.substring(0, 3);
        String mnc = networkOperator.substring(3);
        textMCC.setText("MCC: " + mcc);
        textMNC.setText("MNC: " + mnc);
        mccstr = String.valueOf(mcc);

```

```

// cell_idstr= new Integer(cell_id).toString();

switch ( networktype ){
case TelephonyManager.NETWORK_TYPE_UNKNOWN:
networkTypeString = "UNKNOWN"; break;
case TelephonyManager.NETWORK_TYPE_GPRS:
networkTypeString = "GPRS"; break;
case TelephonyManager.NETWORK_TYPE_EDGE:
networkTypeString = "EDGE"; break;
case TelephonyManager.NETWORK_TYPE_UMTS:
networkTypeString = "UMTS";
case TelephonyManager.NETWORK_TYPE_CDMA:
networkTypeString = "CDMA"; break;
case TelephonyManager.NETWORK_TYPE_EVDO_0:
networkTypeString = "EVDO_0"; break;
case TelephonyManager.NETWORK_TYPE_EVDO_A:
networkTypeString = "EVDO_A"; break;
case TelephonyManager.NETWORK_TYPE_1xRTT:
networkTypeString = "1xRTT"; break;
case TelephonyManager.NETWORK_TYPE_HSDPA:
networkTypeString = "HSDPA"; break;
case TelephonyManager.NETWORK_TYPE_HSUPA:
networkTypeString = "HSUPA"; break;
case TelephonyManager.NETWORK_TYPE_HSPA:
networkTypeString = "HSPA"; break;
}

textNettype.setText("Network type: "+ networkTypeString);

//nettype2=networkTypeString;

if (signalStrength.isGsm()){
int mStrength = signalStrength.getGsmSignalStrength();
int dBm = 2*mStrength-113;
TextView textRSSI = (TextView)
findViewById(R.id.cell_rssi);
textRSSI.setText("RSSI: "+dBm);
}
else{
int strength = -1;
if (signalStrength.getEvdoDbm() < 0){
strength = signalStrength.getEvdoDbm();
TextView textRSSI = (TextView)
findViewById(R.id.cell_rssi);
textRSSI.setText("RSSI: "+strength);
}
else if (signalStrength.getCdmaDbm() < 0){
strength = signalStrength.getCdmaDbm();
TextView textRSSI = (TextView)
findViewById(R.id.cell_rssi);
}
}
}

```

```

        textRSSI.setText("RSSI: "+strength);
    }
    if (strength < 0){
        // convert to asu
        int rssi_signal = Math.round((strength +
113f));
        TextView textRSSI = (TextView)
findViewById(R.id.cell_rssi);
        textRSSI.setText("RSSI: "+ rssi_signal);
    }
}
super.onSignalStrengthsChanged(signalStrength);

mySignal =
String.valueOf(signalStrength.getGsmSignalStrength());
Toast.makeText(getApplicationContext(), "Measuring Cinr =
"
        +
String.valueOf(signalStrength.getGsmSignalStrength()),
Toast.LENGTH_SHORT).show();
String mysignal =
String.valueOf(signalStrength.getGsmSignalStrength());
TextView textCINR =
(TextView)findViewById(R.id.cell_cinr);
textCINR.setText("CINR: "+ mysignal);
neighbor();
print(lac, cellid);
}

@Override
public void onCellLocationChanged(CellLocation location) {
    // TODO Auto-generated method stub
    super.onCellLocationChanged(location);
    loc = (GsmCellLocation) location;
    lac = loc.getLac();
    cellid = loc.getCid();

    //cell_id = loc.getCid();
}
public void print(int Lac, int Cid){
    TextView textLAC = (TextView)
findViewById(R.id.cell_Lac);
    TextView textCID = (TextView)findViewById(R.id.cid);
    textCID.setText("Cell ID: "+ Cid);
    textLAC.setText("LAC: "+ Lac + "\n");
}
}

```

```

        public void neighbor(){
            TextView neighboring =
            (TextView)findViewById(R.id.neighboring);
            List<NeighboringCellInfo> neighboringList =
            telephonyManager.getNeighboringCellInfo();

            String stringNeighboring = "Neighboring List- Lac : Cid :
            RSSI\n";
            for(int i=0; i < NeighboringList.size(); i++){

                String dBm;
                int rssi = NeighboringList.get(i).getRssi();
                if(rssi == NeighboringCellInfo.UNKNOWN_RSSI){
                    dBm = "Unknown RSSI";
                }else{
                    dBm = String.valueOf(-113 + 2 * rssi) + " dBm";
                }

                stringNeighboring = stringNeighboring
                + String.valueOf(NeighboringList.get(i).getLac()) + " : "
                + String.valueOf(NeighboringList.get(i).getCid()) + " : "
                + dBm + "\n";
            }

            neighboring.setText(stringNeighboring);
        }

        public void MccMncCidLac(){

        }

    }

```

DataHelper.java

```

package com.paad.whereami;

import android.content.Context;
import android.database.Cursor;
import android.database.SQLException;
import android.database.sqlite.SQLiteDatabase;
import android.database.sqlite.SQLiteException;
import android.database.sqlite.SQLiteOpenHelper;
import android.database.sqlite.SQLiteStatement;
import android.util.Log;
import com.paad.whereami.XMLwriter;

import java.util.ArrayList;
import java.util.List;

public class DataHelper{

```

```

public List<String> getTrack_list() {
    return track_list;
}

public void setTrack_list(List<String> track_list) {
    this.track_list = track_list;
}

private static final String DATABASE_NAME = "example.db";
private static final int DATABASE_VERSION = 12;
private static final String TABLE_NAME = "table1";
String EXPORTEDFILE="/sdcard/exampladata/samples.xml";

private Context context;
private SQLiteDatabase db;
private static SQLiteDatabase mydb;
XMLwriter pako = new XMLwriter();
List<String> track_list;

private SQLiteStatement insertStmt;

private static final String INSERT = "insert into " + TABLE_NAME +
"(trackname, latitude, longitude, signalstrength, cinr, time, MCC, MNC,
LAC, type, networktype, cellid, speed, Neighborcell1, N1_cellid, N1_rssi,
Neighborcell2, N2_cellid, N2_rssi, Neighborcell3, N3_cellid, N3_rssi,
Neighborcell4, N4_cellid, N4_rssi, Neighborcell5, N5_cellid, N5_rssi,
Neighborcell6, N6_cellid, N6_rssi, Neighborcell7, N7_cellid, N7_rssi,
Neighborcell8, N8_cellid, N8_rssi) values (?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?,
?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ? )";

public DataHelper(Context context) {
    this.context = context;
    OpenHelper openHelper = new OpenHelper(this.context);
    this.db = openHelper.getWritableDatabase();
    this.insertStmt = this.db.compileStatement(INSERT);
}

public SQLiteDatabase getDb() {
    return db;
}

public long insert(String trackname, String latitude, String longitude,
String signalstrength,String cinr, String time, String MCC, String MNC,
String LAC, String type, String networktype, String cellid, String speed,
String Neighborcell1, String N1_cellid, String N1_rssi, String
Neighborcell2, String N2_cellid, String N2_rssi, String Neighborcell3,
String N3_cellid, String N3_rssi, String Neighborcell4, String N4_cellid,
String N4_rssi, String Neighborcell5, String N5_cellid, String N5_rssi,

```

```

String Neighborcell6, String N6_cellid, String N6_rssi, String
Neighborcell7, String N7_cellid, String N7_rssi, String Neighborcell8,
String N8_cellid, String N8_rssi ) {
    this.insertStmt.bindString(1, trackname);
    this.insertStmt.bindString(2, latitude);
    this.insertStmt.bindString(3, longitude);
    this.insertStmt.bindString(4, signalstrength);
    this.insertStmt.bindString(5, cinr);
    this.insertStmt.bindString(6, time);
    this.insertStmt.bindString(7, MCC);
    this.insertStmt.bindString(8, MNC);
    this.insertStmt.bindString(9, LAC);
    this.insertStmt.bindString(10, type);
    this.insertStmt.bindString(11, networktype);
    this.insertStmt.bindString(12, cellid);
    this.insertStmt.bindString(13, speed);
    this.insertStmt.bindString(14, Neighborcell1);
    this.insertStmt.bindString(15, N1_cellid);
    this.insertStmt.bindString(16, N1_rssi);
    this.insertStmt.bindString(17, Neighborcell2);
    this.insertStmt.bindString(18, N2_cellid);
    this.insertStmt.bindString(19, N2_rssi);
    this.insertStmt.bindString(20, Neighborcell3);
    this.insertStmt.bindString(21, N3_cellid);
    this.insertStmt.bindString(22, N3_rssi);
    this.insertStmt.bindString(23, Neighborcell4);
    this.insertStmt.bindString(24, N4_cellid);
    this.insertStmt.bindString(25, N4_rssi);
    this.insertStmt.bindString(26, Neighborcell5);
    this.insertStmt.bindString(27, N5_cellid);
    this.insertStmt.bindString(28, N5_rssi);
    this.insertStmt.bindString(29, Neighborcell6);
    this.insertStmt.bindString(30, N6_cellid);
    this.insertStmt.bindString(31, N6_rssi);
    this.insertStmt.bindString(32, Neighborcell7);
    this.insertStmt.bindString(33, N7_cellid);
    this.insertStmt.bindString(34, N7_rssi);
    this.insertStmt.bindString(35, Neighborcell8);
    this.insertStmt.bindString(36, N8_cellid);
    this.insertStmt.bindString(37, N8_rssi);
    return this.insertStmt.executeInsert();
}

public void deleteAll() {
    this.db.delete(TABLE_NAME, null, null);
}

public void deleteTrack(String NameToDelete){
    // this.db.execSQL("DELETE FROM table1 WHERE trackname =
20111010000657OUTDOORS");
    db.delete(TABLE_NAME, "trackname" + "=?", new String[]
{NameToDelete});
}

```



```

}

public void XmlFile(){

    List<String> list = new ArrayList<String>();
    //pako.writeheader("/sdcard/temp/samplestestmarios8.xml");
    Cursor cursor = this.db.query(TABLE_NAME, new String[] {
"trackname", "latitude", "longtitude", "signalstrength","cinr","time",
"MCC", "MNC", "LAC", "type", "networktype","cellid","speed",
"Neighborcell1", "N1_cellid", "N1_rssi", "Neighborcell2", "N2_cellid",
"N2_rssi", "Neighborcell3", "N3_cellid", "N3_rssi", "Neighborcell4",
"N4_cellid", "N4_rssi", "Neighborcell5", "N5_cellid", "N5_rssi",
"Neighborcell6", "N6_cellid", "N6_rssi", "Neighborcell7", "N7_cellid",
"N7_rssi", "Neighborcell8", "N8_cellid", "N8_rssi" }, null, null, null,
null, "id desc");
    if (cursor.moveToFirst()) {
        do {

                pako.writing(EXPORTEDFILE, cursor.getString(0),
cursor.getString(1), cursor.getString(2), cursor.getString(3),
cursor.getString(4), cursor.getString(5), cursor.getString(6),
cursor.getString(7), cursor.getString(8), cursor.getString(9),
cursor.getString(10),
cursor.getString(11),cursor.getString(12),cursor.getString(13),cursor.getSt
ring(14),cursor.getString(15), cursor.getString(16), cursor.getString(17),
cursor.getString(18), cursor.getString(19), cursor.getString(20),
cursor.getString(21), cursor.getString(22), cursor.getString(23),
cursor.getString(24), cursor.getString(25), cursor.getString(26),
cursor.getString(27), cursor.getString(28), cursor.getString(29),
cursor.getString(30),cursor.getString(31), cursor.getString(32),
cursor.getString(33), cursor.getString(34), cursor.getString(35),
cursor.getString(36));

        } while (cursor.moveToNext());
    }

    if (cursor != null && !cursor.isClosed()) {
        cursor.close();
    }

    //pako.writelast("/sdcard/temp/samplestestmarios7.xml");
}

public List<String> selectAll() {
    List<String> list = new ArrayList<String>();
    Cursor cursor = this.db.query(TABLE_NAME, new String[] { "trackname",
"latitude", "longtitude", "signalstrength","cinr","time", "MCC", "MNC",
"LAC", "type", "networktype","cellid","speed", "Neighborcell1",
"N1_cellid", "N1_rssi", "Neighborcell2", "N2_cellid", "N2_rssi",
"Neighborcell3", "N3_cellid", "N3_rssi", "Neighborcell4", "N4_cellid",
"N4_rssi", "Neighborcell5", "N5_cellid", "N5_rssi", "Neighborcell6",
"N6_cellid", "N6_rssi", "Neighborcell7", "N7_cellid", "N7_rssi",

```

```

"Neighborcell8", "N8_cellid", "N8_rssi" }, null, null, null, null, "id
desc");
    if (cursor.moveToFirst()) {
        do {
            list.add("Trackname: " + cursor.getString(0));
            list.add("Latitude: "+cursor.getString(1));
            list.add("Longitude: "+cursor.getString(2));
            list.add("SignalStrength: "+cursor.getString(3)+" dBm");
            list.add("CINR: "+cursor.getString(4));
            list.add("Time: "+cursor.getString(5));
            list.add("MCC: "+cursor.getString(6));
            list.add("MNC: "+cursor.getString(7));
            list.add("LAC: "+cursor.getString(8));
            list.add("Type: "+ cursor.getString(9));
            list.add("Network Type: "+ cursor.getString(10));
            list.add("Cell id: " + cursor.getString(11));
            list.add("Speed: "+ cursor.getString(12));
            list.add(cursor.getString(13));
            list.add("Neighbor_Cell_1_ID: "+cursor.getString(14));
            list.add("Neighbor_Cell_1_RSSI: "+cursor.getString(15)+" dBm");
            list.add("-----");
            list.add(cursor.getString(16));
            list.add("Neighbor_Cell_2_ID: "+cursor.getString(17));
            list.add("Neighbor_Cell_2_RSSI: "+cursor.getString(18)+" dBm");
            list.add("-----");
            list.add(cursor.getString(19));
            list.add("Neighbor_Cell_3_ID: "+cursor.getString(20));
            list.add("Neighbor_Cell_3_RSSI: "+cursor.getString(21)+" dBm");
            list.add("*****");
        } while (cursor.moveToNext());
    }
    if (cursor != null && !cursor.isClosed()) {
        cursor.close();
    }
    return list;
}

public List<String> selectTrack(){
    track_list = new ArrayList<String>();
    // Cursor cursor = this.db.query(TABLE_NAME, new String[] {
"trackname", "latitude", "longitude", "signalstrength","cinr","time",
"MCC", "MNC", "LAC", "type", "networktype","cellid","speed",
"Neighborcell1", "N1_cellid", "N1_rssi", "Neighborcell2", "N2_cellid",
"N2_rssi", "Neighborcell3", "N3_cellid", "N3_rssi", "Neighborcell4",
"N4_cellid", "N4_rssi", "Neighborcell5", "N5_cellid", "N5_rssi",
"Neighborcell6", "N6_cellid", "N6_rssi", "Neighborcell7", "N7_cellid",
"N7_rssi", "Neighborcell8", "N8_cellid", "N8_rssi" }, null, null, null,
null, "id desc");
    // this.db.execSQL("SELECT DISTINCT trackname FROM table1");

    Cursor cursor = this.db.query(true, TABLE_NAME, new String[]
{"trackname"}, null, null, null, null, "id desc", null);
    if ((cursor.moveToFirst())) {

```

```

        do {
            track_list.add(cursor.getString(0));
        } while (cursor.moveToNext());
    }
    if (cursor != null && !cursor.isClosed()) {
        cursor.close();
    }
    return track_list;
}

private static class OpenHelper extends SQLiteOpenHelper {

    OpenHelper(Context context) {
        super(context, DATABASE_NAME, null, DATABASE_VERSION);
    }

    @Override
    public void onCreate(SQLiteDatabase db) {
        db.execSQL("CREATE TABLE " + TABLE_NAME + " (id INTEGER PRIMARY
KEY, trackname TEXT, latitude TEXT, longitude TEXT, signalstrength TEXT,
cinr TEXT, time TEXT, MCC TEXT, MNC TEXT, LAC TEXT, type TEXT, networktype
TEXT, cellid TEXT, speed TEXT, Neighborcell1 TEXT, N1_cellid TEXT, N1_rssi
TEXT, Neighborcell2 TEXT, N2_cellid TEXT, N2_rssi TEXT, Neighborcell3 TEXT,
N3_cellid TEXT, N3_rssi TEXT, Neighborcell4 TEXT, N4_cellid TEXT, N4_rssi
TEXT, Neighborcell5 TEXT, N5_cellid TEXT, N5_rssi TEXT, Neighborcell6 TEXT,
N6_cellid TEXT, N6_rssi TEXT, Neighborcell7 TEXT, N7_cellid TEXT, N7_rssi
TEXT, Neighborcell8 TEXT, N8_cellid TEXT, N8_rssi TEXT )");
    }

    @Override
    public void onUpgrade(SQLiteDatabase db, int oldVersion, int
newVersion) {
        Log.w("Example", "Upgrading database, this will drop tables and
recreate.");
        db.execSQL("DROP TABLE IF EXISTS " + TABLE_NAME);
        onCreate(db);
    }

    private boolean checkDataBase(){

        SQLiteDatabase checkDB = null;

        try{
            String myPath = "/data/com.paad.whereami/databases/example.db";
            checkDB = SQLiteDatabase.openDatabase(myPath, null,
SQLiteDatabase.OPEN_READONLY);

        }catch(SQLiteException e){
            return false;
            //database doesn't exist yet.
        }
    }
}

```

```

        return true;
    }

    public SQLiteDatabase openDataBase() throws SQLException {
        // Open the database
        String myPath = "/data/com.paad.whereami/databases/example.db";
        mydb = SQLiteDatabase.openDatabase(myPath, null,
            SQLiteDatabase.OPEN_READONLY);
        return mydb;
    }

    public SQLiteDatabase getDb() {
        return mydb;
    }
}

/*
@Override
public void onCreate(SQLiteDatabase db) {
    // TODO Auto-generated method stub
}

@Override
public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {
    // TODO Auto-generated method stub
}
}*/
}
}

```

Email.java

```

package com.paad.whereami;

import java.util.Properties;

import javax.activation.DataHandler;
import javax.activation.DataSource;
import javax.activation.FileDataSource;
import javax.mail.BodyPart;
import javax.mail.Message;
import javax.mail.Multipart;
import javax.mail.Session;
import javax.mail.Transport;
import javax.mail.internet.InternetAddress;
import javax.mail.internet.MimeBodyPart;
import javax.mail.internet.MimeMessage;
import javax.mail.internet.MimeMultipart;

import android.app.Activity;
import android.os.Bundle;
import android.util.Log;

```

```

import android.view.View;
import android.widget.Button;
import android.widget.EditText;

public class Email extends Activity {
    EditText SenderAddress;
    EditText ReceiverAddress;
    EditText SenderPass;
    static String senderstr;
    static String receiverstr;
    static String passstr;
    String GENERATEDFILE="/sdcard/temp/samples.xml";

    /** Called when the activity is first created. */
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.email);

        final Button send = (Button)
this.findViewById(R.id.send_email_button);
        SenderAddress = (EditText) this.findViewById(R.id.emailsender);
        ReceiverAddress = (EditText)
this.findViewById(R.id.emailreceiver);
        SenderPass = (EditText) this.findViewById(R.id.passsender);

        send.setOnClickListener(new View.OnClickListener() {
            public void onClick(View v) {
                // TODO Auto-generated method stub

                try {
                    senderstr =
SenderAddress.getText().toString();
                    receiverstr =
ReceiverAddress.getText().toString();
                    passstr = SenderPass.getText().toString();
                    sendmail();

                    //GmailSender sender = new
GmailSender(senderstr, passstr);
                    //sender.sendMail("whereAmI", "Hello
argentina", senderstr, receiverstr);
                } catch (Exception e) {
                    Log.e("SendMail", e.getMessage(), e);
                }
                sendmail();
            }
        });
    }
}

```

```

    }

    public static void sendmail() {

        String host = "smtp.gmail.com";
        String from = "whereami.application@gmail.com" ;//senderstr;
        String pass = "marios.241865";
        //String filename = "/Users/marioskotsiandres/Desktop/marios.txt";
        String filename = "/sdcard/exampledata/file.xml";
        //get system properties
        Properties props = System.getProperties();
        props.put("mail.smtp.starttls.enable", "true"); // added this line
        props.put("mail.smtp.host", host);
        props.put("mail.smtp.user", from);
        props.put("mail.smtp.password", pass);
        props.put("mail.smtp.port", "587");
        props.put("mail.smtp.auth", "true");
        String[] to = {"whereami.application@gmail.com"}; // added this
line
        //create a new session
        Session session = Session.getDefaultInstance(props, null);
        //System.out.println("exw kanei kai th dhmiourgia tou instance");

        MimeMessage message = new MimeMessage(session);
        try {
            message.setFrom(new InternetAddress(from));
        } catch (Exception e) {
        }

        InternetAddress[] toAddress = new InternetAddress[to.length];

        // To get the array of addresses
        for (int i = 0; i < to.length; i++) { // changed from a while loop
            try {
                toAddress[i] = new InternetAddress(to[i]);
            } catch (Exception e) {
                System.out.println("epesa se exception 1");
            }
        }
        //System.out.println(Message.RecipientType.TO);

        for (int i = 0; i < toAddress.length; i++) { // changed from a
while loop
            try {
                message.addRecipient(Message.RecipientType.TO,
toAddress[i]);
            } catch (Exception e) {
                // System.out.println("epesa se exception 2");
            }
        }
        try {
            message.setSubject("sending in a group");

```

```

        //message.setText("Sto esteila apo mia efarmogh pou
eftiaksa!!!!");

//newly added code
        BodyPart messageBodyPart = new MimeBodyPart();
        messageBodyPart.setText("Here's the file");
//begin the file part
        Multipart multipart = new MimeMultipart();
        multipart.addBodyPart(messageBodyPart);

        messageBodyPart = new MimeBodyPart();

        DataSource source = new FileDataSource(filename);

        messageBodyPart.setDataHandler(new DataHandler(source));

        messageBodyPart.setFileName(filename);

        multipart.addBodyPart(messageBodyPart);

        message.setContent(multipart);

        Transport transport = session.getTransport("smtp");
        transport.connect(host, from, pass);
        transport.sendMessage(message, message.getAllRecipients());
        transport.close();
        //System.out.println("Mail Sent Successfully");

    } catch (Exception e) {
    }
}
}

```

FindLocByAddress.java

```

package com.paad.whereami;

import java.io.IOException;
import java.util.List;
import java.util.Locale;

import android.app.AlertDialog;
import android.content.Context;
import android.content.DialogInterface;
import android.content.Intent;
import android.graphics.Bitmap;
import android.graphics.BitmapFactory;
import android.graphics.Canvas;
import android.graphics.Paint;

```

```
import android.graphics.Point;
import android.graphics.RectF;
import android.location.Address;
import android.location.Criteria;
import android.location.Geocoder;
import android.location.Location;
import android.location.LocationListener;
import android.location.LocationManager;
import android.os.Bundle;
import android.view.KeyEvent;
import android.view.Menu;
import android.view.MenuInflater;
import android.view.MenuItem;
import android.view.View;
import android.view.ViewGroup.LayoutParams;
import android.view.inputmethod.EditorInfo;
import android.view.inputmethod.InputMethodManager;
import android.widget.EditText;
import android.widget.LinearLayout;
import android.widget.TextView;
import android.widget.Toast;

import com.google.android.maps.GeoPoint;
import com.google.android.maps.MapActivity;
import com.google.android.maps.MapController;
import com.google.android.maps.MapView;
import com.google.android.maps.Overlay;
import com.google.android.maps.Projection;

public class FindLocByAddress extends MapActivity {

    class MapOverlay extends com.google.android.maps.Overlay
    {
        @Override
        public boolean draw(Canvas canvas, MapView mapView, boolean
shadow, long when)
        {
            super.draw(canvas, mapView, shadow);

            //---translate the GeoPoint to screen pixels---
            Point screenPts = new Point();
            mapView.getProjection().toPixels(p, screenPts);
            Point point = new Point();

            //---add the marker---
            Bitmap bmp = BitmapFactory.decodeResource(getResources(),
R.drawable.icon);
            canvas.drawBitmap(bmp, screenPts.x-20, screenPts.y-50,
null);

            return true;
        }
    }
}
```



```
    }  
}  
  
public static Double getGeolat() {  
    return geolat;  
}  
  
public void setGeolat(Double geolat) {  
    this.geolat = geolat;  
}  
  
public static Double getGeolng() {  
    return geolng;  
}  
  
public void setGeolng(Double geolng) {  
    this.geolng = geolng;  
}  
  
public static GeoPoint getP() {  
    return p;  
}  
  
public void setP(GeoPoint p) {  
    this.p = p;  
}  
  
MapView mapView;  
MapController mapController;  
//JustFindMeOverlay positionOverlay;  
FindAddressByLocOverLay positionOverlay;  
static GeoPoint p;  
String Address;  
EditText mapSearchBox;  
GeoPoint geo;  
boolean marios=true;  
static Double geolat;  
static Double geolng;  
String in;  
  
public boolean onCreateOptionsMenu(Menu findaddress_menu){
```

```

        super.onCreateOptionsMenu((android.view.Menu)
findaddress_menu);
        MenuInflater m_signal = getMenuInflater();
        m_signal.inflate(R.menu.findaddress_menu,(android.view.Menu)
findaddress_menu);
        return true;
    }

    public boolean onOptionsItemSelected(MenuItem item){
        switch(item.getItemId()){

            case R.id.findaddress:
                AlertDialog          alert          =          new
AlertDialog.Builder(this).create();
                alert.setTitle("Insert Address");
                alert.setMessage("Please insert the address you'd like to
find bellow");
                final EditText input = new EditText(this);
                input.setText("marios");
                alert.setView(input);
                alert.setButton("OK", new DialogInterface.OnClickListener() {
whichButton) {
                    String value = input.getText().toString();
                    Address=value;
                    findLoc(value);
                }
            });

                alert.show();

        }
        return(super.onOptionsItemSelected(item));
    }

    @Override
    protected boolean isRouteDisplayed() {
        // TODO Auto-generated method stub
        return false;
    }

    @Override
    protected void onPause() {
        // TODO Auto-generated method stub
        super.onPause();
    }

    @Override
    protected void onResume() {
        // TODO Auto-generated method stub

```

```

        super.onResume();
    }

    /** Called when the activity is first created. */
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.findmeindex);

        mapView = (MapView) findViewById(R.id.mymMapView);
        mapController = mapView.getController();
        LinearLayout zoomLayout =
        (LinearLayout) findViewById(R.id.zoom);
        View zoomView = mapView.getZoomControls();
        mapView.setStreetView(true);
        // mapView.setSatellite(false);

        zoomLayout.addView(zoomView,
            new LinearLayout.LayoutParams(
                LayoutParams.WRAP_CONTENT,
                LayoutParams.WRAP_CONTENT));
        mapView.displayZoomControls(true);
        mapView.setSatellite(false);

        // mapController.setZoom(17);

        mapController = mapView.getController();
        //String coordinates[] = {"1.352566007", "103.78921587"};
        //double lat = Double.parseDouble(coordinates[0]);
        //double lng = Double.parseDouble(coordinates[1]);
        findLoc(Address);
        //p = new GeoPoint(
        //(int) (lat * 1E6),
        //(int) (lng * 1E6));

        mapController.animateTo(p);
        mapController.setZoom(17);
        mapView.invalidate();

        //---Add a location marker---
        MapOverlay mapOverlay = new MapOverlay();
        List<Overlay> listOfOverlays = mapView.getOverlays();
        listOfOverlays.clear();
        listOfOverlays.add(mapOverlay);

        mapView.invalidate();
    }

```

```

    }

    public void findLoc(String address){
        Geocoder geoCoder = new Geocoder(this, Locale.getDefault());
        if (address==null){
            address="Antifilou 47";
        }
        try {
            List<Address> addresses =
geoCoder.getFromLocationName(address, 5);
            String add = "";
            if (addresses.size() > 0) {
                p = new GeoPoint(
                    (int) (addresses.get(0).getLatitude() *
1E6),
                    (int) (addresses.get(0).getLongitude()
* 1E6));
                mapController.animateTo(p);
                mapView.invalidate();
            }
        } catch (IOException e) {
            e.printStackTrace();
        }
    }
}

```

CopyFile.java

```

package com.paad.whereami.database;

import java.io.File;
import java.io.FileInputStream;
import java.io.FileOutputStream;
import java.io.IOException;
import java.io.InputStream;
import java.io.OutputStream;

public class CopyFileClass {

    public static void copy (String sourcefile,String destinationfile)
    {

        InputStream inStream = null;
        OutputStream outStream = null;

```

```

try{

    File afile =new File(sourcefile);
    File bfile =new File(destinationfile);

    inStream = new FileInputStream(afile);
    outputStream = new FileOutputStream(bfile);

    byte[] buffer = new byte[1024];

    int length;
    //copy the file content in bytes
    while ((length = inStream.read(buffer)) > 0){

        outputStream.write(buffer, 0, length);

    }

    inStream.close();
    outputStream.close();

    //System.out.println("File is copied successful!");

}catch(IOException e){
    e.printStackTrace();
}
}

```

Dectypt.java

```

import javax.crypto.*;
import javax.crypto.spec.*;
import java.io.*;

public class Dectrypt {

    private static FileInputStream inFile;
    private static FileOutputStream outFile;
    private String password = "strong_pass";

    public void execute(String encryptedfilepath, String outputpath) {

        try {
            inFile = new FileInputStream(encryptedfilepath);
            outFile = new FileOutputStream(outputpath);

            PBEKeySpec keySpec = new PBEKeySpec(password.toCharArray());
            SecretKeyFactory keyFactory =
                SecretKeyFactory.getInstance("PBEWithMD5AndDES");
            SecretKey passwordKey = keyFactory.generateSecret(keySpec);

```

```

        byte[] salt = new byte[8];
        inFile.read(salt);
        int iterations = 100;

        PBEPParameterSpec parameterSpec = new PBEPParameterSpec(salt,
iterations);

        Cipher cipher = Cipher.getInstance("PBEWithMD5AndDES");
        cipher.init(Cipher.DECRYPT_MODE, passwordKey, parameterSpec);

        byte[] input = new byte[64];
        int bytesRead;
        while ((bytesRead = inFile.read(input)) != -1) {
            byte[] output = cipher.update(input, 0, bytesRead);
            if (output != null) {
                outFile.write(output);
            }
        }

        byte[] output = cipher.doFinal();
        if (output != null) {
            outFile.write(output);
        }

        inFile.close();
        outFile.flush();
        outFile.close();

    } catch (Exception e) {
    }
}
}

```

WhereAmI.java

```

import javax.swing.*;
import javax.swing.filechooser.*;
import java.awt.*;
import java.awt.event.*;
import java.io.*;
import java.io.File;
/**
 *
 * @author Exile
 */
public class Whereami_Server {
    private JFrame frame;

    public static void main(String[] args) {
        new Whereami_Server().buildGUI();
    }

    public void buildGUI() {
        setUpNetworking();
        frame = new JFrame();

        JLabel status = new JLabel("Waiting to receive file from
device...");
    }
}

```

```

//Make the buttons and 'register' the listeners:
JButton button2 = new JButton("Decrypt a File");
button2.addActionListener(new DecryptListener());

JButton button3 = new JButton("Insert XML Data in DB");
button3.addActionListener(new InsertListener());

//Put the buttons in a panel and put that panel in the frame's
ContentPane:
JPanel panel = new JPanel();
panel.add(status);

JPanel panel2 = new JPanel();
panel2.add(button2);
panel2.add(button3);

frame.getContentPane().add(BorderLayout.CENTER, panel2);
frame.getContentPane().add(BorderLayout.SOUTH, panel);

frame.setTitle("WHERE AM I - Android Application Server");
frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
frame.setSize(350,100);
frame.setLocationRelativeTo(null); //Place the window at the center
of the screen.
frame.setVisible(true);
}

class InsertListener implements ActionListener {
    // This "actionPerformed" method runs when someone clicks the
    "Insert XML Data in DB" button.
    public void actionPerformed(ActionEvent event) {
        // Make a file chooser and set its filters.
        JFileChooser chooser = new JFileChooser();
        chooser.setAcceptAllFileFilterUsed(false); //Disable "All
Files" filter.
        FileNameExtensionFilter filter = new
FileNameExtensionFilter("XML Files", "xml");
        chooser.setFileFilter(filter);

        int returnVal = chooser.showOpenDialog(frame);
        if(returnVal == JFileChooser.APPROVE_OPTION) {
            File file = chooser.getSelectedFile(); // Get the selected
file.

            String path = file.toString();
            new ReadXML().insert_into_db(path);
        }
    }
}

class DecryptListener implements ActionListener {
    // This "actionPerformed" method runs when someone clicks the
    "Decrypt a File" button.
    public void actionPerformed(ActionEvent event) {
        // Make a file chooser and set its filters.
        JFileChooser chooser = new JFileChooser();

        int returnVal = chooser.showOpenDialog(frame);
        if(returnVal == JFileChooser.APPROVE_OPTION) {

```

```
File encrypted = chooser.getSelectedFile(); // Get the selected
file.
    String encrypted_path = encrypted.toString();
    String decrypted_path = encrypted_path+".xml";
    new Decrypt().execute(encrypted_path, decrypted_path);
    }
} //close ConnectListener class

private void setUpNetworking() {
    try{
        ServerListen sl = new ServerListen();
        Thread thr = new Thread(sl);
        thr.start();
    }catch (Exception e){
        System.out.println("Can't create thread: ServerListen");
    }
}

class XMLFilter implements FilenameFilter {
    public boolean accept(File dir, String s) {
        if (s.endsWith(".xml") || s.endsWith(".XML")) {
            return true;
        }
        return false;
    }
}
}
```


7. Βιβλιογραφία - Αναφορές

- [1] Youngkon Lee, “QoS Metrics for Service Level Measurement for SOA Environment”
- [2] Ivica Kostanic, Nenad Mijatovic, “Measurement Based QoS Comparison of Cellular Communication Networks”
- [3] Tanenbaum, Andrew S.: computer Networks, 3rdEdition, Prentice-Hall, Inc.
- [4] R. Morawek, H. Oczelik, "UMTS Basic Network Architecture"
- [5] Jarmo Prokkola, Mikko Hanski, “Measuring WCDMA and HSDPA Delay Characteristics with QoSMeT”
- [6] ITU-T E800, “Terms and Definitions Related to Quality of Service Engineering”,1997
- [7] <http://technet.microsoft.com>
- [8] Yishay Mansour, “Jitter Control in QoS Networks”
- [9] www.cisco.com
- [10] Rob van den Brink, “Muse Test Suite, Test Methods”
- [11] RemtOpt Sales Brochure
- [12] Constantine Drovolis, Krishna Gummadi, ”Measurement Lab:Overview and an Invitation to the Research Community.
- [13] <http://www.6qm.org>
- [14] <http://www.achuver.com/Aspx/Main/Home.aspx>
- [15] Glen Dobson & Alophonso Sánchez-Macián, “Towards unified QoS/SLA ontologies.
- [16] www.zeroshell.net/eng/qos
- [17] www.cse.wustl.edu/~jain/
- [18] <http://gtnoise.net/projects/6-access-networks/5-nano>
- [19] www.M-Lab.com
- [20] Jianchang Yang, D. Manivannan “A Fault Tolerant Dynamic Channel Allocation Scheme for Enhancing QoS in Cellular Networks.
- [21] J. C. Francis and M. Abu El-Ata, Benchmarking Mobile Network QoS
- [22] Emmanuel Marilly, Olivier Martinot, Hélène Papini, Danny Goderis, SERVICE LEVEL AGREEMENTS: A MAIN CHALLENGE for Next Generation Networks
- [23] Majid Ghaderi and Raouf Boutaba, “Call Admission Control in Mobile Cellular”
- [24] Ying Xu, “Individual QoS versus Aggregate QoS: A loss performance study”
- [25] Paul Engelstad, Analysis of Qos in WLAN
- [26] Braun T , End to End Quality of Service Over Heterogeneous Networks
- [27] Orio Mula Valls, A practical retaining mechanism for network traffic classification in operational environments
- [28] Mukarram Bin Tariq, Network Access Neutrality Observatory
- [29] Covad, wireless service Agreement
- [30] Shuja Jamil, Performance Analysis of a Framework for Multi-Interfaced Service Level Agreements on Mobile Devices

-
- [31] Guidelines for Service Level Agreement between Mobile Operators and Carriers
- [32] Judith Myerson, Use SLAs in a Web services context, Part 1: Guarantee your Web service with a SLA, 2004
- [33] Dinesh C.Verma, Service Level Agreements on IP Networks, Proceedings of the IEEE, VOL. 92, NO. 9, September 2004
- [34] Effective Quality-of-Service Renegotiating Schemes for Streaming Video, EURASIP Journal on Applied Signal Processing 2004:2, 280–289
- [35] <http://www.pulsewan.com/>
- [36] <http://download.oracle.com>
- [37] <http://www.slateplate.com/>
- [38] Haibo Wang, Devedra Prasad, “End to End QoS Provisioning in UMTS Networks”
- [39] World Telecommunications Congress 2002 (WRC 2002), QoS Provisioning in Cellular Networks Based on Mobility Prediction Techniques.
- [40] www.ascom.com
- [41] www.anite.com/nemo
- [42] www.3gpp.org
- [43] www.agilent.com