



ΠΑΝΕΠΙΣΤΗΜΙΟ  
ΔΥΤΙΚΗΣ ΜΑΚΕΔΟΝΙΑΣ  
ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ  
ΠΛΗΡΟΦΟΡΙΚΗΣ & ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ

## **ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ**

Υλοποίηση Δικτυακού Ελεγκτή για Εφαρμογές Δικτύων Οριζόμενων από  
Λογισμικό

ΙΩΑΝΝΗΣ ΠΡΑΒΙΝΟΣ Α.Μ 738

ΕΠΙΒΛΕΠΟΝΤΕΣ ΚΑΘΗΓΗΤΕΣ:

ΛΟΥΤΑ ΜΑΛΑΜΑΤΗ & ΒΑΡΔΑΚΑΣ ΙΩΑΝΝΗΣ

ΚΟΖΑΝΗ, ΙΟΥΝΙΟΣ 2019





ΠΑΝΕΠΙΣΤΗΜΙΟ  
ΔΥΤΙΚΗΣ ΜΑΚΕΔΟΝΙΑΣ  
ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ  
ΠΛΗΡΟΦΟΡΙΚΗΣ & ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ

## **ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ**

Υλοποίηση Δικτυακού Ελεγκτή για Εφαρμογές Δικτύων Οριζόμενων από  
Λογισμικό

Implementation of a Controller for Software-Defined Networking (SDN) Applications

ΙΩΑΝΝΗΣ ΠΡΑΒΙΝΟΣ Α.Μ 738

ΕΠΙΒΛΕΠΟΝΤΕΣ ΚΑΘΗΓΗΤΕΣ:

ΛΟΥΤΑ ΜΑΛΑΜΑΤΗ & ΒΑΡΔΑΚΑΣ ΙΩΑΝΝΗΣ

ΚΟΖΑΝΗ, ΙΟΥΝΙΟΣ 2019



## Περίληψη

---

Η πέμπτη γενιά του δικτύου κινητής τηλεφωνίας θα υποστηρίξει ένα ευρύ φάσμα νέων εφαρμογών και υπηρεσιών, η οποία απαιτεί υψηλότερο ρυθμό δεδομένων, μειωμένη κατανάλωση ενέργειας ανά υπηρεσία, αξιόπιστη συνδεσιμότητα με πολύ χαμηλή λανθάνουσα κατάσταση και δυνατότητα χειρισμού πολλών συσκευών. Ωστόσο, με τη βοήθεια του OpenAirInterface5G (OAI), μπορούμε να οικοδομήσουμε και να υποστηρίξουμε ένα ανοιχτό οικοσύστημα που μπορεί να χρησιμοποιηθεί για γενικό σκοπό από το σύστημα LTE για το μέλλον του 5G.

Στη συγκεκριμένη διπλωματική εργασία θα χρησιμοποιηθεί το λογισμικό του OpenAirInterface5G για να προσομοιωθεί ένα πλήρως λειτουργικό δίκτυο με μία κεραία μετάδοσης (evolved NodeB), ένα εξοπλισμό χρήστη (User Equipment) και ένα ολοκληρωμένο κεντρικό δίκτυο (Evolved Packet Core) και να μελετηθούν ταχύτητες μετάδοσης UDP και TCP πακέτων μεταξύ δύο άκρων. Γενικά το OAI υποστηρίζει και τη χρήση εξοπλισμού όπως USRP B210 ή ExpressMIMO2 αλλά για τη συγκεκριμένη πειραματική διαδικασία ολόκληρο το πείραμα και κάθε στοιχείο του προσομοιώθηκε μεμονωμένα.

Στη λειτουργία προσομοίωσης, το OAI μπορεί να προσομοιώσει ένα πλήρες LTE δίκτυο, χρησιμοποιώντας το πακέτο oasisim (OpenAirInterface Simulator). Αρκετά eNB και οι UE μπορούν να προσομοιωθούν στην ίδια μηχανή ή σε διαφορετικές μηχανές μέσω ενσύρματης επικοινωνίας (Ethernet LAN). Τα φυσικά (PHY) και ραδιοφωνικά κανάλια είναι εξ ολοκλήρου προσομοιωμένα το οποίο είναι χρονικά δαπανηρό ή προσεγγίζονται σε μια κατάσταση αφαίρεσης των φυσικών καναλιών (PHY abstraction mode) η οποία είναι σημαντικά ταχύτερη. Και στις δύο περιπτώσεις, η λειτουργία προσομοίωσης τρέχει ολόκληρο το πρωτόκολλο (protocol stack), χρησιμοποιώντας τον ίδιο MAC (Media Access Control) κώδικα όπως και στο κανονικό πείραμα.

Λέξεις κλειδιά: OAI, UE, E-UTRAN, EPC, LTE, SDN



## **Abstract**

---

The fifth generation of the mobile network will support a wide range of new applications and services that require a higher data rate, reduced power consumption per service, reliable connectivity with very low latency and multi-device capability. However, with the help of OpenAirInterface5G(OAI), we can build and support an open ecosystem that can be used for general purpose by the LTE system for the future of 5G.

In this diploma thesis, OpenAirInterface5G software will be used to simulate a fully functional network with an evolved NodeB(eNB), an User Equipment(UE), and an Evolved Packet Core(EPC) and study the transfer rates of UDP and TCP packages between the two ends. In general, OAI also supports the use of equipment such as USRP B210 or ExpressMIMO2 but for the experimental process the whole experiment and each element has been simulated individually.

In simulation mode, OAI can simulate a complete LTE network using the oaisim (OpenAirInterface Simulator) package. Several eNBs and UEs can be simulated on the same machine or on different machines via an Ethernet LAN. Physical (PHY) and radio channels are totally simulated which is time consuming or approximated to a physical channel removal (PHY abstraction mode) which is significantly faster. In both cases, the simulation runs the entire protocol stack, using the same MAC (Media Access Control) code as in the regular simulation.

Keywords: OAI, UE, E-UTRAN, EPC, LTE, SDN





## **Ευχαριστίες**

---

Η διπλωματική αυτή εργασία είναι το κορύφωμα της φοιτητικής μου σταδιοδρομίας στο Πανεπιστήμιο Δυτικής Μακεδονίας και πιο συγκεκριμένα ως φοιτητής στο τμήμα Μηχανικών Πληροφορικής και Τηλεπικοινωνιών, έτσι θα ήθελα εκφράσω τις ευχαριστίες μου στα άτομα που ήταν δίπλα μου στα χρόνια φοίτησης μου.

Αρχικά θέλω να ευχαριστήσω τον επιβλέποντες καθηγητές της διπλωματικής εργασίας μου, Λούτα Μαλαματή και Βαρδάκα Ιωάννη, για τη συνεχόμενη καθοδήγηση και βοήθεια κατά τη διάρκεια της πρακτικής μου άσκησης στο εξωτερικό όπου είχε ως στόχο την εκτέλεση των προσομοιώσεων, που ήταν απαραίτητο να γίνουν για να παρθούν τα αποτελέσματα τους για αυτή τη διπλωματική αλλά και το συνολικό χρόνο που απαιτήθηκε για την ολοκλήρωση της διπλωματικής εργασίας μου.

Ακόμα θα ήθελα να ευχαριστήσω τους φίλους μου που ήταν δίπλα μου αυτά τα χρόνια και πολλοί από αυτούς αποτέλεσαν και συνεργάτες μου στη σχολή αυτή. Τέλος ένα τεράστιο ευχαριστώ στην οικογένεια μου για την συνεχόμενη και ακλόνητη παρουσία τους αλλά και υποστήριξη που μου παρείχαν ως φοιτητή αλλά και ως άνθρωπο.



## Περιεχόμενα

---

<b>Περίληψη</b>	4
<b>Abstract</b>	6
<b>Ευχαριστίες</b>	8
<b>Περιεχόμενα</b>	10
<b>Κατάλογος Σχημάτων</b>	12
1. Δίκτυα Κινητών Επικοινωνιών	14
1.1 Εξέλιξη των δικτύων κινητών επικοινωνιών	14
1.1.1 Πρώτη Γενιά(1G)	14
1.1.2 Δεύτερη Γενιά(2G)	14
1.1.3 Τρίτη Γενιά (3G)	15
1.1.4 Τέταρτη Γενιά(4G)	15
1.1.5 Πέμπτη Γενιά(5G)	15
1.2 Τι είναι το Long Term Evolution	16
1.2.2 Αρχιτεκτονική του Εξοπλισμού Χρήστη (UE)	17
1.2.3 Αρχιτεκτονική E-UTRAN	18
1.2.4 Αρχιτεκτονική του Evolved Packet Core(EPC)	19
1.3 Αρχιτεκτονική Ράδιο-Πρωτοκόλλων	21
1.3.1 Κανάλια Μεταφοράς - Επίπεδο Χρήστη	23
1.3.2 Κανάλια Μεταφοράς - Επίπεδο Ελέγχου	23
1.4 Κανάλια Μετάδοσης Δεδομένων	24
1.4.1 Λογικά Κανάλια	25
1.4.2 Κανάλια Μεταφοράς	25
1.4.3 Φυσικά κανάλια	25
1.5 Τι είναι το Software Defined Networking	25
1.5.1 Βασικά Χαρακτηριστικά SDN	25
1.5.2 Αρχιτεκτονική του SDN	26
2. Εγκατάσταση λογισμικού	28
2.1 Γενική Επεξήγηση του Open Air Interface 5G	28

2.2	Επεξήγηση του OpenAirInterface Scenario Descriptor	29
2.3	Στοιχεία Προσομοίωσης	30
2.4	Προεργασία Πειράματος	32
2.4.1	Χαμηλή λανθάνουσα κατάσταση Kernel(PC1)	32
2.4.2	Χαμηλή λανθάνουσα κατάσταση Kernel(PC2)	33
2.4.3	Διαχείριση της ενέργειας(PC1&PC2)	34
2.4.4	Απενεργοποίηση εναλλασσόμενης ταχύτητας επεξεργαστή(PC1&PC2)	35
2.5	Εγκατάσταση βασικών λειτουργιών	36
2.5.1	Εγκατάσταση OpenAirInterface	36
2.5.1.1	eNodeB	37
2.5.1.2	OpenAirInterface Scenario Descriptor	38
2.5.2	Εγκατάσταση EPC	40
2.5.2.1	Home Subscriber Server(HSS)	41
2.5.2.2	Mobility Management Entity(MME)	44
2.5.2.3	Packet Data Network Gateway και Service Gateway(SPGW)	46
3.	Σενάριο και Εκτέλεση της προσομοίωσης	50
3.1	Περιγραφή Σεναρίου	50
3.2	Ενέργειες πριν την εκτέλεση	52
3.3	Εκτέλεση Προσομοίωσης	53
3.3.1	Επιτυχία Σύνδεσης eNB, UE και EPC	54
3.4	Συνοπτική Εκτέλεση Προσομοίωσης	57
4.	Αποτελέσματα Προσομοίωσης	58
4.1	Μεταφορά Αρχείου-Περίπτωση TCP πακέτων	58
4.2	Μεταφορά Αρχείου-Περίπτωση UDP πακέτων	61
4.3	Προσομοίωση με τρεις εικονικούς χρήστες	63
4.4	Μελλοντική εξέλιξη	70
4.5	Τελικά Συμπεράσματα	70
	<b>Βιβλιογραφία</b>	72

## Κατάλογος Σχημάτων

---

Σχήμα 1.	EPS Δίκτυο .....	17
Σχήμα 2.	UE Αρχιτεκτονική.....	17
Σχήμα 3.	E-UTRAN .....	18
Σχήμα 4.	EPC δίκτυο.....	21
Σχήμα 5.	Αρχιτεκτονική Ράδιο-Πρωτοκόλλων .....	22
Σχήμα 6.	Επίπεδο Χρήστη.....	23
Σχήμα 7.	Επίπεδο Χρήστη.....	24
Σχήμα 8.	Αρχιτεκτονική του SDN .....	27
Σχήμα 9.	OpenAirInterface LTE Software Stack.....	29
Σχήμα 10.	Απεικόνιση σεναρίου προσομοίωσης .....	50
Σχήμα 11.	Μετάδοση αρχείου 5MB και 10MB σε 100x100 διαστάσεις(TCP) .....	59
Σχήμα 12.	Μετάδοση αρχείου 5MB και 10MB σε 1000x1000 διαστάσεις(TCP) .....	60
Σχήμα 13.	Μετάδοση αρχείου 5MB και 10MB σε 100x100 διαστάσεις(UDP).....	61
Σχήμα 14.	Μετάδοση αρχείου 5MB και 10MB σε 1000x1000 διαστάσεις(UDP).....	62



## 1. Δίκτυα Κινητών Επικοινωνιών

---

### 1.1 Εξέλιξη των δικτύων κινητών επικοινωνιών

Οι διαφορετικές γενιές και η εξέλιξη των δικτύων επικοινωνιών στις οποίες θα γίνει αναφορά παρακάτω, για τον διαχωρισμό τους ήταν τα χαρακτηριστικά τους όπως είναι τα πρωτόκολλα που ακολουθούνται σε κάθε γενιά, οι διαφορετικές τεχνολογίες ανά εποχή, ο αριθμός των συνδρομητών και η κινητικότητα τους ανά δίκτυο, οι ταχύτητες αποστολής και λήψης δεδομένων και η αγοράς που καθορίζει τη ζήτηση ανά χρονική περίοδο.

#### 1.1.1 Πρώτη Γενιά(1G)

Τα 1ης γενιάς δίκτυα ασύρματων κινητών τηλεπικοινωνιών πρώτο εμφανίστηκαν στις αρχές της δεκαετίας του 1980. Ήταν αναλογικά συστήματα τα οποία για πρώτη φορά προσέφεραν στον χρήστη υπηρεσίες φωνής σε προσιτές τιμές για την τότε περίοδο. Ως αρχική γενιά χαρακτηρίστηκε από τα πολλά και διαφορετικά ανά τόπους αναλογικά πρότυπα και από την ασυμβατότητα μεταξύ των προτύπων, και αυτό έφερε τον κατακερματισμό της αγοράς των κινητών τηλεπικοινωνιών, αδυναμία αισθητής μείωσης του κόστους των υπηρεσιών φωνής και στη αδυναμία παροχής διεθνούς κάλυψης.

#### 1.1.2 Δεύτερη Γενιά (2G)

Τα δίκτυα κινητών τηλεπικοινωνιών 2ης γενιάς κάνουν την εμφάνισή τους περίπου το 1991 με το πρότυπο παγκόσμιου συστήματος κινητών επικοινωνιών (Global System for Mobile communications-GSM) και αμέσως παρουσιάζονται σημαντικές διαφορές σε σχέση με τα δίκτυα 1<sup>ης</sup> γενιάς. Τα δίκτυα 2<sup>ης</sup> γενιάς είναι ψηφιακά συστήματα παρέχοντας έτσι σημαντικά καλύτερη απόδοση χρήσης του φάσματος συχνοτήτων και της μεταφοράς δεδομένων ενώ επιπλέον προσφέρεται και για πρώτη φορά η υπηρεσία των σύντομων γραπτών μηνυμάτων (Short Message Service-SMS) παρουσιάζοντας μεγάλη ανταπόκριση από την αγορά εκείνης της εποχής.

Τα σύστημα 2<sup>ης</sup> γενιάς αποτέλεσαν κομβικό ρόλο στην ραγδαία ανάπτυξη του κλάδου παροχής κινητών επικοινωνιών με την εκθετική αύξηση των συνδρομητών κινητής τηλεφωνίας που παρουσιάστηκε. Μέχρι και σήμερα τα 2G συστήματα χρησιμοποιούνται ευρέως σε αρκετά μέρη του κόσμου.

### 1.1.3 Τρίτη Γενιά (3G)

Τα δίκτυα κινητών τηλεπικοινωνιών 3<sup>ης</sup> γενιάς κάνουν την εμφάνισή τους στις αρχές της δεκαετίας του 2000. Με το πρότυπο International Mobile Telecommunications(IMT-2000) για τα 3G δίκτυα η Παγκόσμια Ένωση Τηλεπικοινωνιών (International Telecommunication Union-ITU) έκανε τη τοποθέτηση της σχετικά με τις λύσεις που απαιτούνταν για τα προβλήματα του 2G που είχαν προκύψει. Ειδικότερα η 3<sup>η</sup> γενιά δίνει λύσεις σχετικά με τους χαμηλούς ρυθμούς μετάδοσης δεδομένων και τη χρήση ασυμβίβαστων τεχνολογιών.

Από το IMT-2000 προβλέπονται υψηλοί ρυθμοί μετάδοσης δεδομένων της τάξης των 128kbps για κινητούς σταθμούς και 2 Mbps για σταθερές εφαρμογές. Σε γενικές γραμμές οι χρήστες των 3G δικτύων ήταν ένα μικρό ποσοστό σε σύγκριση με της δεύτερης γενιάς εξαιτίας των τεχνικών προβλημάτων που υπήρχαν με τις υπηρεσίες δεδομένων, λόγω της μεγάλης διακύμανσης που παρατηρείται στους ρυθμούς μετάδοσης των δεδομένων, κινούμενοι χρήστες ακόμα και μέσα στην ίδια περιοχή ενός 3G δικτύου γίνεται δύσκολο να δέχονται σωστά υπηρεσίες που χρειάζονται υψηλής ταχύτητας δεδομένα.

### 1.1.4 Τέταρτη Γενιά(4G)

Τα 4<sup>ης</sup> γενιάς δίκτυα ασύρματων κινητών τηλεπικοινωνιών κάνουν την εμφάνισή τους κατά τις αρχές της δεκαετίας του 2010. Στόχος τους με βάση και τις απαιτήσεις που έχουν θεσπιστεί από την ITU και το πρότυπο IMT-advanced είναι να κάνουν τις κινητές υπηρεσίες μία πραγματικότητα για το όλους τους χρήστες ξεπερνώντας έτσι τα στερεότυπα των δικτύων που τα χαρακτηρίζει ως μέσο φωνητικής επικοινωνίας.

Με τα 4G δίκτυα η επικοινωνία ξεπερνά το επίπεδο της φωνής και φτάνει το επίπεδο παροχής υπηρεσιών Internet για το τελικό χρήστη. Η ανάπτυξη και χρήση των 4G δικτύων έχει γίνει μεμονωμένα από άλλες τεχνολογίες και έτσι παρέχει υψηλής ταχύτητας μεταφορά δεδομένων και τη δυνατότητα για ασταμάτητη κινητικότητα του χρήστη για όποια υπηρεσία και αν χρησιμοποιεί.

### 1.1.5 Πέμπτη Γενιά(5G)

Τα δίκτυα κινητής τηλεφωνίας 5<sup>ης</sup> γενιάς αποτελούν την επόμενη άκρως σημαντική φάση στην εξέλιξη των κινητών επικοινωνιών. Οι πιο σύγχρονες τεχνολογίες για τις κινητές επικοινωνίες αναμένονται να εφαρμοστούν μέχρι τέλος του 2019. Βασικά χαρακτηριστικά τους είναι ότι θα παρέχουν εργαλεία και μηχανισμούς ελέγχου των συνδρομητών, μεγάλη μετάδοση των δεδομένων σε Gigabit επίπεδα τα οποία θα υποστηρίζουν σχεδόν 64.000 ταυτόχρονες συνδέσεις και ένας χρήστης θα μπορεί να λάβει μια καλύτερη και ταχύτερη λύση μέσω της απομακρυσμένης διαχείρισης που προσφέρεται και επίσης θα παρέχεται έως 25 Mbps ταχύτητα σύνδεσης.



## 1.2 Τι είναι το Long Term Evolution

Το σύστημα LTE, είναι ευρέως γνωστό και διαδεδομένο ως 4G ή B3G(Beyond 3G) και σχεδιάστηκε από την αρχή με στόχο την εξέλιξη της τεχνολογίας ραδιοπρόσβασης (Radio access) με σκοπό όλες οι υπηρεσίες να στηρίζονται στη μεταγωγή πακέτων και όχι στη μεταγωγή κυκλώματος όπως ισχύει στα προηγούμενα κινητά δίκτυα. Γενικότερα, ο όρος LTE αντιπροσωπεύει την εξέλιξη της ραδιοπρόσβασης και αποκαλείτε Evolved UMTS Terrestrial Radio Access Network (E-UTRAN).

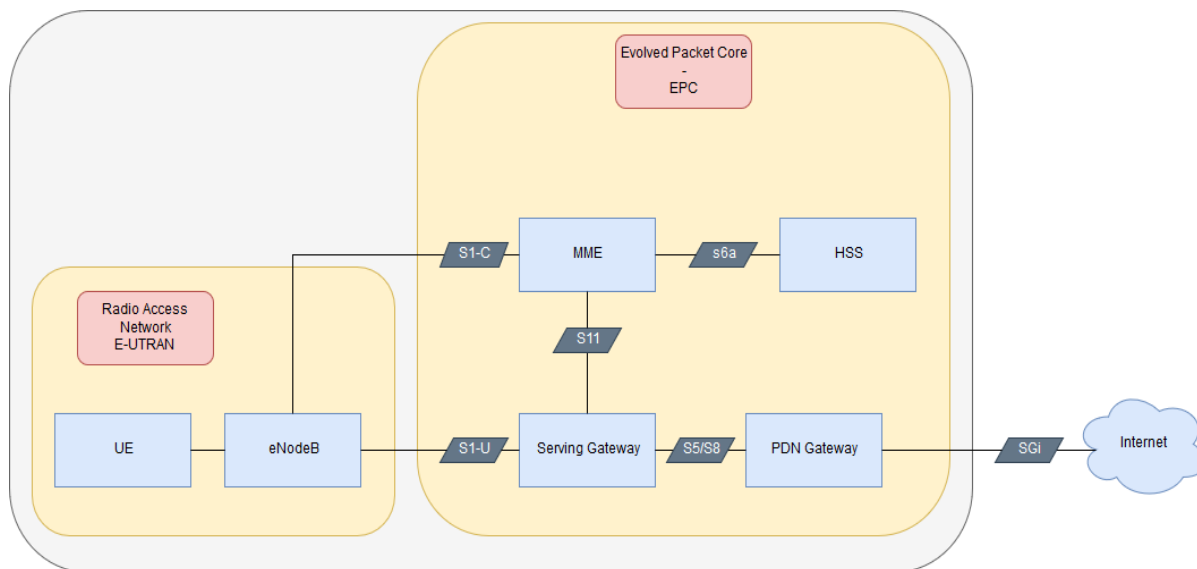
Από την άλλη πλευρά τα στοιχεία του δικτύου που δεν είναι συνδεδεμένα με τη ραδιοεπαφή αποκαλούνται System Architecture Evolution (SAE) στον οποίο εντάσσεται και το Evolved Packet Core (EPC) δίκτυο. Μαζί αυτές οι δύο ορολογίες δηλαδή LTE & SAE, αποτελούν το Evolved Packet System (EPS). Βασικά χαρακτηριστικά του δικτύου που υλοποιείται από το LTE είναι η επίπεδη αρχιτεκτονική αλλά και η χρήση του IP πρωτοκόλλου μαζί με τη μεταγωγή πακέτων για την επικοινωνία είναι λόγοι που βοηθούν να επιτευχθούν των στόχοι που έχουν τεθεί.

Οι στόχοι αυτοί είναι η μείωση καθυστέρησης στο επίπεδο του χρήστη αλλά και η πιο αποτελεσματική απόδοση της κινητικότητας. Ακόμα τεχνολογίες που συναντώνται στο φυσικό επίπεδο είναι η πολυπλεξία με διαίρεσης ορθογώνιας συχνότητας (Orthogonal Frequency Division Multiplexing-OFDM) και οι πολλαπλές εισαγωγές και πολλαπλές εξαγωγές(Multiple-Input Multiple-Output-MIMO) και είναι σημαντικές γιατί βοηθούν να ολοκληρωθούν απαιτήσεις του δικτύου αλλά και είναι παράγοντες που επιτρέπουν την ευέλικτη ανάπτυξη ραδιοφάσματος σε νέα ή ήδη υπάρχον φάσματα συχνοτήτων.

Βασικές υπηρεσίες που προσφέρονται από το LTE είναι η μετάδοση υψηλής ποιότητας περιεχομένου σε πραγματικό χρόνο είτε να αποθηκεύεται και να αναμεταδίδεται κάποια άλλη στιγμή και μέσω της εξέλιξης της τεχνολογίας MBMS σε Evolved MBMS είναι δυνατή η παροχή multicast υπηρεσιών ψηφιακού περιεχομένου με πιο πολλές προοπτικές σχετικά με τη χωρητικότητα και τον αριθμό των καναλιών που είναι διαθέσιμα.

### 1.2.1 Αρχιτεκτονική LTE

Η αρχιτεκτονική υψηλού επιπέδου του EPS αποτελείται κυρίως από τρία συστατικά: εξοπλισμό χρήστη (User Equipment-UE),το δίκτυο πρόσβασης (E-UTRAN) και το δίκτυο κορμού (EPC).Στο Σχήμα 1 φαίνεται ότι το EPC αποτελείται από πολλές λογικές οντότητες ενώ το δίκτυο πρόσβασης έχει μόνο ένα στοιχείο το evolved NodeB (eNB) το οποίο είναι άμεσα συνδεδεμένο με τους εξοπλισμούς χρήστη (UEs). Επιπρόσθετα αναφέρονται και οι διεπαφές μεταξύ των οντοτήτων.



Σχήμα 1. EPS Δίκτυο

### 1.2.2 Αρχιτεκτονική του Εξοπλισμού Χρήστη (UE)

Ο εξοπλισμός χρήστη είναι μια συσκευή που χρησιμοποιείται από έναν τελικό χρήστη για να επικοινωνεί απευθείας με το κινητό δίκτυο. Η κύρια συσκευή που χρησιμοποιείται για την επικοινωνία είναι συνήθως χειρός δηλαδή είτε smartphone είτε Laptop. Ακόμα συμπεριλαμβάνεται το δομοστοιχείο ταυτότητας συνδρομητή (Universal Subscriber Identity Module-USIM) το οποίο είναι διακριτό του UE. Η USIM είναι μία εφαρμογή που τοποθετείται στη κάρτα ολοκληρωμένου κυκλώματος (Universal integrated circuit card-UICC) η οποία είναι μια έξυπνη κάρτα που αποθηκεύει πληροφορίες συνδρομητών όπως τον αριθμό τηλεφώνου του χρήστη και την ταυτότητα του οικιακού δικτύου. Ο εξοπλισμός χρήστη είναι μία πλατφόρμα που παρέχει τη διεπαφή για τον τελικό χρήστη με σκοπό εφαρμογές όπως VoIP να μπορούν να χρησιμοποιηθούν για την πραγματοποίηση μιας φωνητικής κλήσης.



Σχήμα 2. UE Αρχιτεκτονική

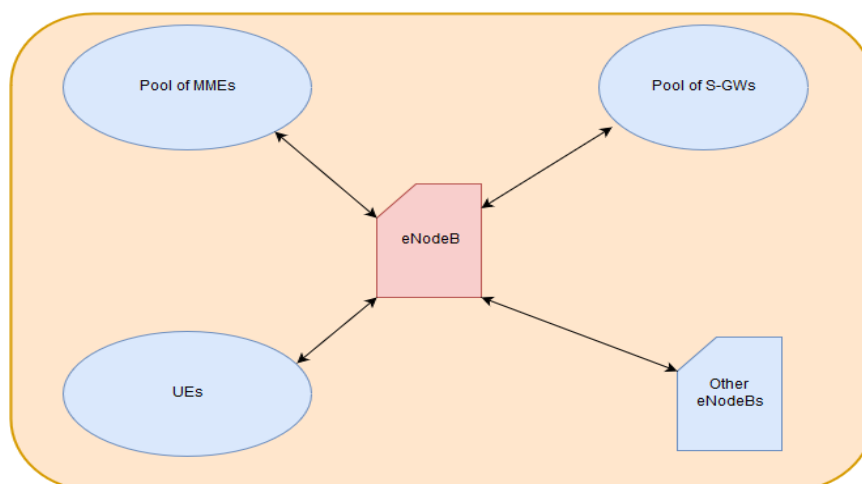
### 1.2.3 Αρχιτεκτονική E-UTRAN

Στο δίκτυο πρόσβασης η μόνη οντότητα είναι το eNodeB το οποίο είναι ένας σταθμός βάσης που είναι υπεύθυνος για τον έλεγχο όλων των ραδιο λειτουργιών που συνδέονται με το δίκτυο κορμού. Γενικά κάθε eNB είναι τοποθετημένος κοντά στις ραδιοκεραίες και σε όλη την περιοχή κάλυψης των δικτύων.

Αυτό γίνεται γιατί αποτελεί το σημείο επικοινωνίας μεταξύ του UE και του EPC. Για το UE είναι σημείο τερματισμού όλων των ραδιο-πρωτοκόλλων αλλά και κρυπτογραφεί και αποκρυπτογραφεί τα δεδομένα του ενώ για το EPC ξανά μεταδίδει αυτά τα δεδομένα στο ίδιο.

Η κύρια λειτουργία του eNB είναι ο έλεγχος και η χρήση της ραδιο-επαφής δηλαδή η κατανομή των πόρων βάση αιτήσεων, τη διαχείριση της κίνησης των δεδομένων και ο έλεγχος και ανάλυση μετρήσεων της έντασης του ραδιοκύματος. Το eNB ελέγχει επίσης το χαμηλό επίπεδο λειτουργίας όλων των κινητών τηλεφώνων, στέλνοντας μηνύματα σηματοδότησης όπως εντολές παράδοσης που σχετίζονται με αυτές τις ασύρματες μεταδόσεις.

Επίσης το γεγονός ότι κάθε στιγμή ένα eNodeB μπορεί να εξυπηρετεί πολλαπλούς UEs στην περιοχή κάλυψης του, ενώ αντίθετα κάθε UE μπορεί να είναι συνδεδεμένο με ένα μόνο eNodeB. Παράλληλα, γειτονικοί eNodeBs είναι αναγκαίο να είναι συνδεδεμένοι μεταξύ τους. Τέλος, κάθε στιγμή σε ένα UE προσφέρονται υπηρεσίες από ένα μόνο MME και S-GW, και ο eNodeB πρέπει να παρακολουθεί αυτή την επικοινωνία. Αυτό σημαίνει ότι ένας eNodeB είναι πιθανό να πρέπει να συνδεθεί με πολλές MMEs και S-GWs.



Σχήμα 3. E-UTRAN

Κάθε σταθμός βάσης είναι συνδεδεμένος με τον EPC μέσω της διεπαφής S1. Μπορεί επίσης να συνδεθεί σε κοντινούς σταθμούς βάσης από τη διεπαφή X2, η οποία χρησιμοποιείται κυρίως για σηματοδότηση και προώθηση πακέτων κατά τη διάρκεια της παράδοσης. Η διεπαφή X2 είναι προαιρετική, στο ότι η διεπαφή S1 μπορεί επίσης να χειριστεί όλες τις λειτουργίες της X2, έστω και έμμεσα και πιο αργά. Συνήθως, οι διεπαφές S1 και X2 δεν έχουν άμεσες φυσικές συνδέσεις, αντίθετα οι πληροφορίες δρομολογούνται σε μια βαθύτερη IP που βασίζεται στο δίκτυο μεταφοράς.

#### 1.2.4 Αρχιτεκτονική του Evolved Packet Core(EPC)

Οι οντότητες που αποτελούν το Δίκτυο Κορμού είναι οι παρακάτω:

##### Home Subscriber Server(HSS):

Είναι η κύρια βάση δεδομένων στην οποία βρίσκονται αποθηκευμένες οι πληροφορίες των χρηστών του δικτύου. Τα δεδομένα που περιέχονται στο HSS περιλαμβάνουν σχετικές πληροφορίες για τη συνδρομή του χρήστη καθώς και πληροφορίες σχετικά με την απόδοση του δικτύου. Επίσης μία υπηρεσία που μπορεί να είναι ενσωματωμένη στο HSS είναι το κέντρο ταυτοποίησης το οποίο παράγει δεδομένα για την ταυτοποίηση και τα κλειδιά ασφαλείας. Γενικότερα όλες οι διαδικασίες που σχετίζονται με αυτές τις λειτουργίες του HSS αλληλεπιδρούν με το MME, άρα η HSS είναι σε θέση να συνδέεται με κάθε MME σε ολόκληρο το δίκτυο έτσι ώστε ο χρήστης να έχει τη δυνατότητα να μετακινείται όπου θέλει.

##### Mobility Management Entity(MME):

Αντιπροσωπεύει την κύρια οντότητα του επιπέδου ελέγχου για να διαχειριστεί την πρόσβαση των χρηστών στο δίκτυο μέσω του E-UTRAN. Υπάρχουν περισσότερα από ένα MME για το δίκτυο και η επιλογή του καθενός (MME) γίνεται με βάση το γεωγραφική θέση των τερματικών σταθμών ή τα κριτήρια εξισορρόπησης φόρτωσης δεδομένων.

Οι κύριες λειτουργίες αυτής της οντότητας είναι οι εξής: Αυθεντικοποίηση και εξουσιοδότηση των χρηστών, διαχείριση της κινητικότητας των χρηστών και των συσκευών τους που βρίσκονται σε αδράνεια. Τέλος το MME είναι το τελικό σημείο του δικτύου για κρυπτογράφηση/προστασία ακεραιότητας για τις NAS διαδικασίες και αναλαμβάνει την διαχείριση του κλειδιού ασφαλείας. Οι Non Access Stratum(NAS) διαδικασίες τερματίζουν στο MME, είναι υπεύθυνες για την παραγωγή και την κατανομή προσωρινών ταυτοτήτων για τους UEs και επιβάλλουν περιορισμούς περιαγωγής στα UE.

##### Packet Data Network Gateway(P-GW):

Αυτή η οντότητα είναι η πύλη για το εξωτερικό δίκτυο που ονομάζεται Public Data Network(PDN), το οποίο παρέχει συνδεσιμότητα μεταξύ του δικτύου LTE και του PDN και υποστηρίζει λειτουργίες όπως η κατανομή διεύθυνσης IP στα τερματικά των χρηστών ή

μηχανισμούς ελέγχου για τις παραμέτρους ποιότητας υπηρεσιών από τα δεδομένα σύσκευσης που δημιουργήθηκαν από το δίκτυο LTE.

Όλα τα πακέτα IP ανταλλάσσονται μεταξύ αυτών των δικτύων κατευθύνονται από αυτήν την πύλη ή διεπαφή που ονομάζεται SGi. Επίσης ένας ακόμα ρόλος της P-GW είναι να ενεργεί ως σημείο αναφοράς για την κινητικότητα μεταξύ 3GPP και μη τεχνολογίες όπως το WiMAX και 3GPP2). Όταν ένας UE μετακινείται από μια S-GW σε άλλη, οι φορείς/κανάλια πρέπει να αλλάξουν στο P-GW και το P-GW θα λάβει ένδειξη για να τροποποιήσει τις ροές δεδομένων από το νέο S-GW.

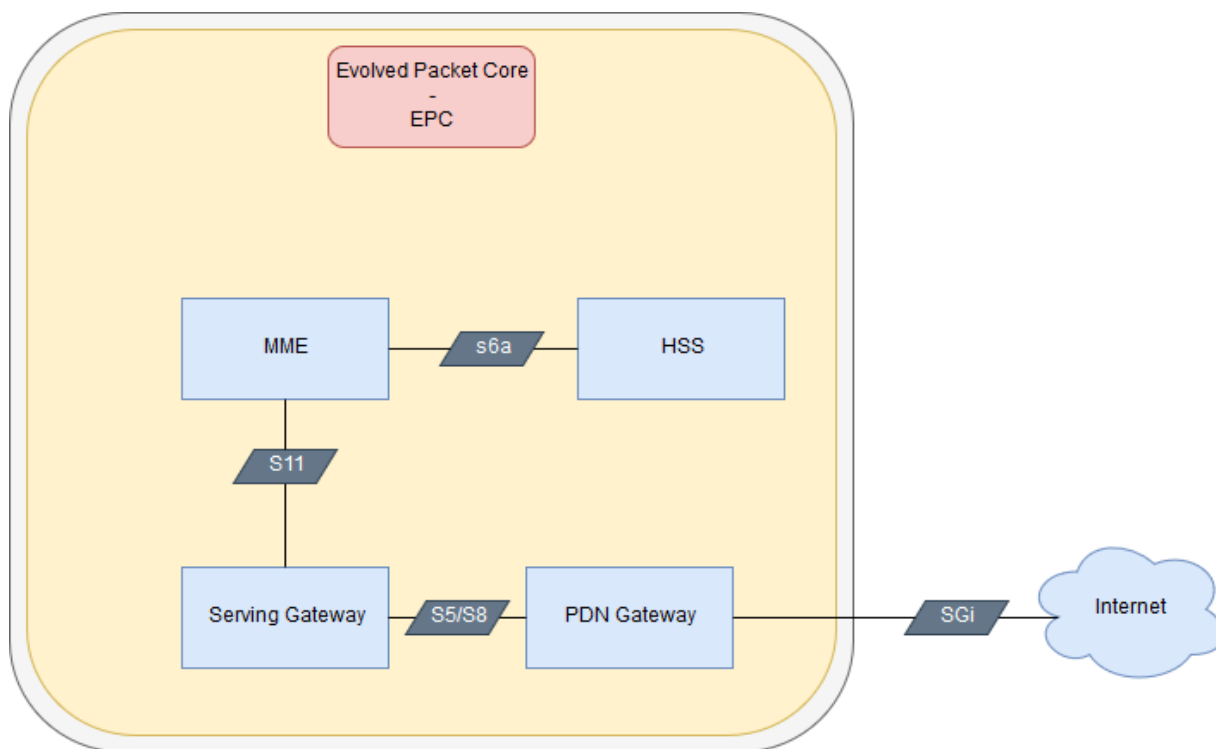
#### Service Gateway(S-GW):

Είναι η πύλη του χρήστη που συνδέει το eNodeB με το Κεντρικό δίκτυο. Όπως και στο MME, όταν ένας χρήστης είναι συνδεδεμένος στο LTE σύστημα ένα S-GW του ανατίθεται ανάλογα με τη γεωγραφική θέση του. Η S-GW παρακολουθεί τα δεδομένα στις συνδέσεις και μπορεί επίσης να συλλέγει δεδομένα που απαιτούνται για τον υπολογισμό της χρέωσης των χρηστών. Επιπρόσθετα περιλαμβάνει τη λειτουργία νόμιμης παρακολούθησης, η οποία δίνει τη δυνατότητα να παρέχονται τα δεδομένα, του χρήστη που παρακολουθείται, στις αρχές για περαιτέρω έλεγχο.

Μια S-GW μπορεί να εξυπηρετεί μόνο μια συγκεκριμένη γεωγραφική περιοχή με ένα περιορισμένο σύνολο eNodeBs και επίσης μπορεί να υπάρχει ένα περιορισμένο σύνολο MMEs που ελέγχουν αυτή την περιοχή. Αυτή η οντότητα παρέχει ένα σημείο προσάρτησης όταν το τερματικό μετακινείται από το ένα eNodeB στο άλλο. Επίσης, είναι υπεύθυνο για τη δρομολόγηση της κυκλοφορίας των χρηστών και τη προσωρινή αποθήκευση των πακέτων IP των χρηστών από τους τερματικούς σταθμούς που βρίσκονται σε αδράνεια. Αυτό συνδέεται με το MME μέσω της S11 και με το P-GW με τη διασύνδεση S5/S8.

Κάθε λειτουργία έχει κάποιο τρόπο σύνδεσης οπότε στη συνέχεια θα υπάρξει η εξήγηση κάθε τύπου σύνδεσης που θα αντιμετωπίσουμε στην συνέχεια σχετικά με τον κεντρικό δίκτυο και το eNodeB για το σενάριο που έχει κατασκευαστεί:

- S1-C: Επικοινωνεί το eNB με το MME και χρησιμοποιείται για τη διαχείριση των δεδομένων ελέγχου.
- S1-U: Επικοινωνεί το eNB με το SPGW και χρησιμοποιείται για τη διαχείριση των δεδομένων του χρήστη.
- S6-a: Επικοινωνεί το MME με το HSS.
- S11: Επικοινωνεί το MME με το SPGW.
- SGi: Επικοινωνεί το SPGW με το εξωτερικό δίκτυο, όπως το Internet.



Σχήμα 4. EPC δίκτυο

### 1.3 Αρχιτεκτονική Ράδιο-Πρωτοκόλλων

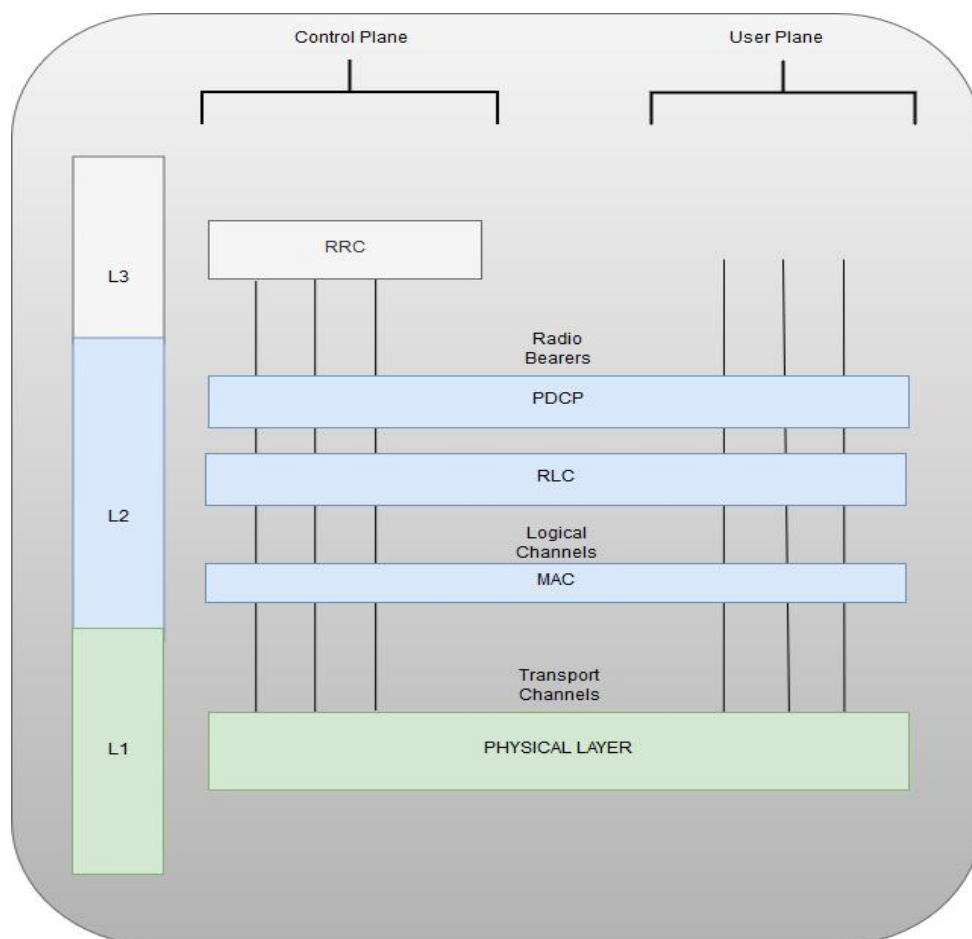
Τα πρωτόκολλα χωρίζονται σε δύο βασικές κατηγορίες, τα πρωτόκολλα επιπέδου χρήστη (User Plane) και τα πρωτόκολλα επιπέδου ελέγχου (Control Plane) τα οποία έχουν να κάνουν με μεταδόσεις δεδομένων του χρήστη και μεταδόσεις σήματος αντίστοιχα. Στο παρακάτω σχήμα παρουσιάζεται η δομή των πρωτοκόλλων από την πλευρά του eNodeB.

Ξεκινώντας από την κορυφή του σχήματος, το επίπεδο ελέγχου ραδιοπόρων (Radio Resource Control-RRC) υποστηρίζει όλες τις διαδικασίες σηματοδότησης μεταξύ του τερματικού και του eNodeB, αυτό περιέχει διαδικασίες κινητικότητας, καθώς και τη διαχείριση της σύνδεσης του τερματικού. Η σηματοδότηση από το EPC Control Plane μεταφέρεται στο τερματικό μέσω του πρωτοκόλλου RRC, έτσι προκύπτει και η σχέση μεταξύ του RRC και των ανώτερων επιπέδων.

Το πρωτόκολλο σύγκλισης δεδομένων πακέτων (Packet Data Convergence Protocol-PDCP) του οποίου βασικός ρόλος είναι η συμπίεση επικεφαλίδων και η εφαρμογή ασφαλείας, όπως η κρυπτογράφηση και η ακεραιότητα, προσφέρεται στους φορείς από τα κατώτερα επίπεδα του E-UTRAN. Το επίπεδο RLC παρέχει σε αυτό του PDCP βασικές υπηρεσίες παρόμοιες με αυτές του επιπέδου 2 του OSI, όπως τμηματοποίηση των πακέτων δεδομένων

και Αυτόματη αίτηση επανάληψης (Automatic Repeat Request-ARQ) ως μηχανισμό διόρθωσης σφαλμάτων.

Υπάρχει ένα-προς-ένα αντιστοίχιση μεταξύ κάθε RLC εισερχόμενης ροής και των λογικών καναλιών που παρέχονται από το επίπεδο RLC στο MAC. Η κύρια λειτουργία του επιπέδου MAC είναι αφού λάβει υπόψη του τις προτεραιότητες στις ροές δεδομένων που παρέλαβε από το επίπεδο RLC, να χαρτογραφήσει και να πολυπλέξει τα λογικά κανάλια στα κανάλια μεταφοράς.



Σχήμα 5. Αρχιτεκτονική Ράδιο-Πρωτοκόλλων

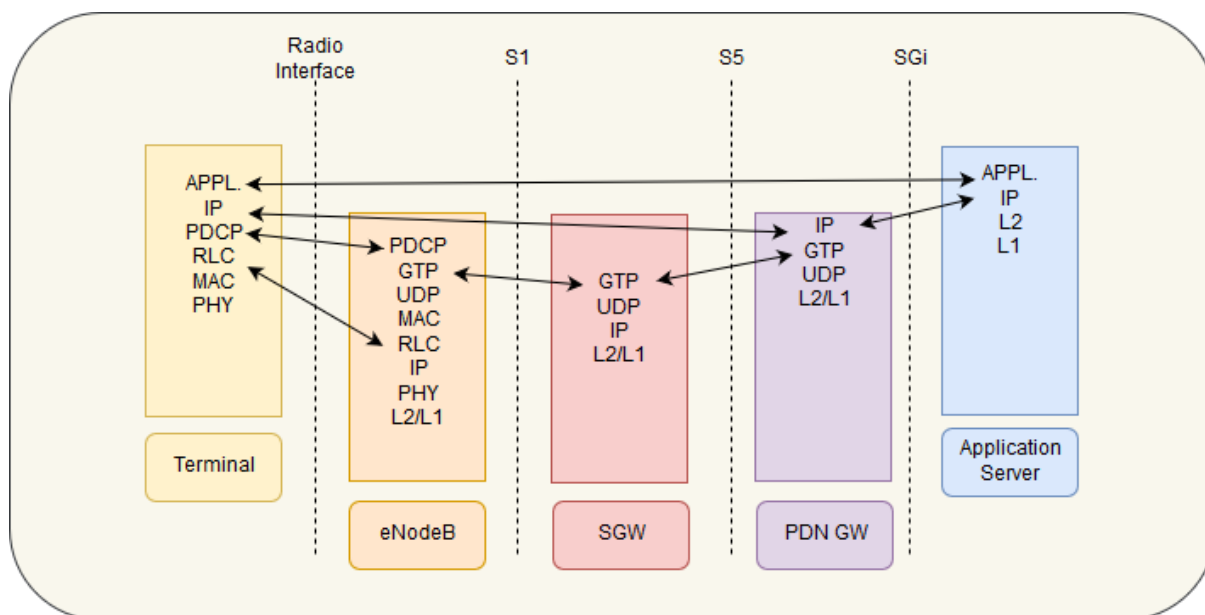
Οι ροές που είναι πολυπλεγμένες σε ένα κανάλι μεταφοράς, μπορεί να προέρχονται είτε από ένα μόνο χρήστη είτε από πολλαπλούς. Τέλος το MAC επίπεδο παραδίδει τις ροές προς μεταφορά στο PHY επίπεδο, το οποίο εφαρμόζει κωδικοποίηση καναλιού και διαμόρφωση πριν κάνει τη μετάδοση μέσω των ραδιο διεπαφών.

### 1.3.1 Κανάλια Μεταφοράς - Επίπεδο Χρήστη

Το ασύρματο δίκτυο του επίπεδο χρήστη περιλαμβάνει όχι μόνο τα δεδομένα του χρήστη, όπως πακέτα φωνής ή Web περιεχόμενο, αλλά επίσης τη σηματοδότηση που σχετίζεται με τις υπηρεσίες εφαρμογών όπως οι εφαρμογές σχετικές με τα πρωτόκολλα Session Initiation Protocol(SIP) και Real Time Control Protocol (RTCP). Ακόμη η σηματοδοσία υψηλού επιπέδου παρά το γεγονός ότι θεωρείται πληροφορία ελέγχου από τα επίπεδα εφαρμογών, μεταδίδεται μέσω του επιπέδου χρήστη.

Στο Σχήμα παρουσιάζεται η στοίβα πρωτοκόλλων User Plane, από το τερματικό (terminal) μέχρι τον εξυπηρετητή εφαρμογών (application server).

Αρχικά το επίπεδο εφαρμογών, που είναι παρόν μόνο στο τερματικό σταθμό και στον εξυπηρετητή εφαρμογών, βασίζεται σε IP μετάδοση δεδομένων όπου τα πακέτα δρομολογούνται μέσω των Packet Core Gateways πριν φτάσουν στον προορισμό τους. Τέλος, οι διεπαφές L1 και L2 αναφέρονται αντίστοιχα στις φυσικές (physical) και ζεύξης δεδομένων (data link) διεπαφές των S1, S5 και SGi διεπαφών δικτύου.



Σχήμα 6. Επίπεδο Χρήστη

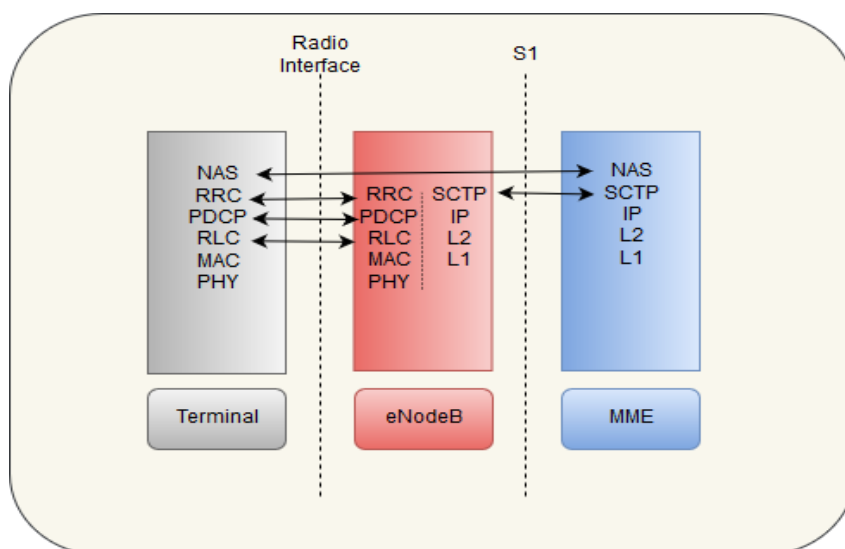
### 1.3.2 Κανάλια Μεταφοράς - Επίπεδο Ελέγχου

Το Control Plane αντιστοιχεί στις ροές πληροφορίας που στην πραγματικότητα θεωρούνται ως ροές σηματοδοσίας από το E-UTRAN και το EPC. Περιλαμβάνει όλες τις ροές σηματοδοσίας RRC E-UTRAN και NAS, που αναφέρονται σε λειτουργίες και υπηρεσίες που



είναι ανεξάρτητες από την τεχνολογία πρόσβασης. Το NAS περιλαμβάνει επίσης τα GPRS Mobility Management (GMM) και Session Management (SM) επίπεδα για λογαριασμό όλων των διαδικασιών σηματοδότησης μεταξύ του τερματικού και του MME για διαχείριση, έλεγχο ασφάλειας και ταυτοποίηση της συνόδου και του EPS φορέα.

Η στοίβα πρωτοκόλλων του επιπέδου ελέγχου παρουσιάζεται στο παρακάτω σχήμα και όπως είναι εμφανές σταματά στην MME, αφού εκεί τερματίζουν τα πρωτόκολλα ανώτατου επιπέδου. Στη ραδιο διεπαφή, το Control Plane χρησιμοποιεί την ίδια στοίβα πρωτοκόλλων (PDCP, RLC, MAC, PHY) για να μεταφέρει τόσο την RRC(Radio Resource Control) όσο την Core Network NAS σηματοδότηση. Όπως φαίνεται και στο σχήμα, τα επίπεδα RLC, MAC και PHY υποστηρίζουν τις ίδιες λειτουργίες για το επίπεδο χρήστη και επίπεδο ελέγχου, ωστόσο, αυτό δε σημαίνει ότι οι πληροφορίες τους μεταδίδονται με τον ίδιο ακριβώς τρόπο.



Σχήμα 7. Επίπεδο Χρήστη

#### 1.4 Κανάλια Μετάδοσης Δεδομένων

Το E-UTRAN για να είναι ευέλικτο και να επιτρέπει διαφορετικούς τρόπους μετάδοσης δεδομένων, υποστηρίζει τα εξής είδη καναλιών: λογικά κανάλια, κανάλια μεταφοράς και Φυσικά κανάλια. Γενικά πρέπει το E-UTRAN να είναι ικανό να μεταδίδει πληροφορίες με υψηλή ταχύτητα και χαμηλή καθυστέρηση με τον πιο αποτελεσματικό τρόπο, τα μηνύματα σηματοδότησης να διαβιβάζονται γρήγορα, χρησιμοποιώντας προστασία από σφάλματα με

κάθε δυνατό τρόπο. Αντίθετα, η φωνή ή οι εφαρμογές ροής δεδομένων μπορεί να έχουν ανοχή σε σφάλματα κατά τη μετάδοσή τους.

#### 1.4.1 Λογικά Κανάλια

Τα λογικά κανάλια σχετίζονται με υπηρεσίες μεταφοράς δεδομένων που παρέχονται από τα πρωτόκολλα ραδιο-διεπαφών. Υπάρχουν δύο είδη λογικών καναλιών: Κάθε ένα από τα παρακάτω κανάλια των δύο αυτών κατηγοριών αντιστοιχεί σε συγκεκριμένο τύπο ροής πληροφοριών. Τα κανάλια ελέγχου για τη μεταφορά των δεδομένων του Control Plane και τα κανάλια δοσοληψίας για τη μεταφορά του User Plane πληροφοριών. Κάθε ένα από τα κανάλια των δύο αυτών κατηγοριών αντιστοιχεί σε συγκεκριμένο τύπο ροής πληροφοριών

#### 1.4.2 Κανάλια Μεταφοράς

Τα κανάλια μεταφοράς χωρίζονται σε δύο υποκατηγορίες, στα κανάλια μεταφοράς κατερχόμενης ζεύξης (downlink) και στα κανάλια μεταφοράς ανερχόμενης ζεύξης (uplink). Γενικότερα περιγράφουν το πώς και με ποια χαρακτηριστικά μεταφέρονται τα δεδομένα μέσω των ραδιο-διεπαφών.

#### 1.4.3 Φυσικά κανάλια

Τα φυσικά κανάλια είναι η ουσιαστικά είναι η εφαρμογή του καναλιού μεταφοράς πάνω στη ραδιο διεπαφή. Είναι γνωστά μόνο στο φυσικό επίπεδο του E-UTRAN και η δομή τους έχει ισχυρή εξάρτηση από τα χαρακτηριστικά της φυσικής διεπαφής του Orthogonal Frequency Division multiplexing (OFDM). Χωρίζονται και πάλι σε δύο υποκατηγορίες: σε αυτά της κατερχόμενης (downlink) και σε αυτά της ανερχόμενης ζεύξης (uplink).

### 1.5 Τί είναι το Software Defined Networking

Σύμφωνα με τον επίσημο ορισμό από το Open Networking Foundation το Software Defined Networking (SDN) είναι μια νέα αρχιτεκτονική δικτύου, όπου το δίκτυο ελέγχου είναι διαχωρισμένο από το δίκτυο προώθησης και είναι άμεσα προγραμματιζόμενο.

Γενικά στοχεύει να υπάρχει η δυνατότητα στους διαχειριστές του κάθε δικτύου να μπορούν άμεσα και απομακρυσμένα να ελέγξουν το δίκτυο μέσω ενός κεντροποιημένου κέντρου ελέγχου. Το SDN μπορεί να υποστηρίξει υποδομή διαφορετικών παρόχων προκειμένου να σχεδιάσει, ελέγξει και διαχειριστεί τα δίκτυα και τον διαχωρισμό του επιπέδου ελέγχου από το επίπεδο δεδομένων.

#### 1.5.1 Βασικά Χαρακτηριστικά SDN

Ένα από τα πιο βασικά χαρακτηριστικά του SDN είναι αρχικά η ευελιξία του γιατί η μεταφορά του επιπέδου ελέγχου σε εξωτερικό περιβάλλον επιτρέπει μέσω ενός ελεγκτή

(Controller) την απευθείας πρόσβαση στις λειτουργίες πολλαπλών Switches που ως δυνατότητα κάνει την διαχείριση και την αλλαγή παραμέτρων σε αυτά πολύ εύκολη σε σχέση με τον προγραμματισμό κάθε μηχανήματος ξεχωριστά.

Στη συνέχεια είναι το γραφικό περιβάλλον διότι πια σε επίπεδο Controller προσφέρεται γραφικό περιβάλλον (GUI) το οποίο προσθέτει σε λειτουργικότητα και αμεσότητα και κάνει πιο εύκολη συσχέτιση με τις λειτουργίες του δικτύου.

Επίσης η τοπολογία, μέσω του Controller γίνεται να υπάρχει ολοκληρωμένη εικόνα του δικτύου με απεικόνιση της τοπολογίας των δικτυακών στοιχείων σε γραφικό περιβάλλον και η ασφάλεια από το SDN διότι δίνεται η δυνατότητα για τοποθέτηση Firewalls λογισμικού στα εξωτερικά σημεία του δικτύου καθώς και δρομολόγησης της κίνησης ευαίσθητων δεδομένων από ασφαλείς κόμβους. Γενικά μπορούν να επιβληθούν εύκολα κανόνες σχετικά με τα πακέτα που κινούνται στο δίκτυο και να αποκλειστούν αυτά που θεωρούνται ύποπτα.

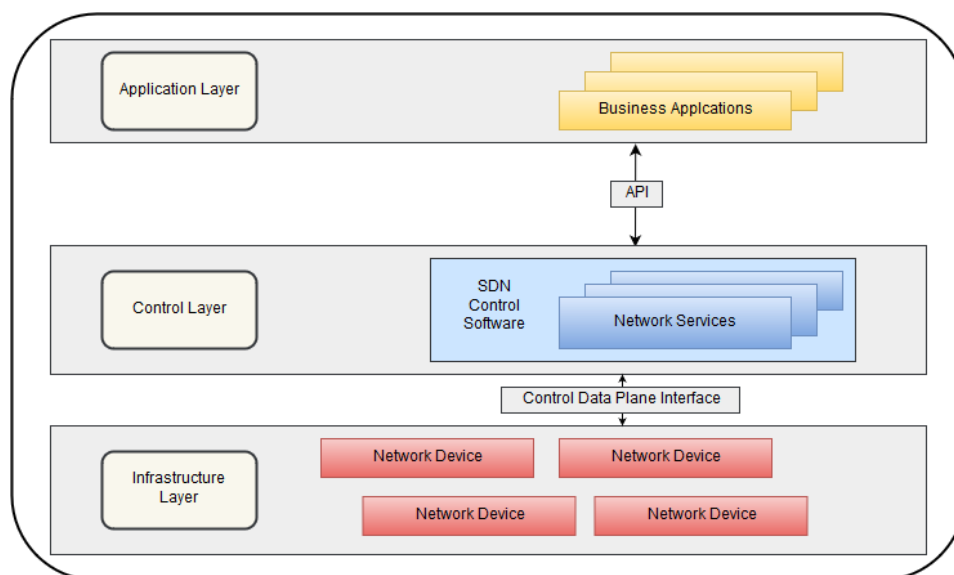
Επιπρόσθετα ένα βασικό χαρακτηριστικό του SDN είναι η υποστήριξη επιπέδων ελέγχου δικτύου κυρίως λόγω της κεντροποιημένης δομής με τον Controller στην κορυφή, η οποία επιτρέπει ολική θέαση του δικτύου και πλήρη έλεγχο στις λειτουργίες κάθε δικτυακού στοιχείου καθώς και της δικτυακής κίνησης. Οι αλλαγές στις ροές του δικτύου γίνεται άμεσα μέσω του Controller ο οποίος συντονίζει τα Switches ώστε να επιβληθούν οι κανόνες που θέτονται στο δίκτυο. Τέλος η SDN αρχιτεκτονική επιτρέπει πολύ καλύτερη αξιοποίηση του εξοπλισμού δικτύου καθώς και υψηλές διαθεσιμότητες σε όλο το εύρος του δικτύου.

### 1.5.2 Αρχιτεκτονική του SDN

Υπάρχουν τρεις βασικές δομές στο SDN, το επίπεδο εφαρμογών, το επίπεδο ελέγχου και το επίπεδο δεδομένων. Στο επίπεδο εφαρμογών ανήκουν οι εφαρμογές που μέσω της NorthBound SDN διεπαφής εκφράζουν στο επίπεδο ελέγχου, δηλαδή στον SDN ελεγκτή, τις ανάγκες του δικτύου. Στο επίπεδο δεδομένων ανήκουν οι δικτυακές συσκευές κι εκεί λαμβάνεται χώρα η προώθηση των πακέτων με τρόπο που έχει ορίσει το επίπεδο ελέγχου. Πιο αναλυτικά η κάθε δομή:

#### Επίπεδο Εφαρμογών (SDN Application):

Πρόκειται για διάφορα προγράμματα μέσω των οποίων ο διαχειριστής του δικτύου μπορεί άμεσα να επικοινωνήσει με τον ελεγκτή ώστε να μάθει τι χρειάζεται το δίκτυο και να ορίσει την απαιτούμενη λειτουργία που απαιτείται να εφαρμοστεί στο δίκτυο.



Σχήμα 8. Αρχιτεκτονική του SDN

#### Επίπεδο Ελέγχου (SDN Controller):

Η πλατφόρμα ελέγχου περιλαμβάνει τον SDN ελεγκτή που αποτελεί μια κεντροποιημένη ενότητα. Σε φυσικό επίπεδο ο ελεγκτής μπορεί να είναι κατανεμημένος σε διαφορετικά φυσικά μηχανήματα έτσι ώστε να υπάρχει μεγαλύτερη κάλυψη και βελτιστοποίηση των διεργασιών. Ωστόσο, ο ελεγκτής συμπεριφέρεται ως αυτόνομη μονάδα, με δικό της λειτουργικό σύστημα και λαμβάνει ενιαίες αποφάσεις. Βασικά λαμβάνει τις απαιτήσεις του επιπέδου ελέγχου και τις μεταφέρει στο επίπεδο δεδομένων και αντίστοιχα αντιλαμβάνεται τυχόν αλλαγές στη λειτουργία του επίπεδο δεδομένων και ενημερώνει το επίπεδο εφαρμογών.

#### Επίπεδο Δεδομένων (SDN Datapath):

Είναι μια δικτυακή υπηρεσία που παρέχει έλεγχο των διαδικασιών προώθησης και επεξεργασίας δεδομένων που λαμβάνουν χώρα στο δίκτυο που βρίσκεται. Αποτελείται από έναν CDPI Agent με σκοπό να επιτυγχάνεται η επικοινωνία με τον ελεγκτή και από ένα σύνολο συσκευών που είναι υπεύθυνες για την προώθηση της κίνησης.

## 2.Εγκατάσταση λογισμικού

---

### 2.1 Γενική Επεξήγηση του Open Air Interface 5G

Το OpenAirInterface είναι μία ευέλικτη πλατφόρμα με γνώμονα ένα ανοιχτό LTE οικοσύστημα. Η πλατφόρμα προσφέρει ένα λογισμικό που εφαρμόζει το LTE μαζί με τη στοίβα πρωτοκόλλου του 3GPP Standard και στο E-UTRAN και στο EPC, και είναι ανοιχτό για ανάπτυξη από το κοινό.

Μπορεί να χρησιμοποιηθεί για να δημιουργηθεί και να τροποποιηθεί όπως είναι επιθυμητό ένα LTE Base Station(OAI eNB),ένας ή περισσότεροι εξοπλισμοί χρήστη (UE) και το κεντρικό δίκτυο(OAI EPC) σε ένα υπολογιστή. Το OAI eNB μπορεί να συνδεθεί σε διάφορους εξοπλισμούς χρήστη για να ελεγχθούν διάφορα σενάρια χρήσης καθώς και να επιβλέπονται οι πόροι στο δίκτυο.

Επί του παρόντος, η πλατφόρμα OAI περιλαμβάνει πλήρη εφαρμογή συστημάτων που συμμορφώνονται με τα πρότυπα 3GPP LTE σε C υπό πραγματικό χρόνο, βελτιστοποιημένο για x86 αρχιτεκτονική.

Στο Physical Layer, παρέχονται διάφορα χαρακτηριστικά, μερικά από αυτά είναι:

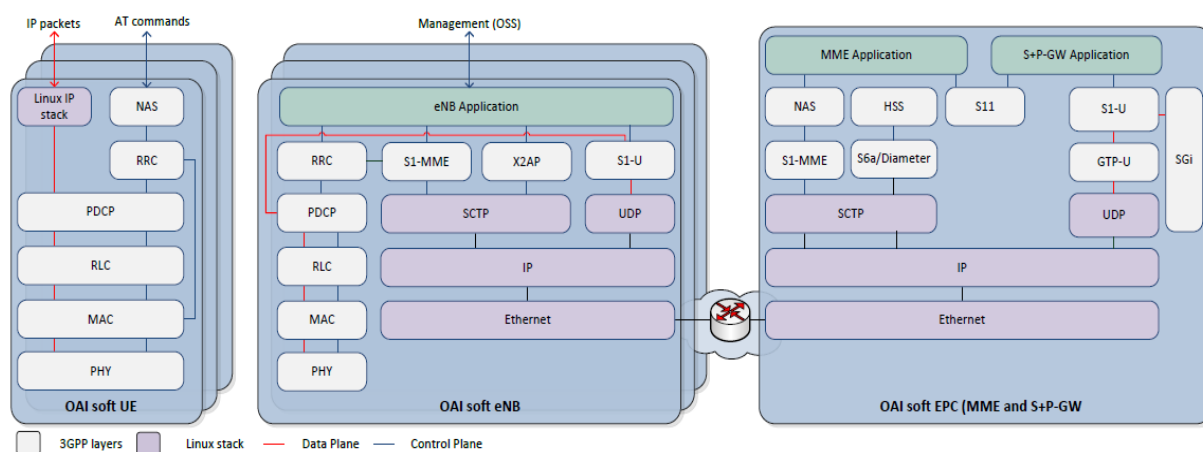
- LTE release 8.6 με ένα υποσύνολο του release 10
- FDD και TDD configurations σε 5, 10, and 20 MHz bandwidth
- Λειτουργία μετάδοσης: 1 (SISO) και 2, 4, 5, και 6 (MIMO 2x2)
- CQI/PMI αναφορές
- DL κανάλια που υποστηρίζονται: PSS, SSS, PBCH, PCFICH, PHICH, PDCCH, PDSCH, PMCH

Για το E-UTRAN πρωτόκολλο μερικά από τα βασικά χαρακτηριστικά είναι:

- LTE release 8.6 με ένα υποσύνολο του release 10
- MAC, RLC, PDCP and RRC layers
- Υπηρεσίες πρωτοκόλλου για όλα τα Rel8 Channels και Rel10 eMBMS
- Πλήρως αναδιαμορφωμένη στοίβα πρωτοκόλλων
- Κρυπτογράφηση χρησιμοποιώντας AES και Sonw3G αλγόριθμους

Για το EPC κάποια βασικά χαρακτηριστικά είναι:

- MME, SGW, PGW and HSS υλοποιήσεις
- Ακεραιότητα NAS και κρυπτογράφηση χρησιμοποιώντας τους αλγόριθμους AES και Snow3G
- Υποστήριξη IPv4 και IPv6



Σχήμα 9. OpenAirInterface LTE Software Stack(7)

## 2.2 Επεξήγηση του OpenAirInterface Scenario Descriptor

Ο Περιγραφέας Σεναρίου για το OpenAirInterface (OpenAirInterface Scenario Descriptor) χρησιμοποιείται με σκοπό να τρέξει πιο ρεαλιστικό σενάριο εξομοίωσης, το oaisim έχει τη δυνατότητα να διαβάζει περιγραφές σεναρίων από xml αρχείο.

Αυτά επιτρέπουν μια πολύ λεπτομερή προσαρμογή του σεναρίου προσομοίωσης, την επεξεργασία παραμέτρων όπως η ισχύς μετάδοσης της κεραίας eNB, το μοντέλο κινητικότητας για τους κόμβους και το προφίλ των κυκλοφοριακών ροών.

Ωστόσο το OSD δεν υποστηρίζει μεταβλητές παραμέτρους, οπότε όταν εκτελείται μια πολλαπλού επιπέδου προσομοίωση πρέπει να προετοιμάζεται διαφορετική OSD για κάθε συνδυασμό τιμών και παραμέτρων. Το OSD αποτελείται από τέσσερα κύρια μέρη, τα οποία είναι μια βασική περιγραφή της δομής δεδομένων για τη προσομοίωση.

Κάθε σενάριο που περιγράφεται από το OSD ορίζει ένα σύνολο των εννοιών που περιγράφονται παρακάτω:

- Περιβάλλον (Environment configuration)
- Τοπολογία (Topology configuration)
- Ρυθμίσεις Εφαρμογής (Application configuration)
- Ρυθμίσεις Προσομοίωσης (Emulation configuration).

```
<OAI_EMULATION>
  <ENVIRONMENT_SYSTEM_CONFIG>
    </ENVIRONMENT_SYSTEM_CONFIG>
  <TOPOLOGY_CONFIG>
    </TOPOLOGY_CONFIG>
  <APPLICATION_CONFIG>
    </APPLICATION_CONFIG>
  <EMULATION_CONFIG>
    </EMULATION_CONFIG>
  <PROFILE>
    </PROFILE>
</OAI_EMULATION>
```

### 2.3 Στοιχεία Προσομοίωσης

Για την πειραματική διαδικασία ως πόροι για να έρθει εις πέρα χρησιμοποιήθηκαν 3 ξεχωριστά Virtual Machines πάνω στον ίδιο Server. Ο ρόλος του πρώτου Virtual Machine ήταν να είναι ο οικοδεσπότης(host) για τον εξοπλισμό του χρήστη (UE) και την κεραία (eNB),το δεύτερο Virtual Machine ήταν ο host για το δίκτυο κορμού (Openair-cn/EPC) και όλες τις λειτουργίες του και τέλος το τρίτο Virtual Machine χρησιμοποιήθηκε ως host για το εργαλείο iperf3 μέσω του οποίου γινόταν η αποστολή πακέτων από τον ένα άκρο στο άλλο.

Κάθε Virtual machine αντίστοιχα είχε:

1. eNB/UE Virtual Machine:
  - 16 Gigabyte(GB) Ram
  - 8 πυρήνες επεξεργαστή (CPU Cores)
  - 40 GB αποθηκευτικού χώρου
2. Openair-cn(EPC) Virtual Machine:
  - 8 Gigabyte(GB) Ram
  - 6 πυρήνες επεξεργαστή
  - 40 GB αποθηκευτικού χώρου
3. Host iperf3 Virtual Machine:
  - 8 Gigabyte(GB) Ram
  - 4 πυρήνες επεξεργαστή
  - 20 GB αποθηκευτικού χώρου.

Το λειτουργικό σύστημα που χρησιμοποιήθηκε για τα Virtual Machines είναι το Ubuntu, το οποίο εντάσσεται στη κατηγορία των συστημάτων Linux δηλαδή ένα λειτουργικό σύστημα ανοιχτού κώδικα που σχεδιάστηκε με βάση το λειτουργικό σύστημα UNIX, το οποίο αναπτύχθηκε κατά τις δεκαετίες του 1960.

Χρησιμοποιήθηκαν 2 διαφορετικές εκδόσεις τους, η έκδοση 14.04 για το OAI και iperf3 Client και η έκδοση 16.04 για το Openair-CN με βασικό λόγο να αποτελούν οι διαφορές σχετικά με την υποστήριξη βασικών λειτουργιών από τον πυρήνα (Kernel) που είναι η κεντρική μονάδα λειτουργίας ενός λειτουργικού συστήματος (OS).

Ο πυρήνας είναι το μέρος του λειτουργικού συστήματος που φορτώνεται πρώτα και παραμένει στην κύρια μνήμη. Επειδή παραμένει στη μνήμη, είναι σημαντικό ο πυρήνας να είναι όσο το δυνατόν μικρός, ενώ παράλληλα παρέχει όλες τις βασικές υπηρεσίες που απαιτούνται από άλλα μέρη του λειτουργικού συστήματος και των εφαρμογών. Ο κώδικας του πυρήνα φορτώνεται συνήθως σε μια προστατευμένη περιοχή μνήμης για να αποφευχθεί η αντικατάστασή του από προγράμματα ή άλλα μέρη του λειτουργικού συστήματος. Συνήθως,



ο πυρήνας είναι υπεύθυνος για τη διαχείριση μνήμης, τη διαχείριση διαδικασιών και εργασιών και τη διαχείριση δίσκων.

## 2.4 Προεργασία Πειράματος

Αρχικά απαιτείται να γίνουν κάποιες ενέργειες στον πυρήνα(Kernel) κάθε Virtual Machine για την βελτίωση της απόδοσης αλλά και την εγκατάσταση πακέτων που είναι απαραίτητα για την ομαλή λειτουργία διαφόρων ενεργειών για το πείραμα.Θα πρέπει να επισημανθεί ότι το Virtual Machine με την εγκατάσταση του OAI (UE και eNB) θα αποκαλείται Υπολογιστής-1(PC1) και το Virtual Machine με την εγκατάσταση του Openair-CN θα αποκαλείται Υπολογιστής-2 (PC2).

Τρία είναι τα βασικά βήματα για κάθε υπολογιστή που είναι αναγκαίο να εκτελεστούν και μόνο το πρώτο αλλάζει από υπολογιστή σε υπολογιστή γιατί θα γίνει εγκατάσταση διαφορετική έκδοση του Kernel κάθε φορά αντίστοιχα. Τα άλλα δύο βήματα είναι σχετικά με διαχείριση της ενέργειας του μηχανήματος και την μεγιστοποίηση της ταχύτητας ανά πυρήνα του επεξεργαστή για να επιτευχθεί η μέγιστη απόδοση.

### 2.4.1 Χαμηλή λανθάνουσα κατάσταση Kernel(PC1)

Αρχικά στο PC1 μέσω του τερματικού θα γίνει η ενημέρωση και αναβάθμιση του λειτουργικού συστήματος με τις εντολές:

```
sudo apt-get update  
sudo apt-get dist-upgrade.
```

Μετά θα ξεκινήσει η εγκατάσταση της χαμηλής λανθάνουσας κατάστασης του Kernel (Low-Latency Kernel) με πρώτο βήμα την αντικατάσταση των ήδη υπάρχων εκδόσεων των Kernel στο σύστημα μας. Πρώτα γίνεται επίβλεψη των ήδη εγκατεστημένων Kernel στο σύστημα:

```
sudo dpkg --get-selections | grep linux-image
```

Στη συνέχεια τους αφαιρούμε όλους:

```
for K in $(sudo dpkg --get-selections | grep linux-image); do sudo apt-get -y purge $K;  
done
```

και διαγράφουμε ότι έχει απομείνει στο σύστημα:

```
sudo apt-get autoremove
```

Το σύστημα είναι έτοιμο να δεχθεί και να κατεβάσει το νέο Kernel μέσω της εντολής:

```
sudo apt-get install linux-image-3.19.0-61-lowlatency linux-headers-3.19.0-61-lowlatency
```

Αμέσως μετά γίνεται επανεκκίνηση του μηχανήματος για την ολοκλήρωση της εγκατάστασης. Πολύ εύκολα μπορεί να γίνει έλεγχος ότι έχει εγκατασταθεί η επιθυμητή έκδοση Kernel στο σύστημα με την εντολή:

```
uname -a
```

#### 2.4.2 Χαμηλή λανθάνουσα κατάσταση Kernel(PC2)

Ξανά αλλά στο PC2 μέσω του τερματικού θα γίνει η ενημέρωση του λειτουργικού συστήματος με τις εντολές:

```
sudo apt-get update  
sudo apt-get dist-upgrade
```

Βασικό αρχικό πάλι βήμα η αντικατάσταση των ήδη υπάρχων εκδόσεων των Kernel στο σύστημα μας. Πρώτα γίνεται επίβλεψη των ήδη εγκατεστημένων Kernel στο σύστημα:

```
sudo dpkg --get-selections | grep linux-image
```

Στη συνέχεια τους αφαιρούμε όλους:

```
for K in $(sudo dpkg --get-selections | grep linux-image); do sudo apt-get -y purge $K;  
done
```

και διαγράφουμε ότι έχει απομείνει στο σύστημα μας:

```
sudo apt-get autoremove
```

Το σύστημα είναι έτοιμο να δεχθεί το νέο Kernel ο οποίος αυτή τη φορά περιέχει τη διεργασία GTP-MODULE η οποία θα χρησιμοποιηθεί από το openair-cn:

```
wget http://kernel.ubuntu.com/~kernel-ppa/mainline/v4.8/linux-headers-4.8.0-040800-generic_4.8.0-040800.201610022031_amd64.deb
και
wget http://kernel.ubuntu.com/~kernel-ppa/mainline/v4.8/linux-image-4.8.0-040800-generic_4.8.0-040800.201610022031_amd64.deb
```

στη συνέχεια αφού γίνει η λήψη τους θα ακολουθήσει και η εγκατάσταση τους με τις εντολές

```
sudo dpkg -i linux-headers-4.8.0-040800-generic_4.8.0-040800.201610022031_amd64.deb
sudo dpkg -i linux-image-4.8.0-040800-generic_4.8.0-040800.201610022031_amd64.deb
```

Αφού γίνει η επανεκκίνηση του συστήματος φορτώνεται το gtp-module στην υπάρχων έκδοση Kernel:

```
sudo modprobe gtp
```

και η επιβεβαίωση ότι τρέχει ομαλά:

```
lsmod | grep gtp
```

### 2.4.3 Διαχείριση της ενέργειας(PC1&PC2)

Πρώτα πρέπει να γίνει η απενεργοποίηση του p-state και το c-state στα Linux, αυτό επιτυγχάνεται ανοίγοντας το φάκελο grub με την εντολή:

```
sudo gedit /etc/default/grub
```

Προσθέτοντας την παρακάτω γραμμή: GRUB\_CMDLINE\_LINUX\_DEFAULT="quiet intel\_pstate=disable processor.max\_cstate=1 intel\_idle.max\_cstate=0 idle=poll", τότε γίνεται και μία ενημέρωση του grub με σκοπό να οριστικοποιηθούν οι αλλαγές που έγιναν.

```
sudo update-grub
```

Τέλος ανοίγουμε το φάκελο `blacklist.conf` με την εντολή:

```
sudo gedit /etc/modprobe.d/blacklist.conf
```

και στο τέλος του αρχείου προσθέτουμε τη γραμμή:

`blacklist intel_powerclamp` με σκοπό να σταματήσουμε την `intel_powerclamp` διεργασία.

#### 2.4.4 Απενεργοποίηση εναλλασσόμενης ταχύτητας επεξεργαστή(PC1&PC2)

Σκοπός είναι να ενεργοποιηθεί η μέγιστη ταχύτητα του κάθε πυρήνα του επεξεργαστή έτσι ώστε να υπάρχει η μέγιστη απόδοση. Αυτό γίνεται εγκαθιστώντας το εργαλείο `cpufrequtils` με την εντολή:

```
sudo apt-get install cpufrequtils
```

Επίσης στο φάκελο `/etc/default/cpufrequtils` προσθέτουμε τη πρόταση: `GOVERNOR="performance"` το οποίο είναι ο τρόπος ενεργοποίησης της πλήρους απόδοσης του επεξεργαστή. Ακόμα για να ολοκληρωθεί η ενέργεια και να μην χαθεί η αλλαγή μετά τη πρώτη επανεκκίνηση εκτελείται η εντολή:

```
sudo update-rc.d ondemand disable
```

Για την επιβεβαίωση ότι κάθε πυρήνας τρέχει στην ίδια ταχύτητα μπορεί να εκτελεστεί η παρακάτω εντολή στο τερματικό:

```
cpufreq-info
```

## 2.5 Εγκατάσταση βασικών λειτουργιών

### 2.5.1 Εγκατάσταση OpenAirInterface

Μετά την ρύθμιση των παραμέτρων που χρειάστηκαν παραπάνω το PC1 με διεύθυνση:10.1.14.250/24 και όνομα ens2 είναι έτοιμο να δεχθεί την εγκατάσταση του OAI. Αρχικά θα γίνει η εγκατάσταση του πακέτου “git” μέσω της εντολής:

```
sudo apt-get install subversion git
```

για να είναι δυνατό να κατέβουν τα επιθυμητά πακέτα στη συνέχεια. Αμέσως μετά γίνεται η λήψη όλου το φακέλου και των θεμελιωδών αρχείων του βασισμένα στον Μάστερ Κλάδο(Master Branch) για το OpenAirInterface με την εντολή:

```
git clone https://gitlab.eurecom.fr/oai/openairinterface5g.git
```

Στη συνέχεια θα γίνει η είσοδος στο φάκελο openairinterface5g:

```
cd openairinterface5g
```

και θα χρησιμοποιηθεί η εντολή: source oaienv με σκοπό να χρησιμοποιηθεί για να φορτώσει οποιοδήποτε αρχείο λειτουργιών στο τρέχων τερματικό. Τέλος θα γίνει είσοδος στο φάκελο cmake\_targets μέσω της εντολής:

```
cd cmake_targets
```

και θα ολοκληρωθεί η εγκατάσταση του OAI με την εντολή:

```
sudo ./build_oai -I --oaisim -t ETHERNET
```

Στη συγκεκριμένη περίπτωση το “sudo” μας δίνει δικαιώματα διαχειριστή, το “I” εγκαθιστά όλα τα απαραίτητα πακέτα για την υποστήριξη των λειτουργιών που θα εγκατασταθούν αργότερα, το “oaisim” εγκαθιστά τον προσομοιωτή και εν τέλη το “-t ETHERNET” αναφέρεται στο ότι η σύνδεση θα γίνει σε τοπικό επίπεδο και με την λειτουργία του τοπικού ενσύρματου δικτύου για να αποφευχθεί η χρήση επιπλέον εξοπλισμού.

### 2.5.1.1 eNodeB

Επιπρόσθετα στο PC1, ένα αρχείο ρυθμίσεων που είναι αναγκαίο να τροποποιηθεί, είναι αυτό της κεραίας. Χρησιμοποιώντας την εντολή:

```
sudo gedit ~/openairinterface5g/targets/PROJECTS/GENERIC-LTE-  
EPC/CONF/enb.band7.generic.oaisim.local_mme.conf
```

Μόλις ανοίξει το αρχείο `enb.band7.generic.local_mme.conf` το οποίο είναι έτοιμο για επεξεργασία. Ουσιαστικά θα γίνει ο ορθός προσδιορισμός των διευθύνσεων των διεπαφών για κάθε σύνδεση μεταξύ κεραίας και του κεντρικού δικτύου. Πιο συγκεκριμένα θα τοποθετηθούν οι σωστές διευθύνσεις στις σωστές διεπαφές για επικοινωνία κεραίας με MME και SGW.

Αρχείο `enb.band7.generic.oaisim.local_mme.conf`:

```
////////// MME parameters:  
mme_ip_address = ( { ipv4 = "10.1.14.251";  
                    ipv6 = "192:168:30::17";  
                    active = "yes";  
                    preference = "ipv4";  
                    });  
NETWORK_INTERFACES :  
{  
    ENB_INTERFACE_NAME_FOR_S1_MME = "ens2";  
    ENB_IPV4_ADDRESS_FOR_S1_MME = "10.1.14.250/24";  
  
    ENB_INTERFACE_NAME_FOR_S1U = "ens2";  
    ENB_IPV4_ADDRESS_FOR_S1U = "10.1.14.250/24";  
    ENB_PORT_FOR_S1U = 2152; # Spec 2152
```

### 2.5.1.2 OpenAirInterface Scenario Descriptor

Η Περιγραφή Σεναρίου (OSD - 1.xml) για το πείραμα που θα εκτελεστεί έχει τη παρακάτω μορφή και βρίσκεται στο παρακάτω φάκελο .../SIMU/EXAMPLE/OSD/WEBXML/.

```
<OAI_EMULATION>
  <ENVIRONMENT_SYSTEM_CONFIG>

    <WALL_PENETRATION_LOSS_dB>5</WALL_PENETRATION_LOSS_dB>
    <SYSTEM_BANDWIDTH_MB>10</SYSTEM_BANDWIDTH_MB>
    <SYSTEM_FREQUENCY_GHz>1.9</SYSTEM_FREQUENCY_GHz>
    <ANTENNA>
      <eNB_ANTENNA>
        <RX_NOISE_LEVEL_dB>1</RX_NOISE_LEVEL_dB>
        <BEAM_WIDTH_dB>10</BEAM_WIDTH_dB>
        <ANTENNA_GAIN_dBi>16</ANTENNA_GAIN_dBi>
        <TX_POWER_dBm>40</TX_POWER_dBm>
      </eNB_ANTENNA>
      <UE_ANTENNA>
        <RX_NOISE_LEVEL_dB>1</RX_NOISE_LEVEL_dB>
        <ANTENNA_GAIN_dBi>5</ANTENNA_GAIN_dBi>
        <TX_POWER_dBm>20</TX_POWER_dBm>
      </UE_ANTENNA>
    </ANTENNA>
  </ENVIRONMENT_SYSTEM_CONFIG>

  <TOPOLOGY_CONFIG>
    <AREA>
      <X_m>1000</X_m>
      <Y_m>1000</Y_m>
    </AREA>
    <MOBILITY>
```

```
<UE_MOBILITY>
<RANDOM_UE_DISTRIBUTION>
  <NUMBER_OF_NODES>1</NUMBER_OF_NODES>
</RANDOM_UE_DISTRIBUTION>
<UE_MOBILITY_TYPE>STATIC</UE_MOBILITY_TYPE>
</UE_MOBILITY>
<eNB_MOBILITY>
<eNB_INITIAL_DISTRIBUTION>STATIC</eNB_INITIAL_DISTRIBUTION>
<RANDOM_eNB_DISTRIBUTION>
  <NUMBER_OF_CELLS>1</NUMBER_OF_CELLS>
</RANDOM_eNB_DISTRIBUTION>
<eNB_MOBILITY_TYPE>STATIC</eNB_MOBILITY_TYPE>
</eNB_MOBILITY>
</MOBILITY>
</TOPOLOGY_CONFIG>
<EMULATION_CONFIG>
  <EMULATION_TIME_ms>15000</EMULATION_TIME_ms> <!-- if 0 set to
infinity-->
  <PERFORMANCE_METRICS>

  <THROUGHPUT>enable</THROUGHPUT> <!-- option: enable, disable. If
enable, throughput measurements are plotted in real time-->
  <LATENCY>enable</LATENCY> <!-- option: enable, disable. If enable, latency
measurements are plotted in real time-->
  <LOSS_RATE>enable</LOSS_RATE>

</PERFORMANCE_METRICS>
<LOG> <!-- set the global log level -->
<LEVEL>info</LEVEL> <!-- info, trace, warning, debug, -->
<VERBOSITY>low</VERBOSITY> <!-- low, medium, high, full -->
</LOG>
```



```
<SEED_VALUE>0</SEED_VALUE>    <!-- value 0 means randomly
generated by OAI -->
</EMULATION_CONFIG>
</OAI_EMULATION>
```

### 2.5.2 Εγκατάσταση EPC

Στο PC2 με διεύθυνση:10.1.14.251/24 αρχικά θα τροποποιηθεί το πεδίο ορισμού(Domain Name) πριν εγκαταστήσουμε το HSS,MME, και SPGW γιατί παίζει άμεσω ρόλο με τις ρυθμίσεις αυτών των τριών λειτουργιών. Είναι καλό να ελεγχθεί το όνομα του υπολογιστή και να επιβεβαιωθεί ότι είναι “oai” με την εντολή:

```
sudo gedit /etc/hostname
```

Έτσι η επεξεργασία του πεδίου ορισμού θα γίνει με την εντολή:

```
sudo gedit /etc/hosts
```

και το αρχείο είναι αναγκαίο να τροποποιηθεί έτσι ώστε να μοιάζει ως εξής :

```
127.0.0.1 localhost
```

```
127.0.1.1 oai.openair4G.eur oai
```

```
127.0.1.1 hss.openair4G.eur hss
```

Αμέσως μετά γίνεται η λήψη του όλων το αρχείων βασισμένα στο Μάστερ Κλάδο του Openair-cn με την εντολή:

```
git clone https://gitlab.eurecom.fr/oai/openair-cn.git
```

Γίνεται η είσοδος στο φάκελο openair-cn

```
cd openair-cn
```

και αμέσως μετά χρησιμοποιείται η εντολή:source oaienv με σκοπό να χρησιμοποιηθεί για να φορτώσει οποιοδήποτε αρχείο λειτουργιών στο τρέχων τερματικό.

Αμέσως μετά θα γίνει η ενημέρωση του κλάδου από Μάστερ σε Ανάπτυξης(Develop) και η ενεργοποίησή του:

```
git checkout develop  
git pull
```

Τέλος πριν ξεκινήσει η εγκατάσταση της κάθε λειτουργίας του κεντρικού δικτύου ξεχωριστά γίνεται η δημιουργία του παρακάτω φακέλου για την αποθήκευση διαφόρων αρχείων στη συνέχεια:

```
sudo mkdir -p /usr/local/etc/oai/freeDiameter
```

#### 2.5.2.1 Home Subscriber Server(HSS)

Πρώτα θα εγκατασταθούν τα απαραίτητα πακέτα στο openair-cn/scripts φάκελο για το HSS με την εντολή:

```
sudo ./build_hss -i
```

και αμέσως μετά θα γίνει η εγκατάσταση του και σε ότι μας ζητηθεί η ανταπόκριση του χρήστη είναι με “Yes”:

```
sudo ./build_hss
```

Αμέσως μετά θα γίνει η αντιγραφή των αρχείων(acl.conf & hss\_fd.conf) με τις ρυθμίσεις του HSS στο φάκελο που δημιουργήθηκε πριν με στόχο να μπορούμε να τα επεξεργαστούμε δίχως πρόβλημα:

```
sudo cp ~/openair-cn/etc/acl.conf /usr/local/etc/oai/freeDiameter  
sudo cp ~/openair-cn/etc/hss_fd.conf /usr/local/etc/oai/freeDiameter
```

Σημαντικό είναι να εκτελεστεί η παρακάτω εντολή έτσι ώστε να δημιουργηθεί το πιστοποιητικό για το HSS.

```
./check_hss_s6a_certificate /usr/local/etc/oai/freeDiameter/ hss.openair4G.eur
```

Το επόμενο βήμα είναι να γίνει επεξεργασία των ρυθμίσεων της βάσης δεδομένων (MySQL) ώστε να γίνει η αλλαγή στη παρακάτω γραμμή για να είναι αποδεκτές όλες οι συνδέσεις:

bind-address 0.0.0.0 και αμέσως μετά η επανεκκίνηση της βάσης δεδομένων μέσω της εντολής:

```
sudo gedit /etc/mysql/mysql.conf.d/mysqld.cnf  
sudo service mysql restart
```

Επίσης είναι αναγκαίο να δημιουργηθεί ένας χρήστης “oai” στη βάση δεδομένων και να έχει άδεια να μπορεί να έχει πρόσβαση στη βάση δεδομένων είτε από τοπικά είτε απομακρυσμένα. Αυτό θα γίνει με την χρήση της εντολής:

```
mysql -uroot -p
```

και στη συνέχεια μέσα στο περιβάλλον της MySQL θα δημιουργηθεί ο χρήστης oai που θα αναγνωρίζεται από τη βάση δεδομένων, η οποία δημιουργείται αμέσως μετά αλλά είναι κενή σε πρώτο χρόνο και παίρνει δικαιώματα για τη χρήση της:

```
>CREATE USER 'oai'@'localhost' IDENTIFIED BY 'oai';  
>CREATE USER 'oai'@'%' IDENTIFIED BY 'oai';  
>CREATE DATABASE oai_db;  
>GRANT ALL ON oai_db.* TO 'oai'@'localhost'  
>GRANT ALL ON oai_db.* TO 'oai'@'%'
```

Αφού εκτελεστεί επιτυχώς η παραπάνω διαδικασία, επεξεργαζόμαστε το αρχείο ~/openair-cn/etc/hss.conf, το τροποποιούμε για την επιτυχή σύνδεση HSS και της βάσης δεδομένων και εν τέλη το αντιγράφουμε ως εξής:

```
sudo gedit ~/openair-cn/etc/hss.conf
sudo cp ~/openairinterface5g/openair-cn/etc/hss.conf /usr/local/etc/oai/
```

Αρχείο hss.conf:

```
HSS :
{
## MySQL mandatory options
MYSQL_server = "10.1.14.251"; # HSS S6a bind address
MYSQL_user = "oai"; # Database server login
MYSQL_pass = "oai"; # Database server password
MYSQL_db = "oai_db"; # Your database name

## HSS options
OPERATOR_key = "1006020f0a478bf6b699f15c062e42b3"; # OP key matching your
database
#OPERATOR_key = "11111111111111111111111111111111"; # OP key matching your
database

RANDOM = "true"; # True random or only pseudo random (for subscriber vector
generation)

## Freediameter options
FD_conf = "/usr/local/etc/oai/freeDiameter/hss_fd.conf";
};
```

Για την ολοκλήρωση εγκατάστασης του HSS θα εκτελεστεί η εντολή:

```
sudo ./run_hss -i ~/openair-cn/src/oai_hss/db/oai_db.sql
```

έτσι ώστε να γίνει η φόρτωση της βάσης δεδομένων, κάτι το οποίο απαιτείται μόνο μία φορά,την πρώτη φορά. Επιπρόσθετα στη βάση δεδομένων θα αλλαχθεί η τιμή του idmmeidentity σε δύο(2) ανάλογα με το κελί που υπάρχει το 'oai.openair4G.eur',αυτό θα γίνει με τις παρακάτω εντολές:

```
mysql -uroot -p
```

```
>use oai_db;  
>UPDATE mmeidentity SET mmehost = 'oai.openair4G.eur' WHERE idmmeidentity = 2;
```

Η επιλογή του 2 είναι η πιο λογική έτσι ώστε να μην χρειαστεί να αλλάξουμε κάτι σε κάποιον άλλο πίνακα της βάσης δεδομένων. Εν τέλη ξανά εγκαθίσταται το HSS έτσι ώστε να γίνουν ενεργές όλες οι διαφορετικές ρυθμίσεις που έχουν εκτελεστεί προηγουμένως:

```
sudo ./build_hss --clean --debug
```

και είναι έτοιμο για εκτέλεση με την εντολή:

```
./run_hss
```

### 2.5.2.2 Mobility Management Entity(MME)

Πρώτα θα εγκατασταθούν τα απαραίτητα πακέτα στο openair-cn/scripts φάκελο για το MME και στη συνέχεια θα γίνει η εγκατάσταση του, σε ότι ζητηθεί η απάντηση γίνεται με “Yes”:

```
sudo ./build_mme -i  
sudo ./build_mme
```

Μετά γίνεται η αντιγραφή του αρχείου(mme\_fd.conf) ρυθμίσεων του MME σχετικά με το HSS:

```
sudo cp ~/openair-cn/etc/mme_fd.conf /usr/local/etc/oai/freeDiameter
sudo gedit /usr/local/etc/oai/mme_fd.conf
```

και η αλλαγή ορισμένων παραμέτρων ώστε το τελικό αποτέλεσμα να είναι ως εξής:

Αρχείο mme\_fd.conf:

```
Identity = "oai.openair4G.eur";
Realm = "openair4G.eur";
ConnectPeer = "hss.openair4G.eur" { ConnectTo = "127.0.1.1"; No_SCTP ; No_IPv6;
Prefer_TCP; No_TLS; port = 3868; realm = "openair4G.eur";};
```

Ομοίως θα αντιγραφεί και θα επεξεργαστεί το αρχείο αποκλειστικά για το MME με τις δύο εντολές:

```
sudo cp ~/openair-cn/etc/mme.conf /usr/local/etc/oai/
sudo gedit /usr/local/etc/oai/mme.conf
```

Στο αρχείο ρυθμίσεων του MME ορίζονται οι διεπαφές(Interfaces) για τις συνδέσεις του SGW(S1) και eNodeB(S11) διεπαφής.

Αρχείο mme.conf:

```
NETWORK_INTERFACES :
{
# MME binded interface for S1-C or S1-MME communication (S1AP), can be ethernet
interface, virtual ethernet interface, we don't advise wireless interfaces
MME_INTERFACE_NAME_FOR_S1_MME = "ens2"; # YOUR NETWORK
CONFIG HERE
MME_IPV4_ADDRESS_FOR_S1_MME = "10.1.14.251/24"; # YOUR
```

```
NETWORK CONFIG HERE

# MME binded interface for S11 communication (GTPV2-C)
MME_INTERFACE_NAME_FOR_S11_MME      = "lo"; # YOUR NETWORK
CONFIG HERE
MME_IPV4_ADDRESS_FOR_S11_MME        = "127.0.0.1/8"; # YOUR
NETWORK CONFIG HERE
MME_PORT_FOR_S11_MME                 = 2123; # YOUR NETWORK
CONFIG HERE
};
S-GW :
{
# S-GW binded interface for S11 communication (GTPV2-C), if none selected the ITTI
message interface is used
SGW_IPV4_ADDRESS_FOR_S11             = "127.0.0.1/8"; # YOUR NETWORK
CONFIG HERE
```

Στο τελευταίο στάδιο γίνεται η εγκατάσταση του πιστοποιητικού:

```
./check_mme_s6a_certificate /usr/local/etc/oai/freeDiameter/ oai.openair4G.eur
```

και η καθαρή επανεγκατάσταση του MME και η εκτέλεση του αντίστοιχα:

```
sudo ./build_mme --clean
./run_mme.
```

### 2.5.2.3 Packet Data Network Gateway και Service Gateway(SPGW)

Για τη τελευταία υπηρεσία του κεντρικού δικτύου ακολουθείτε μία παρόμοια διαδικασία. Πρώτα γίνεται η εγκατάσταση των απαραίτητων πακέτων, στη συνέχεια της υπηρεσίας στο openair-cn/scripts και σε ότι ζητηθεί πληκτρολογείτε “Yes”:

```
sudo ./build_spgw -i
sudo ./build_spgw
```

Αμέσως μετά αντιγράφεται το αρχείο(spgw.conf) ρυθμίσεων και γίνεται η επεξεργασία του ώστε να γίνει σωστά η κατανομή των διευθύνσεων στις αντίστοιχες διεπαφές τους, με eNodeB(S1-U) και MME(S11),με τις παρακάτω εντολές:

```
sudo cp ~/openair-cn/etc/spgw.conf /usr/local/etc/oai/ &
sudo gedit /usr/etc/local/etc/oai/spgw.conf
```

Αρχείο spgw.conf:

```
S-GW :
{
  NETWORK_INTERFACES :
  {
    # S-GW binded interface for S11 communication (GTPV2-C), if none selected the
    ITTI message interface is used
    SGW_INTERFACE_NAME_FOR_S11           = "lo"; # STRING, interface name,
    YOUR NETWORK CONFIG HERE
    SGW_IPV4_ADDRESS_FOR_S11             = "127.0.0.1/8";# STRING, CIDR,
    YOUR NETWORK CONFIG HERE

    #2152 S-GW binded interface for S1-U communication (GTPV1-U) can be ethernet
    interface, virtual ethernet interface, we don't advise wireless interfaces
    SGW_INTERFACE_NAME_FOR_S1U_S12_S4_UP = "ens2"; #
    STRING, interface name, YOUR NETWORK CONFIG HERE, USE "lo" if S-GW run on
    eNB host
    SGW_IPV4_ADDRESS_FOR_S1U_S12_S4_UP   = "10.1.14.251/24"; #
    STRING, CIDR, YOUR NETWORK CONFIG HERE
    SGW_IPV4_PORT_FOR_S1U_S12_S4_UP      = 2152; # INTEGER,
    port number, PREFER NOT CHANGE UNLESS YOU KNOW WHAT YOU ARE
    DOING

    # S-GW binded interface for S5 or S8 communication, not implemented, so leave it to
    none
    SGW_INTERFACE_NAME_FOR_S5_S8_UP      = "none"; #
    STRING, interface name, DO NOT CHANGE (NOT IMPLEMENTED YET)
```



```
    SGW_IPV4_ADDRESS_FOR_S5_S8_UP      = "0.0.0.0/24";          # STRING,
CIDR, DO NOT CHANGE (NOT IMPLEMENTED YET)

};

P-GW =
{
    NETWORK_INTERFACES :
    {
        # P-GW binded interface for S5 or S8 communication, not implemented, so leave it to
none
        PGW_INTERFACE_NAME_FOR_S5_S8    = "none";
        PGW_IPV4_ADDRESS_FOR_S5_S8      = "0.0.0.0/24"; # STRING, interface
name, DO NOT CHANGE (NOT IMPLEMENTED YET)

        # P-GW binded interface for SGI (egress/ingress internet traffic)
        PGW_INTERFACE_NAME_FOR_SGI      = "ens2";              # STRING,
YOUR NETWORK CONFIG HERE
        PGW_MASQUERADE_SGI              = "yes";              # STRING, {"yes",
"no"}. YOUR NETWORK CONFIG HERE, will do NAT for you if you put "yes".
        UE_TCP_MSS_CLAMPING             = "yes";              # STRING, {"yes",
"no"}.
    };

    # Pool of UE assigned IP addresses
    # Do not make IP pools overlap
    # first IPv4 address X.Y.Z.1 is reserved for GTP network device on SPGW
    # Normally no more than 16 pools allowed, but since recent GTP kernel module use,
only one pool allowed (TODO).
    IP_ADDRESS_POOL :
    {
        IPV4_LIST = (
            "172.16.0.0/16"              # STRING, CIDR, YOUR NETWORK CONFIG
HERE(we DIDN'T edit it)
```

```
);  
};  
# DNS address communicated to UEs  
DEFAULT_DNS_IPV4_ADDRESS = "8.8.8.8"; # YOUR  
NETWORK CONFIG HERE  
DEFAULT_DNS_SEC_IPV4_ADDRESS = "8.8.4.4"; # YOUR  
NETWORK CONFIG HERE  
  
# Nonstandard feature, normally should be set to "no", but you may need to set to yes for  
# UE that do not explicitly request a PDN address through NAS signaling  
FORCE_PUSH_PROTOCOL_CONFIGURATION_OPTIONS = "no";  
# STRING, {"yes", "no"}.  
UE_MTU = 1500 # INTEGER  
};
```

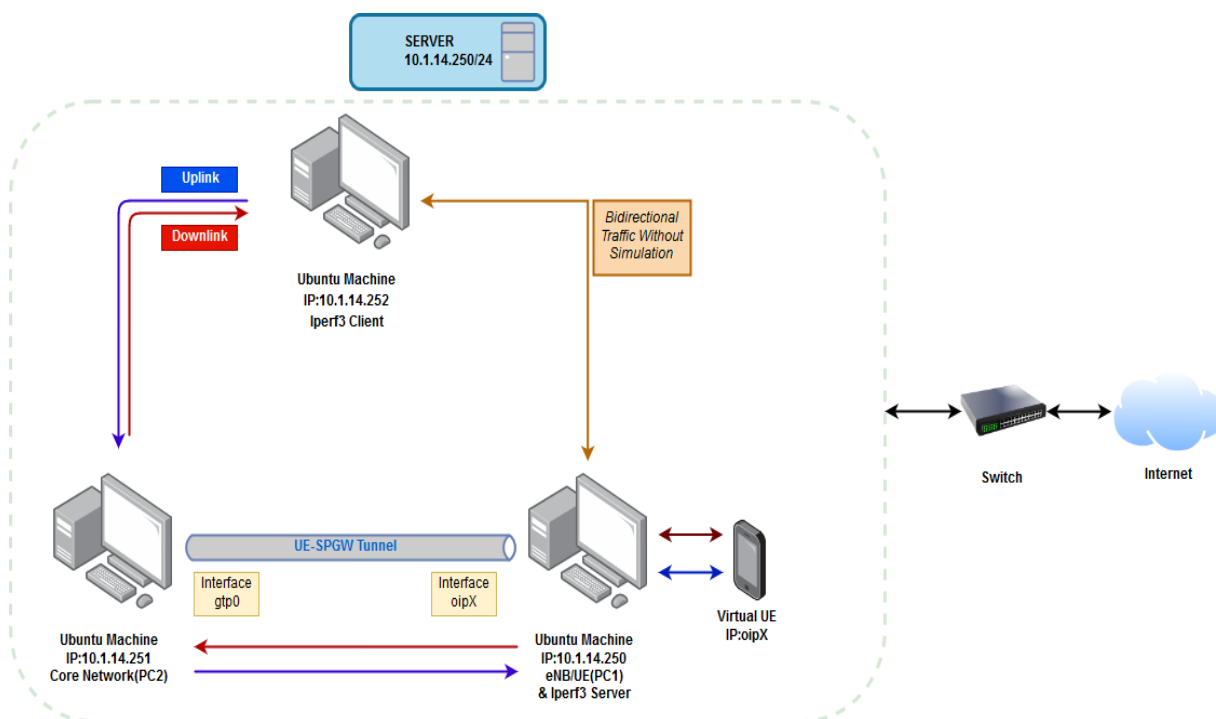
Εν τέλει ξαναγίνεται η εγκατάσταση του, για να οριστικοποιηθεί η αλλαγή στο αρχείο ρυθμίσεων καθαριστεί η προσωρινή μνήμη και είναι έτοιμο για εκτέλεση:

```
sudo ./build_spgw --clean  
./run_spgw
```

### 3.Σενάριο και Εκτέλεση της προσομοίωσης

#### 3.1 Περιγραφή Σεναρίου

Το βασικό σενάριο βασίστηκε γύρω από την μεταφορά πακέτων μεταξύ ενός πομπού και ενός δέκτη χωρίς τη χρήση Hardware, σε αυτή τη περίπτωση όμως εξετάστηκαν και τα δύο σενάρια. Πρώτα χρησιμοποιήθηκε το κενό Virtual Machine ως δέκτης (Χρήστης) με το πρόγραμμα iperf3 το οποίο είναι ικανό να στέλνει και να δέχεται TCP και UDP πακέτα αντίστοιχα από τον πομπό(Server) ο οποίος ήταν ενεργός στο PC1 μαζί με την κεραία(eNodeB) αλλά και τον εικονικό χρήστη(User equipment).Ο λόγος για τον οποίο επιλέχθηκε να είναι έτσι καταναμημένος ο ρόλος πομπού και δέκτη έτσι είναι γιατί όταν είναι ενεργοποιημένη η προσομοίωση από τη πλευρά του πομπού πρέπει να δεσμευτεί η διεύθυνση η οποία στην συγκεκριμένη περίπτωση είναι η εικονική διεύθυνση του User Equipment και όχι η διεύθυνση του μηχανήματος και αντίστοιχα της κεραίας.



Σχήμα 10. Απεικόνιση σεναρίου προσομοίωσης

Σε γενικές γραμμές ο δευτερεύον στόχος της διεξαγωγής του παρακάτω πειράματος ήταν να γίνει αντιληπτό τον αν είναι εφικτό να εκτελεστεί πάνω σε ξεχωριστές εικονικές μηχανές (VMs - Virtual Machines) η εγκατάσταση του χρήστη και της κεραίας(OAISim) και του κεντρικού δικτύου(Openair-cn).

Έτσι θα είναι πιο ευέλικτη η μεταφορά του λογισμικού με την απλή δημιουργία μιας εικόνας του Virtual Machine και η μεταφορά του σε οποιοδήποτε άλλο υπολογιστή. Επιπρόσθετα πραγματοποιείται ένα σύνολο από συγκριτικές μετρήσεις σχετικά με το ποσοστό επιτυχούς μεταφοράς πακέτων, από τον χρήστη (UE) και τη κεραία (eNodeB) προς ένα τρίτο εξυπηρετητή(Server) .

Στο σημείο αυτό θα πρέπει να τονιστεί ότι το TCP (Transmission Control Protocol) είναι το ένα από τα δύο πρωτόκολλα επιπέδου μεταφοράς που μεταφέρουν μηνύματα επιπέδου εφαρμογής. Μερικές σημαντικές υπηρεσίες που παρέχει το TCP είναι: α) η εγγυημένη παράδοση μηνυμάτων, μέσω επιβεβαίωσης από τον παραλήπτη των τμημάτων που παρελήφθησαν σωστά και επαναμετάδοσης όσων χάθηκαν ή υπέστησαν αλλοίωση κατά τη μεταφορά, β) η παράδοση τμημάτων στη σωστή σειρά, δηλαδή αυτή με την οποία στάλθηκαν από τον αποστολέα, γ) ο έλεγχος ροής, δηλαδή το ταίριασμα ανάμεσα στην ταχύτητα του αποστολέα και του παραλήπτη, δ) ο έλεγχος συμφόρησης, δηλαδή η ρύθμιση του ρυθμού μετάδοσης ώστε να ταιριάζει με τις συνθήκες συμφόρησης του δικτύου, και, ε) ο τεμαχισμός των μηνυμάτων αποστολέα σε μικρότερα τμήματα.

Πριν ξεκινήσει η ανταλλαγή μηνυμάτων μέσω TCP είναι απαραίτητη η εγκαθίδρυση της σύνδεσης μέσω μιας ειδικής διαδικασίας που ονομάζεται χειραψία (TCP handshake) που καθορίζεται από το πρωτόκολλο και εκτελείται από τα τερματικά. Από την άλλη το UDP (User Datagram Protocol) είναι το δεύτερο πρωτόκολλο επιπέδου μεταφοράς (το πρώτο είναι το TCP) και ουσιαστικά κάνει τα ελάχιστα που μπορεί να κάνει ένα πρωτόκολλο επιπέδου μεταφοράς.

Το UDP δεν προσφέρει εγγυημένη παράδοση μηνυμάτων στη σωστή σειρά ή επιβεβαιώσεις παράδοσης, κανένα έλεγχο ροής και συμφόρησης και δεν χρειάζεται ειδική διαδικασία εγκαθίδρυσης σύνδεσης πριν την ανταλλαγή μηνυμάτων. Πρακτικά η μόνη υπηρεσία που παρέχει είναι η πολυπλεξία μέσω του μηχανισμού των θυρών οπότε μια εφαρμογή που χρησιμοποιεί το UDP είναι σαν να χρησιμοποιεί το επίπεδο Διαδικτύου απευθείας.

Το πρωτόκολλο UDP αφού δεν παρέχει όλες τις χρήσιμες υπηρεσίες του TCP επιλέγεται επειδή: α) έχει μικρή καθυστέρηση αποστολής και η άμεση ανταλλαγή μηνυμάτων εξαιτίας της πολύ μικρής επικεφαλίδας και του μη καθορισμού σύνδεσης, β) η εξοικονόμηση πόρων στα τερματικά επειδή δεν χρειάζεται η διατήρηση κατάσταση σύνδεσης και γ) ο καλύτερος έλεγχος για το ποια δεδομένα αποστέλλονται και τότε, κυρίως επειδή δεν υπάρχει υποχρεωτική επαναμετάδοση πακέτων και αγνοούνται οι συνθήκες συμφόρησης του δικτύου.

Τέλος τα αποτελέσματα στα πλαίσια της διπλωματικής εργασίας παρήχθησαν από την μεταφορά πακέτων μεταξύ του Server και του Client του iperf3 χωρίς να είναι ενεργοποιημένη η προσομοίωση. Και από το Client προς το Server για να παρθούν όλες οι απαιτούμενες μετρήσεις μέσω του Κεντρικού Δικτύου και χρήση του OaiSim Tunnel για να φτάσουν τα δεδομένα στο προορισμό τους.

### 3.2 Ενέργειες πριν την εκτέλεση

Για το συγκεκριμένο σενάριο είναι σημαντικό να προηγηθούν κάποια βασικά βήματα πριν την εκτέλεση των εντολών που θα ξεκινήσουν τη προσομοίωση. Αρχικά, είναι σημαντικό στο PC1 να γίνει και η εγκατάσταση του ue\_ip.ko πακέτου για τον πυρήνα του συστήματος, με σκοπό να δημιουργήσει την εικονική διεύθυνση για το Εξοπλισμό Χρήστη (UE).

Η διεύθυνση που θα παίρνει κάθε φορά θα ορίζεται από το SPGW μέσω του MME. Αρχικά στο φάκελο του openairinterface5g χρησιμοποιείται η εντολή:

```
source oaienv
```

και γίνεται η μετακίνηση μέσα στο φάκελο cmake\_targets και tools. Εκεί βρίσκεται το αρχείο με όνομα: init\_nas\_s1 και είναι αναγκαίο να του δοθούν όλα τα δικαιώματα και να μπορεί να χρησιμοποιηθεί ως εκτελέσιμο αρχείο, αυτό θα επιτευχθεί με την παρακάτω εντολή:

```
sudo chmod 777 init_nas_s1
```

Μόλις γίνει το παρακάτω βήμα στον ίδιο φάκελο εκτελείται η εντολή και θα γίνει φανερό το ότι έχει δημιουργηθεί η διεύθυνση για τον Εξοπλισμό Χρήστη.

```
sudo -E ./init_nas_s1 UE
```

Τέλος στο PC2 εκτελούνται σε 3 διαφορετικά τερματικά για το κάθε ένα και οι λειτουργίες του κεντρικού δικτύου στο φάκελο openair-cn/scripts με την παρακάτω σειρά η οποία είναι σημαντικό να μην αλλάζει για να μην προκύψουν προβλήματα στην προσομοίωση:

- 1) ./run\_hss
- 2) ./run\_mme
- 3) ./run\_spgw

### 3.3 Εκτέλεση Προσομοίωσης

Η εκτέλεση της προσομοίωσης γίνεται στο PC1 και είναι σχετικά απλή συγκριτικά με την προεργασία που χρειάστηκε να γίνει για να ολοκληρωθεί η ρύθμιση κάθε υπηρεσίας που χρησιμοποιηθεί.

Γίνεται η είσοδος στον υποφάκελο του `openairinterface5g/cmake_targets/oaisim_build_oai/build` και εκτελείται η εντολή:

```
sudo-E ./oaisim -O ~/openairinterface5g/targets/PROJECTS/GENERIC-LTE-  
EPC/CONF/enb.band7.generic.oaisim.local_mme.conf -c 1
```

Η παραπάνω εντολή ενεργοποιεί τον προσομοιωτή και η παράμετρο `-c` προσδιορίζει ένα αρχείο τύπου `.xml` το οποίο ονομάζεται `OpenAirInterface Scenario Descriptor(OSD)`.

Τη στιγμή που ξεκινάει η προσομοίωση ανατίθεται και η διεύθυνση για την διεπαφή του εξοπλισμού του χρήστη που δημιουργήθηκε πριν, οπότε με την εκτέλεση της εντολής(PC1) `ifconfig` είναι δυνατό να φανεί η διεύθυνση της κεραίας(eNodeB) η οποία είναι η βασική διεύθυνση του ενσύρματου δικτύου και η εικονική για τον Εξοπλισμό του χρήστη(UE).

### 3.3.1 Επιτυχία Σύνδεσης eNB, UE και EPC

#### 1) Επιτυχία Σύνδεσης - Home Subscriber Server (HSS)

```
0 rows affected
IMSI: 2081000001236Key: 8b.af.47.3f.2f.8f.d0.94.87.cc.cb.d7.09.7c.68.62.
OPc: e7.34.f8.73.40.07.d6.c5.ce.7a.05.08.80.9e.7e.9c.
RijndaelKeySchedule: K 8BAF473F2F8FD09487CCCB7097C6862
Compute opc:
  K:      8BAF473F2F8FD09487CCCB7097C6862
  In:     1006020F0A478BF6B699F15C062E42B3
  Rinj:   F732FA7C4A405D3378E3F45486B03C2F
  Out:    E734F8734007D6C5CE7A0508809E7E9C
Query: UPDATE `users` SET `OPc`=UNHEX('e734f8734007d6c5ce7a0508809e7e9c') WHERE `users`.`imsi`='2081000001236'
IMSI 2081000001236 Updated OPc e734f8734007d6c5ce7a0508809e7e9c -> e734f8734007d6c5ce7a0508809e7e9c
0 rows affected
Initializing s6a layer
11/26/18,10:50:28.195082  NOTI  libfdproto '1.2.0' initialized.
11/26/18,10:50:28.195128  NOTI  libgnutls '3.4.10' initialized.
11/26/18,10:50:28.195542  DBG   Core state: 0 -> 1
11/26/18,10:50:28.195555  NOTI  libfdcore '1.2.0' initialized.
11/26/18,10:50:28.196896  DBG   Generating fresh Diffie-Hellman parameters of size 1024 (this takes some time)...
11/26/18,10:50:28.303177  DBG   Loading : /usr/local/lib/freeDiameter/acl_wl.fdx
11/26/18,10:50:28.303383  NOTI  Extension ACL_wl initialized with configuration: '/usr/local/etc/oai/freeDiameter/acl.conf'
11/26/18,10:50:28.303398  DBG   Loading : /usr/local/lib/freeDiameter/dict_nas_mip6.fdx
11/26/18,10:50:28.303493  DBG   Dictionary Extension 'MIPv6 NAS-to-HAAA Interaction' initialized
11/26/18,10:50:28.303501  DBG   Loading : /usr/local/lib/freeDiameter/dict_s6a.fdx
11/26/18,10:50:28.304576  NOTI  Dictionary Extension 'S6A from 3GPP standard v.10.5' initialized
11/26/18,10:50:28.304590  NOTI  All extensions loaded.
11/26/18,10:50:28.304618  NOTI  freeDiameter configuration:
11/26/18,10:50:28.304624  NOTI  Default trace level .... : +1
11/26/18,10:50:28.304629  NOTI  Configuration file .... : /usr/local/etc/oai/freeDiameter/hss_fd.conf
11/26/18,10:50:28.304634  NOTI  Diameter Identity ..... : hss.openair4G.eur (1:17)
11/26/18,10:50:28.304643  NOTI  Diameter Realm ..... : openair4G.eur (1:13)
11/26/18,10:50:28.304647  NOTI  Tc Timer ..... : 30
11/26/18,10:50:28.304651  NOTI  Tw Timer ..... : 30
11/26/18,10:50:28.304656  NOTI  Local port ..... : 3868
11/26/18,10:50:28.304660  NOTI  Local secure port ..... : 5868
11/26/18,10:50:28.304664  NOTI  Number of SCTP streams . : 3
11/26/18,10:50:28.304669  NOTI  Number of clients thr . . : 5
11/26/18,10:50:28.304673  NOTI  Number of app threads . . : 4
11/26/18,10:50:28.304677  NOTI  Local endpoints ..... : Default (use all available)
11/26/18,10:50:28.304681  NOTI  Local applications ..... : (none)
11/26/18,10:50:28.304686  NOTI  Flags : - IP ..... : Enabled
11/26/18,10:50:28.304690  NOTI  - IPv6 ..... : DISABLED
11/26/18,10:50:28.304694  NOTI  - Relay app .... : DISABLED
11/26/18,10:50:28.304698  NOTI  - TCP ..... : Enabled
11/26/18,10:50:28.304702  NOTI  - SCTP ..... : DISABLED
11/26/18,10:50:28.304724  NOTI  - Pref. proto .. : TCP
11/26/18,10:50:28.304733  NOTI  - TLS method ... : Separate port
11/26/18,10:50:28.304738  NOTI  TLS : - Certificate .. : /usr/local/etc/oai/freeDiameter/hss.cert.pem
11/26/18,10:50:28.304743  NOTI  - Private key .. : /usr/local/etc/oai/freeDiameter/hss.key.pem
11/26/18,10:50:28.304748  NOTI  - CA (trust) ... : /usr/local/etc/oai/freeDiameter/hss.cacert.pem (1 certs)
11/26/18,10:50:28.304759  NOTI  - CRL ..... : (none)
11/26/18,10:50:28.304764  NOTI  - Priority ..... : (default: 'NORMAL')
11/26/18,10:50:28.304769  NOTI  - DH bits ..... : 1024
11/26/18,10:50:28.304773  NOTI  Origin-State-Id ..... : 1543225828
11/26/18,10:50:28.304788  NOTI  Loaded extensions: '/usr/local/lib/freeDiameter/acl_wl.fdx' [/usr/local/etc/oai/freeDiameter/acl.conf], loaded
11/26/18,10:50:28.304793  NOTI  Loaded extensions: '/usr/local/lib/freeDiameter/dict_nas_mip6.fdx' [(no config file)], loaded
11/26/18,10:50:28.304799  NOTI  Loaded extensions: '/usr/local/lib/freeDiameter/dict_s6a.fdx' [(no config file)], loaded
11/26/18,10:50:28.304813  DBG   Core state: 1 -> 2
11/26/18,10:50:28.305194  NOTI  Local server address(es): 10.1.14.251{---L-} 172.16.0.1{---L-}
11/26/18,10:50:28.305223  DBG   Core state: 2 -> 3
Initializing s6a layer: DONE
```

## 2)Επιτυχία Σύνδεσης - Mobility Management Entity(MME)

```

000151 000000:746329 7F75AD7FA700 ALERT S6A ai/openair-cn/src/s6a/s6a_task.c:0080 'hss.openair4G.eur' in state 'STATE_CLOSED' waiting for next event.
000152 000000:746343 7F75D52EC700 ALERT S6A ai/openair-cn/src/s6a/s6a_task.c:0080 Core state: 2 -> 3
000153 000000:746356 7F75D52EC700 DEBUG S6A ai/openair-cn/src/s6a/s6a_task.c:0219 fd core start done
000154 000000:746365 7F75D52EC700 DEBUG S6A ai/openair-cn/src/s6a/s6a_task.c:0229 fd core waitstartcomplete done
000155 000000:746398 7F75AD7FA700 ALERT S6A ai/openair-cn/src/s6a/s6a_task.c:0080 'STATE_CLOSED' <-- 'FDEVP_BSM_TIMEOUT' (nil),0 'hss.openair4G.eur'
000156 000000:746409 7F75AD7FA700 ALERT S6A ai/openair-cn/src/s6a/s6a_task.c:0080 hss.openair4G.eur: Connecting...
000157 000000:746419 7F75AD7FA700 ALERT S6A ai/openair-cn/src/s6a/s6a_task.c:0080 'STATE_CLOSED' -> 'STATE_WAITCNACK' 'hss.openair4G.eur'
000158 000000:746429 7F75AD7FA700 ALERT S6A ai/openair-cn/src/s6a/s6a_task.c:0080 Peer timeout reset to 10 seconds
000159 000000:746435 7F75D52EC700 DEBUG S6A ai/openair-cn/src/s6a/s6a_task.c:0236 s6a_fd_inir_dicit_objs done
000160 000000:746481 7F75AD7FA700 ALERT S6A ai/openair-cn/src/s6a/s6a_task.c:0080 'hss.openair4G.eur' in state 'STATE_WAITCNACK' waiting for next event.
000161 000000:746532 7F75D52EC700 DEBUG S6A ai/openair-cn/src/s6a/s6a_task.c:0243 Initializing S6a interface: DONE
000162 000000:746560 7F75D52EC700 DEBUG MME-AP openair-cn/src/oa_mme/oa1_mme.c:0149 MME app initialisation complete
000163 000000:746584 7F7555FBF700 ALERT S6A ai/openair-cn/src/s6a/s6a_task.c:0080 Prepared 1 setx of connection parameters to peer hss.openair4G.eur
000164 000000:746603 7F7555FBF700 ALERT S6A ai/openair-cn/src/s6a/s6a_task.c:0080 Connecting to TCP 127.0.1.1(3868)...
000165 000000:747033 7F75AD7FA700 ALERT S6A ai/openair-cn/src/s6a/s6a_task.c:0080 'STATE_WAITCNACK' <-- 'FDEVP_CNK_ESTABLISHED' (0x7f753000a10,0) 'hss.openair4G.eur'
000166 000000:747055 7F75AD7FA700 ALERT S6A ai/openair-cn/src/s6a/s6a_task.c:0080 hss.openair4G.eur: Connection established, (----) TCP,#38->127.0.1.1(3868)
000167 000000:747115 7F75AD7FA700 ALERT S6A ai/openair-cn/src/s6a/s6a_task.c:0080 SENT to 'hss.openair4G.eur': 'Capabilities-Exchange-Request'/'0/257 f:R--- src:'(nil)' len:216
000168 000000:747121 7F75AD7FA700 ALERT S6A ai/openair-cn/src/s6a/s6a_task.c:0080 C:264/1:25,C:266/1:12,C:278/1:12,C:257/1:14,C:257/1:14,C:266/1:12,C:269/1:20,C:267/1:12,C:259/1:12,C:260/1:32,C:265/1:12)
000169 000000:747126 7F75AD7FA700 ALERT S6A ai/openair-cn/src/s6a/s6a_task.c:0080 Sending 216b data on connection (----) TCP,#38->127.0.1.1(3868)
000170 000000:747172 7F75AD7FA700 ALERT S6A ai/openair-cn/src/s6a/s6a_task.c:0080 'STATE_WAITCNACK' -> 'STATE_WAITCEA' 'hss.openair4G.eur'
000171 000000:747179 7F75AD7FA700 ALERT S6A ai/openair-cn/src/s6a/s6a_task.c:0080 Peer timeout reset to 10 seconds
000172 000000:747185 7F75AD7FA700 ALERT S6A ai/openair-cn/src/s6a/s6a_task.c:0080 'hss.openair4G.eur' in state 'STATE_WAITCEA' waiting for next event.
000173 000000:747913 7F7554FEF700 ALERT S6A ai/openair-cn/src/s6a/s6a_task.c:0080 Thread terminated
000174 000000:747913 7F75AD7FA700 ALERT S6A ai/openair-cn/src/s6a/s6a_task.c:0080 'STATE_WAITCEA' <-- 'FDEVP_CNK_MSG_RECV' (0x7f75340008f0,200) 'hss.openair4G.eur'
000175 000000:747942 7F75AD7FA700 ALERT S6A ai/openair-cn/src/s6a/s6a_task.c:0080 RCV from 'hss.openair4G.eur': (no model)0/257 f:---- src:'hss.openair4G.eur' len:200 (C:268/1:12,C:264/1:25,C:266/1:12,C:278/1:12,C:257/1:14,C:266/1:12,C:269/1:20,C:267/1:12,C:260/1:32,C:265/1:12)
000176 000000:747961 7F75AD7FA700 ALERT S6A ai/openair-cn/src/s6a/s6a_task.c:0080 Iterating on rules of AVP: 'Vendor-Specific-Application-Id'.
000177 000000:747969 7F75AD7FA700 ALERT S6A ai/openair-cn/src/s6a/s6a_task.c:0080 Iterating on rules of COMMAND: 'Capabilities-Exchange-Answer'.
000178 000000:748020 7F75AD7FA700 ERROR S6A ai/openair-cn/src/s6a/s6a_task.c:0080 'hss.openair4G.eur' claims support for a subset of vendor 10415 features.
000179 000000:748027 7F75AD7FA700 ERROR S6A ai/openair-cn/src/s6a/s6a_task.c:0080 Connected to 'hss.openair4G.eur' (TCP,sock#38), remote capabilities:
000180 000000:748035 7F75AD7FA700 ERROR S6A ai/openair-cn/src/s6a/s6a_task.c:0080 'Capabilities-Exchange-Answer'
000181 000000:748042 7F75AD7FA700 ERROR S6A ai/openair-cn/src/s6a/s6a_task.c:0080 Version: 0x01
000182 000000:748046 7F75AD7FA700 ERROR S6A ai/openair-cn/src/s6a/s6a_task.c:0080 Length: 200
000183 000000:748054 7F75AD7FA700 ERROR S6A ai/openair-cn/src/s6a/s6a_task.c:0080 Flags: 0x00 (----)
000184 000000:748060 7F75AD7FA700 ERROR S6A ai/openair-cn/src/s6a/s6a_task.c:0080 Command Code: 257
000185 000000:748065 7F75AD7FA700 ERROR S6A ai/openair-cn/src/s6a/s6a_task.c:0080 ApplicationId: 0
000186 000000:748072 7F75AD7FA700 ERROR S6A ai/openair-cn/src/s6a/s6a_task.c:0080 Hop-by-Hop Identifier: 0x759EB86A
000187 000000:748079 7F75AD7FA700 ERROR S6A ai/openair-cn/src/s6a/s6a_task.c:0080 End-to-End Identifier: 0x03457880
000188 000000:748086 7F75AD7FA700 ERROR S6A ai/openair-cn/src/s6a/s6a_task.c:0080 (internal data): src:hss.openair4G.eur(17) rxb:(nil) rt:2 cb:(nil), (nil) (nil) qry:0x7f5c013010 ssa:0 sba:0
000189 000000:748093 7F75AD7FA700 ERROR S6A ai/openair-cn/src/s6a/s6a_task.c:0080 AVP: 'Result-Code' (268) l=12 f=M val='DIAMETER_SUCCESS' (2001 (0x7d1))
000190 000000:748099 7F75AD7FA700 ERROR S6A ai/openair-cn/src/s6a/s6a_task.c:0080 AVP: 'Origin-Host' (264) l=25 f=M val='hss.openair4G.eur'
000191 000000:748105 7F75AD7FA700 ERROR S6A ai/openair-cn/src/s6a/s6a_task.c:0080 AVP: 'Origin-Realm' (296) l=21 f=M val='openair4G.eur'
000192 000000:748111 7F75AD7FA700 ERROR S6A ai/openair-cn/src/s6a/s6a_task.c:0080 AVP: 'Origin-State-Id' (278) l=12 f=M val='543225150 (0x5bfbfb3e)'
000193 000000:748117 7F75AD7FA700 ERROR S6A ai/openair-cn/src/s6a/s6a_task.c:0080 AVP: 'Host-IP-Address' (257) l=14 f=M val='10.1.14.251'
000194 000000:748123 7F75AD7FA700 ERROR S6A ai/openair-cn/src/s6a/s6a_task.c:0080 AVP: 'Vendor-Id' (266) l=12 f=M val=0x0 (0x0)
000195 000000:748129 7F75AD7FA700 ERROR S6A ai/openair-cn/src/s6a/s6a_task.c:0080 AVP: 'Product-Name' (269) l=20 f=- val='freeDiameter'
000196 000000:748135 7F75AD7FA700 ERROR S6A ai/openair-cn/src/s6a/s6a_task.c:0080 AVP: 'Firmware-Revision' (267) l=12 f=- val=10200 (0x27d8)
000197 000000:748142 7F75AD7FA700 ERROR S6A ai/openair-cn/src/s6a/s6a_task.c:0080 AVP: 'Vendor-Specific-Application-Id' (260) l=32 f=M val=16777251 (0x1000023)
000198 000000:748148 7F75AD7FA700 ERROR S6A ai/openair-cn/src/s6a/s6a_task.c:0080 AVP: 'Vendor-Id' (266) l=12 f=M val=10415 (0x28af)
000199 000000:748155 7F75AD7FA700 ERROR S6A ai/openair-cn/src/s6a/s6a_task.c:0080 AVP: 'Supported-Vendor-Id' (263) l=12 f=M val=10415 (0x28af)
002000 000000:748165 7F75AD7FA700 ERROR S6A ai/openair-cn/src/s6a/s6a_task.c:0080 No TLS protection negotiated with peer 'hss.openair4G.eur'.
002001 000000:748184 7F75AD7FA700 ERROR S6A ai/openair-cn/src/s6a/s6a_task.c:0080 'STATE_WAITCEA' -> 'STATE_OPEN' 'hss.openair4G.eur'
002002 000000:748216 7F75AD7FA700 ALERT S6A ai/openair-cn/src/s6a/s6a_task.c:0080 Peer timeout reset to 30 seconds (+/- 2)
002003 000000:748226 7F75AD7FA700 ALERT S6A ai/openair-cn/src/s6a/s6a_task.c:0080 'hss.openair4G.eur' in state 'STATE_OPEN' waiting for next event.
002004 000000:746606 7F7555FBF700 DEBUG S6A ai/openair-cn/src/s6a/s6a_peer.c:0115 Diameter identity of MME: epc.openair4G.eur with length: 17
002005 000000:746619 7F7555FBF700 DEBUG S6A ai/openair-cn/src/s6a/s6a_peer.c:0150 S6a peer connection attempt 1 / 8
002006 000000:746628 7F7555FBF700 DEBUG S6A ai/openair-cn/src/s6a/s6a_peer.c:0163 Peer hss.openair4G.eur is now connected...

```

Η παρακάτω εικόνα εμφανίζεται κατά αυτό το τρόπο αφού γίνει και η εκτέλεση ολόκληρης της προσομοίωσης:

Current Status	Added since last display	Removed since last display
Connected eNBs	1	0
Attached UEs	1	0
Connected UEs	1	0
Default Bearers	1	0
SI-U Bearers	1	0



### 3)Επιτυχία Σύνδεσης - Packet Data Network Gateway και Service Gateway(SPGW)

```
000091 00000:585617 7F7E95DF8700 INFO SPGW-A /openair-cn/src/sgw/pgw_config.c:0388 S5_S8 iface .....: ens3
000092 00000:585622 7F7E95DF8700 INFO SPGW-A /openair-cn/src/sgw/pgw_config.c:0389 S5_S8 ip (read).....: 10.1.14.251
000093 00000:585627 7F7E95DF8700 INFO SPGW-A /openair-cn/src/sgw/pgw_config.c:0390 S5_S8 MTU (read).....: 1500
000094 00000:585632 7F7E95DF8700 INFO SPGW-A /openair-cn/src/sgw/pgw_config.c:0391 - SGI;
000095 00000:585637 7F7E95DF8700 INFO SPGW-A /openair-cn/src/sgw/pgw_config.c:0392 SGI iface .....: ens3
000096 00000:585642 7F7E95DF8700 INFO SPGW-A /openair-cn/src/sgw/pgw_config.c:0393 SGI ip (read).....: 10.1.14.251
000097 00000:585647 7F7E95DF8700 INFO SPGW-A /openair-cn/src/sgw/pgw_config.c:0394 SGI MTU (read).....: 1500
000098 00000:585652 7F7E95DF8700 INFO SPGW-A /openair-cn/src/sgw/pgw_config.c:0396 - MSS clamping: .....: 1
000099 00000:585657 7F7E95DF8700 INFO SPGW-A /openair-cn/src/sgw/pgw_config.c:0397 - Masquerading: .....: 1
000100 00000:585662 7F7E95DF8700 INFO SPGW-A /openair-cn/src/sgw/pgw_config.c:0398 - Push FCO: .....: 0
Initializing MSC logs
Initializing MSC logs Done
000101 00000:618587 7F7E95DF8700 DEBUG UDP /src/udp/udp_primitives_server.c:0356 Initializing UDP task interface
000102 00000:618630 7F7E95DF8700 DEBUG UDP /src/udp/udp_primitives_server.c:0364 Initializing UDP task interface: DONE
000103 00000:618648 7F7E95DF8700 DEBUG S11 oai/openair-cn/src/s11/s11_sgw.c:0259 Initializing S11 interface
000104 00000:618963 7F7E95DF8700 INFO GTPv2- 2-c/nwgtprv2c-0.11/src/NwGtprv2c.c:0218 *-----*
000105 00000:618969 7F7E95DF8700 INFO GTPv2- 2-c/nwgtprv2c-0.11/src/NwGtprv2c.c:0219 * *
000106 00000:618977 7F7E95DF8700 INFO GTPv2- 2-c/nwgtprv2c-0.11/src/NwGtprv2c.c:0220 * n w - g t p v 2 c *
000107 00000:618982 7F7E95DF8700 INFO GTPv2- 2-c/nwgtprv2c-0.11/src/NwGtprv2c.c:0221 * G P R S T u n n e l i n g P r o t o c o l v 2 c S t a c k *
000108 00000:618986 7F7E95DF8700 INFO GTPv2- 2-c/nwgtprv2c-0.11/src/NwGtprv2c.c:0222 * *
000109 00000:618989 7F7E95DF8700 INFO GTPv2- 2-c/nwgtprv2c-0.11/src/NwGtprv2c.c:0223 * *
000110 00000:618995 7F7E95DF8700 INFO GTPv2- 2-c/nwgtprv2c-0.11/src/NwGtprv2c.c:0224 * Copyright (c) 2010-2011 Amit Chawre *
000111 00000:619001 7F7E95DF8700 INFO GTPv2- 2-c/nwgtprv2c-0.11/src/NwGtprv2c.c:0225 * All rights reserved. *
000112 00000:619005 7F7E95DF8700 INFO GTPv2- 2-c/nwgtprv2c-0.11/src/NwGtprv2c.c:0226 * *
000113 00000:619009 7F7E95DF8700 INFO GTPv2- 2-c/nwgtprv2c-0.11/src/NwGtprv2c.c:0227 * Redistribution and use in source and binary forms, with or without *
000114 00000:619014 7F7E95DF8700 INFO GTPv2- 2-c/nwgtprv2c-0.11/src/NwGtprv2c.c:0228 * modification, are permitted provided that the following conditions *
000115 00000:619019 7F7E95DF8700 INFO GTPv2- 2-c/nwgtprv2c-0.11/src/NwGtprv2c.c:0229 * are met: *
000116 00000:619021 7F7E95DF8700 INFO GTPv2- 2-c/nwgtprv2c-0.11/src/NwGtprv2c.c:0230 * *
000117 00000:619026 7F7E95DF8700 INFO GTPv2- 2-c/nwgtprv2c-0.11/src/NwGtprv2c.c:0231 * 1. Redistributions of source code must retain the above copyright *
000118 00000:619031 7F7E95DF8700 INFO GTPv2- 2-c/nwgtprv2c-0.11/src/NwGtprv2c.c:0232 * notice, this list of conditions and the following disclaimer. *
000119 00000:619035 7F7E95DF8700 INFO GTPv2- 2-c/nwgtprv2c-0.11/src/NwGtprv2c.c:0233 * 2. Redistributions in binary form must reproduce the above copyright *
000120 00000:619038 7F7E95DF8700 INFO GTPv2- 2-c/nwgtprv2c-0.11/src/NwGtprv2c.c:0234 * notice, this list of conditions and the following disclaimer in the *
000121 00000:619040 7F7E95DF8700 INFO GTPv2- 2-c/nwgtprv2c-0.11/src/NwGtprv2c.c:0235 * documentation and/or other materials provided with the distribution. *
000122 00000:619045 7F7E95DF8700 INFO GTPv2- 2-c/nwgtprv2c-0.11/src/NwGtprv2c.c:0236 * 3. The name of the author may not be used to endorse or promote products *
000123 00000:619049 7F7E95DF8700 INFO GTPv2- 2-c/nwgtprv2c-0.11/src/NwGtprv2c.c:0237 * derived from this software without specific prior written permission. *
000124 00000:619054 7F7E95DF8700 INFO GTPv2- 2-c/nwgtprv2c-0.11/src/NwGtprv2c.c:0238 * *
000125 00000:619058 7F7E95DF8700 INFO GTPv2- 2-c/nwgtprv2c-0.11/src/NwGtprv2c.c:0239 * THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS'' AND ANY EXPRESS OR *
000126 00000:619061 7F7E95DF8700 INFO GTPv2- 2-c/nwgtprv2c-0.11/src/NwGtprv2c.c:0240 * IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES *
000127 00000:619063 7F7E95DF8700 INFO GTPv2- 2-c/nwgtprv2c-0.11/src/NwGtprv2c.c:0241 * OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. *
000128 00000:619067 7F7E95DF8700 INFO GTPv2- 2-c/nwgtprv2c-0.11/src/NwGtprv2c.c:0242 * IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT, *
000129 00000:619072 7F7E95DF8700 INFO GTPv2- 2-c/nwgtprv2c-0.11/src/NwGtprv2c.c:0243 * INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT *
000130 00000:619077 7F7E95DF8700 INFO GTPv2- 2-c/nwgtprv2c-0.11/src/NwGtprv2c.c:0244 * NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, *
000131 00000:619081 7F7E95DF8700 INFO GTPv2- 2-c/nwgtprv2c-0.11/src/NwGtprv2c.c:0245 * DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY *
000132 00000:619086 7F7E95DF8700 INFO GTPv2- 2-c/nwgtprv2c-0.11/src/NwGtprv2c.c:0246 * THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT *
000133 00000:619088 7F7E95DF8700 INFO GTPv2- 2-c/nwgtprv2c-0.11/src/NwGtprv2c.c:0247 * (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF *
000134 00000:619092 7F7E95DF8700 INFO GTPv2- 2-c/nwgtprv2c-0.11/src/NwGtprv2c.c:0248 * THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE. *
000135 00000:619097 7F7E95DF8700 INFO GTPv2- 2-c/nwgtprv2c-0.11/src/NwGtprv2c.c:0249 *-----*
000136 00000:619140 7F7E95DF8700 DEBUG S11 oai/openair-cn/src/s11/s11_sgw.c:0244 Tx UDP_INIT IP addr 127.0.11.2
000137 00000:619149 7F7E95DF8700 DEBUG S11 oai/openair-cn/src/s11/s11_sgw.c:0301 Initializing S11 interface: DONE
000138 00000:619155 7F7E95DF8700 DEBUG SPGW-A ai/openair-cn/src/sgw/sgw_task.c:0148 Initializing SPGW-APP task interface
000139 00000:619158 7F7E95DF8700 DEBUG GTPv1- air-cn/src/gtpv1-u/gtpv1u_task.c:0096 Initializing GTPV1U interface
000140 00000:619164 7F7E95DF8700 DEBUG UDP /src/udp/udp_primitives_server.c:0126 Creating new listen socket on address 127.0.11.2 and port 2123
000141 00000:619735 7F7E95DF8700 DEBUG UDP /src/udp/udp_primitives_server.c:0171 Inserting new descriptor for task 6, sd 31
000142 00000:619790 7F7E95DF8700 DEBUG UDP /src/udp/udp_primitives_server.c:0187 Received 1 events
000143 00000:672203 7F7E95DF8700 NOTICE GTPv1- -cn/src/gtpv1-u/gtp_mod_kernel.c:0081 Using the GTP kernel mode (genl ID is 27)
000144 00000:677067 7F7E95DF8700 DEBUG GTPv1- -cn/src/gtpv1-u/gtp_mod_kernel.c:0104 Setting route to reach UE net 172.16.0.0 via gtp0
000145 00000:677147 7F7E95DF8700 NOTICE GTPv1- -cn/src/gtpv1-u/gtp_mod_kernel.c:0111 GTP kernel configured
000146 00000:677207 7F7E95DF8700 DEBUG GTPv1- air-cn/src/gtpv1-u/gtpv1u_task.c:0124 Initializing GTPV1U interface: DONE
000147 00000:682250 7F7E95DF8700 DEBUG SPGW-A ai/openair-cn/src/sgw/sgw_task.c:0208 Initializing SPGW-APP task interface: DONE
```

### 3.4 Συνοπτική Εκτέλεση Προσομοίωσης

Αφού γίνει η εκτέλεση όλων των παραπάνω βημάτων την πρώτη φορά, από τη δεύτερη φορά κάποια βήματα μπορούν να παραλειφθούν. Υποθετικά θα θεωρηθεί ότι έχει γίνει επανεκκίνηση του Virtual Machine και για το PC1 και το PC2 και είναι επιθυμητό να γίνει από την αρχή η εκτέλεση της προσομοίωσης.

1. Στο PC1 εκτελούνται οι παρακάτω εντολές στο τερματικό:

```
cd openairinterface
source oaienv
cd cmake_targets
cd tools
sudo -E ./init_nas_s1 UE // Με αυτή την εντολή ενεργοποιείται ξανά η εικονική
διεύθυνση για τον εξοπλισμό χρήστη. Συγκεκριμένη IP θα ανατεθεί μόλις ξεκινήσει η
προσομοίωση (βήμα 3).
```

2. Στο PC2 εκτελούνται οι παρακάτω εντολές σε τρία διαφορετικά τερματικά αντίστοιχα:

```
./run_hss
./run_mme
./run_sprg //με την ίδια σειρά όπως και στην πρώτη εκτέλεση
```

3. Στο PC1 εκτελείται η παρακάτω εντολή στο τερματικό για την έναρξη της προσομοίωσης:

```
sudo-E ./oaisim -O ~/openairinterface5g/targets/PROJECTS/GENERIC-LTE-
EPC/CONF/enb.band7.generic.oaisim.local_mme.conf -c 1 //σε νέο τερματικό
```

## 4. Αποτελέσματα Προσομοίωσης

---

### 4.1 Μεταφορά Αρχείου-Περίπτωση TCP πακέτων

Με την έναρξη της προσομοίωσης όπως αναφέρθηκε προηγουμένως θέτουμε σε χρήση το iperf3 εργαλείο. Αρχικά στο PC1 δηλαδή το Virtual Machine με το eNB και το User Equipment εκτελείται η παρακάτω εντολή σε ένα τερματικό έτσι ώστε να ενεργοποιηθεί ως iPerf3 Server το μηχάνημα:

```
iperf3 -s -B 172.16.0.X -p 9999
```

-s (IP):λειτουργία ως server της συγκεκριμένης διεύθυνσης

-B:Δέσμευση της διεύθυνσης που ακολουθεί ως host

-p 9999:Χρήση του Port 9999 για επικοινωνία

-172.16.0.X: Είναι η εικονική διεύθυνση του Χρήστη, στη θέση του X είναι ένας αριθμός ο οποίος ξεκινάει από το 0(πρώτη εκτέλεση) και από την επόμενη αυξάνεται με κάθε φορά που εκτελείται η προσομοίωση. Μπορεί να βρεθεί με τη χρήση της εντολής ifconfig και το όνομα της διεπαφής του είναι oipX.

Στο Virtual Machine(iPerf3 Client) δίχως κάποια επιπρόσθετη εγκατάσταση λογισμικού, χρησιμοποιείται η παρακάτω εντολή έτσι ώστε να σταλεί το επιθυμητό αρχείο στο server iperf3.

```
iperf3 -p 9999 -t 10 -i 1 -c 172.16.0.X -F testX -b 10M
```

-c (IP):λειτουργία ως client του συγκεκριμένου Virtual Machine και αποστολή στη διεύθυνση IP

-t 10/20(sec):Χρόνος αποστολής πακέτων

-i 1:Εμφάνιση ποσοστού αποστολής και bandwidth ανά 1 δευτερόλεπτο

-testX:Αρχείο 5 ή 10 MB(test5/test10)

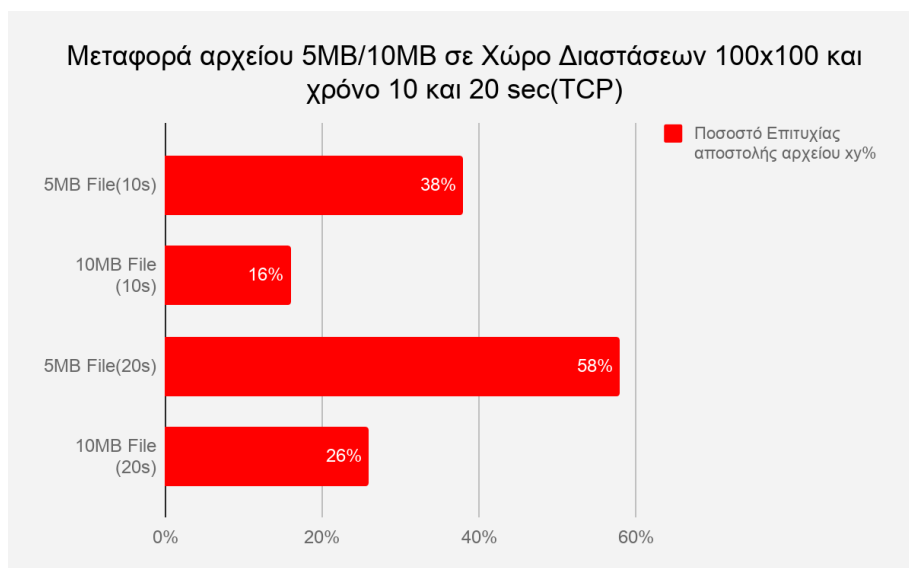
-b 10M:Bandwidth 10MB

-p 9999:Χρήση του Port 9999 για επικοινωνία

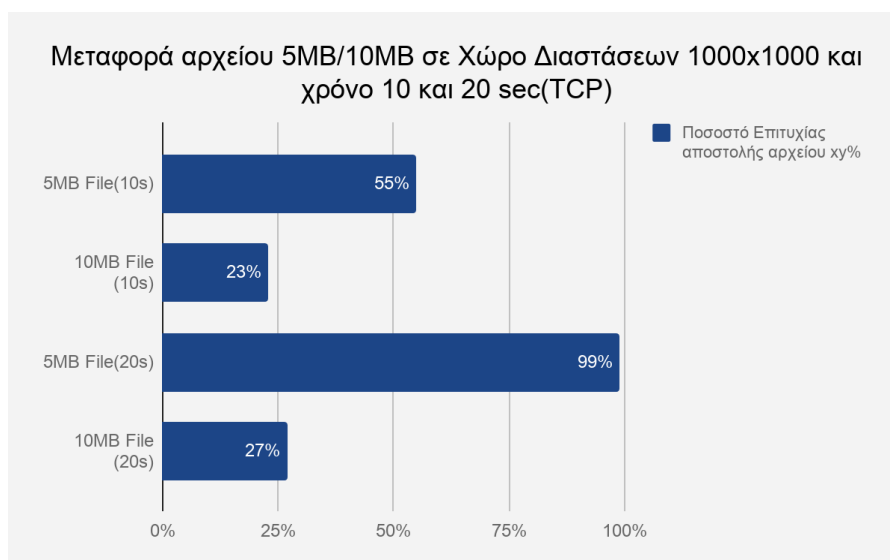
Η απόσταση 100x100 ή 1000x1000 είναι μία μεταβλητή η οποία έχει ενταχθεί στο OSD και εκτελείται μαζί με τη προσομοίωση. Αυτό σημαίνει ότι ο πομπός δέκτης έχουν μία απόσταση 141.2 μέτρα και 1.414,2 μέτρα αντίστοιχα απόσταση γιατί είναι ακίνητα. Τα αποτελέσματα για τη πρώτη περίπτωση είναι τα παρακάτω. Παρατηρείται απώλεια δεδομένων κατά την αποστολή των δεδομένων και εν τέλη δεν ολοκληρώνεται η αποστολή του αρχείου στον απαιτούμενο χρόνο.

Λόγος απώλειας πακέτων προς μεταφορά μπορεί να θεωρηθεί η συμφόρηση του δικτύου από τις υπερ-χρησιμοποιούμενες συσκευές ή προγράμματα. Αυτό σημαίνει ότι μια συσκευή ή πρόγραμμα λειτουργεί για χωρητικότητα για την οποία δεν έχει σχεδιαστεί καταλλήλως. Σε ένα δίκτυο, τα πακέτα μπορεί να φτάνουν ταχύτερα από ό, τι μπορούν να επεξεργαστούν και να αποστέλλονται ξανά. Για να διαχειριστεί αυτό το είδος κατάστασης οι συσκευές ή το πρόγραμμα διαθέτουν τα πακέτα προσωρινά αποθηκευμένα, μέχρι να μπορέσουν να επεξεργαστούν και να σταλούν.

Ωστόσο, στην περίπτωση μιας υπερ χρησιμοποιούμενης συσκευής ή προγράμματος, η προσωρινή μνήμη θα γεμίσει πιθανότατα γρήγορα, με αποτέλεσμα την “πτώση” των πακέτων. Επίσης είναι φυσικό τα πακέτα να χάνονται σε ασύρματα δίκτυα, καθώς τα ασύρματα δίκτυα είναι ανοιχτά σε κάποια απρόβλεπτα στοιχεία, όπως παρεμβολές από άλλα ασύρματα δίκτυα ή απόσταση μεταξύ πομπού δέκτη.



Σχήμα 11. Μετάδοση αρχείου 5MB και 10MB σε 100x100 διαστάσεις (TCP)



Σχήμα 12. Μετάδοση αρχείου 5MB και 10MB σε 1000x1000 διαστάσεις (TCP)

Οι εντολές για κάθε αποστολή Πακέτου από το Client στο Iperf server ενώ η εντολή για την έναρξη του server είναι η ίδια και η απόσταση εξαρτάται από το OSD:

Server(PC1):

```
iperf3 -s -B 172.16.0.X -p 9999
```

1)5MB/10sec:

```
iperf3 -p 9999 -t 10 -i 1 -c 172.16.0.1 -F test5 -b 10M
```

2)10MB/10sec:

```
iperf3 -p 9999 -t 10 -i 1 -c 172.16.0.1 -F test10 -b 10M
```

3)5MB/20sec:

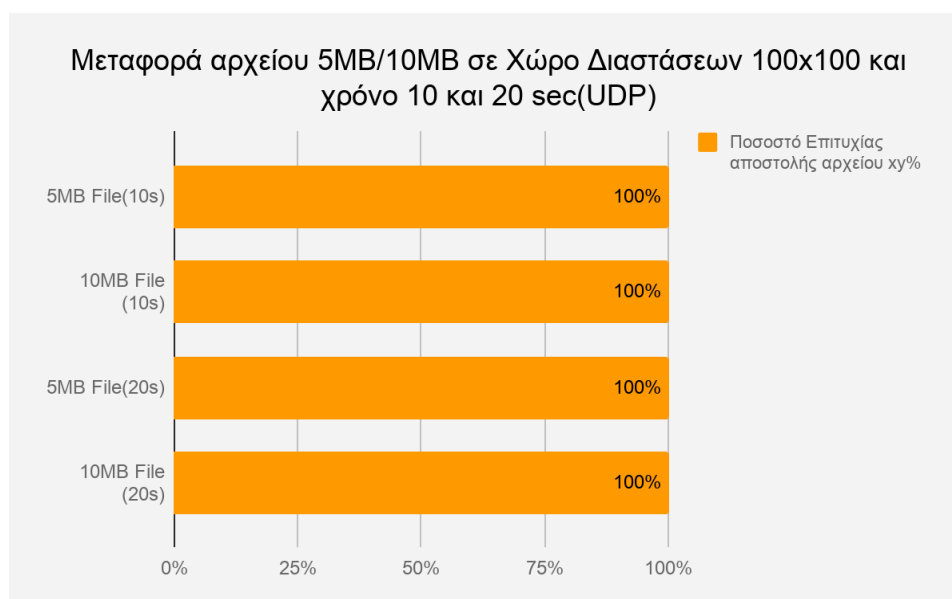
```
iperf3 -p 9999 -t 20 -i 1 -c 172.16.0.2 -F test5 -b 10M
```

4)10MB/20sec:

```
iperf3 -p 9999 -t 20 -i 1 -c 172.16.0.2 -F test10 -b 10M
```

## 4.2 Μεταφορά Αρχείου-Περίπτωση UDP πακέτων

Αντίθετα με τα TCP πακέτα τα UDP είχαν σε κάθε περίπτωση αποστολή 100% του πακέτου και βασικός λόγος είναι ότι δεν υπάρχει υποχρεωτική επαναμετάδοση πακέτων και αγνοούνται οι συνθήκες συμφόρησης του δικτύου.



Σχήμα 13. Μετάδοση αρχείου 5MB και 10MB σε 100x100 διαστάσεις(UDP)

Ομοίως οι εντολές για κάθε αποστολή Πακέτου από το Client στο Iperf server ενώ η εντολή για την έναρξη του server είναι η ίδια και η απόσταση εξαρτάται από το OSD. Σημαντικό είναι να σημειωθεί ότι χρησιμοποιείται και η εντολή -u για την μεταφορά πακέτων τύπου UDP.

Server(PC1):

```
iperf3 -s -B 172.16.0.X -p 9999
```

1)5MB/10sec:

```
iperf3 -p 9999 -t 10 -i 1 -c 172.16.0.3 -F test5 -b 10M -u
```

2)10MB/10sec:

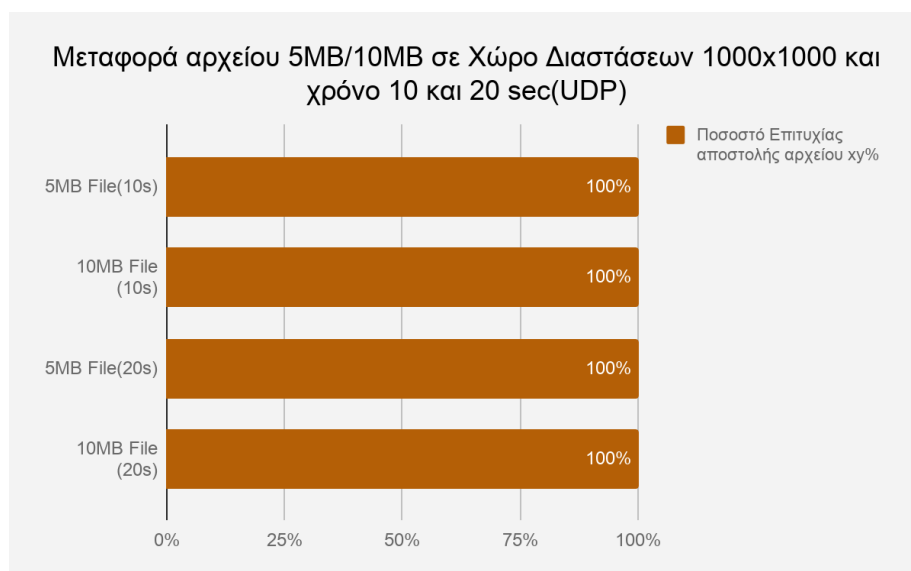
```
iperf3 -p 9999 -t 10 -i 1 -c 172.16.0.3 -F test10 -b 10M -u
```

3)5MB/20sec:

```
iperf3 -p 9999 -t 20 -i 1 -c 172.16.0.4 -F test5 -b 10M -u
```

4)10MB/20sec:

```
iperf3 -p 9999 -t 20 -i 1 -c 172.16.0.4 -F test10 -b 10M -u
```



Σχήμα 14. Μετάδοση αρχείου 5MB και 10MB σε 1000x1000 διαστάσεις(UDP)

### 4.3 Προσομοίωση με τρεις εικονικούς χρήστες

Το ίδιο σενάριο προσομοίωσης εκτελέστηκε και με τρεις συνολικά συνδεδεμένους εικονικούς χρήστες ο ένας δίπλα στον άλλο από προεπιλογή εξαρχής. Φυσικά δεν είναι δυνατό να επιτευχθεί αυτό δίχως την επεξεργασία κάποιων ακόμα παραμέτρων για να επιτευχθεί το επόμενο αυτό βήμα. Αρχικά στο PC1 γίνεται επεξεργασία του αρχείου ue\_eurecom\_test\_sfr.conf το οποίο βρίσκεται στο φάκελο openairinterface5g/openair3/NAS/TOOLS/ :

```
sudo gedit ue_eurecom_test_sfr.conf
```

Αρχείο ue\_eurecom\_test\_sfr.conf:

```
# List of known PLMNS
PLMN: {
  PLMN0: {
    FULLNAME="Test network";
    SHORTNAME="OAI4G";
    MNC="01";
    MCC="001";
  };
  PLMN1: {
    FULLNAME="SFR France";
    SHORTNAME="SFR";
    MNC="10";
    MCC="208";
  };
  PLMN2: {
    FULLNAME="SFR France";
    SHORTNAME="SFR";
```



```
MNC="11";
MCC="208";
};
PLMN3: {
    FULLNAME="SFR France";
    SHORTNAME="SFR";
    MNC="13";
    MCC="208";
};
PLMN4: {
    FULLNAME="OAI LTEBOX";
    SHORTNAME="OAI ALU";
    MNC="93";
    MCC="208";
};
PLMN5: {
    FULLNAME="T-Mobile USA";
    SHORTNAME="T-Mobile";
    MNC="280";
    MCC="310";
};
PLMN6: {
    FULLNAME="FICTITIOUS USA";
    SHORTNAME="FICTITIO";
    MNC="028";
    MCC="310";
};
```

```
PLMN7: {  
    FULLNAME="Vodafone Italia";  
    SHORTNAME="VODAFONE";  
    MNC="10";  
    MCC="222";  
};  
PLMN8: {  
    FULLNAME="Vodafone Spain";  
    SHORTNAME="VODAFONE";  
    MNC="01";  
    MCC="214";  
};  
PLMN9: {  
    FULLNAME="Vodafone Spain";  
    SHORTNAME="VODAFONE";  
    MNC="06";  
    MCC="214";  
};  
PLMN10: {  
    FULLNAME="Vodafone Germ";  
    SHORTNAME="VODAFONE";  
    MNC="02";  
    MCC="262";  
};  
PLMN11: {  
    FULLNAME="Vodafone Germ";  
    SHORTNAME="VODAFONE";
```

```
MNC="04";
MCC="262";
});
UE0:
{
  USER: {
    IMEI="35525105157726";
    MANUFACTURER="EURECOM";
    MODEL="LTE Android PC";
    PIN="0000";
  };

  SIM: {
    MSIN="0100001111";
    USIM_API_K="8baf473f2f8fd09487cccbd7097c6862";
    OPC="e734f8734007d6c5ce7a0508809e7e9c";
    MSISDN="33638020001";
  };

  # Home PLMN Selector with Access Technology
  HPLMN= "20893";
  # User controlled PLMN Selector with Access Technology
  UCPLMN_LIST = ();
  # Operator PLMN List
  OPLMN_LIST = ("00101", "20810", "20811", "20813", "20893", "310280", "310028");
  # Operator controlled PLMN Selector with Access Technology
  OCPLMN_LIST = ("22210", "21401", "21406", "26202", "26204");
```

```
# Forbidden plmns
FPLMN_LIST = ();

# List of Equivalent HPLMNs
#TODO: UE does not connect if set, to be fixed in the UE
#  EHPLMN_LIST= ("20811", "20813");
  EHPLMN_LIST= ();
};
UE1:
{ USER: {
    IMEI="35508908384580";
    MANUFACTURER="EURECOM";
    MODEL="LTE Android PC";
    PIN="0000";
};
SIM: {
    MSIN="0100001112";
    USIM_API_K="8baf473f2f8fd09487ccbd7097c6862";
    OPC="e734f8734007d6c5ce7a0508809e7e9c";
    MSISDN="33638020002";
};

# Home PLMN Selector with Access Technology
HPLMN= "20893";
# User controlled PLMN Selector with Access Technology
UCPLMN_LIST = ();
# Operator PLMN List
OPLMN_LIST = ("00101", "20810", "20811", "20813", "20893", "310280", "310028");
```

```
# Operator controlled PLMN Selector with Access Technology
OCPLMN_LIST = ("22210", "21401", "21406", "26202", "26204");

# Forbidden plmns
FPLMN_LIST = ();

# List of Equivalent HPLMNs
#TODO: UE does not connect if set, to be fixed in the UE
#  EHPLMN_LIST= ("20811", "20813");
  EHPLMN_LIST= ();
};
UE2:
{USER: {
  IMEI="356113022094149";
  MANUFACTURER="EURECOM";
  MODEL="LTE Android PC";
  PIN="0000"  };
SIM: {
  MSIN="0100001113";
  USIM_API_K="8baf473f2f8fd09487cccbd7097c6862";
  OPC="e734f8734007d6c5ce7a0508809e7e9c";
  MSISDN="33611123458";
};
# Home PLMN Selector with Access Technology
HPLMN= "20893";
# User controlled PLMN Selector with Access Technology
UCPLMN_LIST = ();
# Operator PLMN List
OPLMN_LIST = ("00101", "20810", "20811", "20813", "20893", "310280", "310028");
```

```
# Operator controlled PLMN Selector with Access Technology
OCPLMN_LIST = ("22210", "21401", "21406", "26202", "26204");
# Forbidden plmns
FPLMN_LIST = ();
# List of Equivalent HPLMNs
#TODO: UE does not connect if set, to be fixed in the UE
#  EHPLMN_LIST= ("20811", "20813");
  EHPLMN_LIST= ();
};
```

Αμέσως μετά τρέχουμε την παρακάτω εντολή στον υποφάκελο `openairinterface5g/targets/bin` για να γίνει ανανέωση της λίστας με τους χρήστες:

```
sudo -E ./conf2uedata -c
```

Και στη συνέχεια εκτελείται η προσομοίωση όπως έχει οριστεί πιο πριν και παρατηρείται ότι στο terminal του MME θα έχουμε συνδεδεμένες 3 χρήστες και αντίστοιχα ο καθένας μία μοναδική εικονική διεπαφή, το οποίο μπορεί να ελεγχθεί με τη χρήση της εντολής `ifconfig` στο PC1. Τα αποτελέσματα της προσομοίωσης αναφορικά με τα ποσοστά επιτυχούς μετάδοσης πακέτων είναι ίδια με περιθώριο διαφοράς 1-2% σε σχέση με την περίπτωση ενός τελικού χρήστη. Δηλαδή στα TCP πακέτα είχαμε απώλειες ενώ στα UDP όχι.

#### 4.4 Μελλοντική εξέλιξη

Σε αυτό το στάδιο έχει γίνει κατανοητό το πως έχει στηθεί το δίκτυο και όλο το κομμάτι εγκατάστασης και ρύθμισης κάθε τμήματος του OpenAirInterface5g για την επιτυχής εκτέλεση της προσομοίωσης του πειράματος αυτού.

Επόμενο βήμα είναι να χρησιμοποιηθεί πραγματικός εξοπλισμός για να γίνει το πείραμα σε αληθινές συνθήκες και πιθανώς να παρθούν πιο ακριβής αποτελέσματα. Τέτοιος εξοπλισμός είναι το USRP B210 για παράδειγμα και παρέχει μια πλατφόρμα με κάλυψη συχνότητας από 70 MHz έως 6 GHz. Είναι σχεδιασμένο για πειράματα χαμηλού κόστους, η υποστήριξη του λογισμικού USRP Hardware Driver™ από κάθε λειτουργικό σύστημα επιτρέπει να ξεκινήσουν άμεσα οι πειραματικές διαδικασίες. Τέλος, θα μπορούσαν να χρησιμοποιηθούν και κενές κάρτες SIM έτσι ώστε μαζί με ένα Card Reader και ένα πρόγραμμα επεξεργασίας των πληροφοριών τους θα είναι δυνατό να προσομοιωθούν χρήστες για το δίκτυο.

#### 4.5 Τελικά Συμπεράσματα

Εν κατακλείδι, θα ήταν χρήσιμο να τονιστεί το γεγονός ότι ο χρόνος προσομοίωσης παίζει σημαντικό ρόλο όσον αφορά το ποσοστό αποστολής του αρχείου όπως φαίνεται κυρίως στα TCP πακέτα όπου και παρατηρήθηκαν κυρίως απώλειες. Η καλύτερη επίδοση TCP παρατηρήθηκε στα 1000x1000 μέτρα, η οποία είναι για 20 δευτερόλεπτα για το αρχείο των 5MB όπου είχαμε την ολοκληρωτική του αποστολή του. Αντίθετα, στα UDP πακέτα είχαμε την πλήρη αποστολή τους κάθε φορά δίχως απώλειες. Λόγος της απώλειας στα TCP πακέτα θα μπορούσε να θεωρηθεί η σύγχυση στο δίκτυο μεταφοράς των πακέτων, κάποιο πρόβλημα με το ίδιο το πρόγραμμα, πιο συγκεκριμένα με την έκδοση που ήταν διαθέσιμη την χρονική περίοδο που εκτελέστηκε το πείραμα, σχετικά με το πόσα πακέτα υποστηρίζονται και τι τύπου, ή κάποιος άλλος που αναφέρθηκε στην Ενότητα 4.1 .





## Βιβλιογραφία

---

1. Trois, C., Del Fabro, M. D., de Bona, L. C., & Martinello, M. (2016). A survey on SDN programming languages: Toward a taxonomy. *IEEE Communications Surveys & Tutorials*, 18(4), 2687-2712.
2. Investigation, Analysis and Implementation of Open Source Mobile Communication Software , Suresh Paudel
3. Andreas Gladisch and Wolfgang Kellerer Software defined networking and network function virtualization
4. OpenAirInterface Large-Scale Wireless Emulation Platform and Methodology, Ben Romdhanne Bilel; Nikaein Navid; Knopp Raymond; Bonnet Christian
5. <https://gitlab.eurecom.fr/oai/openairinterface5g/wikis/OpenAirKernelMainSetup#ubuntu-14043-ltslinux-kernel-version-319-use-ubuntu-1404-for-master-branch>
6. <https://gitlab.eurecom.fr/oai/openairinterface5g/wikis/HowtorunoaisimwithvirtualMMEonnsamemachine>
7. <https://gitlab.eurecom.fr/oai/openairinterface5g/wikis/HowToConnectOAIENBWithOAIUEWithS1Interface>
8. <https://gitlab.eurecom.fr/oai/openairinterface5g/wikis/OpenAirLTEEmulation>
9. Adachi, F, Wireless Past and Future--Evolving Mobile Communications Systems--. *IEICE transactions on fundamentals of electronics, communications and computer sciences*, 84(1), 55-60.
10. Callegati, F., Cerroni, W., Contoli, C., Cardone, R., Nocentini, M., & Manzalini, A. (2016). SDN for dynamic NFV deployment. *IEEE Communications Magazine*, 54(10), 89-95.
11. Chávez-Santiago, Raúl, et al. "5G: The convergence of wireless communications." *Wireless Personal Communications* 83.3 (2015): 1617-1642. Overview of LTE-A technology published by IEEE
12. Lee, H., Vahid, S., & Moessner, K., "A survey of radio resource management for spectrum aggregation in LTE-advanced", *IEEE Communications Surveys & Tutorials*, 16(2), 745-760.
13. Computer Networks UDP and TCP Saad Mneimneh Computer Science Hunter College of CUNY
14. Agiwal, M., Roy, A., & Saxena, N. (2016). Next generation 5G wireless networks: A comprehensive survey. *IEEE Communications Surveys & Tutorials*, 18(3), 1617-1655.

15. Nguyen, V. G., Brunstrom, A., Grinnemo, K. J., & Taheri, J. (2017). SDN/NFV-based mobile packet core network architectures: A survey. *IEEE Communications Surveys & Tutorials*, 19(3), 1567-1602.