ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΑΤΡΩΝ

ΠΟΛΥΤΕΧΝΙΚΗ ΣΧΟΛΗ

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΗΛΕΚΤΡΟΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ & ΠΛΗΡΟΦΟΡΙΚΗΣ

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

« CORD σε Δίκτυα Κινητών Επικοινωνιών »

Συγγραφέας
Μιχαλόπουλος Μάριος, ΑΜ: 4638

Υπεύθυνος Καθηγητής
Χρήστος Ι. Μπούρας, Καθηγητής

Επιβλέποντες
Κόλλια Αναστασία, Παπαζώης Ανδρέας

Πάτρα, Ιούλιος 2019

Πανεπιστήμιο Πατρών, Τμήμα Μηχανικών Ηλεκτρονικών Υπολογιστών και Πληροφορικής
Μιχαλόπουλος Μάριος

Πρόλογος

Αρχικά θέλω να ευχαριστήσω τον υπεύθυνο καθηγητή της παρούσας διπλωματικής εργασίας και καθηγητή του τμήματος Μηχανικών Ηλεκτρονικών Υπολογιστών και Πληροφορικής της Πολυτεχνικής σχολής του Πανεπιστημίου Πατρών, κ. Χρήστο Ι. Μπούρα, για την ευκαιρία που μου έδωσε και την εμπιστοσύνη που μου έδειξε με την ανάθεση της μελέτης αυτής, καθώς και για την πολύτιμη καθοδήγηση που μου προσέφερε.

Επίσης, θέλω να ευχαριστήσω θερμά τους επιβλέποντες αυτής της προσπάθειάς μου, για την διεκπεραίωση αυτής της διπλωματικής εργασίας, Αναστασία Κόλλια και Ανδρέα Παπαζώη, οι οποίοι μου παρείχαν συνεχή βοήθεια και καθοδήγηση, κάτω από άριστη συνεργασία.

Κλείνοντας, θα ήθελα να ευχαριστήσω ολόψυχα την οικογένειά μου, τους φίλους μου και όλους όσους στάθηκαν δίπλα μου και με στήριξαν στην φοιτητική μου πορεία.

Μάριος Μιχαλόπουλος

Πάτρα, Ιούλιος 2019

Περίληψη

Στόχος της συγκεκριμένης διπλωματικής εργασίας είναι να περιγραφεί και να αναδειχθεί η τεχνολογία του Central Office Re-architected as a Datacenter (CORD), καθώς και να αποφανθούν τα σημαντικά οφέλη της εφαρμογής της. Το CORD αποτελεί τη νέα προσέγγιση για το Central Office (CO), το οποίο αποτελεί το βασικό τμήμα των τηλεπικοινωνιακών παρόχων για την επικοινωνία με τα εξωτερικά δίκτυα και τους συνδρομητές, δηλαδή το τμήμα που παρέχει τις δικτυακές υπηρεσίες στους τελικούς χρήστες.

Τα παραδοσιακά Central Offices είναι δομημένα με πολλαπλά υλικά και λογισμικά στοιχεία διαφορετικών εταιρειών και τεχνολογιών, πράγμα που δυσκολεύει τη συνολική διαχείριση και επεκτασιμότητα του δικτύου. Καθώς όμως τα τελευταία χρόνια οι απαιτήσεις για δικτυακές υπηρεσίες αυξάνονται ραγδαία, καθίσταται επιτακτική η ανάγκη για την αντιμετώπιση αυτών των προβλημάτων και περιορισμών. Το CORD έρχεται για να δώσει τη λύση στο πρόβλημα αποδίδοντας ένα αποδοτικό, ευέλικτο και κεντρικά διαχειρίσιμο μοντέλο βασισμένο σε συσκευές κοινού τύπου, συνδυάζοντας τις Cloud, SDN και NFV σύγχρονες τεχνολογίες.

Θα αναλυθεί το CORD τόσο ως ένα γενικό μοντέλο, μέσα από τις τεχνολογίες και τα δομικά του στοιχεία, καθώς επίσης θα παρουσιαστεί και η εφαρμογή του στην κινητή δικτύωση, που αποτελεί μία καλή βάση για την επερχόμενη τεχνολογία πέμπτης γενιάς (5G).

Abstract

This diploma thesis aims to describe and feature the Central Office Re-architected as a Datacenter (CORD) project, as well as to present the significant benefits of its deployment. CORD is a new approach for implementing Central Office (CO). A Central Office is the main part at a telecommunications service provider's premises, which is responsible for the communication with the external networks and providing subscribers with network services.

The conventional Central Offices are built up on multiple hardware and software components of different brands and technologies, making the network hard in management and scalability. Additionally, the recent years, the requirements for network services are rapidly increasing, compelling us to find a way to overcome those restrictions. The CORD technology which is presented here, constitutes an essential solution, rendering an efficient, agile and central controlled model, built on commodity devices and based on Cloud, SDN and NFV innovative concepts.

In this work, CORD is elaborated as a general model through its technologies and components. Furthermore this thesis presents the mobile implementation of this technology, called M-CORD, and how it paves the way for deploying the upcoming 5G mobile technology.

Περιεχόμενα

Κατάλογος σχημάτων

Ακρωνύμια

AMQP - Advanced Message Queuing Protocol
API - Application Programming Interface
AS - α)Access Stratum, β)Autonomous System
BBU - Bare Band Unit
BGP - Border Gateway Protocol
BNG - Broadband Network Gateway
BSC - Base Station Controller
BSS - Business Support System
BTS - Base Transceiver Station
CAPEX - Captal Expenditure
CDN - Content Delivery Network
CLI - Command Line Interface
CO - Central Office
CORD - Central Office Re-architected as a Datacenter
CPE - Customer Premises Equipment
DCTCP - Data Center Transmission Control Protocol
DRS - Dynamic Resource Scheduling
DU - Digital Unit
ECMP - Equal-Cost Multi-Path routing
EDGE - Enhanced Data rates for GSM Evolution
EPC - Evolved Packet Core
EPON - Ethernet Passive Optical Network
EPS - Evolved Packet System
E2E - End-to-End
E-UTRAN - Evolved Universal Terrestrial Radio Access Network
FDMA - Frequency Division Multiple Access
GPON - Gigabit-capable Passive Optical Networks
GPRS - General Packet Radio Service
GTP - GPRS Tunneling Protocol
HSS - Home Subscriber Server
HSPA - High Speed Packet Access
IaaS - Infrastructure as a Service
IGMP - Internet Group Management Protocol
IoT - Internet of Things
LTE - Long Term Evolution
LXC - Linux Container
MAC - Medium Access Control
MIMO - Multiple In – Multiple Out
MBB – mobile broadband
MME - Mobility Management Entity
MPLS - Multiprotocol Label Switching
NAS - Non Access Stratum
NAT - Name Address Translation
NBI - Northbound Interface
NCP - Network Control Point
NFV - Network Functions Virtualisation
NFVI - Network Functions Virtualisation Infrastructure
NFV MANO - Network Functions Virtualisation Management and Network Orchestration

NG-PON - Next-Generation Passive Optical Network
NIC - Network Interface Card
ODN - Optical Distribution Network
OLT - Optical Line Terminal
ONF - Open Networking Foundation
ONOS - Open Network Operating System
ONT - Optical Network Terminal
ONU - Optical Network Unit
OPEX - Operating Expenditure
OSFP - Open Shortest Path First (protocol)
OSS - Operations Support System
OvS - Open Virtual Switch
PaaS - Platform as a Service
PDCP - Packet Data Convergence Protocol
PGW – Public Data Network Gateway
PIM-SSM - Protocol Independent Multicast Source-Specific Multicast
PoC - Proof of Concept
PoD - Point of Delivery
PON - passive optical network
QAM - Quadrature Amplitude Modulation
QoE - Quality of Experience
QoS - Quality of Service
RAN - Radio Access Network
REST - Representational State Transfer
RG - Residential Gateway
RLC - Radio Link Control
RNC - Radio Network Controller
RRC - Radio Resources Control
RRH - Remote Radio Head
RRU - Remote Radio Unit
RSTP - Rapid Spanning Tree Protocol
RU - Radio Unit
SaaS - Software as a Service
SBI - Southbound Interface
SDN - Software Defined Networking
SG - Subscriber Gateway
SGW - Serving Gateway
SLA - Service Level Agreement
TDMA - Time Division Multiple Access
ToR - Top of Rack (Switch)
UE - User Equipment
UMTS - Universal Mobile Telecommunication System
VXLAN - Virtual Extensible LAN
VM - Virtual Machine
VN - Virtual Network
VNF - Virtual Network Function
VOLTHA - Virtual OLT Hardware Abstraction
VPN - Virtual Private Network
VTN - Virtual Tenant Network
WAN - Wide Area Network
XaaS - Everything as a Service

1. Εισαγωγή

Τα τελευταία χρόνια οι χειριστές δικτύων (network operators) έρχονται αντιμέτωποι με τις σημαντικές προκλήσεις, για διαρκώς αυξανόμενες, με εκθετικό ρυθμό, απαιτήσεις σε εύρος ζώνης (bandwidth) καθώς και την κάλυψη των διαρκώς αυξανόμενων υπηρεσιών (π.χ. κίνηση πολυμεσικών δεδομένων). Ακόμα η ανάπτυξη και η εγκατάσταση νέων τεχνολογιών υλικού και υλικού ειδικού σκοπού, όπως ταχύτερων οπτικών ινών, δρομολογητών (routers) και switches, καθώς και ισχυρότερων εξυπηρετητών (servers) συνήθως απαιτεί μεγάλο χρόνο και κόστος, καθιστώντας έτσι πιο δύσκολη την αντιμετώπιση του προβλήματος.

Έχοντας αυτά κατά νου, οι διαχειριστές δικτύων, για να αντιμετωπίσουν αυτές τις προκλήσεις, στρέφονται σε λύσεις σχετικά με την αποδοτικότερη διαχείριση των ήδη υπαρχόντων και κοινών δικτυακών πόρων. Τέτοιες λύσεις βασίζονται σε τεχνικές όπως ή αποδοτική επέκταση απλών δικτυακών πόρων και μπλοκ αυτών καθώς και στην εφαρμογή και επέκταση υπηρεσιών (services) με ελαστικό τρόπο (ευκολία στην κλιμάκωση). Αυτές οι τεχνικές όπως θα δούμε προσφέρονται από τους σύγχρονους εμπορικούς παρόχους "νέφους" (cloud providers). Εκμεταλλευόμενοι των δυνατοτήτων που προσφέρει το cloud, οι πάροχοι δικτύων στρέφονται στην οργάνωση του Central Office (CO).

Το Central Office είναι ο όρος που αναφέρεται στο τμήμα των τηλεπικοινωνιακών παρόχων αλλά και των μεγάλων εταιριών και οργανισμών, που λειτουργεί ως διεπαφή με τη διαδικτυακή επικοινωνία στον "έξω κόσμο". Πιο συγκεκριμένα είναι το σημείο στο οποίο γίνεται η τελική διαχείριση και ανάθεση των τηλεπικοινωνιακών συνδέσεων και ροών. Στην ουσία δηλαδή πρόκειται για το σύνολο των πόρων διαχείρισης της τηλεπικοινωνιακής κίνησης, όπως δρομολογητές (routers) και διακόπτες (switches), από τους παρόχους διαδικτύου-τηλεπικοινωνιών καθώς και άλλων οργανισμών που παρέχουν διαδικτυακές υπηρεσίες προς τους πολίτες και αντίστροφα. Όμως στο πέρασμα των χρόνων στις περισσότερες περιπτώσεις, τα συγκεκριμένα τμήματα αναπτύχθηκαν με χρήση υλικού και πόρων διαφορετικών τεχνολογιών και επωνυμιών με "άναρχο" τρόπο και έλλειψη αρμονίας, καθιστώντας τα δύσκολα στην διαχείριση, στην συντήρηση καθώς και στην αναβάθμισή τους. Ακόμα αυτή η χαοτική δυσαρμονία είναι ένας πολύ περιοριστικός παράγοντας για τη βέλτιστη χρήση των πόρων, αφήνοντας έτσι ένα μεγάλο μέρος της υπολογιστικής ισχύος αυτών των πόρων ανεκμετάλλευτο. Και για να μιλήσουμε με τεχνική ορολογία, αυξάνεται το OPEX (operating expenditure) και το CAPEX (capital expenditure), δηλαδή το κόστος για την παροχή των υπηρεσιών και το κόστος για τη συντήρηση-διαχείριση των πόρων αντίστοιχα.

Λαμβάνοντας υπόψιν όλα τα παραπάνω, γίναται αντιληπτή η σαφής και επιτακτική ανάγκη για τη δημιουργία ενός προτύπου το οποίο θα "ομογενοποιεί" και θα βελτιώνει μέσω συγκεκριμένων προτύπων, τεχνολογιών και αρχιτεκτονικής το σύνολο της δικτυακής διαχείρισης και απόδοσης υπηρεσιών από τους αναμενόμενους παρόχους και οργανισμούς υπηρεσιών internet. Μία από τις επικρατέστερες λύσεις που βρίσκει ολοένα και περισσότερο εφαρμογή στις τηλεπικοινωνιακές εταιρείες είναι το **CORD (Central Office Re-architected as a Datacenter)**. Το CORD είναι ένα πρότζεκτ ανοιχτού κώδικα (open source) του ONF (Open Network Function), που αποτελεί μία κοινοπραξία για την μεταφορά των υπαρχόντων δικτύων σε νέας γενιάς που θα υπόκεινται σε σαφή πρότυπα. Το CORD έχει επίσης αποδεχθεί ως προτζεκτ και από το Linux Foundation. Όπως δηλώνει το όνομά του το CORD αναδιοργανώνει το Central Office ως data-center (κέντρο δεδομένων). Στην ουσία δηλαδή το CORD αποτελεί μια αρχιτεκτονική για τα τηλεπικοινωνιακά Central Offices, που συνδυάζει το Software Defined Networking (SDN), το Network Functions Virtualization (NFV) και τις ελαστικές υπηρεσίες που παρέχει το Cloud

networking, τα οποία όλα τρέχουν πάνω σε commodity hardware (δηλαδή σε κοινό, εμπορικό υλικό γενικού σκοπού), με σκοπό την εγκαθίδρυση αποδοτικών, χαμηλού κόστους και ευέλικτων δικτύων με ουσιαστικά μειωμένο CAPEX/OPEX και τη δυνατότητα γρήγορης δημιουργίας και εγκατάστασης υπηρεσιών (δικτυακών λειτουργιών με τη μορφή εκτελέσιμων προγραμματιστικών εφαρμογών).

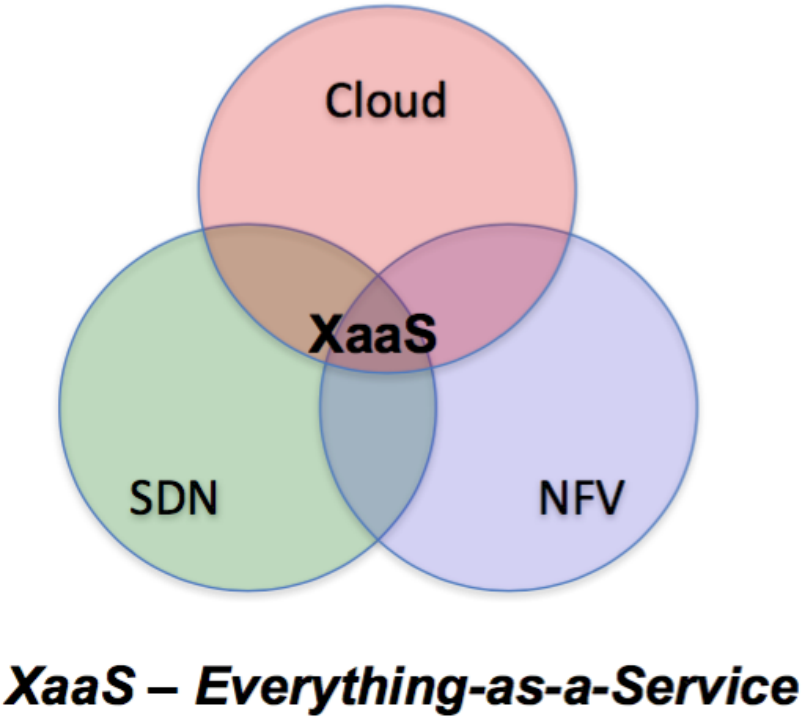
Αρχικά παρουσιάζονται και αναλύονται, σε γενικό βαθμό, οι τρεις συσχετιζόμενες, αλλά διακριτές παρά ταύτα, τεχνολογίες οι οποίες αποτελούν τα βασικά συστατικά του CORD που αναφέρθηκαν παραπάνω (SDN, NFV, Cloud Networking). Έτσι, όπως είναι λογικό, θα είμαστε σε θέση, να κατανοήσουμε πως εφαρμόζεται το CORD, αλλά και πως ενδέχεται να αναπτυχθεί και να εξελιχθεί στο άμεσο μέλλον.

Ύστερα θα γίνει ανάλυση του CORD παρουσιάζοντας τα δομικά στοιχεία του, την διασύνδεσή τους και τις βασικές εφαρμογές και υπηρεσίες του. Με αυτόν τον τρόπο θα έχουμε μία σαφή εικόνα και θα γίνει κατανοητή σε βάθος η έννοια του CORD, η συνολική λειτουργία του και οι εφαρμογές και υπηρεσίες του.

Τέλος θα γίνει αναφορά στο κομμάτι του CORD που αφορά την κινητή δικτύωση, δηλαδή στο M-CORD. Εκεί θα τεθεί μία σύντομη επισκόπηση των προηγούμενων τεχνολογιών κινητής δικτύωσης, θα παρουσιαστεί το LTE, δηλαδή η τρέχουσα τεχνολογία κινητής δικτύωσης. Ύστερα θα αναφερθούν κάποια βασικά χαρακτηριστικά του 5G, μιας και αποτελεί την επόμενη μεγάλη τεχνολογική εξέλιξη στον τομέα της κινητής δικτύωσης, αλλά και επειδή η λειτουργία του θα σχετίζεται με το M-CORD. Τέλος θα παρουσιαστεί η ιδέα και η αρχιτεκτονική του M-CORD, καθώς και κάποιες αναδεικνυόμενες εφαρμογές του.

2. Βασικές τεχνολογίες στις οποίες βασίζεται το CORD

Σε αυτό το κεφάλαιο θα παρουσιαστούν οι βασικές τεχνολογίες σύγχρονης δικτύωσης, Cloud, SDN και VNF, στις οποίες βασίζεται το CORD για να επιτελέσει το έργο του. Στην ουσία όπως θα γίνει αντιληπτό και στα επόμενα κεφάλαια, το CORD είναι ο συνδυασμός αυτών των τριών τεχνολογιών για να επιφέρει την έννοια του XaaS (Everything-as-a-Service – τα πάντα ως υπηρεσία).



Σχημα 1: Το XaaS ως συνδυασμός των Cloud, SDN, NFV τεχνολογιών [73]

2.1 Cloud Networking

Το Cloud αποτελεί τη βάση για την ανάπτυξη των υπόλοιπων τεχνολογιών που θα μελετήσουμε. Καταδεικνύει ένα σύνολο από καινοτόμες τεχνολογίες για την ανάπτυξη επεκτάσιμων υπηρεσιών (scalable services), την ανάδειξη λύσεων βασισμένων σε λογισμικό (software based solutions), αρχιτεκτονική μικροϋπηρεσιών (micro-service architecture), εικονικοποιημένες κοινές πλατφόρμες εμπορίου (virtualized commodity platforms), ελαστική επεκτασιμότητα και σύνθεση υπηρεσιών (service composition), επιτρέποντας έτσι στους διαχειριστές δικτύων ευελιξία και ταχεία εξέλιξη.

Το Cloud βασίζεται άρρηκτα με τα data-centers στα οποία υλοποιούνται οι τεχνολογίες, οι λειτουργίες και η διαχείριση του Cloud. Πρόκειται στην ουσία για τους παρόχους του Cloud. Γι' αυτόν τον λόγο, στη συνέχεια θα περιγραφεί η δομή, οι λειτουργίες και η διαχείριση που εφαρμόζουν τα σύγχρονα data-centers.

2.1.1 Υλική υποδομή

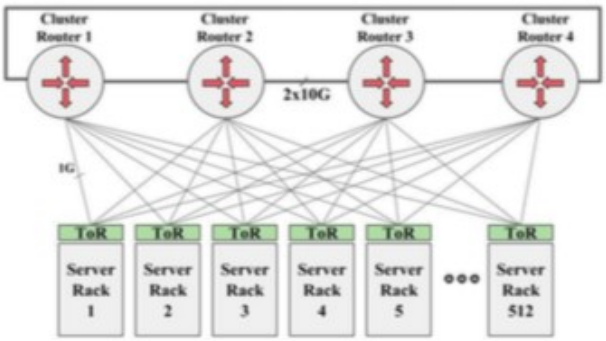
Αρχικά θα δούμε ποια οργάνωση - δομή του υλικού (physical structure) ακολουθείται και τη βασική αρχιτεκτονική που μεγιστοποιεί την απόδοση, την επεκτασιμότητα αλλά και την αξιοπιστία στα data-centers.

Τα data-centers είναι εγκαταστάσεις οι οποίες δύναται να στεγάζουν έναν τεράστιο αριθμό μηχανών - εξυπηρετητών (servers) οι οποίοι είναι οργανωμένοι σε racks, δηλαδή σε στήλες τοποθετημένοι ο ένας πάνω στον άλλον. Κάθε rack συνήθως αποτελείται από μερικές δεκάδες servers οι οποίοι επικοινωνούν άμεσα με ένα απλού τύπου switch που βρίσκεται στην κορυφή του rack και γι' αυτό ονομάζονται ToR switches (Top of Rack). Ένα data-center μπορεί να περιέχει μερικές εκατοντάδες racks.

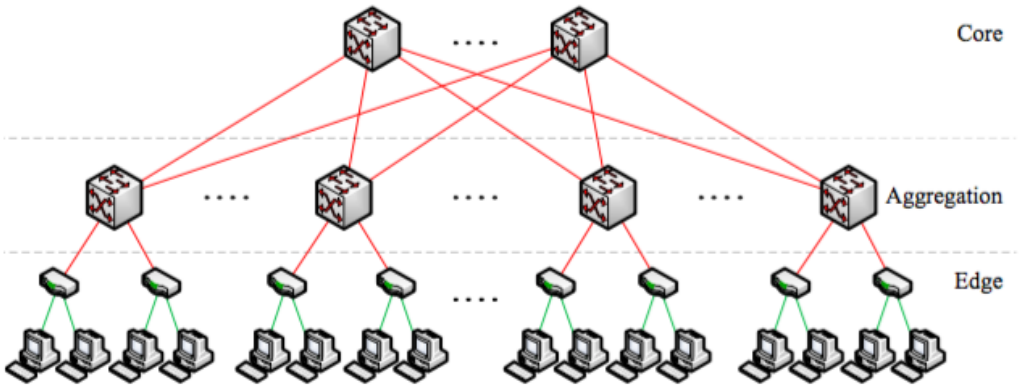
Έτσι γίνεται άμεσα αντιληπτό ότι θα πρέπει να υπάρχει μία σωστή δομή που να επιτρέπει την κυκλοφορία των δεδομένων μεταξύ των servers και κατ' επέκτασιν των racks με αποδοτικό τρόπο. Ακόμα υπάρχει η μεγάλη πρόκληση της ραγδαίας αύξησης της κίνησης που απαιτείται μέσα στα data-centers, πολλαπλάσια της αντίστοιχης του Internet ανοιχτής περιοχής (wide area Internet), δηλαδή της κίνησης από τους τερματικούς πελάτες μέχρι τα data-centers και αντίστροφα. Αυτό οφείλεται σε μεγάλο βαθμό στο φαινόμενο του scatter-gather (διάχυση-περισυλλογή) των web αιτήσεων, δηλαδή τεμαχισμός και αντιγραφή των αιτήσεων για διαμοιρασμό σε διάφορους servers μέσα στο κέντρο δεδομένων (μηχανισμοί map reducing όπως το Hadoop), για την πιο γρήγορη και αποδοτική εξυπηρέτηση (διαμοιρασμός φόρτου εργασίας και αντιμετώπιση σφαλμάτων). Μάλιστα μεγάλοι οργανισμοί όπως το Facebook αναφέρουν ότι η εσωτερική κίνηση διπλασιάζεται σε διάστημα λίγο μικρότερο του ενός έτους. Συνεπώς θα πρέπει η οργάνωση του υλικού των data-centers να παρουσιάζει ανεκτικότητα σε αυξανόμενη κίνηση (μεγάλο bandwidth), ελαστικότητα για αποφυγή προβλημάτων λόγω συγκρούσεων ή κάποιου προβλήματος υλικού και επίσης επεκτασιμότητα. Επιπροσθέτως ένας σημαντικός παράγοντας του σχεδιασμού της δομής αποτελεί φυσικά το κόστος του υλικού που επιλέγεται να χρησιμοποιηθεί.

Μέχρι πρόσφατα τα παραδοσιακά data-centers, η συνήθης τακτική που ακολουθούσαν ήταν τα racks να συνδέονται μεταξύ τους μέσω ενός επιπέδου (ή και περισσότερων δενδρικής μορφής - leaf spine fabric) ειδικών, μεγάλου τύπου switches (Big Switches) όπως φαίνεται στην παρακάτω εικόνα (θεωρήστε τα Cluster Routers ως μεγάλα switches πολλών θυρών). Όμως όπως γίνεται αντιληπτό, αυτή η οργάνωση δημιουργεί κάποια προβλήματα και περιορισμούς.

Αρχικά η επιλογή διάφορων τύπων ειδικής μορφής switches δημιουργεί προβλήματα ασυνέπειας τεχνολογίας και δυσκολεύει το έργο της διαχείρισης και του προγραμματισμού τους, πράγματα που όπως θα δούμε είναι απαραίτητα και για τη χρήση του SDN. Επίσης σε περίπτωση βλαβών και σφαλμάτων δημιουργούνται σοβαρά προβλήματα στην κυκλοφορία καθώς χάνονται ολοκληρωτικά οι εμπλεκόμενοι δίαυλοι επικοινωνίας. Ακόμα σε περίπτωση επεκτάσεων γίνεται ανάγκη αγοράς επιπλέον μεγάλων switches, τα οποία στοιχίζουν πολλές φορές περισσότερο από τα κοινά, πράγμα που επιφέρει σοβαρό κόστος.



Σχήμα 2: Παραδοσιακή spine-leaf δομή ενός επιπέδου [66]

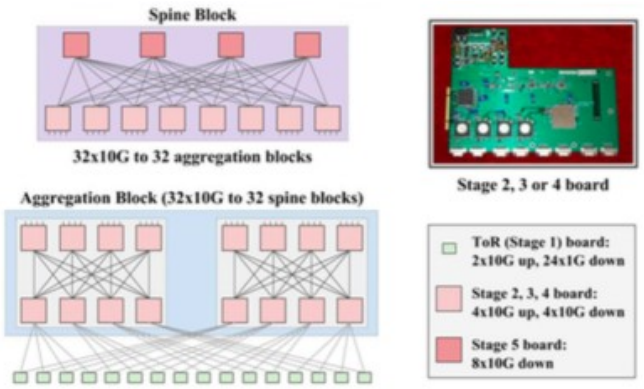


Σχήμα 3: spine-leaf δομή δόμή με δύο επίπεδα (Core/Aggregation) υλοποιημένα με μεγάλες ειδικού σκοπού συσκευές [2]

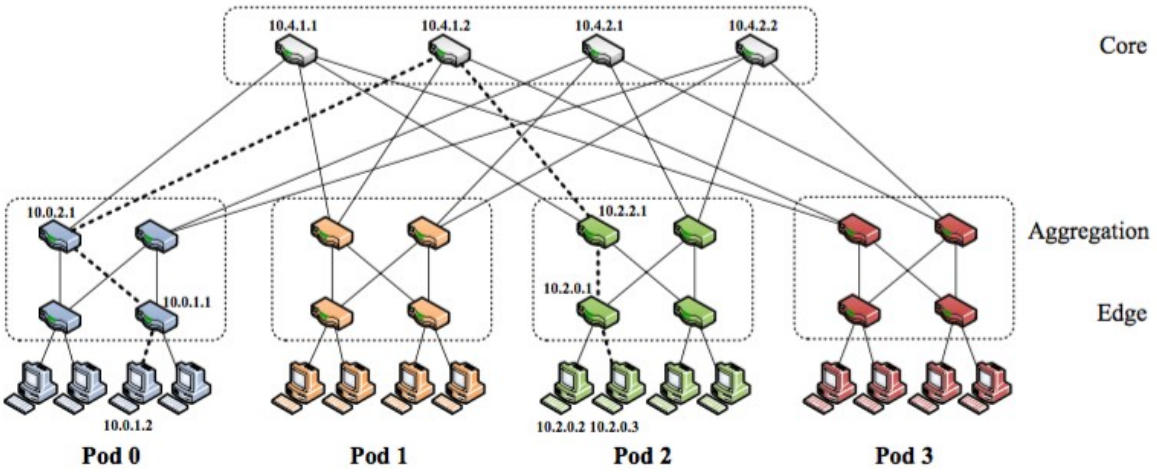
Έτσι η λύση που πλέον εφαρμόζεται είναι ο σχεδιασμός τοπολογιών τύπου Clos (Clos Topologies – Fat Trees) που είναι έτσι σχεδιασμένη ώστε να αντιμετωπίζει τα προβλήματα που μόλις αναφέραμε. Πιο συγκεκριμένα τα μεγάλα switches που βρίσκονται σε επίπεδα πάνω από τα ToR, αντικαθίστανται από ένα σύμπλεγμα δυο επιπέδων από απλά (commodity) switches (ένα για επικοινωνία με το κάτω επίπεδο και ένα για επικοινωνία με το από πάνω (aggregation/edge layer). Το κάθε switch από το κάθε ένα από τα 2 επίπεδα συνδέεται με κάθε ένα από το άλλο. Ως παράδειγμα της οργάνωσης παρουσιάζονται οι επόμενες εικόνες (σχήμα 4 και σχήμα 5). Αυτό που παρατηρείται είναι ότι επιτυγχάνεται αντικατάσταση των ειδικών-μεγάλων switches με έναν αριθμό απλών, commodity, προσφέροντας σημαντική μείωση κόστους, την συμβατότητα μεταξύ όλων των switches (ευκολία στον προγραμματισμό τους και κατ' επέκτασιν χρήση SDN και Virtualization). Αυτά τα σύνθετα switches υλοποιημένα από commodity switches αναφέρονται ως οντότητα και ως "White Box".

Ακόμα γίνεται εύκολα αντιληπτό πως με την αύξηση των ακμών λόγω σχεδιασμού έπεται αύξηση του συνόλου των διαδρομών (paths), οπότε σε περίπτωση σφάλματος ή διακοπής

λειτουργίας ενός switch υπάρχουν εναλλακτικές διαδρομές, εξαλείφοντας ουσιαστικά το πρόβλημα με μόνο κόστος μια μικρή μείωση στη χωρητικότητα του δικτύου (network capacity).



Σχήμα 4: Αντικατάσταση των μεγάλων switches με σύμπλεγμα μικρότερων commodity switches [66]



Σχήμα 5: Spine-leaf δομή υλοποιημένη με συμπλέγματα από commodity switches [2]

Στη συνέχεια παρουσιάζονται διάφορες αρχές και λειτουργίες που ακολουθούν τα data-centers, έτσι ώστε να ανταποκρίνονται αποδοτικά στις απαιτήσεις του Cloud.

2.1.2 Network Virtualization

Καθώς οι επιχειρήσεις και οι ιδιώτες μεταφέρουν τις υπολογιστικές τους λειτουργίες στα data-centers, επιθυμούν το ίδιο και για τις δικτυακές τροποποιήσεις τους. Αυτό μάλιστα κρίνεται

αναγκαίο για τους παρόχους τηλεπικοινωνιών, που αυτοσκοπός τους είναι η παροχή δικτυακών υπηρεσιών.

Το network virtualization [3] (δικτυακή εικονικοποίηση) αποτελεί σημαντικό χαρακτηριστικό του Cloud και της λειτουργίας των data-centers και αυτό γιατί πάνω σε μια υλική υποδομή μπορούν να αναπτυχθούν και να "συμβιώσουν" πολλά εικονικά δίκτυα για την εξυπηρέτηση των διαφορετικών αναγκών των πολλαπλών ενοίκων (tenants). Επίσης, η εφαρμογή νέων υπηρεσιών που θα απαιτούσαν πολύ χρόνο λόγω της τροποποίησης που χρειάζεται σε υλικό, τώρα γίνεται πάρα πολύ γρήγορα απλά με τη δημιουργία Virtual Machines (εικονικών μηχανών) και της λογικής (software) διαμόρφωσής τους ώστε να παραχθούν τα εικονικά δίκτυα που θα εξυπηρετήσουν τις υπηρεσίες. Ακόμα σε περίπτωση μετανάστευσης (migration) ενός φόρτου εργασίας, δηλαδή μεταφορά σε άλλη φυσική υπολογιστική μονάδα - server, για παράδειγμα λόγω σφάλματος ή λόγω συντήρησης, με τη χρήση VMs γίνεται άμεσα και χωρίς κίνδυνο διακοπής της εκτελούμενης λειτουργίας.

Όπως αναφέρθηκε τα εικονικά δίκτυα εφαρμόζονται πάνω σε κάποια υλική υποδομή – βάση. Η διαχείριση της υλικής υποδομής γίνεται μέσω ενός επιπέδου λογισμικού (software layer) που ονομάζεται **hypervisor** (πχ. Zen, KVM, ESXI) και διαδρά μαζί της μέσω μιας φυσικής κάρτας (κάρτα υλικού) διεπαφής δικτύου (physical Network Interface Card – **pNIC**). Από την άλλη πλευρά ο hypervisor τρέχει κάποιες εικονικές μηχανές που εκτελούνται στον χώρο χρήστη (namespace) και με τις οποίες αλληλεπιδρά μέσω εικονικών καρτών διεπαφής δικτύου (virtual Network Interface Card – vNIC). Με αυτόν τον τρόπο ο hypervisor παρέχει μία εξομοίωση της υποκείμενης υλικής υποδομής σε κάθε εικονική μηχανή (VM). Η λειτουργικότητα παρέχεται μέσω μιας εικονικής λειτουργίας που εκτελεί ο hypervisor όπως ένα εικονικό switch – vSwitch, το οποίο προωθεί τα πακέτα μεταξύ των εικονικών μηχανών και του υλικού.

Οι εικονικές μηχανές όμως όπως είναι γνωστό τρέχουν από μία προσομοίωση του λειτουργικού συστήματος η κάθε μία. Έτσι, για την εκτέλεση κάθε υπηρεσίας απαιτείται η δημιουργία μιας εικονικής μηχανής και εκτέλεση ολόκληρου λειτουργικού συστήματός της. Συνολικά αυτό, όπως αντιλαμβανόμαστε προκαλεί μεγάλο overhead (περίσσεια πληροφορίας). Έτσι γίνεται χρήση εναλλακτικών πιο ελαφρών δομών, όπως είναι η χρήση **containers** και η διαχείρισή τους από το **Docker**. Το κάθε container αντλεί τους απαραίτητους πόρους για την εκτέλεση της ανατιθέμενης εφαρμογής του και τρέχει πάνω στην στοίβα του λειτουργικού συστήματος της μηχανής που το φιλοξενεί και όχι σε ολόκληρο λειτουργικό σύστημα αφοσιωμένο σε αυτό όπως τα VMs. Στη συνέχεια καθώς θα παρουσιαστούν οι SDN και NFV τεχνολογίες θα γίνει καλύτερα κατανοητό, πώς μέσω αυτών επιτυγχάνεται η ζητούμενη εικονικοποίηση.

2.1.3 Migration

Το migration (μετανάστευση), αναφέρεται στην περίπτωση όπου μία εικονική μηχανή (ή και περισσότερες) που εκτελεί μία λειτουργία – υπηρεσία εκτελείται σε έναν φυσικό υπολογιστικό κόμβο - server (host), μεταφέρεται σε άλλον φυσικό υπολογιστικό κόμβο (host), εντός της δομής του data-center (ή σε κάποιες περιπτώσεις και εκτός).

Αυτό είναι ένα πολύ συχνό φαινόμενο στα multi-tenant συστήματα (τα οποία εξηγούνται στην επόμενη παράγραφο) καθώς, ανάλογα με τις ανάγκες και τους περιορισμούς σε πόρους γίνονται μεταφορές των εικονικών μηχανών μεταξύ των servers της υποδομής για λόγους εξισορρόπησης φόρτου (load balancing).

Θέτει σημαντικά ζητήματα συνέπειας (consistency), συγχρονισμού, καθώς και της ανάγκης μιας φαινομενικά συνεχούς (seamless) εκτέλεσης των λειτουργιών των εν λόγω μηχανών, δηλαδή η μεταφορά να γίνεται με "διάφανο" τρόπο.

2.1.4 Multi-tenancy

Ως **tenant** (ένοικος) ορίζεται μία ομάδα χρηστών που μοιράζεται την ίδια όψη πάνω σε μία εφαρμογή που χρησιμοποιούν. Στην όψη αυτή περιλαμβάνονται τα δεδομένα τα οποία προσπελαύνουν, η τροποποίηση (configuration), η διαχείριση χρήστη, συγκεκριμένη λειτουργικότητα (functionality), καθώς και σχετικές μη-λειτουργικές ιδιότητες. Συνήθως οι ομάδες αποτελούν μέλη διαφορετικών νομικών οντοτήτων (legal entities). Αυτό συνεπάγεται κάποιους περιορισμούς όπως είναι στην ασφάλεια δεδομένων και την ιδιωτικότητα[1].

Έτσι όταν αναφερόμαστε στον όρο multi-tenancy εννοούμε την τεχνολογία όπου ένα στιγμιότυπο μιας εφαρμογής (application instance) μοιράζεται μεταξύ πολλαπλών tenants. Ο κάθε tenant αποκτά το δικό του αντίγραφο από το στιγμιότυπο, με την έννοια της απομόνωσης. Δηλαδή, ο κάθε tenant μπορεί να τροποποιήσει κάποια χαρακτηριστικά του δικού του στιγμιότυπου, χωρίς να επηρεάζει τα στιγμιότυπα των άλλων και σε καμία περίπτωση τον κώδικα της κεντρικής εφαρμογής[1].

Μια ακόμα σημαντική ιδέα είναι αυτή του χώρου ενοίκου (tenant space), στην οποία ο tenant μπορεί να καθορίσει έναν χώρο από πόρους για να τρέξει διάφορες εφαρμογές. Ως παράδειγμα αναφέρουμε το IaaS (Infrastructure-as-a-Service – υποδομή ως υπηρεσία), όπου ο πελάτης αγοράζει πόρους για να τρέξει τις εφαρμογές που επιθυμεί[1].

Επίσης δε θα πρέπει να υπάρχει σύγχυση με το multiple application deployment (εγκατάσταση πολλαπλών εφαρμογών) το οποίο αναφέρεται στην εκτέλεση πολλαπλών εφαρμογών σε ένα στιγμιότυπο ενός περιβάλλοντος χρόνου εκτέλεσης.

Παρατηρούμε ότι με τη χρήση του Virtualization και της απομακρυσμένης πρόσβασης, το multi-tenancy αποκτά έναν καίριο ρόλο στα σημερινά συστήματα cloud για παροχή SaaS (Software-as-a-Service) καθώς και IaaS που είδαμε παραπάνω με το tenant space.

2.1.5 Microservice architecture

Με τον όρο microservices [5] αναφερόμαστε στη λογική όπου η υλοποίηση μιας εφαρμογής γίνεται μέσω μιας σειράς συνδεδεμένων εκτελέσιμων επιμέρους τμημάτων προγράμματος με την μορφή υπηρεσιών.

Κάθε τμήμα-υπηρεσία εκτελεί μια συγκεκριμένη λειτουργία, μέρος της συνολικής λειτουργίας της εφαρμογής. Επικοινωνεί με τις υπόλοιπες υπηρεσίες της εφαρμογής μέσω API interface, έτσι ώστε να συντονίζονται και να αποδίδουν τη συνολική λειτουργία-αποτέλεσμα της εφαρμογής. Κάθε επιμέρους υπηρεσία είναι απομονωμένη (isolated) από τις υπόλοιπες με την

έννοια ότι εκτελείται και μπορεί να διαχειριστεί ανεξάρτητα των υπολοίπων. Με αυτό τον τρόπο, παρατηρούμε ότι η εφαρμογή που εκτελείται με αυτήν την αρχιτεκτονική, αποκτά κατανεμημένο τρόπο εκτέλεσης, ακολουθώντας την ιδέα του διαίρει και βασίλευε.

Παρατηρούνται πολλά πλεονεκτήματα από τη χρήση των microservices. Πρώτον, ο τεμαχισμός της εφαρμογής σε μικρότερες υπηρεσίες, δίνει τη δυνατότητα κατανομής αυτών των υπηρεσιών σε διαφορετικές φυσικές υπολογιστικές μονάδες – εξυπηρετητές, έτσι το σύνολο της εφαρμογής μπορεί να υλοποιηθεί πολύ πιο γρήγορα καθώς εκτελείται σε μία μορφή παραλληλισμού. Αυτή η δυνατότητα κατανομής βοηθάει ακόμα στην εξομάλυνση του φόρτου εργασίας μέσα στο data-center, εφόσον χρησιμοποιηθούν κατάλληλες τεχνικές διαχείρισης για αποδοτική κατανομή των υπηρεσιών μεταξύ των servers. Ακόμα είναι πιο αποτελεσματικό στα σφάλματα, αφού ένα σφάλμα σε μία υπηρεσία δεν επηρεάζει την εκτέλεση των υπολοίπων, καθώς επίσης λόγω της τμηματοποίησης της εφαρμογής, ένα σφάλμα μπορεί να εντοπιστεί και να αντιμετωπιστεί πιο εύκολα και γρήγορα. Άλλα πλεονεκτήματα είναι η γρήγορη επεκτασιμότητα (scale), η επαναχρησιμοποίηση, αλλά και η αποδοτική χρήση VMs και containers που οφείλεται στη δυνατότητα να τοποθετούμε τις υπηρεσίες σε αυτά.

Από την άλλη μεριά, υπάρχουν κάποια προβλήματα που ενδέχεται να προκύψουν σε μία υλοποίηση microservice αρχιτεκτονικής. Κάποια σημαντικά από αυτά είναι: η δυσκολία στον συγχρονισμό και συντονισμό μεταξύ των υπολογιστικών μονάδων που εκτελούν τις υπηρεσίες μιας εφαρμογής, η αυξημένη κίνηση μέσα στο δίκτυο λόγω του διασκορπισμού και της επικοινωνίας των υπηρεσιών, καθώς και η πολυπλοκότητα. Είναι προκλήσεις οι οποίες αναφέρθηκαν στη λειτουργία του scatter-gather.

2.1.6 Δρομολόγηση

Συνεχίζοντας, θα εστιαστεί τώρα το ενδιαφέρον μας στην δρομολόγηση των δεδομένων κάνοντας μια μικρή γενική ανάλυση στις προκλήσεις που παρατηρούνται και στις λύσεις και πρωτόκολλα που προτείνονται.

Όπως αναφέρθηκε και προηγουμένως οι αιτήσεις εξυπηρέτησης οι οποίες καταφθάνουν από τον πελάτη (client) στο data-center ακολουθούν μια διαδικασία scatter-gather (διασκορπισμού-συλλογής), δηλαδή δημιουργούνται αντίγραφα των αιτήσεων και των δεδομένων τους καθώς επίσης αναλύονται και σε μικρότερα κομμάτια και διαμοιράζονται σε διάφορους φυσικούς servers. Αυτό γίνεται πρώτον για την αποφυγή σφαλμάτων, π.χ. αν ένας server “πέσει” ένα αντίγραφο της αίτησης θα εξυπηρετηθεί από κάποιον άλλον server. Δεύτερον, ακολουθείται η λογική του διαίρει και βασίλευε, καθώς οι αιτήσεις τεμαχίζονται σε μικρότερες πιο απλές που διαμοιράζονται σε διαφορετικούς servers και έτσι το καθένα από αυτά τα κομμάτια επιλύεται-εξυπηρετείται τάχιστα με αποτέλεσμα το συνολικό request, που θα προκύψει από την συγχώνευση των επί μέρους τμημάτων, να εξυπηρετείται και αυτό πολύ πιο γρήγορα.

Εδώ παρατηρείται ότι αυτή η διαδικασία δημιουργεί σημαντικές προκλήσεις και προβλήματα καθώς δημιουργείται ραγδαία κίνηση πληροφορίας μέσα στο Data Centers μεταξύ των servers και συνεπώς αυξημένες πιθανότητες συγκρούσεων των πακέτων. Ακόμα όπως είναι φυσικό τίθενται και θέματα συγχρονισμού - σωστής ενημέρωσης στις μηχανές. Στα παραπάνω προστίθεται και ο κυκλοφοριακός φόρτος από το migration.

Κάποια σημαντικά από τα πρωτόκολλα δρομολόγησης που χρησιμοποιούνται στις συγκεκριμένες αρχιτεκτονικές για την αποφυγή των παραπάνω προβλημάτων είναι το MPLS, το ECMP, το iBGP/eBGP και το DCTCP. Ακολουθεί η λειτουργία τους επιγραμματικά:

MPLS: Το MPLS είναι ένα πρωτόκολλο δρομολόγησης, που όπως αναφέρει και το όνομά του (Multi-Protocol Label Switching), βασίζει την μετάδοση των πακέτων σε ετικέτες (labels), υπό την μορφή επέκτασης της επικεφαλίδας τους, οι οποίες υποδεικνύουν τη συντομότερη διαδρομή που θα ακολουθήσουν μέσα στο δίκτυο μέχρι τον τελικό προορισμό.

ECMP: Όπως αναφέρει το όνομά του (Equal-Cost Multi-Path routing**),** το πρωτόκολλο αυτό, διενεργεί δρομολόγηση, με μετάδοση του πακέτου στον επόμενο κόμβο μέσω πολλαπλών διαδρομών μετάδοσης ίσου κόστους. Αποτελεί μία τεχνική που επιφέρει load balancing, και όπως γίνεται αντιληπτό, ταιριάζει στις απαιτήσεις των clos τοπολογιών που αναφέρθηκαν σε προηγούμενη ενότητα.

iBGP/eBGP: Αποτελούν εκδοχές του BGP (Border Gateway Protocol). Το πρώτο αναφέρεται στην εκδοχή του πρωτοκόλλου, υπεύθυνο για την επικοινωνία εντός ενός αυτόνομου συστήματος (autonomous system), ενώ το δεύτερο για την επικοινωνία μεταξύ διαφορετικών αυτόνομων συστημάτων. Ως αυτόνομο σύστημα (autonomous system – AS) ορίζεται ένα δίκτυο ή μία συλλογή από δίκτυα που υπόκεινται στον έλεγχο και την διαχείριση μιας μονάδας ή ενός οργανισμού.

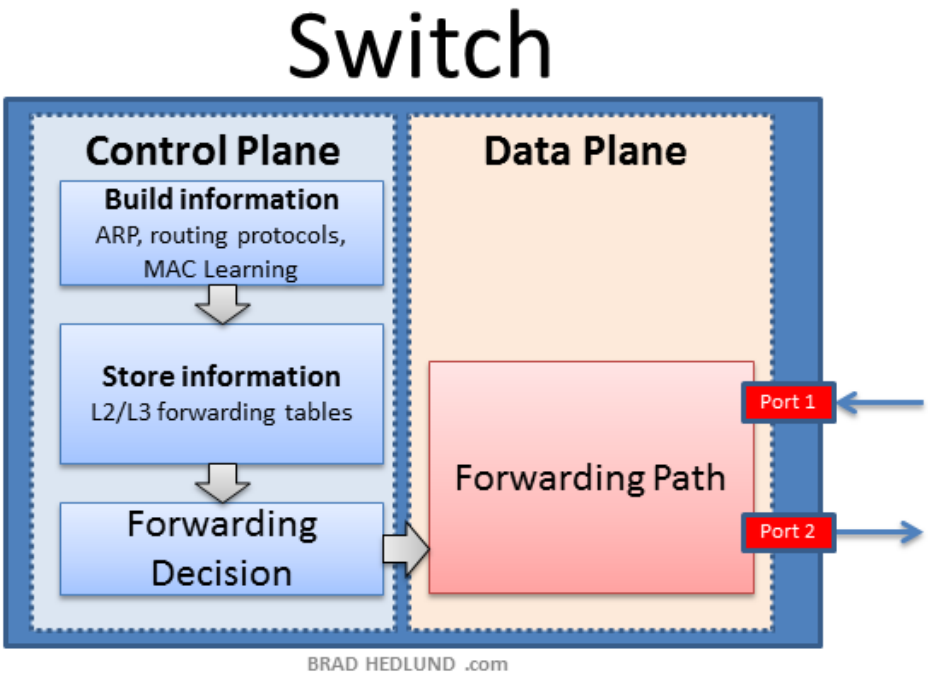
DCTCP: Το DCTCP (Data Center TCP) αποτελεί πρωτόκολλο που βασικός του στόχος είναι η αντιμετώπιση της κυκλοφοριακής συμφόρησης (congestion) του δικτύου. Χρησιμοποιεί την τεχνική του ECN (Explicit Congestion Notification), δηλαδή μιας ρητής επισήμανσης συμφόρησης, που εφαρμόζεται στις ουρές των μέσων προώθησης πακέτων, όπου όταν ξεπεραστούν κάποια προκαθορισμένα όρια μέσα στον buffer της ουράς, ρυθμίζεται κατάλληλα ο ρυθμός αποστολής των δεδομένων για την αποφυγή της συμφόρησης.

2.2 Software - Defined Networking (SDN)

Το SDN αποτελεί μία αναδεικνυόμενη τεχνολογία στον τομέα των δικτύων η οποία έχει ως βασικό στοιχείο ότι διαχωρίζει τον έλεγχο του δικτύου (network control) από την απτή μετάδοση των δεδομένων (forwarding) του υλικού, δίνοντας έτσι την δυνατότητα της περιγραφής και διαχείρισης του δικτύου μέσω προγραμματισμού υψηλού επιπέδου.

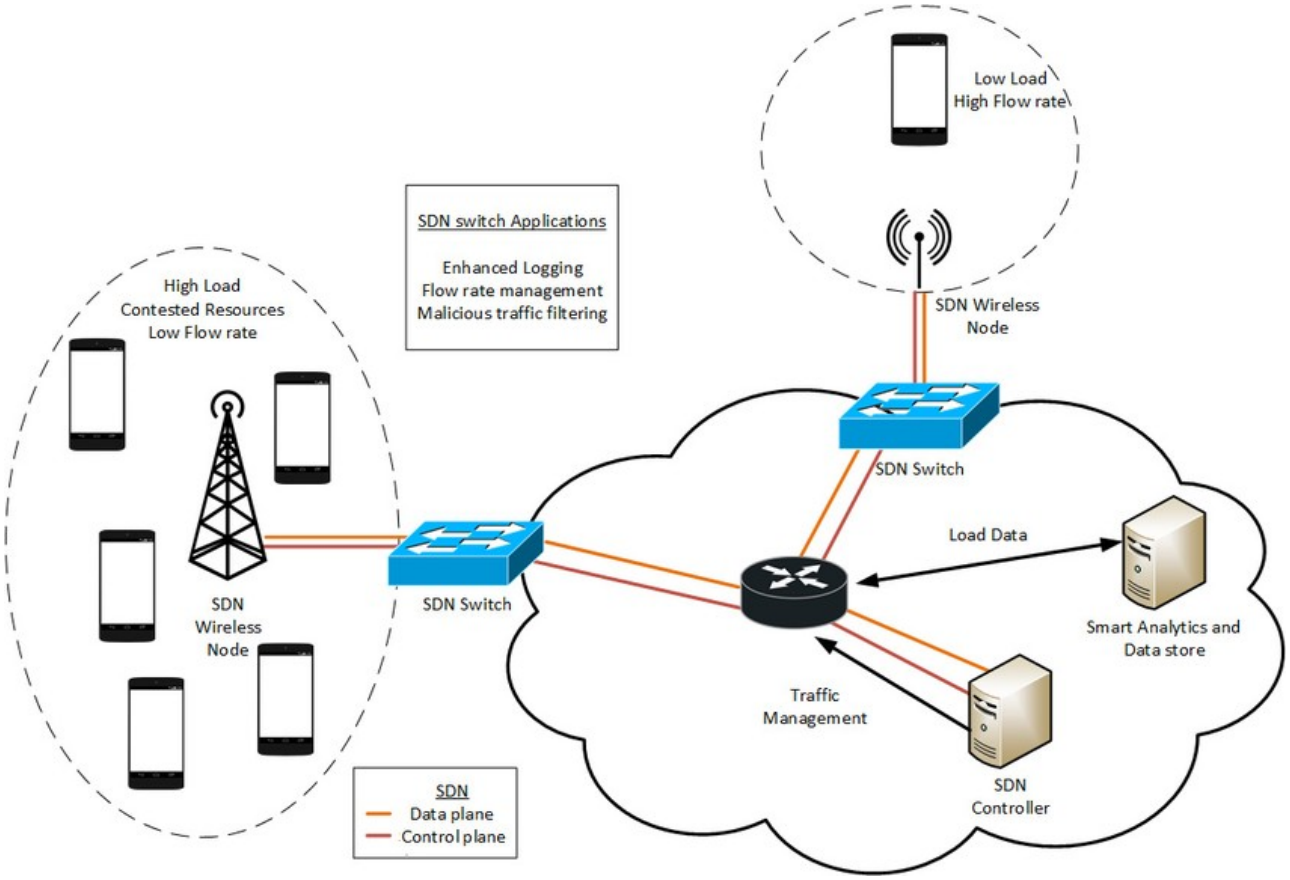
Έτσι ο έλεγχος του δικτύου που ως πρότινος ήταν ισχυρά συνδεδεμένος με την κάθε συσκευή δικτύου, με την έννοια ότι κάθε συσκευή-κόμβος αυτοβούλως αποφάσιζε για το πώς θα κατευθύνει την διερχόμενη κίνηση βασισμένη στα route tables της, γίνεται τώρα διαχειρίσιμος σε κεντρικές υπολογιστικές συσκευές (computing devices) ή αλλιώς NCPs (network control points), δημιουργώντας έτσι ένα αφαιρετικό μοντέλο για τις εφαρμογές και τις υπηρεσίες δικτύου, προσδίδοντας στο δίκτυο μια προγραμματιστική virtual εικόνα.

Παρατηρούμε ότι χάρις το SDN, από τα παραδοσιακά δίκτυα όπου η αρχιτεκτονική των δικτύων είναι αποκεντρωμένη (decentralised) και πολύπλοκη εισερχόμαστε σε μία αρχιτεκτονική κεντρικοποιημένου (centralised) ελέγχου, αποσυνδέοντας τη διαδικασία της μετάδοσης των πακέτων (network packet forwarding) δηλαδή το data plane από τη διαδικασία της δρομολόγησης, το λεγόμενο control plane.



Σχήμα 6: Διαχωρισμός του control-plane από το data-plane στο switch [74]

Με άλλα λόγια το control plane αποτελεί το ευφυές μέρος του δικτύου όπου λαμβάνονται αποφάσεις για δρομολόγηση με βέλτιστο τρόπο, διαχείριση και αποφυγή σφαλμάτων, και ανεπιθύμητων καταστάσεων στο υποκείμενο δίκτυο. Η διαχείριση και η καθοδήγηση του control plane γίνεται από ένα κεντρικό σημείο, τον SDN controller. Με αυτόν τον τρόπο θα μπορούσε κανείς να περιγράψει το σύνολο του δικτύου ως ένα γενικό προγραμματίσιμο switch.



Σχήμα 7: Ενδεικτική οργάνωση δικτύωσης κατά SDN [74]

Στο σχήμα από πάνω βλέπουμε τη γενική δομή ενός δικτύου βασισμένο στο SDN με κινητούς τελικούς χρήστες.

Βασικό πρωτόκολλο με ευρεία χρήση στα SDN δίκτυα είναι το **OpenFlow**. Το OpenFlow επιτρέπει απομακρυσμένη διαχείριση, στο επίπεδο 3 της στοίβας πρωτοκόλλων δικτύου, των λιστών των πακέτων προώθησης (switch's packet forwarding tables) διαχειρίζοντας τους κανόνες ταιριάσματος και τις ενέργειες των πακέτων. Έτσι επιτυγχάνεται ο απομακρυσμένος έλεγχος για τη δρομολόγηση από τον controller εισάγοντας στα επιμέρους switches τις εντολές και τους κανόνες δρομολόγησης της κίνησης και των πακέτων του δικτύου στις λίστες ροών τους (flow tables), επιτρέποντας έτσι κεντρική προγραμματιστική διαχείριση του δικτύου, δηλαδή το SDN.

2.2.1 Αρχιτεκτονική του SDN

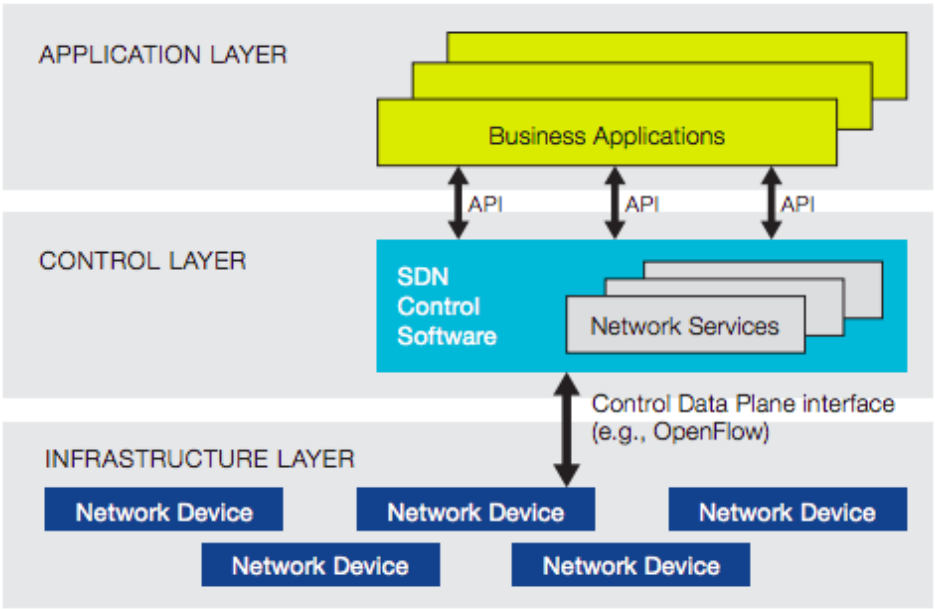
Η αρχιτεκτονική του SDN έχει τα εξής βασικά συστατικά μέρη[10]:

- Τα **SDN Applications** τα οποία είναι προγράμματα που περιγράφουν τις επιθυμητές προδιαγραφές και λειτουργία του δικτύου και επικοινωνούν με τον controller μέσω μιας northbound διεπαφής (northbound interface - NBI). Τέτοια παραδείγματα SDN εφαρμογών

(applications) είναι τα: firewall, access control, IDS/IPS, quality of service (QoS), routing, proxy service, and monitoring balancer. Κάθε SDN Application αποτελείται από μια μονάδα λογικού (SDN Application Logic) και από έναν ή περισσότερους NBI οδηγούς (NBI Drivers).

- Τον **SDN Controller** ο οποίος αποτελεί μία λογικά κεντρικοποιημένη μονάδα που δέχεται τις πληροφορίες που δίνουμε από τα SDN Applications, και τις μεταφέρει στο data plane, δηλαδή στις δικτυακές μονάδες που επιτελούν τη μετάδοση της πληροφορίας στο δίκτυο. Από την άλλη, ο Controller παρουσιάζει το δίκτυο με αφαιρετικό (abstract) τρόπο καθώς και διάφορες πληροφορίες σχετικά με αυτό στα SDN Applications. Αποτελείται από μονάδα ελέγχου και από μονάδες NBI Agent, για την επικοινωνία με το επίπεδο εφαρμογών (application layer), και για επικοινωνία με το data plane, δηλαδή το Infrastructure layer, μέσω του Control to Data Plane Interface, καλούμενο και ως Southbound Interface (SBI), με πιο γνωστό πρωτόκολλο τέτοιας διεπαφής το OpenFlow.

- Το **Infrastructure** (υποδομή) το οποίο συμπεριλαμβάνει τις φυσικές δικτυακές συσκευές και συσκευές λογισμικού, δηλαδή τα switches/vSwitches, routers και access points του δικτύου, που ο ρόλος τους είναι να προωθούν τα πακέτα με τον τρόπο που τους υποδεικνύει το SDN controller.



Σχήμα 8: Βασική αρχιτεκτονική του SDN [7]

Από τα παραπάνω συμπεραίνουμε ότι σημαντικά χαρακτηριστικά της αρχιτεκτονικής SDN είναι τα εξείς:

- Απευθείας προγραμματίσιμο: εφόσον ο έλεγχος του δικτύου έχει διαχωριστεί από τις λειτουργίες προώθησης (forwarding functions), έχουμε την δυνατότητα άμεσου προγραμματισμού του.

- Ευελιξία (agility): η αφαίρεση του ελέγχου από το υποκείμενο δίκτυο προώθησης, παρέχει τη δυνατότητα δυναμικής προσαρμογής της ροής κίνησης (traffic flow) μέσα στο δίκτυο, για την ικανοποίηση των εκάστοτε αναγκών.

- Προγραμματιστικά τροποποιήσιμη: οι διαχειριστές δικτύων μπορούν να τροποποιούν, να διαχειρίζονται, να εξασφαλίζουν και να βελτιώνουν τους πόρους δικτύου πολύ γρήγορα μέσω δυναμικών, αυτοματοποιημένων προγραμμάτων τα οποία μπορούν και οι ίδιοι να έχουν γράψει, αφού τα προγράμματα δεν εξαρτώνται από ιδιωτικό λογισμικό (proprietary software).

- Βασισμένο σε γενικά ανοιχτά πρότυπα: Εφαρμοζόμενο το SDN από ανοιχτά πρότυπα (open standards) και κοινά προγραμματιστικά περιβάλλοντα, καθιστά απλούστερο τον σχεδιασμό του δικτύου και την λειτουργία του, καθώς οι παρεχόμενες οδηγίες παρέχονται από τους SDN controllers αντί πολλαπλών συσκευών και πρωτοκόλλων εξειδικευμένων βάσει των εμπόρων (vendor specific).

Μερικά ακόμα από τα πλεονεκτήματα που επιφέρει η χρήση SDN είναι:

- Μεγαλύτερη αξιοπιστία και ασφάλεια: λόγω του κεντρικοποιημένης και αυτοματοποιημένης διαχείρισης των δικτυακών συσκευών, της επιβολή ενιαίας πολιτικής, και των λιγότερων σφαλμάτων τροποποίησης.

- Μείωση των CAPEX/OPEX: καθώς το control plane διαχωρίζεται και μεταφέρεται σε μεγάλο μέρος στον SDN controller, μειώνεται η πολυπλοκότητα των δικτυακών συσκευών. Έτσι οι απαιτήσεις για το υλικό μικραίνουν και στρεφόμαστε σε συσκευές πιο γενικής χρήσης (commodity devices) με πολύ χαμηλότερο κόστος. Επίσης η απλοποίηση της συνολικής λειτουργίας του δικτύου που επιφέρει η χρήση SDN, όπως είναι εμφανές, μειώνει κατά πολύ το διαχειριστικό κόστος.

2.3 Network Functions Virtualization (NFV)

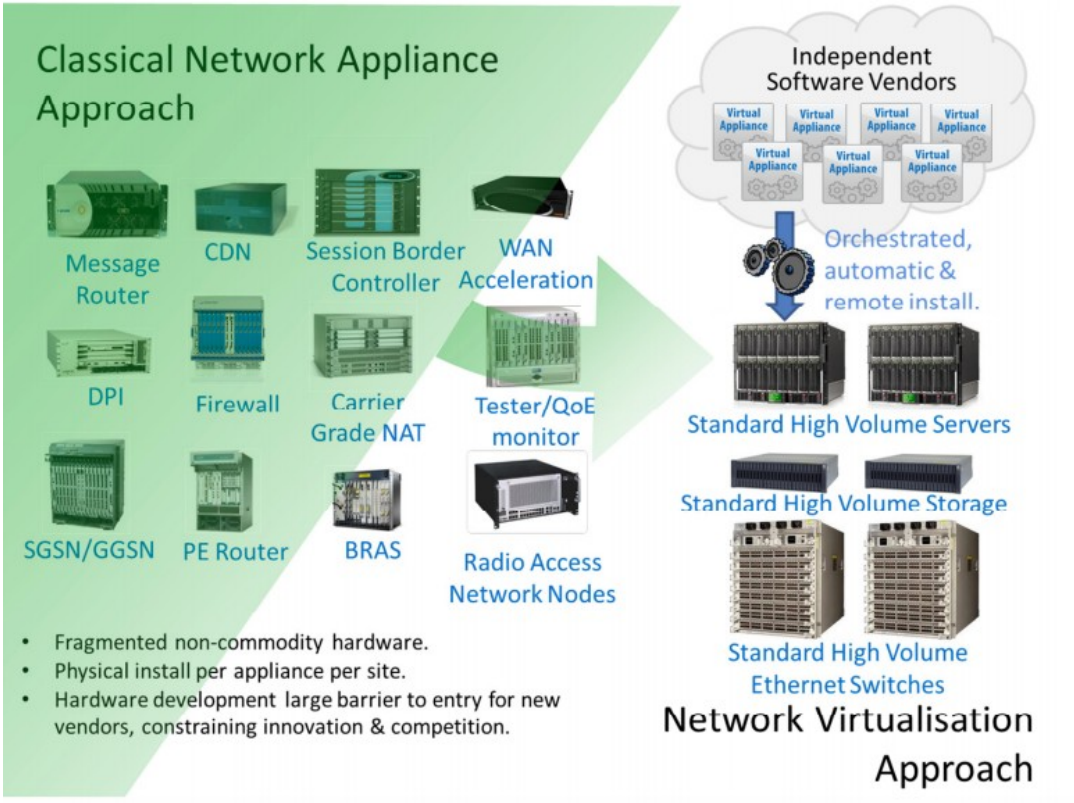
Το NFV (Network Functions Virtualization - Εικονικοποίηση Λειτουργιών Δικτύου) αναφέρεται στην τεχνολογία εικονικοποίησης των διαφόρων λειτουργιών του δικτύου που παραδοσιακά βασίζονταν πάνω σε συσκευές υλικού (hardware devices). Έτσι ολόκληρες κλάσεις από λειτουργίες δικτύου εικονικοποιούνται σε κατασκευαστικά στοιχεία τα οποία μπορούν να συνδέονται με διάφορους τρόπους όπως την διαδικασία αλυσίδας υπηρεσιών (service chaining), ώστε να επιφέρουν τις επιθυμητές υπηρεσίες.

Η υλοποίηση του NFV βασίζεται στη δημιουργία και χρήση των VNFs (Virtualized Network Functions – εικοινικοποιημένες λειτουργίες δικτύου). Ένα VNF αποτελεί την εικονική (virtual) μορφή μεμονωμένων εργασιών που μέχρι πρότινος εκτελούνταν με τη χρήση υλικού, εξειδικευμένου στη συγκεκριμένη λειτουργία της κάθε εργασίας. Η χρήση υλικού αποτελούσε πιο απλή προσέγγιση στο παρελθόν καθώς επίσης εξυπηρετούσε προσφέροντας ένα ισχυρά συνεκτικό και αξιόπιστο δίκτυο, με την έννοια του carrier grade, αποτελούμενο από ένα σύνολο διαφορετικών μονάδων υλικού, αφοσιωμένων σε συγκεκριμένες λειτουργίες (routing, DNS, firewall, NAT, caching, load balancers, κλπ.) το καθένα. Όμως αυτό δημιουργεί μια ισχυρή εξάρτηση από το υλικό καθιστώντας το δίκτυο μη ευέλικτο, δύσκολο και κοστοβόρο στην ανάπτυξη, την εξέλιξή και τη διαχείρισή του. Έτσι, λόγω της ταχείας εξέλιξης του διαδικτύου και των τηλεπικοινωνιών,

στραφήκαμε σε μία πιο ευέλικτη, καλύτερη στη διαχείριση δικτυακών πόρων, υλοποίηση, αυτή των εικονικών ισοδύναμων λειτουργιών, που εφαρμόζονται και διαχειρίζονται με προγραμματιστικό τρόπο και δεν απαιτούν ακριβό υλικό ειδικού σκοπού, παρά τη χρήση φθηνότερων μακροβιότερων μηχανών γενικού σκοπού (commodity devices-servers) στις οποίες τρέχουν.

Κατ' αυτόν τον τρόπο μεταφέρουμε την διεκπεραίωση των υπηρεσιών (που τώρα υλοποιούνται από ένα σύνολο VNFs) από τις παραδοσιακές συσκευές ειδικού σκοπού, σε data-centers, σε γενικούς κόμβους δικτύου και στις συσκευές στην πλευρά του χρήστη (user premises). Έτσι τα VNFs, ως λογικές μονάδες μπορούν να τοποθετούνται ή να μετακινούνται σε διάφορες τοποθεσίες και σε διάφορες συσκευές εντός, εκτός ή και πέραν του ενός data-center, ώστε να είναι πιό αποδοτικό το δίκτυο (efficiency, load balance, κλπ). Με αυτόν τον τρόπο παρατηρούμε ότι το NFV παίρνει μία κατανεμημένη μορφή.

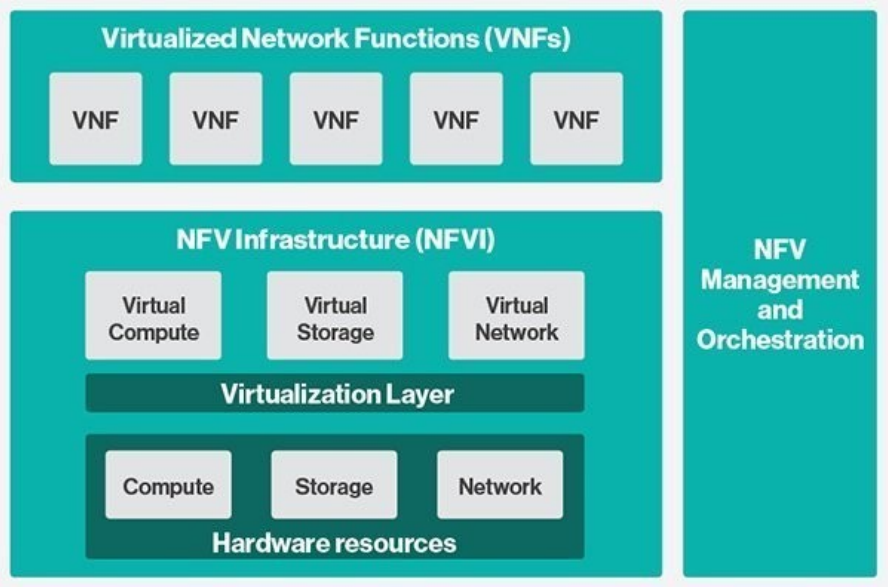
Ένα VNF αποτελείται από ένα ή περισσότερα Virtual Machines, τα οποία τρέχουν διαφορετικά προγράμματα πάνω σε καθιερωμένους υψηλού όγκου εξυπηρετητές (standard high-volume servers), μεταγωγείς (switches), συσκευές αποθήκευσης και σε υποδομή νέφους.



Σχήμα 9: Μεταφορά από τις παραδοσιακές δικτυακές συσκευές ειδικού σκοπού στις αντίστοιχες εικονικές λειτουργίες τους μέσω του NFV [70]

2.3.1 Αρχιτεκτονική του NFV

Στην ακόλουθη εικόνα βλέπουμε την αρχιτεκτονική του NFV σε απλοποιημένη μορφή.



Σχήμα 10: Βασική αρχιτεκτονική του NFV [13]

Ακολουθεί μια σύντομη περιγραφή των στοιχείων της παραπάνω αρχιτεκτονικής NFV.

- **VNF (Virtualized Network Function)**: είναι όπως αναφέρθηκε οι εφαρμογές σε λογισμικό (software implementation) των λειτουργιών δικτύου και εφαρμόζονται πάνω σε μια NFV υποδομή (NFV Infrastructure).

- **NFVI (Network Function Virtualization Infrastructure)**: η υποδομή NFV είναι το σύνολο των στοιχείων υλικού και λογισμικού τα οποία μπορεί να βρίσκονται σε διαφορετικές τοποθεσίες και συνιστούν το περιβάλλον όπου εφαρμόζονται τα VNFs.

- **NFV-MANO (Network Function Virtualization Management and Orchestration)**: Είναι, όπως δηλώνει και το όνομά του, το NFV μέρος διαχείρισης και οργάνωσης. Αποτελείται από όλα τα λειτουργικά μέρη, τις αποθήκες δεδομένων (data storage), καθώς και τα σημεία αναφοράς (reference points) και τις διεπαφές (interfaces) με τις οποίες αυτά τα μέρη συνδιαλέγονται με σκοπό τη διαχείριση και την οργάνωση του NFVI και των VNFs. Επίσης συνδέεται με στοιχεία του NFVI ώστε να επικοινωνούν με τα λειτουργικά και χρεωστικά (billing) συστήματα υποστήριξης (OSS/BSS).

Το σύνολο NFVI και NFV-MANO συνιστά την NFV πλατφόρμα. Με το NFVI, όπως φαίνεται στην εικόνα να απαρτίζεται από τους εικονικούς (virtual), φυσικούς (υλικούς-physical) πόρους επεξεργασίας και αποθήκευσης, καθώς και το virtualization software. το NFV-MANO αποτελείται από VNF και NFVI managers, καθώς και virtualization software που λειτουργούν σε ελεγκτή υλικού (hardware controller). Ιδιότητες, απαραίτητες για το δημόσιο δίκτυο μεταφοράς, οι οποίες εφαρμόζονται από τη NFV πλατφόρμα αφορούν τη διαχείριση και καταγραφή (monitor) των στοιχείων της, ανασύσταση (recover) από τυχόν αποτυχίες και παροχή ασφάλειας.

Ένα VNF (εικονικοποιημένη λειτουργία δικτύου) δεν αποτελεί από μόνο του μία χρήσιμη υπηρεσία προς τους πελάτες του παρόχου, αλλά όπως αναφέρθηκε πιο γενικά στην αρχή, χρησιμοποιείται μία σύνδεση πολλαπλών VNFs σε σειρά (in sequence) που ονομάζεται service chaining (αλυσίδα υπηρεσίας), ώστε να παραχθεί μία χρήσιμη υπηρεσία δικτύου.

2.3.2 NFV και SDN

Όσον αφορά τη σχέση του NFV με το SDN, δεν θα πρέπει να υπάρχει σύγχυση μεταξύ των δύο εννοιών. Αποτελούν δυο διαφορετικές τεχνολογίες.

Το SDN βασίζεται στην αρχή του διαχωρισμού του control plane από το data plane, με το control plane να μεταφέρεται κεντρικά και το data plane (τα στοιχεία προώθησης των πακέτων) να παραμένει κατανεμημένο. Το control plane παρέχει μέσω NBIs (northbound interfaces) μια κοινή αφαιρετική όψη του δικτύου προς τις εφαρμογές υψηλού επιπέδου, ενώ μέσω των SBIs (southbound interfaces) προγραμματίζει τη συμπεριφορά προώθησης (forward) του data plane, δηλαδή του φυσικού εξοπλισμού του δικτύου.

Από την άλλη είδαμε ότι το VNF έχει να κάνει με την εικονικοποίηση (virtualization) των λειτουργιών του δικτύου. Έτσι, είναι φανερό ότι οι δύο τεχνολογίες είναι ανεξάρτητες. Είναι δυνατή η εφαρμογή μόνο NFV ή μόνο SDN πάνω σε ένα δίκτυο. Ωστόσο, η χρήση SDN αρχών στην εφαρμογή και διαχείριση του NFVI προσφέρει αρκετά πλεονεκτήματα. Για παράδειγμα μπορεί ένας κεντρικοποιημένος ελεγκτής (centralized controller) να ελέγχει μία κατανεμημένη λειτουργία προώθησης, ή οποία μπορεί να γίνει virtualized πάνω σε υπάρχων εξοπλισμό επεξεργασίας ή δρομολόγησης.

2.3.3 Εφαρμογές του NFV στο CORD

Σχετικά με το CORD, θα φανεί ο καθοριστικός ρόλος που παίζει το NFV, καθώς για την υλοποίηση της τεχνολογίας του CORD μεταφερόμαστε αποκλειστικά στην λογική του Everything-as-a Service (Xaas), και το VNF μας επιτρέπει την παροχή παραδοσιακών λειτουργιών του υποκείμενου υλικού δικτύου ως virtualized εκδοχές. Τέτοια παραδείγματα που θα παρουσιαστούν σε επόμενα κεφάλαια είναι τα vSwitch, vRouter, vSG, και οι mobile υπηρεσίες vBBU, vSGW, vPGW και άλλα.

3 CORD: Δομικά στοιχεία, βασικές υπηρεσίες και εφαρμογές

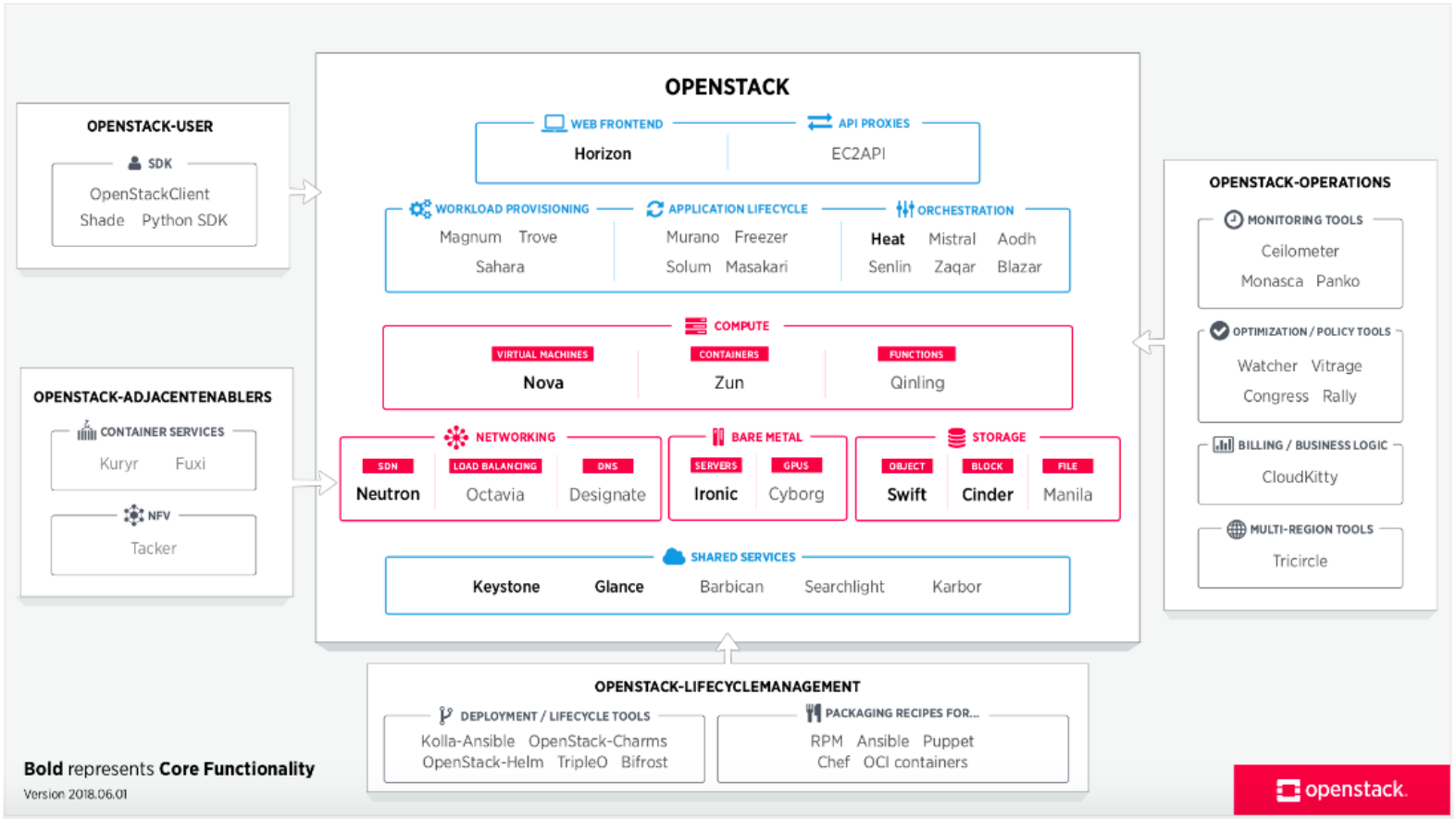
Σε αυτό το κεφάλαιο θα διεισδύσουμε στο CORD, αναλύοντας τα βασικά δομικά του μέρη: OpenStack, Docker, ONOS και XOS. Μέσα από αυτή την ανάλυση θα γίνουν κατανοητές οι επιμέρους λειτουργίες τους, το πως διασυνδέονται μεταξύ τους, αλλά και η συνολική λειτουργία του CORD που παρέχεται μέσω του συνδυασμού τους. Ακόμα θα παρουσιαστούν οι βασικότερες υπηρεσίες του CORD, vRouter, vSG και vOLT, το Trellis το οποίο αποτελεί ένα ενοποιημένο πρότζεκτ για τη διαχείριση του υποκείμενου δικτύου, η VTN εφαρμογή για παροχή εικονικών δικτύων και πως συνδυάζεται με το XOS και τα υπόλοιπα εργαλεία για να παράξει τις τελικές υπηρεσίες. Έτσι έχοντας μελετήσει τα παραπάνω, θα καταστεί σαφής η εικόνα του τι είναι το CORD, πως λειτουργεί και τις δυνατότητες που ορίζει για εφαρμογή επιπλέον υπηρεσιών, αλλά και το ρόλο του στην αποδοτικότερη απόδοση του Central Office.

3.1 OpenStack

Το OpenStack είναι μία πλατφόρμα ανοιχτού κώδικα (open source) για cloud, το οποίο διαχειρίζεται και προσφέρει με αποδοτικό τρόπο τους πόρους της υποδομής σε cloud συστήματα, εφαρμόζει δηλαδή την ιδέα της υποδομής ως υπηρεσία (Infrastructure-as-a-Service – IaaS), ώστε εικονικοί servers και άλλοι πόροι να γίνονται διαθέσιμοι και διαχειρίσιμοι στους πελάτες. Η πλατφόρμα αποτελείται από συσχετιζόμενα στοιχεία - συστατικά τα οποία ελέγχουν μία ποικιλία από multi-vendor επεξεργαστικά σύνολα υλικού, αποθήκευσης, και δικτυακούς πόρους σε ένα data-center. Οι χρήστες το διαχειρίζονται είτε μέσω μιας web εφαρμογής με command line εργαλεία, είτε μέσω REST web υπηρεσιών (dashboard).

3.1.1 Βασικά συστατικά στοιχεία του OpenStack

Στο επόμενο σχήμα παρουσιάζονται το σύνολο των στοιχείων που απαρτίζουν το OpenStack, καθώς επίσης και την βασική οργάνωσή τους σε αυτό. Στη συνέχεια εξετάζονται τα πιο σημαντικά από αυτά, που θα φανούν χρήσιμα για την κατανόηση του υπολοίπου της εργασίας.



Σχήμα 11: Τα συστατικά στοιχεία του OpenStack [16]

- **Compute (Nova)**: Πρόκειται για έναν ελεγκτή για cloud computing fabric, ο οποίος είναι το κύριο τμήμα ενός IaaS συστήματος. Είναι σχεδιασμένο να διαχειρίζεται και να αυτοματοποιεί σύνολα (pools) από υπολογιστικούς πόρους και μπορεί να δουλέψει με ένα ευρύ αριθμό διαθέσιμων τεχνολογιών για virtualization, καθώς και τροποποιήσεις (configurations) bare metal και υπολογισμού υψηλής απόδοσης (high performance computing – HPC). Τα KVM, VMware και Xen αποτελούν διαθέσιμες επιλογές για hypervisor (επόπτης εικονικών μηχανών), μαζί με το Hyper-V και τα Linux Containers (LXC)[17].

  Είναι γραμμένο σε Python και χρησιμοποιεί διάφορες εξωτερικές βιβλιοθήκες, όπως το Eventlet για σύγχρονο (concurrent) προγραμματισμό, Kambu για AMQP επικοινωνία και SQLAlchemy για πρόσβαση σε βάση δεδομένων. Είναι αρχιτεκτονικά σχεδιασμένο ώστε να επεκτείνεται "οριζόντια" σε προκαθορισμένο υλικό χωρίς απαιτήσεις για υλικό ή λογισμικό και έχει την ικανότητα να ενσωματώνεται με legacy συστήματα και third party τεχνολογίες[17].

  Λόγω της ευρείας ενσωμάτωσης σε εγκαταστάσεις επιχειρησιακού επιπέδου (enterprise-level infrastructures), η καταγραφή (monitoring) της απόδοσης του OpenStack και του Nova συγκεκριμένα είναι σημαντικό ζήτημα[17].

  Η καταγραφή από άκρη σε άκρη (end-to-end) της απόδοσης απαιτεί tracking metrics από τα Nova, Keystone, Neutron, Cinder, Swift και άλλων υπηρεσιών, καθώς και του RabbitMQ που χρησιμοποιείται από τις υπηρεσίες του OpenStack για πέρασμα μηνυμάτων.

Όλες αυτές οι υπηρεσίες δημιουργούν αρχεία καταχώρησης τα οποία με τη σειρά τους πρέπει να καταγράφονται (monitored)[17].

- **Networking (Neutron)**: Είναι σύστημα για διαχείριση δικτύων και IP διευθύνσεων (static ή DHPC). Το OpenStack Networking εξασφαλίζει ότι στο δίκτυο δεν γίνεται συμφόρηση και ότι δεν αποτελεί περιοριστικό παράγοντα σε μία cloud εγκατάσταση, και δίνει τη δυνατότητα στους χρήστες για self-service, ακόμα και πάνω σε τροποποιήσεις (configurations) δικτύου[17].

  Παρέχει δικτυακά μοντέλα για διάφορες εφαρμογές και ομάδες χρηστών. Στα προκαθορισμένα μοντέλα περιλαμβάνονται επίπεδα δίκτυα (flat networks) ή VLANs τα οποία διαχωρίζουν τους servers από την κίνηση[17].

  Οι χρήστες μπορούν να δημιουργούν δικά τους δίκτυα, έλεγχο κίνησης (control traffic) και να συνδέουν servers και συσκευές σε ένα ή περισσότερα δίκτυα. Οι διαχειριστές (administrators) μπορούν να χρησιμοποιούν SDN τεχνολογίες όπως το OpenFlow για υποστήριξη υψηλού επιπέδου multi-tenancy και επεκτασιμότητας. Επίσης το OpenStack networking (Neutron) παρέχει ένα framework το οποίο μπορεί να εφαρμοστεί και να διαχειριστεί επιπλέον δικτυακές εφαρμογές όπως ειναι οι: instruction detection systems (IDS), load balancing, firewall και VPN[17].

- **Block Storage (Cinder)**: Παρέχει συσκευές αποθήκευσης σε block level για χρήση με OpenStack υπολογιστικά στιγμιότυπα (compute instances). Το block storage σύστημα διαχειρίζεται τη δημιουργία, την προσθήκη και την απόσπαση των block συσκευών σε servers. Block αποθηκευτικοί όγκοι (storage volumes) είναι πλήρως ενσωματωμένοι στο OpenStack Compute και στο Dashboard, επιτρέποντας στους χρήστες cloud να διαχειρίζονται τις δικές τους ανάγκες αποθήκευσης. Πέραν του local Linux server storage μπορεί να γίνει χρήση μία σειρά σειρά από διαθέσιμες πλατφόρμες αποθήκευσης, όπως είναι τα Ceph, CloudByte. IBM storage κλπ[17].

- **Identity (Keystone)**: Παρέχει έναν κεντρικό κατάλογο με τους χρήστες χαρτογραφημένους (mapped) με τις OpenStack υπηρεσίες που μπορούν να έχουν πρόσβαση. Λειτουργεί ως ένα κοινό σύστημα πιστοποίησης (authentication) στο cloud OS και μπορεί να ενσωματωθεί με υπάρχοντα backend directory services όπως το LDAP. Υποστηρίζει πολλαπλούς τρόπους πιστοποίησης όπως username – password, token-based συστήματα, AWS logins[17].

- **Image (Glance**): Παρέχει υπηρεσίες εύρεσης (discovery), καταχώρησης (registration), και μεταβίβασης (delivery) για εικόνες δίσκου και server. Οι αποθηκευμένες εικόνες μπορούν να χρησιμοποιηθούν ως πρότυπα (templates). Ακόμα μπορεί να χρησιμοποιηθεί για αποθήκευση και εκχώρηση σε κατάλογο ενός απεριόριστου αριθμού από backups. Η υπηρεσία εικόνας (Image Service) μπορεί να αποθηκεύσει εικόνες δίσκου και server σε ποικίλα back-ends, συμπεριλαμβανομένου του Swift. Το API της υπηρεσίας εικόνας παρέχει μία προκαθορισμένη REST διεπαφή για αναζήτηση πληροφοριών σχετικά με τις εικόνες δίσκου και επιτρέπει στους πελάτες (clients) να μεταδίδουν (stream) εικόνες σε νέους servers[17].

  Ακόμα το Glance προσθέτει πολλές βελτιώσεις στις ήδη υπάρχουσες υποδομές. Για παράδειγμα, εάν ενσωματωθεί με VMware, επιφέρει προχωρημένα χαρακτηριστικά στην

οικογένεια του vShere, όπως vMotion, μεγάλη διαθεσιμότητα και δυναμικό χρονοπρογραμματισμό (scheduling) πόρων (DRS). Το vMotion είναι η ζωντανή μετεγκατάσταση (live migration) μιας λειτουργίας εικονικής μηχανής, από έναν φυσικό server σε άλλον, χωρίς διακοπή υπηρεσίας. Έτσι επιτρέπεται ένα δυναμικό, αυτοβελτιούμενο data-center, επιτρέποντας τη συντήρηση υλικού σε υπολειτουργούντες servers χωρίς καθυστερήσεις[17].

Άλλα τμήματα του OpenStack που χρειάζονται να αλληλεπιδρούν με εικόνες όπως το Heat, επικοινωνούν με τα metadata των εικόνων μέσω του Glance. Επιπλέον, το Nova δύναται να παρουσιάσει πληροφορίες για τις εικόνες και να τροποποιήσει μια παραλλαγή πάνω σε μία εικόνα για να παράξει ένα στιγμιότυπο (instance). Ωστόσο το Glance είναι το μόνο τμήμα που έχει τη δυνατότητα να προσθέτει, διαγράφει, μοιράζεται και να αντιγράφει εικόνες[17].

- **Object storage (Swift)**: Είναι ένα επεκτάσιμο, πλεονάζον (redundant) σύστημα αποθήκευσης. Τα αρχεία και τα αντικείμενα είναι γραμμένα σε πολλαπλούς δίσκους (disk drives), "διασκορπισμένους" σε servers μέσα στο data-center, με το OpenStack λογισμικό να είναι υπεύθυνο να διασφαλίζει την αντιγραφή δεδομένων (data replication) και την ακεραιότητα μέσα στη συστάδα (cluster). Οι συστάδες αποθήκευσης επεκτείνονται "οριζόντια", δηλαδή, απλώς προσθέτοντας νέους servers. Εάν κάποιος server ή σκληρός δίσκος αποτύχει, το OpenStack αντιγράφει τα περιεχόμενά του από άλλους ενεργούς κόμβους σε νέες τοποθεσίες εντός της συστάδας. Λόγω του ότι χρησιμοποιεί λογική λογισμικού για διασφάλιση της αντιγραφής των δεδομένων και για την κατανομή μεταξύ διάφορων συσκευών, μπορεί να γίνει χρήση φθηνών, κοινού τύπου σκληρών δίσκων (hard drives) και servers[17].

- **Dashboard (Horizon)**: παρέχει στους διαχειριστές (administrators) και στους χρήστες μια διεπαφή γραφικών, για πρόσβαση παροχή και αυτοματοποιημένη εγκατάσταση πόρων cloud. Η σχεδίασή του περιέχει προϊόντα και υπηρεσίες "τρίτων" (third party), όπως χρέωση (billing), καταγραφή (monitoring) και διάφορα άλλα εργαλεία διαχείρισης. Αποτελεί έναν από τους δυνατούς τρόπους, ώστε οι χρήστες να μπορούν να αλληλεπιδρούν με τους πόρους του OpenStack. Οι προγραμματιστές (developers) μπορούν με αυτοματοποιημένο τρόπο να προσπελαύνουν και να δημιουργούν εργαλεία διαχείρισης πόρων χρησιμοποιώντας API του OpenStack ή αντίστοιχο του EC2[17].

- **Orchestration (Heat)**: Είναι το βασικό στοιχείο του OpenStack για την ενορχήστρωση πολλαπλών σύνθετων εφαρμογών του cloud που χρησιμοποιούν πρότυπα (templates), μέσω ενός παρεχόμενου, ενδογενές στο OpenStack, REST API και ενός Query API συμβατό για Cloud Formation[17].

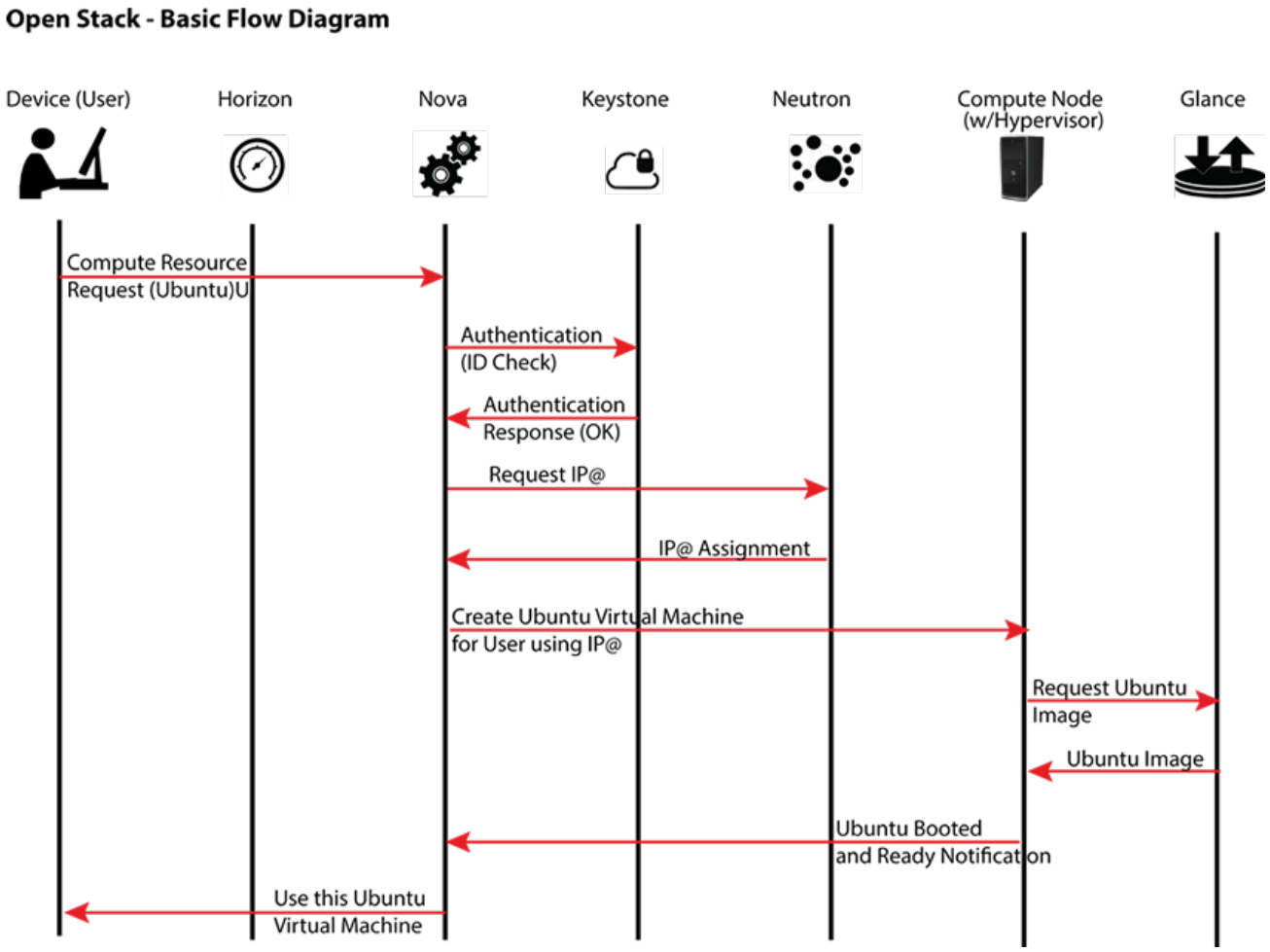
- **Bare-metal (Ironic)**: Αποτελεί ένα ενσωματωμένο πρόγραμμα του OpenStack που προμηθεύει bare-metal μηχανές αντί για εικονικές μηχανές. Πρόκειται στην ουσία για ένα Hypervisor API για bare-metal μαζί με ένα σύνολο από plugins που διαδρούν με bare-metal hypervisors.

3.1.2 Λειτουργία του OpenStack

Ακολουθεί μία περιγραφή της λειτουργίας του OpenStack μέσω ενός ενδεικτικού παραδείγματος, όπου ένας χρήστης θέλει να αποκτήσει μία λειτουργία cloud, όπως είναι μία υπολογιστική υπηρεσία.

Αρχικά ο χρήστης στέλνει ένα αίτημα στο Nova δια μέσω του Horizon που αποτελεί στην ουσία τη διεπαφή με του OpenStack με τον χρήστη. Στη συνέχεια το Nova δημιουργεί μία εικονική μηχανή και πιστοποιεί την αίτηση μέσω του Keystone. Ακολούθως γίνεται αίτηση απόκτησης των απαιτούμενων δικτύων από το Neutron, το οποίο και τα στέλνει πίσω στο Nova. Ύστερα μπορεί να δημιουργηθεί από έναν hypervisor (ο οποίος είναι υπεύθυνος για τον διαμοιρασμό της υλικής υποδομής) μία εικονική μηχανή για την απόδοση ενός tenant. Ο hypervisor στέλνει αίτημα στο Glance για την επιστροφή μιας εικόνας λειτουργικού συστήματος για το εικονικό σύστημα.

Ακολουθεί μία εικόνα που αποσαφηνίζει τη λειτουργία που περιγράφηκε.



Σχήμα 12: Ενδεικτική λειτουργία του OpenStack [80]

3.1.3 OpenStack ως μέρος του CORD

Όπως γίνεται κατανοητό το OpenStack αποτελεί ένα βασικό εργαλείο για διαχείριση των πόρων που προσφέρει η υποκείμενη υλική υποδομή σε ένα cloud σύστημα, με την έννοια της υποδομής ως υπηρεσία (Infrastructure-as-a-Service – IaaS). Καθώς το CORD αποτελεί μία υλοποίηση σε cloud, υιοθετεί το OpenStack ως ένα στοιχείο του, το οποίο εξυπηρετεί στη δημιουργία εικονικών μηχανών και εικονικών δικτύων, καθώς και στη διαχείριση των προσφερόμενων πόρων τους.

3.2 Docker και containerization

Το Docker είναι ένα λογισμικό που σκοπός του είναι η παροχή του virtualization με χρήση containers σε επίπεδο λειτουργικού συστήματος, γνωστή και ως “containerization”.

Το Docker λειτουργεί βάσει τμημάτων λογισμικού που τρέχει και ονομάζονται containers. Το κάθε container δεσμεύει τη δική του εφαρμογή (application), εργαλεία, βιβλιοθήκες και αρχεία τροποποίησης. Όλα τα containers εκτελούνται από έναν πυρήνα λειτουργικού συστήματος και έτσι μπορούν να αποδοθούν ως μια ελαφριά εκδοχή, εικονικών μηχανών. Είναι απομονωμένα μεταξύ τους και δημιουργούνται από εικόνες (images) που προσδιορίζουν τα ακριβή περιεχόμενά τους.

Το Docker είναι σχεδιασμένο έτσι ώστε να προσφέρει ένα γρήγορο και ελαφρύ περιβάλλον, όπου ο κώδικας μπορεί να τρέχει αποτελεσματικά, και ακόμα παρέχει έναν επιπλέον μηχανισμό ώστε να παίρνει τον κώδικα από τον υπολόγιστή και να τον εξετάζει προτού τον θέση σε παραγωγή[81].

3.2.1 Τεχνολογία του Docker

Το Docker χρησιμοποιεί χαρακτηριστικά απομόνωσης πόρων του πυρήνα (kernel) του Linux, όπως τα cgroups και οι χώροι ονομάτων (namespaces). Επιπλέον χρησιμοποιεί σύστημα αρχείων ενοποίησης (union capable file system), έτσι ώστε να υπάρχει η δυνατότητα, ανεξάρτητα containers να μπορούν να τρέχουν σε ένα μόνο Linux στιγμιότυπο (instance), αποφεύγοντας έτσι το μεγάλο επιπρόσθετο φορτίο (overhead) που επιφέρει η εκκίνηση και διατήρηση των εικονικών μηχανών, καθώς κάθε VM εκτελεί ολόκληρο δικό του λειτουργικό σύστημα. Τα namespaces απομονώνουν την όψη της εφαρμογής (application view) από το περιβάλλον λειτουργίας (operating environment), όπως τα δένδρα επεξεργασίας (process trees), το δίκτυο (network), τα IDs χρήστη, και τα προσαρτημένα συστήματα αρχείων (mounted file systems). Τα cgroups του kernel από την άλλη, παρέχουν όρια στη χρήση πόρων, όπως τη μνήμης και τη CPU[18].

Το container, όπως είδαμε, δημιουργείται πάνω σε παροχές του Linux kernel, όπως τα cgroups και namespaces, εν αντιθέσει με μία εικονική μηχανή που απαιτεί ένα ξεχωριστό λειτουργικό σύστημα. Βασίζεται στη λειτουργικότητα του kernel, χρησιμοποιώντας απομόνωση πόρων (resource isolation) για CPU και μνήμη, και διαχωρίζει τους χώρους ονομάτων, ώστε να

απομονώσει την όψη του λειτουργικού συστήματος προς την εφαρμογή. Το Docker χρησιμοποιεί virtualization του Linux kernel, είτε άμεσα μέσω μιας βιβλιοθήκης libcontainer που κατέχει, είτε έμμεσα μέσω των libvirt, LXC (Linux containers) και system-nspawn[18].

3.2.2 Δομικά στοιχεία του Docker

- **Λογισμικό (Software)**: Το dockerd (Docker deamon), είναι μια διεργασία που διαχειρίζεται τα containers και τα αντικείμενα των containers. Επίσης περιμένει αιτήσεις που στέλνονται από το Docker EngineAPI. Το πρόγραμμα πελάτη του Docker ονομάζεται απλά docker και περιέχει μία διεπαφή τύπου command line που επιτρέπει την αλληλεπίδραση των χρηστών με τα dockerd[18].

- **Αντικείμενα (Objects)**: πρόκειται για διάφορες οντότητες που συναθροίζονται για να παράξουν μία εφαρμογή στο Docker. Οι κύριες κατηγορίες Docker αντικειμένων είναι[18]:

  - Docker container: ένα καθορισμένο περιβάλλον που τρέχει εφαρμογές και διαχειρίζεται μέσω Docker API ή CLI (Command Line Interaction).

  - Docker image: είναι ένα read-only πρότυπο (template) που χρησιμοποιείται για οικοδόμηση containers. Χρησιμοποιούνται για αποθήκευση (store) και μεταφορά (ship) των εφαρμογών.

  - Docker service: επιτρέπει στα containers να επεκτείνονται μεταξύ πολλαπλών deamons, με αποτέλεσμα ένα σύνολο από συνεργαζόμενα deamons να επικοινωνούν μέσω Docker API.

  - Registries: αποτελούν στην ουσία αποθήκες για Docker images. Από εκεί οι πελάτες (clients) παίρνουν έτοιμες κτισμένες εικόνες. Υπάρχουν δημόσια (public) registries, όπως τα Docker Hub (προκαθορισμένο) και Docker Cloud, αλλα και ιδιωτικά.

3.2.3 Εργαλεία του Docker

Docker Compose: Ορίζει και εκτελεί εφαρμογές πολλαπλών containers. Μέσω YAML αρχείων ορίζει και τροποποιεί υπηρεσίες εφαρμογής και δημιουργεί και εκκινεί όλα τα εμπλεκόμενα containers με μία εντολή. Το docker-compose CLI επιτρέπει στους χρήστες να εκτελούν εντολές σε πολλαπλά containers, όπως δημιουργία εικόνων, επέκταση των containers που έχουν σταματήσει και άλλα. Οι εντολές που αφορούν τη χειραγώγηση εικόνων (image manipulation), ή τις επιλογές αλληλεπίδρασης χρήστη, δεν σχετίζονται με το Docker Compose, διότι απευθύνονται σε ένα container[18].

Docker Swarm: Παρέχει εντός συστάδας (native cluster) λειτουργικότητα για τα Docker containers, τα οποία μετατρέπουν μία ομάδα από μηχανές Docker (Docker engines) σε μία εικονική

μηχανή Docker (virtual Docker engine). Το swarm CLI επιτρέπει στους χρήστες να τρέχουν swarm containers (containers σμήνους), να δημιουργούν discovery tokens, να τοποθετούν σε λίστα τους κόμβους της συστάδας, και άλλα. Το docker node CLI επιτρέπει στους χρήστες να τρέχουν εντολές διαχείρισης κόμβων εντός του σμήνους (swarm), όπως τοποθέτηση σε λίστα των κόμβων μέσα στο swarm, ενημέρωση κόμβων και αφαίρεση κόμβων από το swarm[18].

3.2.4 Λειτουργία του Docker

Το Docker προσφέρει ένα API υψηλού επιπέδου για παροχή containers τα οποία εκτελούνται απομονωμένα[18].

Είναι ένα εργαλείο το οποίο πακετάρει μία εφαρμογή καθώς και τις εξαρτήσεις της μέσα σε ένα εικονικό container, το οποίο δύναται να τρέχει σε οποιονδήποτε Linux server. Έτσι προάγεται ευελιξία (flexibility), φορητότητα (portability) για το που μπορεί να εκτελείται η εφαρμογή (π.χ. public/private cloud, bare metal premises).[18]

Η χρήση containers απλοποιεί τη δημιουργία εκτενώς κατανεμημένων συστημάτων (highly distributed systems) επιτρέποντας σε πολλαπλές εφαρμογές να τρέχουν αυτόνομα πάνω σε μία φυσική μηχανή ή μεταξύ πολλαπλών εικονικών μηχανών. Αυτό επιτρέπει την εγκατάσταση των κόμβων, όταν οι πόροι γίνονται διαθέσιμοι ή όταν απαιτούνται περισσότεροι κόμβοι, αναδεικνύοντας μία PaaS (Platform-as-a-Service) υλοποίηση.[18]

3.2.5 Docker και CORD

Συνοψίζοντας, το Docker αποτελεί ένα μέσο για παροχή containers, πάνω στα οποία εκτελούνται οι υπηρεσίες, αλλά και τα δομικά εργαλεία του CORD, όπως το XOS, OpenStack και το ONOS, τα οποία δημιουργούνται από το Docker πάνω στα POD head nodes (POD επικεφαλής κόμβοι).

Σημαντικά πλεονεκτήματα για την εφαρμογή του Docker ως μέρος του CORD είναι[81]:

- Ταχύτητα: Από τα πιο σημαντικά πλεονεκτήματα του Docker. Ο χρόνος που απαιτείται για την εγκατάσταση ενός container αλλά και για την διαχείρησή του και εκτέλεσή του είναι πολύ μικρός.

- Φορητότητα: Η εγκατάσταση των εφαρμογών μέσα στα containers μας δίνει φορητότητα καθως μπορούμε να τα μετακινήσουμε, χωρίς επιπτώσεις στην απόδοση.

- Επεκτασιμότητα: Το Docker μπορεί να εγκατασταθεί σε πολλαπλούς φυσικούς servers, data servers και σε πλατφόρμες νέφους. Ακόμα έχει τη δυνατότητα να τρέχει σε οποιοδήποτε Linux σύστημα. Τα containers μεταφέρονται εύκολα και γρήγορα από ένα περιβάλλον νέφους σε local host και το αντίστροφο. Προσαρμογές που αφορούν την επέκταση γίνονται απλά από τον χρήστη βάσει των αναγκών του.

- Γρήγορη διανομή (rapid delievery): Το Docker παρέχει ένα αξιόπιστο, συνεπές και βελτιωμένο περιβάλλον, ώστε τα προβλέψιμα αποτελέσματα να μπορούν να επιτευχθούν όταν ο κώδικας μεταφέρεται μεταξύ συστημάτων ανάπτυξης, εξέτασης και παραγωγής.

- Πυκνότητα: Το Docker χρησιμοποιεί τους διαθέσιμος πόρους πιο αποδοτικά, καθώς δεν χρησιμοποιεί hypervisor. Έτσι συγκριτικά με τα VMs, περισσοτερα containers μπορούν να τρέξουν σε έναν host. Η απόδοση του Docker είναι υψηλότερη, λόγω της υψηλής πυκνότητας και της έλλειψης overhead από σπατάλες πόρων.

3.2.6 Vagrant

Είναι ένα ανοιχτού κώδικα λογισμικό για δημιουργία και συντήρηση φορητών (portable) εικονικών περιβαλλόντων εφαρμοσμένα σε υλικό, όπως σε VirtualBox, Hyper-v, Docker containers, VMware και AWS, το οποίο στοχεύει στην απλοποίηση της διαχείρισης λογισμικού (software configuration management) και του virtualization ώστε να αυξήσει την παραγωγικότητα ανάπτυξης (development productivity). Είναι γραμμένο σε Ruby γλώσσα, αλλά το οικοσύστημα υποστηρίζει προγραμματισμό σε αρκετές γλώσσες[19].

Χρησιμοποιείται ως έκδοση του Docker, κυρίως για εικονικά περιβάλλοντα υλοποίησης (π.χ. CORD-in-a-Box).

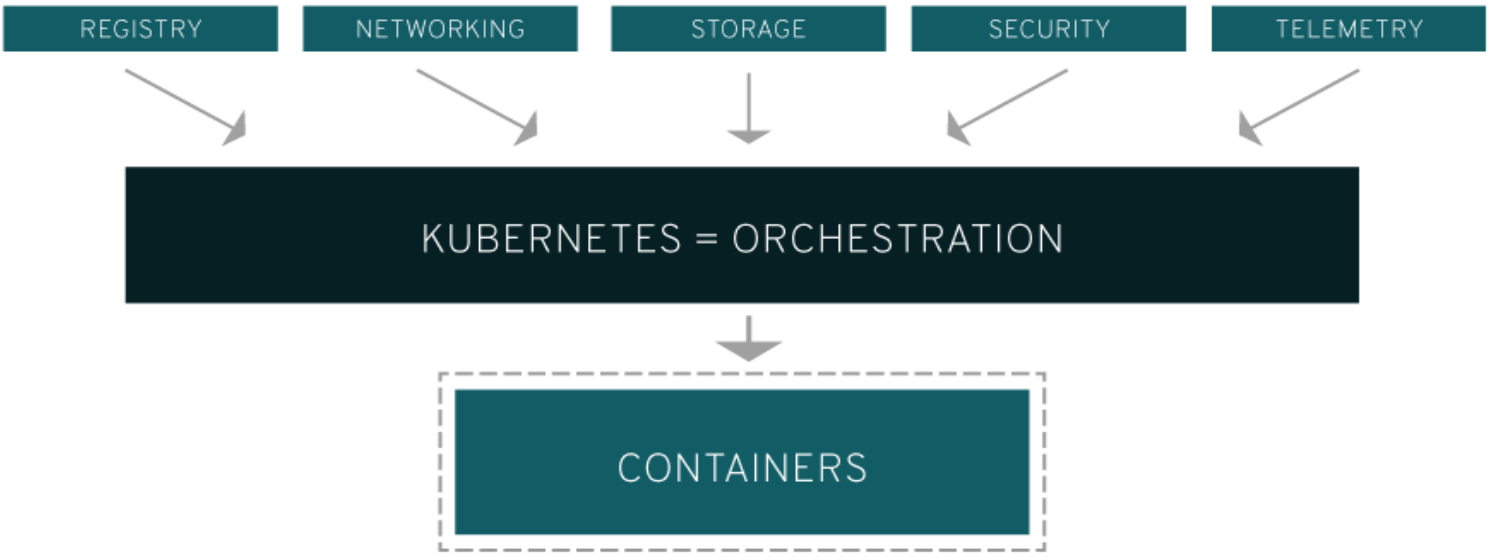
3.2.7 Kubernetes (K8)

Θα γίνει μια σύντομη αναφορά στο Kubernetes, το οποίο είναι ένα νέο σύστημα διαχείρισης των Linux containers, που υιοθετείται και από το CORD. Συνδυάζεται με το Docker και παρέχει αποδοτική εφαρμογή και λειτουργία των containers και κατ' επέκτασιν των υπηρεσιών που χτίζονται πάνω σε αυτά.

Δίνει τη δυνατότητα για καλύτερη εκμετάλλευση υπολογιστικής ισχύος για εκτέλεση εφαρμογών λογισμικού. Αυτοματοποιεί την εγκατάσταση (deployment), τον χρονοπρογραμματισμό (scheduling) και τη λειτουργία των container εφαρμογών σε μεγάλες συστάδες από μηχανές, όπως είναι στις cloud υποδομές. Επίσης, προσφέρει τη δυνατότητα στους προγραμματιστές να δημιουργούν περιβάλλοντα με βάση τα containers, με χρήση container εικόνων (container images) εγκατεστημένων πάνω σε kubernetes ή ενσωματωμένα σε ένα σύστημα συνεχούς ενσωμάτωσης (CI – system)[20].

Η λειτουργία του και η χρήση του μπορεί να συνδυαστεί και με άλλες πλατφόρμες παροχής IaaS και PaaS όπως εν προκειμένω με το Docker.

Το Kubernetes ενσωματώνει διάφορες υπηρεσίες για την εξαγωγή ενός λειτουργικού container όπως φαίνεται στην παρακάτω εικόνα.



Σχήμα 13: Οργάνωση πόρων και λειτουργιών για την παραγωγή containers μέσω του kubernetes [26]

Ακόμα δίνει τη δυνατότητα της οργάνωσης των διαρκώς αυξανόμενων containers, σε ομάδες που καλούνται Pod.

Συνοπτικά οι δυνατότητες που προσφέρει το Kubernetes [21]:

- Οργανώνει τα containers τα οποία μπορεί να εκτελούνται μεταξύ πολλαπλών hosts
- Καλύτερη διαχείριση των υποκείμενων υλικών πόρων
- Αυτοματοποιεί και ελέγχει την εγκατάσταση και ενημέρωση των εφαρμογών
- Κάνει διαχείριση αποθηκευτικού χώρου για τις εφαρμογές
- Επεκτείνει (scale) τις κεντρικοποιημένες εφαρμογές και τους πόρους τους
- Διαχείριση υπηρεσιών για σωστή εκτέλεση των εφαρμογών

3.3 ONOS (Open Network Operating System)

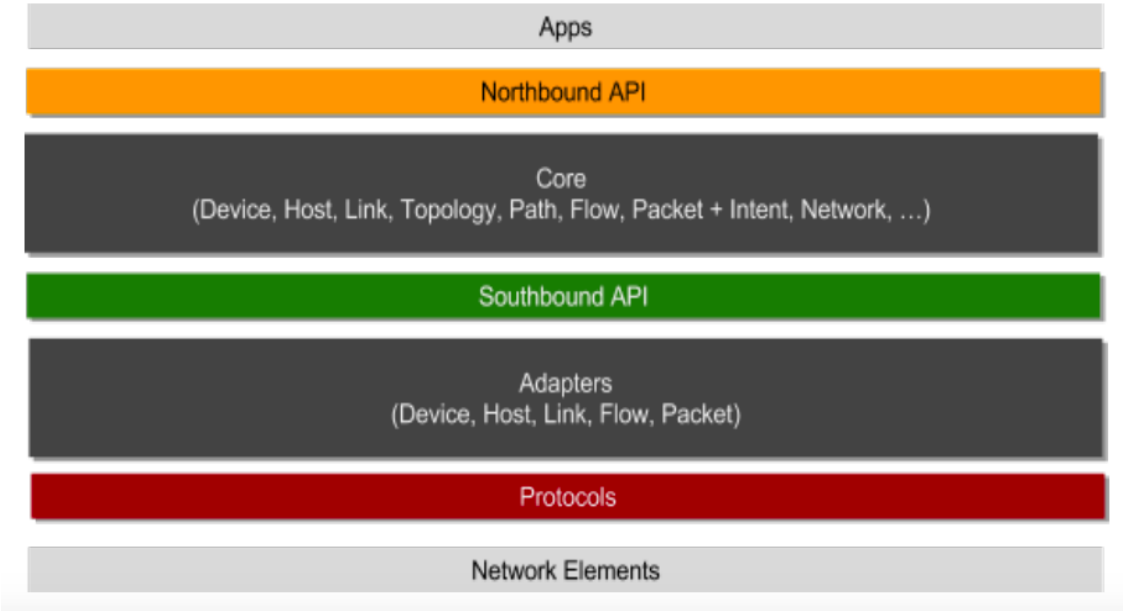
Το ONOS είναι μία SDN controller πλατφόρμα η οποία στοχεύει στην εξυπηρέτηση των σύγχρονων παρόχων υπηρεσιών (service providers), παρέχοντας υψηλή διαθεσιμότητα,

επεκτασιμότητα και υψηλή απόδοση στο SDN control plane. Ακόμα επιτρέπει την μετεγκατάσταση των δικτύων τους σε υποδομή με μηχανήματα γενικού σκοπού (white boxes) όπως στα data-centers, καθώς και την μείωση του CAPEX και OPEX.

Όπως δηλώνει και το όνομά του αποτελεί ένα είδος λειτουργικού συστήματος για τα δίκτυα, το οποίο διαχειρίζεται τους δικτυακούς πόρους (network resources) και παρέχει τις κατάλληλες αφαιρέσεις και διεπαφές, επιτρέποντας έτσι τη διαχείριση και τον προγραμματισμό των συσκευών και τη δημιουργία χρήσιμων και καινοτόμων δικτυακών εφαρμογών πάνω από αυτές, αποκρύπτοντας την πολυπλοκότητα της υποκείμενης υποδομής υλικού.

Το ONOS αποδίδει προς τα πάνω northbound αφαιρέσεις και APIs για ευκολία στην ανάπτυξη εφαρμογών και κανόνων από τους χρήστες και τους διαχειριστές, και αντίστοιχα προς τα κάτω southbound αφαιρέσεις και διεπαφές που να επιτρέπουν τον έλεγχο στις συσκευές, μέσω του OpenFlow, για την εκτέλεση της επιθυμητής δρομολογησης.

3.3.1 Αρχιτεκτονική του ONOS



Σχήμα 14: Τα επίπεδα της αρχιτεκτονικής του ONOS [22]

Η αρχιτεκτονική του ONOS χαρακτηρίζεται από τα εξείς βασικά μέρη[22]:

- **Distributed core (κατανεμημένος πυρήνας)**

  Το ONOS εφαρμόζεται κατανεμημένα πάνω σε μία συστάδα από servers, με τον καθένα να εκτελεί το ίδιο λογισμικό του ONOS. Κάθε στιγμιότυπο του ONOS

συνεργάζεται με τα υπόλοιπα και έτσι παρουσιάζονται εξωτερικά ως μία ενιαία πλατφόρμα. Αυτή η αφαιρετική όψη, προσφέρει τη δυνατότητα διάφανης επεκτασιμότητας σε περίπτωση που απαιτηθούν περισσότεροι πόροι (όπως CPU, memory). Τον συντονισμό και την κεντρική διαχείριση των στιγμιοτύπων τα αναλαμβάνει ο κατανεμημένος πυρήνας[22].

Ο συντονισμός των ONOS στιγμιοτύπων παρέχεται από το distributed core μέσω λειτουργιών, όπως είναι τα μηνύματα μεταξύ των στιγμιοτύπων, η διαχείριση κατάστασης και η εκλογή μοναδικού ηγέτη (leader election) - στιγμιότυπο για τη συστάδα (cluster). Έτσι οι πιθανές ενημερώσεις (updates) μεταδίδονται στα στιγμιότυπα μέσω ταχέων μηνυμάτων. Επίσης χρησιμοποιούνται πρωτόκολλα ανάκαμψης (recovery protocols) για τη σωστή ενημέρωση σε περίπτωση σφάλματος σε κάποιο instance. Όσον αφορά τις καταστάσεις διαχείρισης των στιγμιοτύπων, ποικίλουν για διάφορες περιπτώσεις, όπως είναι: εφαρμογές, η βάση δεδομένων της τοπολογίας, και οι πίνακες ροής (flow tables). Όλες αυτές οι λειτουργίες συντονισμού αποφέρουν υψηλή ρυθμαπόδοση (throughput), χαμηλή καθυστέρηση μετάδοσης (latency) και υψηλή διαθεσιμότητα πόρων[22].

- **Northbound Abstractions (Northbound αφαιρέσεις)**

Υπάρχουν δύο βασικές northbound αφαιρέσεις: το πλαίσιο πρόθεσης (Intent Framework) και η καθολική όψη δικτύου (Global Network View)[22].

Το πλαίσιο πρόθεσης (Intent Framework) αναφέρεται στην αφαίρεση που προσφέρεται στις εφαρμογές όταν αυτές ζητούν υπηρεσίες. Αποκρύπτει τις λεπτομέρειες πίσω από τις υπηρεσίες, προσφέροντας έτσι τη δυνατότητα υψηλού προγραμματισμού του δικτύου στους προγραμματιστές εφαρμογών και στους λειτουργούς δικτύων (network operators). Έτσι το μόνο που χρειάζεται να κάνουν είναι να θέσουν τις απαιτήσεις τους για το δίκτυο, όπως είναι ο ορισμός σύνδεσης μεταξύ συγκεκριμένων hosts, το είδος σύνδεσης και διάφοροι κανόνες και περιορισμούς πάνω σε αυτά[22].

Η καθολική όψη δικτύου (Global Network View) αναφέρεται στην παρουσίαση μιας όψης του δικτύου στις εφαρμογές. Αυτή η όψη παρουσιάζει τα συστατικά του δικτύου όπως, switches, links και hosts, ως γράφο, επιτρέποντας έτσι τον εύκολο προγραμματισμό του δικτύου στις εφαρμογές μέσω χρήσης APIs. Ο προγραμματισμός του δικτύου εδώ αφορά λειτουργίες όπως η επιθυμητή δρομολόγηση, η καταγραφή της απόδοσης του δικτύου και τροποποιήσεις των paths (μονοπατιών) για βελτιστοποίηση, καθώς και διάφορους άλλους κανόνες και πολιτικές που αφορούν την κίνηση προς και εντός του δικτύου. Έτσι αποκρύπτει λεπτομέρειες του δικτύου που δεν αφορούν τις εφαρμογές, καθώς επίσης απομονώνει τις εφαρμογές από το λειτουργικό σύστημα, αφήνοντάς το να ασχοληθεί με τις λειτουργίες που το αφορούν, όπως είναι η διαχείριση πολλαπλών εφαρμογών[22].

- **Southbound abstractions (Southbound αφαιρέσεις)**

Με τις southbound αφαιρέσεις αναφερόμαστε στην υποκείμενη υποδομή του δικτύου και πως παρουσιάζονται με αφαιρετικό τρόπο προς τα επάνω, στον πυρήνα του ONOS. Έτσι οι διάφορες συσκευές και στοιχεία της υποκείμενης υποδομής (όπως είναι τα switches, οι hosts και τα links), παρουσιάζονται μέσω πρωτοκόλλων και ενός API προσαρμογέα (adapter API), με αφαιρετικό τρόπο ως ενός είδους αντικείμενα, αποκρύπτοντας τις λεπτομέρειες και ιδιαιτερότητες των διαφόρων αυτών στοιχείων. Με αυτόν τον τρόπο

γίνεται ευκολότερη διαχείριση διαφορετικών συσκευών με πιθανώς διαφορετικά πρωτόκολλα, η προσθήκη νέων συσκευών καθώς και η "μετανάστευση" (migration) των παραδοσιακών συσκευών και πρωτοκόλλων σε white boxes που χρησιμοποιούν OpenFlow[22].

- **Software modularity (Δομή στοιχείων λογισμικού)**

  Εδώ γίνεται αναφορά σε όλα τα δομικά στοιχεία λογισμικού που παράγονται και εκτελούνται ώστε να λειτουργεί συνολικά το ONOS με τις αφαιρέσεις που αναφέραμε. Τέτοια στοιχεία αποτελούν οι διεπαφές (northbound/southbound APIs) με τις παραγόμενες εφαρμογές, καθώς όμως και λογισμικά στοιχεία μέσα στον πυρήνα του ONOS και οι μηχανισμοί για την σωστή οργάνωση, επικοινωνία, και συντονισμό όλων αυτών[22].

Συνεχίζουμε εστιάζοντας πιο συγκεκριμένα στα συστατικά μέρη που είναι απαραίτητα για τη διεκπεραίωση λειτουργιών του ONOS.

3.3.2 Συστατικά συστήματος του ONOS

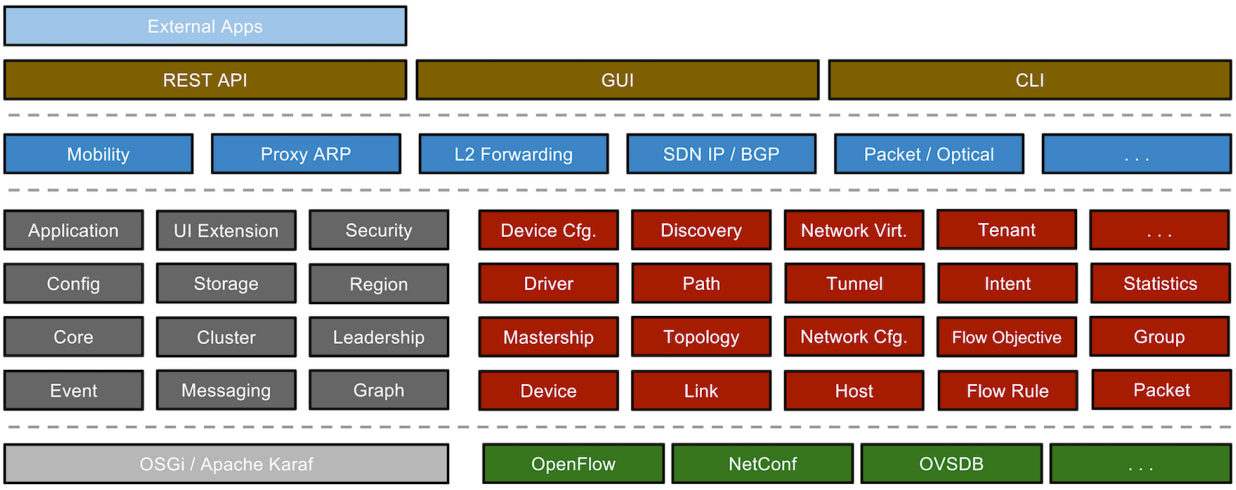
3.3.2.1 Υπηρεσίες και υποσυστήματα (services and subsystems)

Μία **υπηρεσία** (service) στο ONOS καλείται μία λειτουργία που αποτελείται από μία σειρά συστατικών στοιχείων (components) από κάθε επίπεδο της αρχιτεκτονικής του ONOS όπως παρουσιάστηκε προηγουμένως.[25]

Ένα **υποσύστημα** (subsystem) καλείται το σύνολο όλων αυτών των συστατικών που χρησιμοποιούνται για τη δημιουργία και την εκτέλεση της υπηρεσίας.[25]

Μερικές βασικές υπηρεσίες του ONOS είναι οι εξείς[25]:

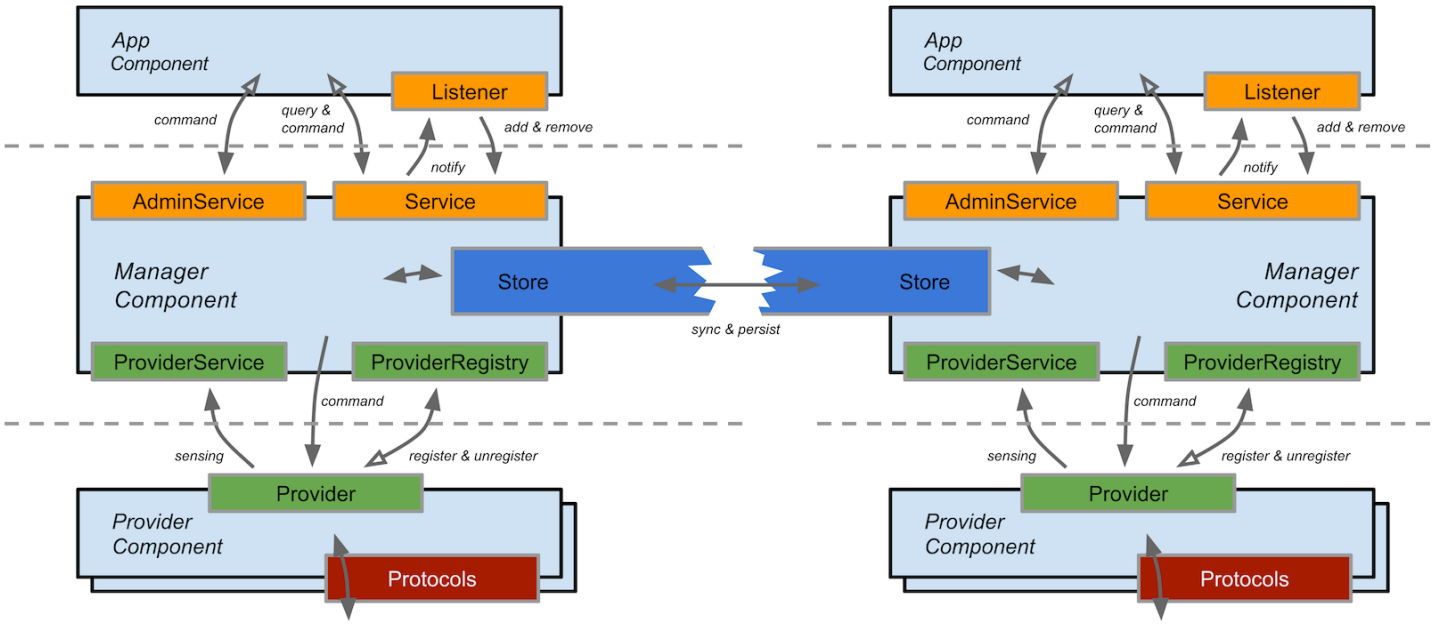
- Υποσύστημα συσκευής (device subsystem)
- Υποσύστημα σύνδεσης (link subsystem)
- Υποσύστημα host (host subsystem)
- Υποσύστημα τοπολογίας (topology subsystem)
- Υπηρεσία διαδρομής (path service)
- Υποσύστημα κανόνων ροής (flowRule Subsystem)
- Υποσύστημα πακέτων (packet subsystem)



Σχήμα 15: Δομικά στοιχεία – υποσυστήματα του ONOS [25]

3.3.2.2 Δομή Υποσυστήματος

Ακολουθούν τα βασικά συστατικά ενός υποσυστήματος, χωρίς ωστόσο να σημαίνει ότι κάθε υποσύστημα χρησιμοποιεί όλα αυτά τα συστατικά.[25]



Σχήμα 16: Βασικά συστατικά υποσυστήματος του ONOS [25]

- **Provider (πάροχος)**: Βρίσκεται στο κατώτερο επίπεδο του ONOS και μέσω χρήσης βιβλιοθηκών πρωτοκόλλου, ενεργεί σαν διεπαφή του υποκείμενου δικτύου με τον πυρήνα. Συλλέγει επίσης δεδομένα από άλλα υποσυστήματα και τα μετατρέπει σε συγκεκριμένα δεδομένα για χρήση από την υπηρεσία. Κάθε provider κατέχει μία ταυτότητα (providerID) για να ξεχωρίζει και να αναγνωρίζεται από τους υπόλοιπους providers διάφορων υπηρεσιών. Ένα υποσύστημα μπορεί να παρέχει διάφορους παρόχους για διαφορετικές λειτουργίες/απαιτήσεις[25].

- **Manager (διαχειριστής)**: Βρίσκεται στον πυρήνα και αντλεί πληροφορίες από τους providers, ώστε να εξυπηρετήσει τις εφαρμογές. Λειτουργεί με διάφορες διεπαφές όπως είναι τα: northbound Service interface, AdminService interface, southbound ProviderRegistry interface, southbound ProviderService interface. Οι πληροφορίες που λαμβάνονται μέσω των διεπαφών του manager από την έκαστη ενδιαφερόμενη λειτουργία, γίνεται είτε σύγχρονα εξετάζοντας την υπηρεσία, είτε ασύγχρονα ως παθητικός δέκτης γεγονότος (event listener) μέσω μιας διεπαφής υπηρεσίας που λέγεται ListenerService interface[25].

- **Store (αποθήκη)**: Βρίσκεται επίσης στον πυρήνα και σκοπός του είναι να κατατάσσει σε πίνακες (indexing), να διατηρεί και να συγχρονίζει τις πληροφορίες που λαμβάνει από τον διαχειριστή (manager)[25].

- **Application (εφαρμογή)**: Οι εφαρμογές βρίσκονται στην κορυφή της αρχιτεκτονικής και είναι στην ουσία το παραγωγικό αποτέλεσμα των υποκείμενων στοιχείων, από τα οποία δέχεται τις απαιτούμενες πληροφορίες μέσω του διαχειριστή (manager), που τις μεταβιβάζει στις εφαρμογές χρησιμοποιώντας διεπαφές όπως είναι το AdminService interface και το Service interface. Οι εφαρμογές ταυτοποιούνται και αναγνωρίζονται μέσω ID (ApplicationId) που τους ανατίθενται, ώστε να λαμβάνουν τα σωστά περιεχόμενα πληροφορίας[25].

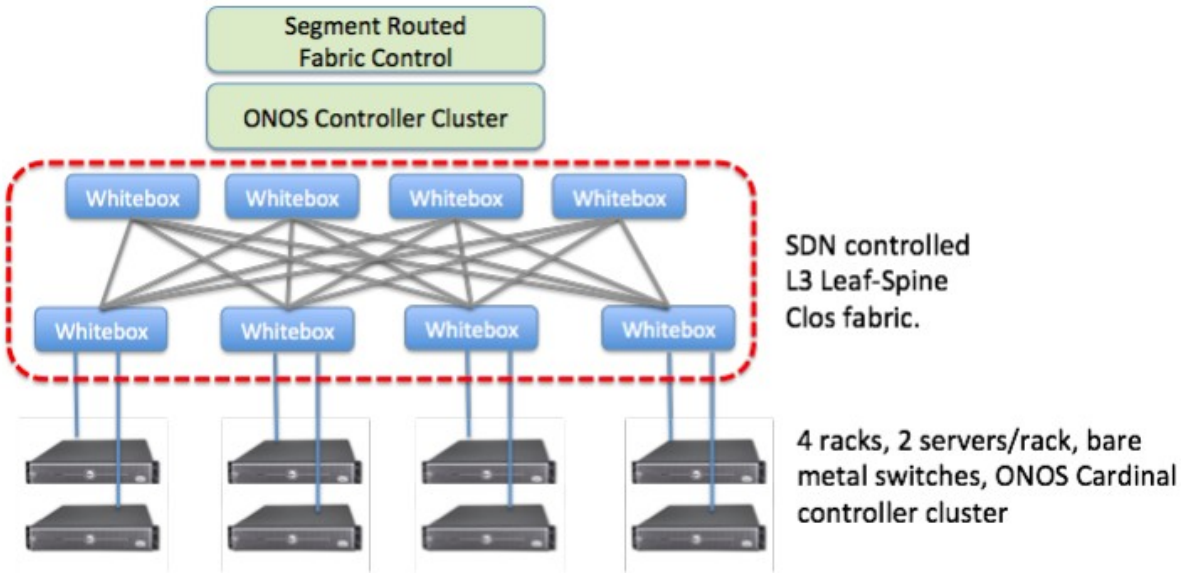
3.3.2.3 Events and Descriptions

Descriptions (περιγραφές): Ο ρόλος των περιγραφών είναι να περνούν πληροφορίες των στοιχείων του υποκείμενου δικτύου προς τα επάνω, μέσου του southbound, και αφορούν διάφορες χρήσιμες πληροφορίες όπως, τις IP και MAC διευθύνσεις ή την τοποθεσία μέσα στο δίκτυο του κάθε στοιχείου[25].

Events (γεγονότα): Μέσω των events οι διαχειριστές (Managers) ενημερώνουν τους listeners τους, καθώς και τα Stores με σκοπό την ειδοποίηση των ομοτίμων τους σε μία κατανεμημένη διάταξη. Για παράδειγμα ένα DeviceEvent ειδοποιεί DeviceListeners για ενημέρωση σχετικά με κάποια αλλαγή συσκευής (προσθήκη/αφαίρεση/ ενημέρωση)[25].

3.3.3 Εφαρμογή του ONOS στο CORD

Το ONOS αποτελεί ένα σημαντικό κομμάτι του CORD καθώς είναι υπεύθυνο για τη λογική διαχείριση της white-box δομής και είναι το μέσο που επιφέρει την SDN λειτουργικότητα και ευελιξία πάνω στην υποκείμενη υποδομή. Παρέχει εφαρμογές ελέγχου των υπηρεσιών και των εικονικών λειτουργιών, την εγκατάσταση εικονικών δικτύων και διαχείριση της ροής σε αυτά. Ακόμα αποτελεί το μέρος στο οποίο εκτελούνται προγράμματα ελέγχου που εφαρμόζουν σημαντικές υπηρεσίες του CORD όπως είναι το vOLT και το vRouter. Ο ρόλος του θα γίνει ακόμα πιο κατανοητός παρακάτω καθώς θα περιγραφούν τα υπόλοιπα εμπλεκόμενα μέρη και υπηρεσίες του CORD.



Σχήμα 17: SDN έλεγχος μέσω του ONOS σε spine-leaf τοπολογία, εφαρμοσμένη από white-boxes [71]

3.4 XOS

Το XOS είναι ένα είδος λειτουργικού συστήματος που εστιάζει στη διαχείριση και λειτουργία συστημάτων τεχνολογίας νέφους πολλαπλών επιπέδων (multi-tiered cloud), και βασίζεται στη λογική των υπηρεσιών. Θεωρεί τις υπηρεσίες λογισμικού ως αντικείμενα πρώτης τάξης (first-class objects), αποδίδοντας έτσι τη δυνατότητα για αύξηση του επιπέδου αφαίρεσης μιας IaaS (Infrastructure-as-a-Service) αρχιτεκτονικής νέφους.

Αυτό επιτυγχάνεται μέσω των τριών αρχών:[26]

- Τα πάντα ως υπηρεσία (Everything-as-a-Service – XaaS), όπου οι υπηρεσίες είναι δομικές μονάδες, και ο συνδυασμός αυτών των μονάδων συνιστά επίσης υπηρεσία.

- Αρχιτεκτονική πολλαπλών ενοίκων (multi-tenancy).
- Διαχωρισμός του control-plane από το data-plane.

Το XOS ορίζει ένα σύνολο από αφαιρέσεις (abstractions), που υποστηρίζουν την κατασκευή multi-tenant υπηρεσιών, που μπορούν να ενσωματωθούν πίσω στο XOS ως διαθέσιμες δομικές μονάδες, καθώς επίσης επεκτείνουν τις δυνατότητες του IaaS.

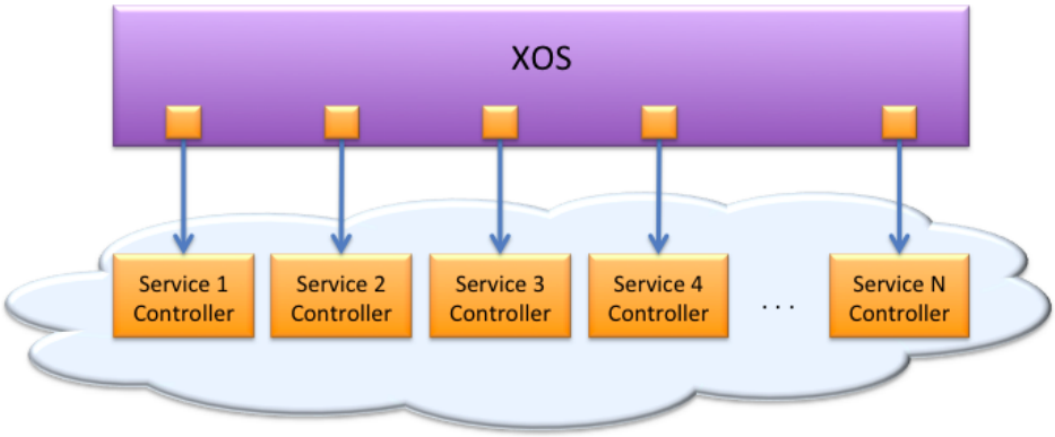
Οι αφαιρέσεις του ενοποιούν την πρόσβαση στην υποδομή του cloud μεταξύ πολλαπλών θέσεων (sites) από παραδοσιακά clouds εως ιδιωτικά data-centers και δίκτυα ευρείας περιοχής (WAN), κέντρα δρομολόγησης και άκρα πρόσβασης (edge access sites). Το XOS επεκτείνει τα υπάρχοντα συστήματα διαχείρισης cloud - datacenter εφαρμόζοντας χρονοπρογραμματισμό χαμηλού επιπέδου και μηχανισμούς διευθέτησης πόρων για κάθε αυτόνομο σύστημα – site, καθώς και τροποποίηση σε SDN για τη σύνδεση των sites.[26]

Το XOS παρέχει στήριξη για multi-tenant υπηρεσίες, καθιστώντας δυνατόν να δημιουργεί, διαχειρίζεται και να συνθέτει υπηρεσίες ως πρώτης τάξης λειτουργίες. Τα συμβατικά multi-tenant clouds είναι σχεδιασμένα να φιλοξενούν εφαρμογές, μεταχειρίζοντας ‘τες ως single-tenant υπηρεσίες που τρέχουν στην κορυφή του cloud, προς όφελος του χρήστη. Αντιθέτως το XOS παρέχει ένα framework για την εφαρμογή multi-tenant υπηρεσιών που γίνονται μέρος του cloud, μειώνοντας έτσι το φράγμα για τις υπηρεσίες να χτίζονται η μία πάνω στην άλλη.[26]

3.4.1 Το XOS ως λειτουργικό σύστημα

Όπως το Unix είναι ένα λειτουργικό σύστημα για υπολογιστές, αντίστοιχα το XOS ακολουθεί την ίδια φιλοσοφία προσανατολισμένη στο cloud.

Στο Unix η βασική ιδέα είναι ότι όλα λαμβάνονται ως αρχεία, ενώ αντίστοιχα στο XOS όλα είναι υπηρεσίες (Everything-as-a-Service – XaaS). Και τα δύο συστήματα αποσκοπούν στο να έχουν έναν περιορισμένο (minimal) πυρήνα (kernel) και να είναι εύκολα επεκτάσιμα σε νέες λειτουργίες. Έτσι στο XOS ο πυρήνας είναι λιτός και η ενδιαφέρουσα λειτουργικότητα παρέχεται από μία συλλογή από υπηρεσίες.



Σχήμα 18: Το XOS ως λειτουργικό σύστημα πάνω σε ένα σύνολο υπηρεσιών (ελεγκτών υπηρεσιών) [26]

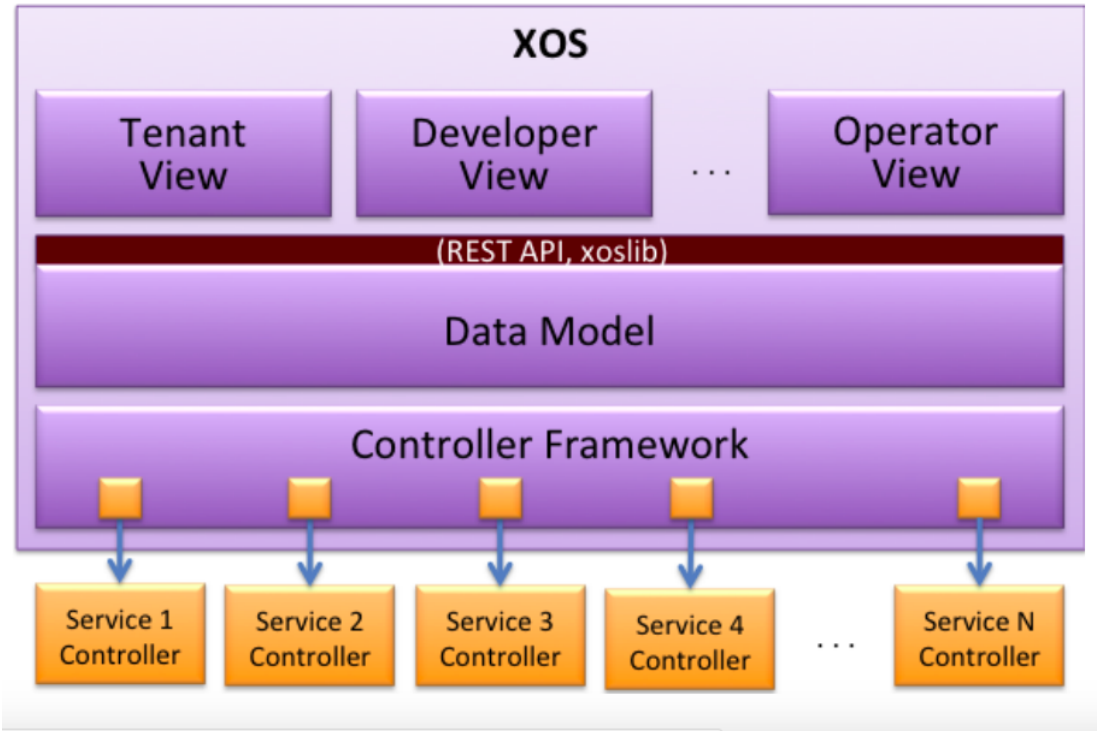
Κάθε υπηρεσία ενσωματωμένη στο XOS παρέχει έναν ελεγκτή υπηρεσίας (service controller) που αποδίδει μία προγραμματίσιμη διεπαφή σε λειτουργικότητα εύρους δικτύου (network-wide functionality), με την υπηρεσία να αποτελείται από έναν αριθμό από στιγμιότυπα υπηρεσίας (service instances) όπως είναι για παράδειγμα οι εικονικές μηχανές (VMs) και οι συσκευές (devices), τα οποία μπορεί να βρίσκονται διασκορπισμένα σε ένα κατανεμημένο σύνολο από συστάδες. Για παράδειγμα κάποιες εικονικές μηχανές μπορεί να είναι συγκεντρωμένες σε ένα ή περισσότερα data-centers, ενώ άλλες να είναι κατανεμημένες σε διάφορα edge sites.[26]

Ένα άλλο καθοριστικής σημασίας σημείο, είναι ο διαχωρισμός του ελεγκτή υπηρεσίας από τα στιγμιότυπα υπηρεσίας. Ο ελεγκτής διατηρεί όλη την κατάσταση πιστοποίησης (authoritative state) για την υπηρεσία. Οι χρήστες υπηρεσίας (service users, service operators) αλληλεπιδρούν με τον ελεγκτή που παρέχει μία καθολική διεπαφή. Ενώ κάθε διεπαφή προς στιγμιότυπο είναι εφαρμοσμένη λειτουργία, κρυμμένη πίσω από τον ελεγκτή.[26]

Όπως αναφέρθηκε και παραπάνω ένα βασικό χαρακτηριστικό στο οποίο βασίζεται η χρήση και λειτουργία του XOS είναι ο διαχωρισμός του control-plane από το data-plane. Καθώς αναφέρθηκε, οι ελεγκτές εκπροσωπούν υπηρεσίες, όπου κάθε υπηρεσία ελέγχει ένα επεκτάσιμο σύνολο από στιγμιότυπα (instances) που είναι κατανεμημένα γύρω από το δίκτυο. Τα στιγμιότυπα αυτά αλληλεπιδρούν μεταξύ τους δίχως τον ελεγκτή να είναι απευθείας πάνω στο data path.[26]

3.4.2 Δομή λογισμικού του XOS

Το XOS οργανώνεται σε τρία επίπεδα: μοντέλο δεδομένων (data model), όψεις χρηστών (user views) και ελεγκτής framework (controller framework).



Σχήμα 19: Απεικόνιση της δομής του XOS [26]

Εντός του πυρήνα βρίσκεται ένα μοντέλο δεδομένων (Data Model) το οποίο καταγράφει τη λογικά κεντρικοποιημένη κατάσταση του συστήματος. Εκεί συγκροτούνται όλες οι υπηρεσίες και τις κάνει να συνεργάζονται αξιόπιστα και αποδοτικά. Ο λογικός συγκεντρωτισμός (logical centralisation) σε αυτό το επίπεδο επιτυγχάνεται μέσω ενός καθαρού διαχωρισμού μεταξύ κατάστασης πιστοποίησης (authoritative state) και κατάστασης λειτουργίας (operational state). Αυτή η διάκριση του συστήματος σε αυτά τα δύο επίπεδα, αποτελεί καινοτομία του XOS.[26]

Το Data Model περιλαμβάνει αφηρημένα αντικείμενα, σχέσεις μεταξύ αυτών των αντικειμένων, καθώς και τις λειτουργίες σε αυτά, οι οποίες εμφανίζονται ως REST-HTTP API, καθώς και μέσω της βιβλιοθήκης πελάτη-εξυπηρέτη του XOS που καλείται xoslib, η οποία παρέχει μια προγραμματιστική διεπαφή υψηλού επιπέδου.[26]

Πάνω από το Data Model βρίσκεται ένα σύνολο από όψεις (views) οι οποίες εκφράζουν πως οι χρήστες διαδρούν με το XOS. Οι όψεις είναι στην πραγματικότητα javascript προγράμματα που τρέχουν στον φυλλομετρητή (browser) του χρήστη.

Το Controller Framework είναι ένα σημαντικό κομμάτι του XOS το οποίο είναι υπεύθυνο για την κατανεμημένη διαχείριση κατάστασης (distributed state management), δηλαδή να διατηρεί την κατάσταση που αντιπροσωπεύεται από ένα κατανεμημένο σύνολο ελεγκτών υπηρεσίας (service controllers) σε συγχρονισμό με την κατάσταση πιστοποίησης (authoritative state) που διατηρείται από το Data Model.[26]

Το XOS εκτελείται πάνω σε ένα σύνολο από ελεγκτές υπηρεσίας, όπου κάποιοι από αυτούς δημιουργούνται ως μέρος του XOS και κάποιοι διατίθενται από commodity παρόχους, όπως για παράδειγμα το OpenStack και το ONOS. [26]

3.4.3 Resource containers - slices

Σημαντικά στοιχεία στα οποία βασίζει τη χρήση του το XOS είναι τα container resources που παρέχει και καλούνται slices, μέσα στα οποία τρέχουν οι υπηρεσίες. Όπως έχει επισημανθεί οι υπηρεσίες είναι πρώτης τάξης αφαίρεση-αντικείμενο στο XOS. Δηλαδή οι υπηρεσίες δημιουργούνται, ονομάζονται και διαχειρίζονται με ρητό τρόπο. Έτσι δίνεται η δυνατότητα να ορίσουμε σχέσεις μεταξύ συλλογών από υπηρεσίες.[26]

Η μορφή των αντικειμένων που συνιστούν μία υπηρεσία (service) έχει ως εξής[26]:

Service = ({slice,…}, Controller)

Slice = ({VM,…}, {VN,…}

VM = (Placement, Image, Resources)

VN = (Topology, NetworkOS, Resources)

Controller = (URL, Credentials, Plugin)

Ας εξηγήσουμε λίγο αυτά τα αντικείμενα[26];

- Μία **υπηρεσία (service)**, όπως παρατηρούμε, αποτελείται από έναν controller (ελεγκτή) και από έναν αριθμό (συνήθως ένα) από slices, μέσα στα οποία τρέχουν τα στιγμιότυπα της υπηρεσίας.

- Το κάθε **slice** αποτελείται με τη σειρά του από ένα σύνολο εικονικών μηχανών (virtual machines – VMs) τα οποία λειτουργούν πάνω σε κάποιους φυσικούς servers, και ένα σύνολο από εικονικά δίκτυα (virtual networks – VNs) τα οποία βασίζονται στο υποκείμενο φυσικό δίκτυο και διασυνδέουν μεταξύ τους τα VMs του slice.

- Οι **εικονικές μηχανές (VMs)** αποτελούνται από κάποια χαρακτηριστικά που δίνουν τη δυνατότητα στον προγραμματιστή των υπηρεσιών να επιλέξει κατάλληλα. Το placement αφορά την τοποθεσία του VM, το image είναι η εικόνα που εκκινεί το κάθε VM, και το resources που αναφέρεται στους πόρους που μπορεί να θέσει στο καθένα. Η εφαρμογή των VMs – containers όπως αναφέραμε γίνεται από τα OpenStack και Docker.

- Αντίστοιχα το ίδιο ισχύει για τα **εικονικά δίκτυα (VNs)** με τα πεδία: topology για την τοπολογία του δικτύου, NetworkOS για το λειτουργικό του δικτύου όπως το ONOS στην περίπτωσή μας, και τα resources για τους πόρους του δικτύου όπως είναι οι συνδέσεις (links), το εύρος ζώνης κλπ.

- Ο **ελεγκτής (Controller)** της κάθε υπηρεσίας καταγράφει την κατάσταση τροποποίησης (configuration state), ώστε να επικαλεστεί λειτουργίες του back-end controller.

3.4.4 Σύνθεση υπηρεσιών στο XOS

Αρχικά επισημαίνεται ότι ο όρος της σύνθεσης υπηρεσιών (service composition) αναφέρεται στην τεχνική της συλλογής και διασύνδεσης επιμέρους υπηρεσιών, ώστε να παραχθεί μία μεγαλύτερη. Όπως γίνεται κατανοητό είναι βασική λειτουργία των σύγχρονων cloud συστημάτων και κατ' επέκταση του CORD, που λειτουργούν βάσει της αρχιτεκτονικής των micro-services.

Μία σχέση υπηρεσιών στο XOS Data Model ως ένα tenant αντικείμενο έχει την εξής μορφή:

Tenant = (ServiceT, ServiceP, Connect, Attributes)[26]

όπου,

ServiceT: υπηρεσία του ένοικου (tenant).

ServiceP: υπηρεσία του παρόχου (provider).

Connect: τρόπος διασύνδεσης των δύο υπηρεσιών πάνω στο υποκείμενο δίκτυο.

Attributes: καταγραφή κατάστασης για εγκατάσταση του tenancy.

Όλες οι υπηρεσίες είναι αυτόνομες, δηλαδή κάθε υπηρεσία μπορεί να λειτουργεί ανεξάρτητα σε "κάποιο" cloud. Όμως από την άλλη το XOS είναι σχεδιασμένο να υποστηρίζει υπηρεσίες ως μέρος του cloud γενικότερα.

Για καλύτερη απόδοση και ελαχιστοποίηση του κόστους, το XOS παρέχει μηχανισμούς οι οποίοι ενεργοποιούν τη σύνθεση (composition) στο data plane και σύνθεση στο control plane.

Στο data plane η συνδεσιμότητα περιγράφει τη δυνατότητα μιας υπηρεσίας να συνδέεται με μία άλλη (ανταλλαγή αντικειμένων) μέσω ενός ή περισσοτέρων εικονικών δικτύων, τα οποία προσφέρουν απομόνωση (isolation) μεταξύ των διάφορων συνδέσεων.

Στο control plane γίνεται η διαχείριση των σχέσεων και των αλληλεπιδράσεων μεταξύ των υπηρεσιών (π.χ. tenancy).

3.4.5 Προγραμματιστικό περιβάλλον στο XOS

Το XOS προσφέρει ένα προγραμματιστικό περιβάλλον για δημιουργία εναλλακτικών όψεων (views) του συνόλου των υπηρεσιών που αναπαρίστανται στο XOS. Οι όψεις σχηματίζονται με τρόπο που να εξυπηρετεί αυτόν στον οποίο απευθύνονται. Για παράδειγμα κάποιες όψεις μπορεί να είναι: traditional cloud tenant view, service developer view, service operator view και cloud operator view.[26]

Οι όψεις αναπαρίστανται στο Data Model και ορίζονται από ένα URL και έναν τύπο που υποδεικνύει αν η όψη αποτελεί ένα javascript ή μία εγκατάσταση, από κάποια άλλη σελίδα web, μιας εντός πλαισίου απόκρισης (inframe). Έτσι η τυπική μορφή της όψης έχει την εξής μορφή:[26]

View = (Type, URL)

Συμπερασματικά οι όψεις αποτελούν διεπαφές οι οποίες τρέχουν στον browser του τελικού χρήστη.

3.4.6 Εισαγωγή πόρων

Στο XOS πρέπει να υπάρχει δυνατότητα διαχείρισης σε ένα σύνολο υλικού που βρίσκεται από κάτω. Αυτό με άλλα λόγια σημαίνει την ενσωμάτωση του IaaS (Infrastructure-as-a-Service) στο XOS. Αυτό επιτυγχάνεται ακολουθώντας την κλασική λογική υπηρεσίας (service) του XOS, απλώς κάνοντας τους πόρους υποδομής (infrastructure resourses) διαθέσιμους μεταξύ διαφόρων συνόλων σημείων (sites).

3.4.7 XOS και CORD

Καθώς όπως αναφέρθηκε και στην εισαγωγή η έννοια του CORD είναι η μεταφορά του central office, δηλαδή της υποδομής των τηλεπικοινωνιακών παρόχων στην άκρη (edge) του δικτύου τους, σε υποδομές παροχής νέφους (cloud), δηλαδή στα σύγχρονα data-centers. Η υλοποίηση των λειτουργιών γίνεται πάνω σε κοινού τύπου εμπορικές συσκευές, και έτσι οι παραδοσιακές συσκευές συγκεκριμένου σκοπού που παρείχαν συγκεκριμένες υπηρεσίες δικτύωσης αντικαθίστανται με αντίστοιχες εικονικές εκδοχές αυτών που εκτελούνται τώρα πάνω σε αυτές τις συσκευές κοινού τύπου και διαχειρίζονται από το XOS.

Οι βασικές εικονικές υπηρεσίες που αντικαθιστούν τα αντίστοιχα υλικά τμήματα του central office είναι οι εξείς:

vOLT (virtual Optical Line Termination) ← OLT (Optical Line Termination)

vBNG (virtual Broadband Network Gateway) ← BNG (Broadband Network Gateway)

vCPE (virtual Customer Premises Equipment) ← CPE (Customer Premises Equipment)

Δηλαδή αυτά τα παραδοσιακά μέρη του central office που αποτελούνταν από συγκεκριμένο υλικό, τώρα μετατρέπονται, σε multi-tenant υπηρεσίες, εγκατεστημένες σε μια spine-leaf δομή απαρτιζόμενη από white-box switches και επεκτάσιμο λογισμικό που τρέχει σε εικονικές μηχανές σε κοινού τύπου εξυπηρετητές (commodity servers).

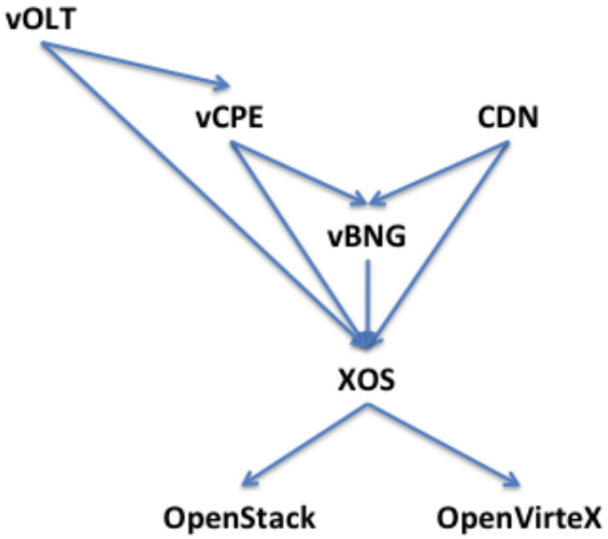
Ας δούμε συνοπτικά τις υπηρεσίες αυτές[26].

vOLT: είναι το σημείο που λήγει το optical link στο central office και εξουσιοδοτεί (authenticates) πρόσβαση στον πελάτη. Κάθε ένοικος (tenant) εδώ, αντιστοιχεί σε έναν συνδρομητή (subscriber).

vCPE: εκτελεί ένα σύνολο από λειτουργίες όπως firewall, DHCP, NAT, VoIP εκ μέρους ενός συνδρομητή. Εδώ κάθε tenant αντιστοιχεί σε μία δέσμη συνδρομητή (subscriber bundle).

vBNG: ο ρόλος του vBNG, ως αντίστοιχος του BNG, είναι να συνδέει τους συνδρομητές και άλλες cloud υπηρεσίες που τρέχουν στο central office, με το δημόσιο διαδίκτυο (public Internet). Κάθε συνδρομητής εδώ αντιστοιχεί σε ένα δρομολογήσιμο υποδίκτυο (routable subnet).

Θα παρουσιάσουμε παρακάτω αυτές τις υπηρεσίες αναλυτικά. Επισημαίνουμε εδώ ότι στο CORD η υπηρεσία vCPE καλείται **vSG** και η υπηρεσία vBNG καλείται **vRouter**.



Σχήμα 20: Σχέση ενοίκου εντός ενός συνόλου εκτελούμενων υπηρεσιών ενός CO [26]

Ακολουθώντας τη ροή αυτών των υπηρεσιών συνολικά παρατηρούμε ότι ένας συνδρομητής γίνεται ένας tenant στο vOLT, όπου γίνεται ένας tenant του vCPE, και με τη σειρά του, γίνεται tenant του vBNG[26].

Επίσης μία CDN υπηρεσία χρησιμοποιεί vBNG για να αποκτήσει δρομολογήσιμες διευθύνσεις (routable addresses) και έτσι δύναται να αποκτήσει περιεχόμενο από γονικούς (origin) servers σε περίπτωση αστοχίας της κρυφής μνήμης (cache miss)[26].

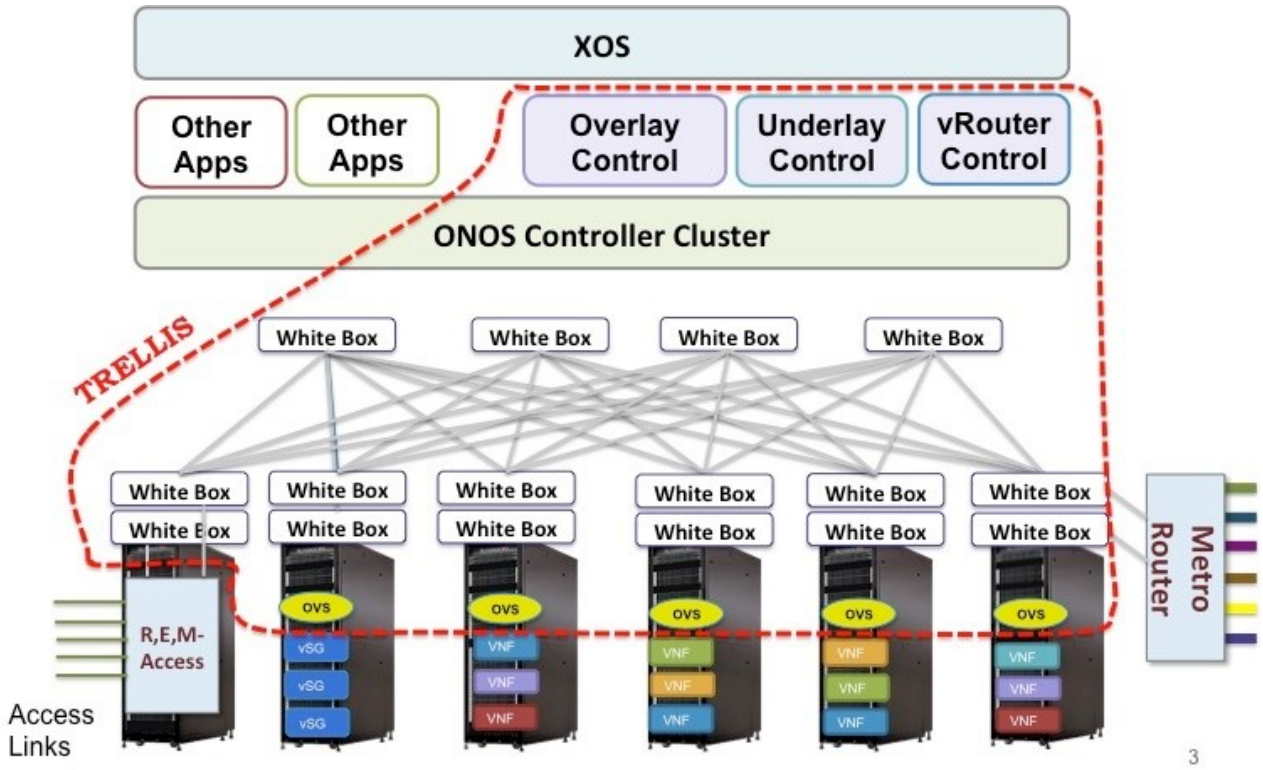
Σημαντική είναι και η δυνατότητα για απομόνωση (isolation) των υπηρεσιών σε ιδιωτικά εικονικά δίκτυα, όσον αφορά την ασφάλεια στο central office. Οι υπηρεσίες vOLT και vCPE συνδέονται μεταξύ τους, συνδέοντας τις εικονικές μηχανές (VMs) που τις εκτελούν, μέσω ενός εικονικού δικτύου (VN), το οποίο δεν είναι δημόσια δρομολογήσιμο. Ενώ, από την άλλη, οι vCPE και vBNG συνδέονται μεταξύ τους με ένα δημόσια δρομολογήσιμο εικονικό δίκτυο. Το vBNG είναι μία επεκτάσιμη υπηρεσία που διενεργεί τη δρομολόγηση του δεύτερου εικονικού δικτύου που αναφέραμε. Πρόκειται ουσιαστικά για ένα λογικό δρομολογητή, εφαρμοσμένο ως μια εφαρμογή

ελέγχου που εκτελείται στο ONOS, το οποίο με τη σειρά του εγκαθιστά κανόνες ροής στην υποκείμενη εγκατάσταση από switches που συνδέουν άλλες υπηρεσίες που τρέχουν μέσα στο data-center, με τις υπόλοιπες στο Internet. Το BGN γενικεύεται ώστε να υποστηρίζει QoS, VPNs και διάφορες μορφές tunneling[26].

Έχοντας παρουσιάσει γενικά το σύνολο αυτών των υπηρεσιών, γίνεται αντιληπτό ότι το XOS είναι αρκετά γενικό ώστε να βρίσκει εφαρμογή σε σημαντικά διαφορετικές περιπτώσεις, με πιο αξιοσημείωτη την XaaS (τα πάντα ως υπηρεσία) αρχή οργάνωσης μαζί με το multi-tenancy, σε ότι παραδοσιακά οργανωνόταν και λειτουργούσε στο επίπεδο των συσκευών και έτσι το XOS αναδεικνύεται ως το πλέον κατάλληλο εργαλείο που λειτουργεί ως ελεγκτής-διαχειριστής αυτών των υπηρεσιών του CORD, καθώς και πολλών άλλων ακόμα που θα δούμε στη συνέχεια καθώς θα αναλύουμε τα συστατικά του CORD.

3.5. TRELLIS

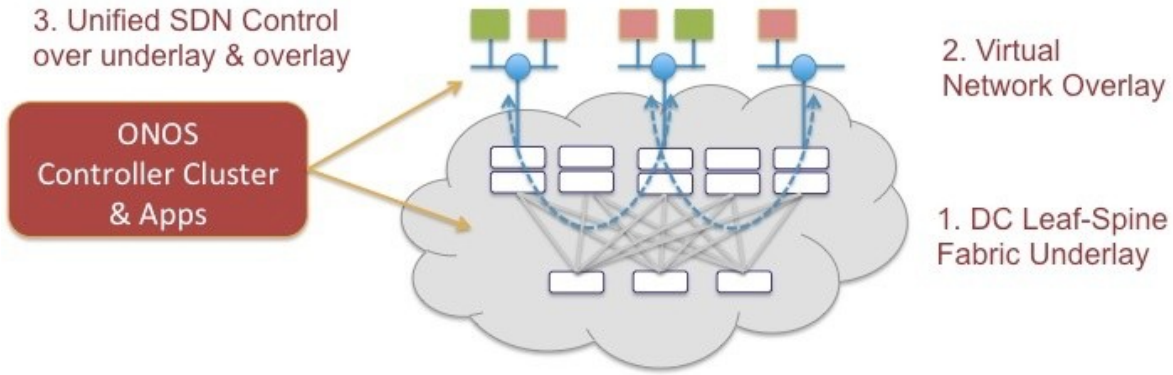
Το Trellis αποτελεί την κυρίαρχη πλατφόρμα ανοιχτού κώδικα (open source) για "φιλοξενία" (hosting) εικονικών δικτύων, η οποία βασίζεται στο SDN και εφαρμόζεται πάνω σε L2/L3 spine-leaf switching αρχιτεκτονική γενικού σκοπού (commodity hardware - white boxes) για δικτύωση σε data-center. Αποτελεί στην ουσία ένα πρότζεκτ πλαισιοποίησης επιμέρους υπηρεσιών και λειτουργιών του υποκείμενου δικτύου του CORD, όπως φαίνεται στην επόμενη εικόνα[27][28].



Σχήμα 21: Τα μέρη της CORD υποδομής που ενσωματώνει το Trellis [29]

Σε αντίθεση με τους παραδοσιακούς τρόπους δικτύωσης, το Trellis μεταφέρει όλον τον λογικό έλεγχο από τα white boxes, τις επιμέρους συσκευές γενικού σκοπού, σε κεντρικό σημείο, στον οργανωμένο κατά συστάδες (onos-cord και onos-fabric) ελεγκτή ONOS. Έτσι αντικαθιστά τα κλασικά πρωτόκολλα ελέγχου όπως το BGP, OSPF, και RTS που τρέχουν στο επιμέρους υλικό (switches, routers, etc), μεταφέροντας τη λογική να τρέχει ως εφαρμογή στο ONOS. Συνεπώς αποκτάται μία αφαιρετική αντίληψη της υλικής υποδομής του δικτύου, δίνοντας τη δυνατότητα για καλύτερη καθολική διαχείριση και βελτίωση, χωρίς να χρειάζεται αναβάθμιση κάθε επιμέρους μηχανήματος (switch). Ακόμα αποφεύγεται το πρόβλημα της ανάγκης για αύξηση της CPU και RAM αυτών των επιμέρους στοιχείων[27].

Όπως αναφέρθηκε το Trellis εφαρμόζεται πάνω σε μια spine-leaf δομή υλικού, το οποίο αναφέρεται ως υπόστρωμα (underlay) του Trellis. Από την άλλη υπάρχει το ανώτερο στρώμα (overlay) του Trellis, που αναφέρεται στην εικονική δικτύωση. Επίσης σημαντικό στοιχείο αποτελεί και ο ενοποιημένος, κοινός SDN έλεγχος που εφαρμόζεται και στα δύο στρώματα.



Σχήμα 22: Ενοποιημένος SDN έλεγχος των overlay και underlay δικτυακών στρωμάτων [29]

Το Trellis είναι ένα σημαντικό κομμάτι του CORD που παρέχει την υλική δομή για όλη τη συνδεσιμότητα στο CORD POD (point of delivery)[27].

Ο κοινός SDN έλεγχος επιτρέπει την αποτελεσματική εφαρμογή κατανεμημένης εικονικής δρομολόγησης και τη βελτιστοποιημένη αποστολή των multicast ροών κίνησης (traffic streams).

Όπως αναφέρθηκε παραπάνω, ο ελεγκτής ONOS είναι οργανωμένος κατά συστάδες. Εδώ εννοούμε ότι αποτελείται ουσιαστικά από δυο σύνολα ONOS ελεγκτών, ένα για κάθε στρώμα του Trellis. Έτσι η μία συστάδα onos-cord είναι υπεύθυνη για το πάνω στρώμα (overlay) που παρέχει την εικονική δικτύωση και τη σύνθεση υπηρεσιών, καθώς και για την πρόσβαση στην υλική υποδομή. Εδώ περιέχονται οι VTN και vOLT εφαρμογές, οι οποίες αναλύονται παρακάτω. Η άλλη ONOS συστάδα είναι υπεύθυνη για τον έλεγχο της υποδομής (fabric) καθώς και της διεπαφής με συμβατικά routers ανόδου (upstream). Εφαρμογές που περιέχονται εδώ είναι το Fabric Control και το vRouter. Αυτός ο διαχωρισμός στις δυο συστάδες επιφέρει καλύτερη απομόνωση και διαχωρισμό των θεμάτων που πρέπει να αντιμετωπιστούν, ώστε η διαχείρισή τους και η επιμέρους ανάπτυξη εφαρμογών να γίνεται πιο εύκολα και αποτελεσματικά[29].

Εδώ επισημαίνεται ότι ο multicast έλεγχος γίνεται μέσω των εφαρμογών IGMP snooping που εκτελείται στο onos-cord και PIM-SSM που εκτελείται στο onos fabric[29].

3.5.1 Trellis Underlay Fabric (υπόστρωμα υποδομής)

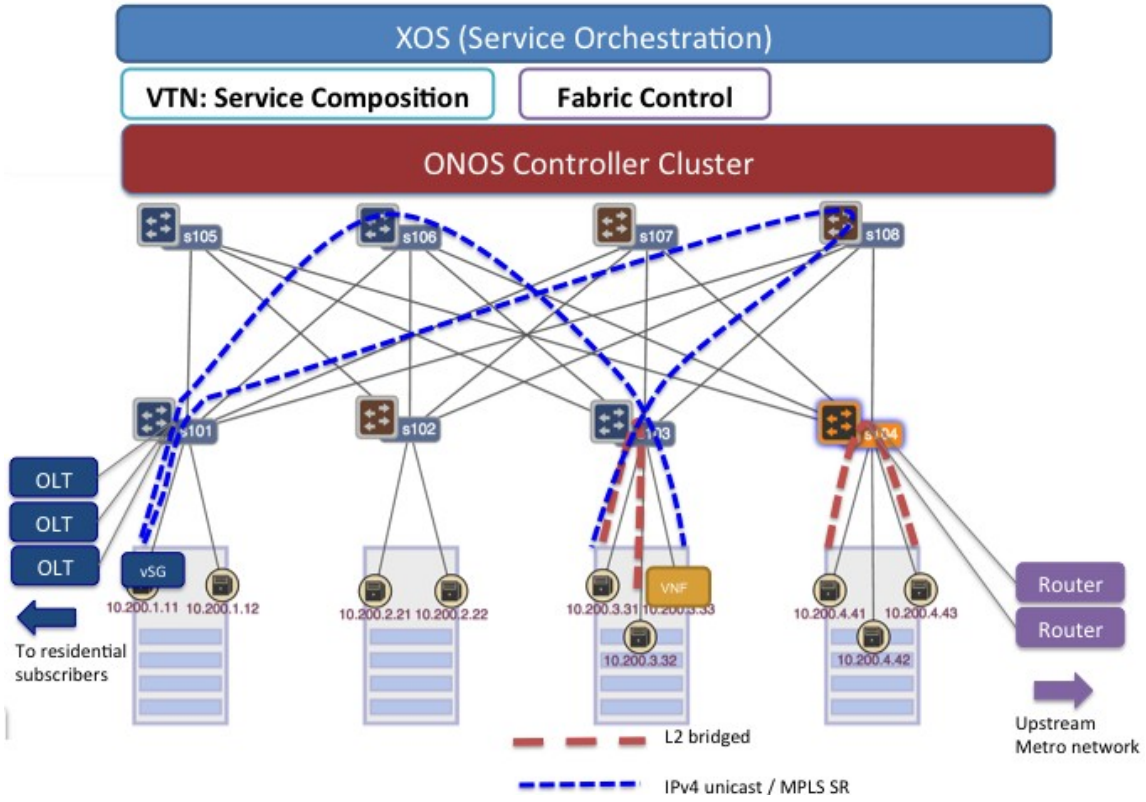
Εδώ γίνεται αναφορά στο υλικό-τεχνικό (feabric) του Trellis που σχηματίζει το δίκτυο υποστρώματος (underlay network). Ο έλεγχος του υποστρώματος γίνεται από μία εφαρμογή στο ONOS που καλείται segmentrouting και αλληλεπιδρά με ένα σύνολο άλλων εφαρμογών όπως το vRouter, vOLT και εφαρμογές multicast ώστε να παρέχει CORD υπηρεσίες[32].

Η αρχιτεκτονική δικτύου του CORD βασισμένη στο Trellis περιλαμβάνει τα εξείς[32]:

- Μία leaf-spine δομή βασισμένη σε SDN με ελεγκτή το ONOS και υλοποιημένη με bare-metal υλικό και λογισμικό ανοιχτού κώδικα (open source switch software). Πρωτόκολλα που χρησιμοποιούνται είναι τα ARP, MAC, VLAN, IPv4, MPLS, VXLAN καθώς και άλλες data plane επικεφαλίδες, χωρίς όμως τη χρήση κατανεμημένων πρωτοκόλλων παραδοσιακών δικτύων, όπως είναι τα: RSTP, OSPF, και BGP, αφού το ευφυές μέρος έχει μεταφερθεί και εκτελείται στο ONOS.

- Όσον αφορά την τεχνική δομή (fabric) περιλαμβάνει τα εξείς στοιχεία: Μία δομή L2 switching εντός των racks, που λειτουργεί κάτω από ένα leaf-switch (ToR-Top of Rack). Ένα L3 σύστημα για την προώθηση (forwarding) δεδομένων μεταξύ των racks, με χρήση **ECMP** hashing (κατακερματισμού) και **MPLS** segment routing (δρομολόγηση τμημάτων). Γίνεται χρήση της λειτουργίας συγκέντρωσης (integration) του vRouter, για διεπαφή με το upstream metro - router για εξωτερική σύνδεση και πρόσβαση με δημόσια δρομολογήσιμες IP διευθύνσεις.

- Από την άλλη μεριά όπως θα παρουσιαστεί παρακάτω, το άνω στρώμα overlay (ή αλλιώς outer fabric) βασίζεται και αυτό σε SDN και χρησιμοποιεί εικονικά switches όπως το OvS (Open Virtual Switch) με τροποποιήσιμο pipeline για αλυσίδωση υπηρεσιών (service chaining). Επίσης διαθέτει λειτουργία κατανεμημένης εξισορρόπησης φορτίου (load-balancing) για κάθε υπηρεσία μέσα σε κάθε OvS. Ακόμα στα OvS γίνεται VxLAN tunneling για εικονικά δίκτυα βασισμένα στο άνω στρώμα (overlay). Το overlay θα εξεταστεί αναλυτικότερα πιο κάτω.



Σχήμα 23: Η υποδομή ως underlay (παράδειγμα δρομολόγησης) [32]

3.5.2 Virtual Network Overerlay (άνω στρώμα εικονικού δικτύου)

3.5.2.1 Virtual Tenant Networks & Σύνθεση υπηρεσιών

Όπως αναφέρθηκε και προηγουμένως, στο CORD οι υπηρεσίες δημιουργούνται με χρήση του XOS από τον network operator . Εν συνεχεία, το XOS παρουσιάζει ένα γράφημα υπηρεσιών (service graph) στο ONOS για την κίνηση συνδρομητή (subscriber traffic). Εκεί ο γράφος αναλύεται χρησιμοποιώντας την VTN εφαρμογή (παρουσιάζεται αναλυτικότερα παρακάτω) του ONOS, σε κανόνες ροής που είναι προγραμματισμένοι στην υποδομή δικτύωσης του CORD (CORD networking infrastructure)[33].

Για να γίνει η σύνθεση υπηρεσιών στο CORD όπως είναι αναμενόμενο, χρησιμοποιούνται τα δίκτυα άνω στρώματος και η τεχνική της εικονικής δικτύωσης[33].

Σύμφωνα με το NFV, κάθε υπηρεσία κατέχει το δικό της virtual network, με τα virtual machines και τα containers που χρησιμοποιούνται για τη διεξαγωγή μιας υπηρεσίας να ανήκουν στο ίδιο virtual network, ενώ μπορούν να δημιουργηθούν και να τρέχουν σε διάφορα σημεία (compute

nodes-servers) μέσα στην υποδομή του cloud του CORD. Επίσης, ένα σημαντικό χαρακτηριστικό για την επεκτασιμότητα του συστήματος είναι η δυνατότητα μεταβολής του αριθμού των VMs/containers, και κατ' επέκταση του μεγέθους του VN, που αποδίδουν μία υπηρεσία[33].

Από την πλευρά των υπολογιστικών κόμβων, κάθε ένας δύναται να φιλοξενεί έναν αριθμό από VMs/containers, που ανήκουν με τη σειρά τους σε πολλαπλά VNs υπηρεσιών. Αυτά τα VMs/containers είναι συνδεδεμένα σε OVS (Open Virtual Switch), το οποίο ενεργεί ως ένα hypervisor switch ελεγχόμενο από το OpenFlow προγραμματίσιμο σε υψηλό επίπεδο[33].

Κάθε VN (ή υπηρεσία) έχει το δικό του load balancer (εξισορροποιητή φορτίου), ο οποίος βρίσκεται κατανεμημένος σε κάθε OVS μέσα στο δίκτυο. Το έργο των load balancers είναι να επιλέγουν κατάλληλα ένα VM instance από τα διάφορα VMs εντός του VN της υπηρεσίας, με σκοπό την καλύτερη κατανομή του φόρτου. Για παράδειγμα έστω ότι ένα VM s1a, στιγμιότυπο της υπηρεσίας service1 VN, το οποίο πρέπει να μεταφέρει κίνηση σε μία άλλη υπηρεσία service 2. Τότε αντί να σταλεί η απαιτούμενη κίνηση σε κάποιο VM του service2, στέλνεται πρώτα στον load balancer του service2 και αποφασίζει αυτός ένα συγκεκριμένο VM του service2 που θεωρεί κατάλληλο, ας πούμε το s2b. Βέβαια υπάρχει η δυνατότητα από τον operator να ορίσει απευθείας ο ίδιος ένα συγκεκριμένο VM που θα μεταφέρει την κίνηση παρακάμπτοντας τον load balancer[33].

Όσον αφορά τον συντονισμό του συστήματος, το XOS και η VTN εφαρμογή του ONOS συνεργάζονται κατάλληλα ώστε να διατηρούν την κατάσταση της εικονικής υποδομής (virtual infrastructure) ενημερωμένη. Σε περιπτώσεις προσθαφαίρεσης VMs/containers για κάποια υπηρεσία και δημιουργία αλυσίδων υπηρεσίας (service chain), το VTN ενημερώνει τους πίνακες (tables) σε συγκεκριμένου σκοπού OVS pipeline για αναθεώρηση της επιθυμητής ροής της υπηρεσίας του συνδρομητή[33].

Από τη μεριά των συνδρομητών, το vSG (Virtual Security Gateway) αναλαμβάνει τη μετάφραση ονόματος-διεύθυνσης - NAT (Name Address Translation) της κίνησης συνδρομητή (subscriber traffic), η οποία στη συνέχεια διοχετεύεται εξωτερικά στο Internet, ή διέρχεται μέσω μιας σειράς υπηρεσιών προτού κατευθυνθεί τελικά έξω στο διαδίκτυο. Τώρα σχετικά με τη μεταφορά της κίνησης μεταξύ των υπηρεσιών, γίνεται χρήση VxLAN "ενθυλάκωσης" (encapsulation) και μέσω VxLAN tunnels γίνεται η διασύνδεση μεταξύ των VMs και των υπηρεσιών. Η ενθυλακωμένη κίνηση δρομολογείται από το υπόστρωμα με χρήση IP/MAC διευθύνσεων της υποδομής[33].

Θα παρουσιαστεί η υλοποίηση των παραπάνω σε μεγαλύτερο βάθος πιο κάτω που θα εξετάσουμε το VTN και τη λειτουργία του αναλυτικότερα[33].

3.5.2.2 vRouter

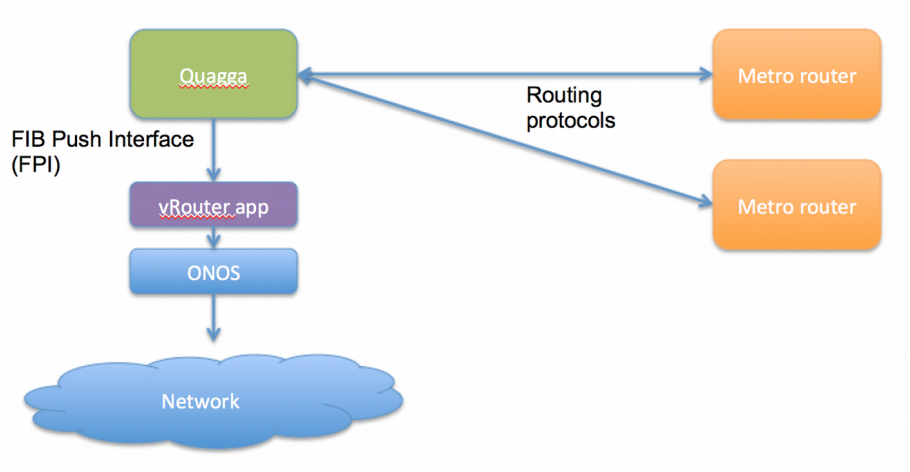
Καθώς σκοπός του CORD είναι η σχεδίαση των δικτυακών υπηρεσιών να αποδίδονται ως εικονικές πάνω σε υλικό γενικού σκοπού, έτσι αντί για τη χρήση BNG (Broadband Network Gateway) συσκευής για τη συγκέντρωση των συνδέσεων των συνδρομητών και της δρομολόγησης της κίνησής τους προς και από τον πυρήνα του δικτύου (core network), εφαρμόζεται μια υπηρεσία virtual router – vRouter (εικονικός δρομολογητής), που παρέχει τη σύνδεση και δρομολόγηση μεταξύ Internet και CORD. Στην ουσία πρόκειται για ένα gateway μεταξύ της CORD υποδομής και του ανωτέρου (upstream) δικτύου και παρέχει πρόσβαση στο διαδίκτυο στους συνδρομητές και τις υπηρεσίες εντός του CORD. Από άποψη λογικού σχεδιασμού, πρόκειται για την τελευταία

υπηρεσία στην αλυσίδα της κίνησης χρήστη, προτού εξέλθει από το σύστημα του CORD. Από υλικής απόψεως, είναι το interface μεταξύ του CORD και του παρόχου του upstream δικτύου. Παρέχει δηλαδή IaaS (Internet-as-a-Service) στις υπόλοιπες υπηρεσίες του Central Office. Το vRouter αποτελεί μία εφαρμογή ελέγχου δικτύου (network control application) η οποία εκτελείται στο ONOS[34].

Η υπηρεσία vRouter χωρίζεται σε control plane και data plane.

Control Plane: Βασική λειτουργία του vRouter είναι να επικοινωνεί χρησιμοποιώντας πρωτόκολλα δρομολόγησης με εξωτερικά routers. Για την αποφυγή της εγκατάστασης των πρωτοκόλλων δρομολόγησης σε μία ONOS εφαρμογή, χρησιμοποιείται μία ανοιχτού κώδικα στοίβα δρομολόγησης, το Quagga, το οποίο υποστηρίζει μεγάλο εύρος πρωτοκόλλων δρομολόγησης, επιτρέποντας έτσι κατ' επέκτασιν την υποστήριξή τους από το vRouter, εξαλείφοντας έτσι την ανάγκη για επανεγκατάστασή τους. Σε αυτήν την προσέγγιση το vRouter πραγματεύεται σχετικά μικρό αριθμό από διαδρομές (routes) και έτσι δεν αφορά υλοποιήσεις εκτάσεως διαδικτύου[34].

Για να επικοινωνεί με upstream routers το Quagga τροποποιείται και χρησιμοποιεί τα OSPF και iBGP πρωτόκολλα δρομολόγησης. Το Quagga ορίζει μία FIP διεπαφή (FIB Plane Interface) για την προώθηση διαδρομών (routes) σε εξωτερική συσκευή και έτσι επικοινωνεί με το ONOS προωθώντας του τις διαδρομές. Ύστερα μία ONOS vRouter εφαρμογή δρά ως διαχειριστής μέσου προώθησης (forwarding plane manager), το οποίο λαμβάνει τις διαδρομές από το Quagga και τα χρησιμοποιεί για να προγραμματίσει αναλόγως το data plane. Το παρακάτω γράφημα περιγράφει αυτή τη λειτουργία[34].



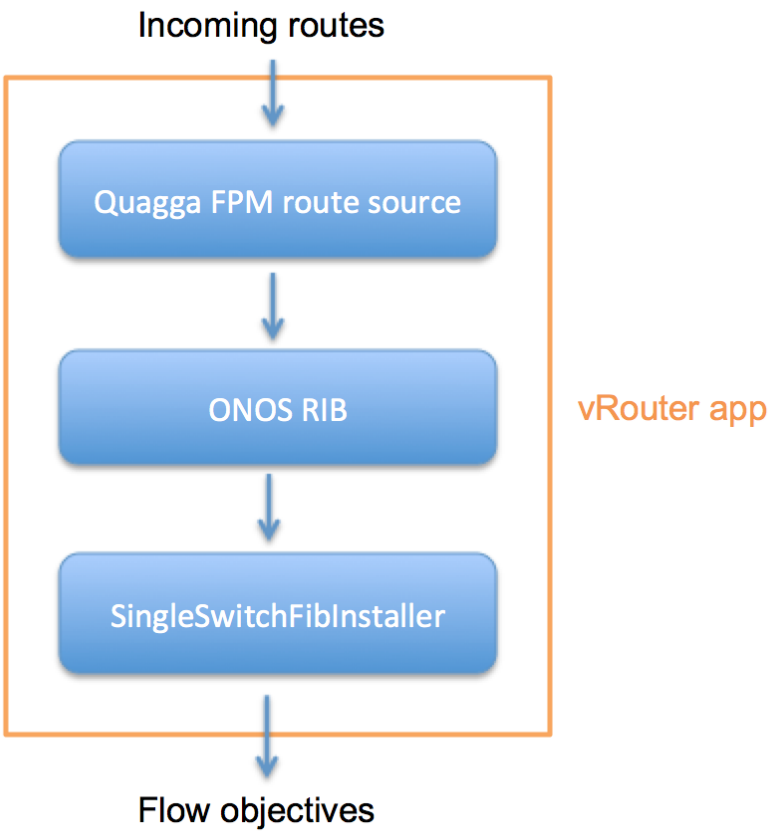
Σχήμα 24: Η control-plane αρχιτεκτονική στο vRouter [31]

Για την επικοινωνία μεταξύ Quagga και εξωτερικού router ανταλλάσσεται μεταξύ τους η κίνηση ελέγχου δρομολόγησης (routing control traffic). Για να επιτευχθεί όμως αυτό θα πρέπει πρώτα το vRouter να προγραμματίσει κατάλληλα το data plane ώστε να επιτρέψει αυτήν την ροή. Το Quagga για να ανταλλάσσει τα δεδομένα με το data plane του vRouter, συνδέει τον server του με ένα port του data plane, αποφεύγοντας έτσι τη λειτουργία δρομολόγησης του vRouter, αφού αφορά control plane κίνηση που προορίζεται για το ίδιο το router[34].

Η multicast σηματοδότηση για upstream γίνεται μέσω υποστήριξης **PIM-SSM** του vRouter το οποίο εφαρμόζεται αποκλειστικά από το ONOS και διαχειρίζεται τα PIM μηνύματα[34].

Data Plane: Το vRouter εγκαθιστά ένα μεγάλο κατανεμημένο router πάνω στα φυσικά switches της υποδομής, και με αυτόν τον τρόπο εξωτερικά το CORD εμφανίζεται ως ένα και μόνο router. Το vRouter επικοινωνεί με τα leaf switches στο data plane τα οποία συνδέονται με uplinks στους upstream routers. Έτσι το vRouter, προγραμματίζει ανάλογα τις διαδρομές (routes) μέσα σε αυτά τα switches. Τα υπόλοιπα switches της υποδομής έχουν προκαθορισμένες διαδρομές οι οποίες δείχνουν προς αυτά τα ακραία (edge) switches[34].

Ας δούμε τώρα γενικά τη συνολική λειτουργία του vRouter. Όπως είπαμε το vRouter λειτουργεί ως ONOS εφαρμογή για δρομολόγηση. Τα εισερχόμενα routes προερχόμενα από το Quagga λαμβάνονται από ένα FPM (Forwarding Plane Manager), το οποίο αποκωδικοποιεί τα routes και τα στέλνει εντός του ONOS RIB. Εκεί αποφασίζεται η MAC διεύθυνση για το επόμενο hop και στη συνέχεια ωθεί μία FIB ενημέρωση στο FIB installer. Στη συνέχεια κάνουμε χρήση ενός SingleSwitchFibinstaller στοιχείο το οποίο όπως δηλώνει και το όνομά του εγκαθιστά διαδρομές μέσα σε ένα single switch, μετατρέποντας στη συνέχεια αυτό το switch σε router. Έπειτα δημιουργεί FlowObjectives και τα υποβάλει στο ONOS[34].



Σχήμα 25: Τα συστατικά που συνιστούν μία vRouter εφαρμογή [31]

3.6 Virtual Tenant Network - VTN

Σε αυτήν την ενότητα θα γίνει ανάλυση μίας σημαντικής εφαρμογής στο CORD, του VTN που θα μας επιτρέψει να κατανοήσουμε καλύτερα τη λειτουργία του CORD, καθώς και τον τρόπο διασύνδεσης των επιμέρους δομικών στοιχείων του CORD.

Το VTN αποτελεί μια εφαρμογή παροχής εικονικού δικτύου πολλαπλών ενοίκων (multi-tenant virtual network) πάνω σε έναν SDN ελεγκτή. Η χρήση των συμβατικών δικτύων για υλοποίηση της λογικής του multi-tenancy επιφέρει αρκετά προβλήματα, καθώς οι ένοικοι στην επικοινωνία τους παρουσιάζουν διαφορετικές ανάγκες δικτύωσης, όπως επίσης απαιτούν να υπάρχει ιδιωτικότητα - απομόνωση μεταξύ τους. Έτσι επιλέγεται η τεχνολογία του virtualization, που γίνεται εφικτή καθώς εφαρμόζεται διαχωρισμός του λογικού (control plane) από το φυσικό (data plane) του δικτύου. Συνεπώς οι χρήστες μπορούν να σχεδιάζουν και να εφαρμόζουν τα επιθυμητά δίκτυα, χωρίς να γνωρίζουν ή να ενδιαφέρονται για την υποκείμενη υλική τοπολογία του δικτύου, ή τους διάφορους περιορισμούς του όπως το εύρος ζώνης κλπ[35].

Με το VTN οι χρήστες ορίζουν το δίκτυο ως ένα συμβατικό L2/L3 δίκτυο. Με το που σχεδιαστεί ένα δίκτυο σε VTN, γίνεται mapped (αντιστοιχίζεται) με το υποκείμενο φυσικό δίκτυο. Με την εφαρμογή του control plane αποκρύπτεται η πολυπλοκότητα του δικτύου, καθώς επίσης γίνεται και καλύτερη διαχείριση των δικτυακών πόρων. Επιτυγχάνει τη μείωση χρόνου επανατροποποίησης των δικτυακών υπηρεσιών και ελαχιστοποιεί τα σφάλματα τροποποίησης[35].

3.6.1 Ο ρόλος του VTN στη σύνθεση υπηρεσιών

Εδώ θα δούμε τον σημαντικό ρόλο του VTN στο CORD και πως συνδέεται με το XOS και τη λογική του.

3.6.1.1 Σύνθεση υπηρεσιών

Κάθε τροποποίηση CORD αποτελείται από ένα σύνολο υπηρεσιών σε ένα **γράφημα υπηρεσιών (Service Graph)**. Τα άκρα (edges) του γράφου αναπαριστούν εξαρτήσεις ένοικου (tenant), όπου μία υπηρεσία πελάτης (client service) είναι ένας ένοικος της υπηρεσίας παρόχου (provider service). Αυτή η σχέση ενοίκου επιτρέπει στην υπηρεσία πελάτη να αποκτά ένα στιγμιότυπο υπηρεσίας (service instance) από την υπηρεσία παρόχου, ακολουθώντας έτσι την πολιτική ασφάλειας που έχει ορίσει ο operator, και καθορίζει πως οι δύο υπηρεσίες πρόκειται να διασυνδέονται στο data-plane του δικτύου. Το VTN παίζει ρόλο κλειδί σε αυτήν τη διασύνδεση[36].

Το μοντέλο δεδομένων (data-model) ορίζει μία υπηρεσία ως ένα σύνολο από εμφωλευμένες κλάσεις αντικειμένων τα οποία αναπαρίστανται ως εξής[36]:

- Service = (Controller. Slices[])
- Slice = (Instances[], Network[])
- ServiceDependency = (ServiceC, ServiceP, ConnectMethod, ConnectService, ConnectNetwork, ConnectAddress)

Το μοντέλο εξάρτησης-υπηρεσίας (SerciceDependency) καθορίζει αποτελεσματικά ένα άκρο (edge) στο Service Graph, να συνδέει μία υπηρεσία πελάτη (customer service – ServiceC) με μία υπηρεσία παρόχου (ServiceP). Τα πεδία σύνδεσης (Connect) υποδεικνύουν πως τα αντίστοιχα δίκτυα των δύο υπηρεσιών διασυνδέονται στο data-plane. Ενώ υπάρχει η δυνατότητα για κάθε υπηρεσία να έχει πολλαπλά slices και κάθε slice να έχει πολλαπλά δίκτυα, συνήθως ακολουθείται η τακτική να υπάρχει ένα (data) δίκτυο για κάθε slice και ένα slice για κάθε υπηρεσία, έτσι για λόγους απλότητας υποθέτουμε στη συνέχεια ότι ισχύει αυτή η ένα-προς-ένα σχέση. Γενικά, για την εγκαθίδρυση μιας ServiceDependency σχέσης, απαιτείται να καθοριστεί ποιο δίκτυο, από ποιο slice, ποιας υπηρεσίας είναι να διασυνδεθεί[36].

3.6.1.2 Δίκτυα παρεχόμενα από VTN

Κάθε υπηρεσία συνδέεται με ένα ή δύο δίκτυα διαχείρισης (management networks) και με ένα ή περισσότερα δίκτυα δεδομένων (data networks). Το VTN εφαρμόζει και τους δύο αυτούς τύπους δικτύων[36].

Το VTN ορίζει δύο δίκτυα διαχείρισης στα οποία τα στιγμιότυπα (instances) μπορούν να ενταχθούν[36]:

- MANAGEMENT_LOCAL: τοποθετεί το στιγμιότυπο στο 172.27.0.1 δίκτυο, το οποίο είναι στον τοπικό υπολογιστικό κόμβο, του οποίου το περιεχόμενο του root είναι πάντα 172.27.0.1. σε αυτό το τοπικό δίκτυο. Οι συγχρονιστές (synchronizers) χρησιμοποιούν αυτό το δίκτυο για SSH μέσα σε ένα στιγμιότυπο για να το διαμορφώσουν. Κάνουν SSH στο περιεχόμενο στο root του υπολογιστικού κόμβου και από εκεί μέσα στο στιγμιότυπο.

- MANAGEMENT_HOST: τοποθετεί το στιγμιότυπο στο 10.1.0.0/24 δίκτυο και γεφυρώνει υπολογιστικούς κόμβους και προσφέρει απ΄ άκρη σε άκρη (end-to-end) συνδεσιμότητα στον επικεφαλής κόμβο (head node). Αυτό το δίκτυο τρέχει πάνω σε ένα φυσικό (physical) δίκτυο διαχείρισης.

Αυτά τα δύο δίκτυα είναι εντελώς ανεξάρτητα. Ένα slice μπορεί να επιλέξει αν θέλει να συμμετέχει σε κάποιο από αυτά, σε κανένα από αυτά ή και στα δύο. Στην τελευταία περίπτωση κάθε στιγμιότυπο έχει δύο διεπαφές διαχείρισης (management interfaces), μία με διεύθυνση 172.27.0.0/24 και μία με διεύθυνση 10.1.0.0/24. Στιγμιότυπα σε διαφορετικούς υπολογιστικούς κόμβους μπορούν να επικοινωνούν μεταξύ τους με το MANAGEMENT_HOST, αλλά όχι με το MANAGEMENT_LOCAL[36].

Τα δύο αυτά δίκτυα διαχείρισης είναι πλήρως τροποποιήσιμα: τα 172.27.0.0/24 και 10.1.0.0/24 είναι αυτά που έχουν τεθεί στο CORD-in-a-BOX αλλά δεν είναι απαραίτητα τα ίδια σε ένα POD πολλαπλών κόμβων[36].

Στη συνέχεια εστιάζουμε στο Data Network, το οποίο συνδέει τα στιγμιότυπα σε μία υπηρεσία μεταξύ τους και προαιρετικά μπορεί επίσης να συνδέσει αυτό το δίκτυο με κάποιο άλλο δίκτυο στο CORD. Το VTN είναι υπεύθυνο για την εφαρμογή της βάσης του Data Network καθώς και τη σύνδεση αυτού του δικτύου με άλλα δίκτυα τα οποία παρέχονται από το VTN καθώς και από άλλες CORD υπηρεσίες όπως τις vOLT, vRouter, vCarrierEthernet κλπ[36].

Κάθε Data Network όταν δημιουργείται είναι ιδιωτικό (private), αναλογικά με ένα ιδιωτικό δίκτυο στο OpenStack (όπως να συνδέει μόνο τα instances στην υπηρεσία). Το δίκτυο μπορεί να παραμείνει ιδιωτικό ή να συνδεθεί σε ένα ή περισσότερα δίκτυα, τα οποία μπορεί να είναι Data Networks άλλων υπηρεσιών, ή μπορεί ένα Data Network να συνδεθεί με το δημόσιο (public) Internet, καθιστώντας τα στιγμιότυπα ως διαδικτυακά-δρομολογούμενα (Internet-routable). Το να μπορείς να συνδέεις μία υπηρεσία στο δημόσιο Internet είναι ασύνηθες, αλλά είναι χρήσιμο στην αντίληψη ότι το δημόσιο Internet είναι απλά σαν κάθε άλλο δίκτυο στο CORD που παρέχεται από κάποια υπηρεσία. Σε αυτήν την περίπτωση, του δημοσίου διαδικτύου, η υπηρεσία παρέχεται από vRouter[36].

Σε προηγούμενη εφαρμογή ένα στιγμιότυπο εμφανιζόταν να συνδέεται απευθείας με το δημόσιο δίκτυο λόγω της εμπλοκής του Neutron του OpenStack. Στη νέα προσέγγιση, το ιδιωτικό δίκτυο είναι αυτό που εγκαθιδρύεται πρώτα και είναι δυνατόν να δωθεί μία δεύτερη δημόσια διεύθυνση σε κάθε port. Έτσι καθίσταται εφικτό ένα ιδιωτικό δίκτυο να γίνεται δημόσιο οποιαδήποτε στιγμή, και γενικά το να μπορούν να του ανατεθούν πολλαπλές διευθύνσεις σε ένα port είναι απαιτούμενο[36].

3.6.2 Διασυνδετικά δίκτυα

Τα υποκείμενα switches λογισμικού προγραμματίζονται ώστε να προωθούν πακέτα, προς και από τις θήρες (Ports) των στιγμιοτύπων της υπηρεσίας. Όταν ένα δίκτυο συνδέεται σε ένα υπάρχον Data Network, τα στιγμιότυπα των δύο δικτύων μπορούν να ανταλλάσουν πακέτα σύμφωνα με τις παραμέτρους που έχουν τεθεί από τη λειτουργία σύνδεσης (splicing operation), πράγμα που μπορεί να οδηγήσει σε ανάθεση νέας διεύθυνσης των θηρών[36].

Το XOS εφαρμόζει, όπως έχει αναφερθεί, ένα Service Graph το οποίο καθοδηγεί τη λειτουργία του VTN. Υπάρχουν οι εξείς περιπτώσεις διασύνδεσης, ανάλογα με το αν οι υπηρεσίες που έχουν συνδεθεί είναι εφαρμοσμένες στο control-plane ή στο data-plane[36].

Αρχικά εξετάζουμε την περίπτωση δύο υπηρεσιών που εφαρμόζουν εικονικές δικτυακές λειτουργίες (VNFs) στο data-plane[36].

Υποθέτουμε την εγκατάσταση μιας εξάρτησης μεταξύ δύο υπηρεσιών στο data-plane, έστω S_A - S_B. Αυτό καθιστά τα Data Networks των A και B να διασυνδέονται, ακολουθώντας τις προδιαγραφές που έχουν καταχωρηθεί σε αυτά τα Data Planes (Έχουμε υποθέσει ότι κάθε υπηρεσία κατέχει ένα slice και κάθε slice ένα δίκτυο)[36].

Τα διανύσματα προδιαγραφών (attribute vectors) είναι τα εξείς δύο και ορίζουν[36]:

1. Άμεσα ή έμμεσα (Direct vs Indirect): Ορίζει τον τρόπο επικοινωνίας (addressing) μεταξύ των A και B. Άμεση με χρήση διεύθυνσης για την κάθε υπηρεσία, ή έμμεση χρησιμοποιώντας διεύθυνση εύρους - υπηρεσίας (service-wide address) με την οποία το VTN εφαρμόζει εξισορρόπηση φορτίου (load balancing), προωθώντας το πακέτο σε συγκεκριμένο στιγμιότυπο.

2. Μη-κατευθυνόμενη ή δικατευθυνόμενη (Undirectional vs Bidirectional): Ορίζει αν η επικοινωνία είναι μη-κατευθυνόμενη, δηλαδή μόνο η A να στέλνει πακετα προς τη B, ή

αν είναι δικατευθυνόμενη, δηλαδή τόσο η Α όσο και η Β να στέλνουν η μία προς την άλλη.

Καθώς, κατά τη διασύνδεση, το δίκτυο της Α προσαρτάται στο δίκτυο της Β, είναι το δεύτερο (της Β) που υποδεικνύει τους όρους της διασύνδεσης. Όμως επεκτείνοντας τις ικανότητες, μπορούμε να καθορίσουμε τα χαρακτηριστικά της διασύνδεσης μέσω ενός μοντέλου σύνδεσης (connection model) αντί της προηγούμενης προσέγγισης[36].

Αυτό αντιστοιχεί σε μία υπηρεσία εφαρμοσμένη από ένα σύνολο από στιγμιότυπα τα οποία έχουν συνδεθεί με μία υπηρεσία εφαρμοσμένη από ένα σύνολο από στιγμιότυπα. Έτσι έχουμε δύο συνδεδεμένα Data Networks εφαρμοσμένα από VTN. Όσον αφορά τις διευθύνσεις, εφόσον το VTN διευθετεί ένα ασύνδετο (disjoint) μπλοκ διευθύνσεων προερχόμενες από κοινό χώρο ιδιωτικών διευθύνσεων σε κάθε Data Network, ένα νέο μπλοκ διευθύνσεων γίνεται δρομολογήσιμο από κάθε υπηρεσία , με το κάθε στιγμιότυπο να έχει την ίδια ιδιωτική διεύθυνση στα Ports του[36].

Στη γενική περίπτωση, η μία ή και οι δύο υπηρεσίες S_A - S_B μπορεί να είναι στο control-plane, όπου εγκαθίσταται ένα δίκτυο, παρά χρήσεις ενός δικτύου. Για παράδειγμα τα vOLT και vRouter, έχοντας δηλαδή μικτές διασυνδέσεις υπηρεσιών[36]:

vOLT → vSG (μια control-plane υπηρεσία συνδέεται με μία data-plane υπηρεσία)

vSG → vRouter (μια data-plane υπηρεσία συνδέεται σε μία control-plane υπηρεσία)

Δηλαδή αντί να συνδεθούν (splice) δύο Data Networks του VTN όπως έγινε προηγουμένως, τώρα μέσω του VTN συνδέεται με κάποιο Data Network με κάποιο άλλο είδος δικτύου.

Έτσι, με αυτήν τη λογική παράγονται και άλλα "δίκτυα ως υπηρεσία" (Networks-as-a-Service), όπως:[36]

vSG → vCarrierEthernet (ένα container συνδέεται με μία κατ' απαίτηση υπηρεσία ενός εικονικού δικτύου αστικής έκτασης)

Η επόμενη περίπτωση είναι αυτή της σύνδεσης μίας control-plane υπηρεσίας με μία άλλη control-plane υπηρεσία. Αυτό το σενάριο δεν υποστηρίζεται ακόμα από το VTN, εφόσον κανένα από τα δύο δίκτυα δεν εφαρμόζεται με VTN. Παραδείγματα τέτοιου τύπου είναι:[36]

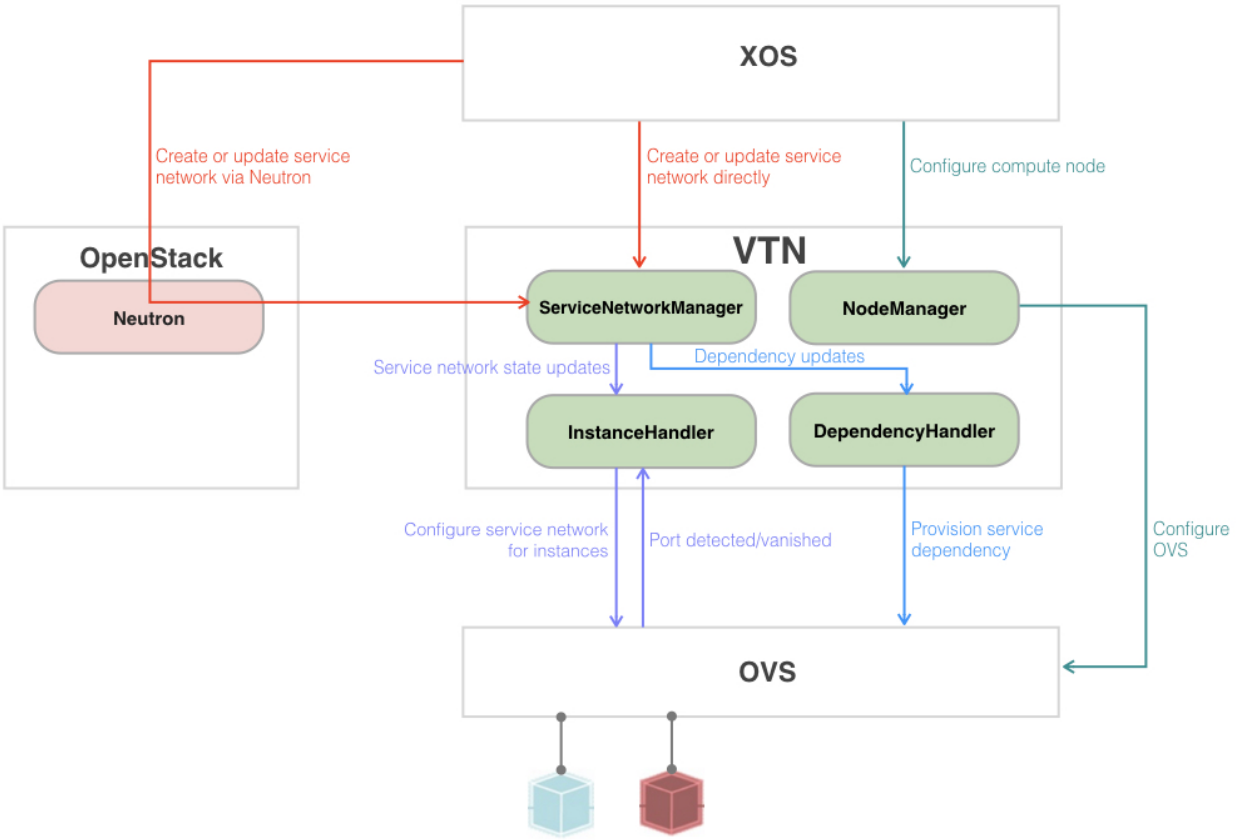
vOLT → vRouter

vSG → vRouter (όπου εδώ το vSG είναι εφαρμοσμένο με SDN και όχι από container)

Άρα ουσιαστικά υπάρχουν δύο περιπτώσεις διασύνδεσης στο VTN. Η μία μεταξύ δύο δικτύων εφαρμοσμένων από το VTN, ενώνοντας στην ουσία τα δύο αρχικά δίκτυα με τους περιορισμούς τους. Η άλλη όταν διασυνδέεται μία υπηρεσία με κάποιο δίκτυο εφαρμοσμένο στο control-plane, δηλαδή παρεχόμενο από το ONOS, με αποτέλεσμα την προσθήκη των στιγμιοτύπων της υπηρεσίας στο νέο δίκτυο, με συνέπεια τα στιγμιότυπα να συνδέονται με νέα διεύθυνση σε αυτό το δίκτυο[36].

Τα παραπάνω είναι αποτέλεσμα των δύο βασικών λειτουργιών του VTN:

- να συνδέει στιγμιότυπα στο δίκτυο.
- να παρέχει ένα ιδιωτικό δίκτυο για ένα σύνολο από στιγμιότυπα.



Σχήμα 26: Συστατικά στοιχεία του VTN [11]

3.7 Virtual Subscriber Gateway – vSG

Ο οικιακός εξοπλισμός καταναλωτή (Consumer Promises Equipment – CPE), γνωστός και ως οικιακός δρομολογητής (home router ή residential gateway), εκτελεί λειτουργίες όπως είναι το DHCP (Dynamic Host Configuration Protocol), NAT (Network Address Translation) καθώς και επιπλέον προαιρετικές υπηρεσίες όπως είναι τα Firewall, Parental Control και VoIP. Σε επιχειρησιακό επίπεδο, επίσης εκτελούνται εξειδικευμένες λειτουργίες όπως WAN Acceleration και IDS (Intrusion Detection System). Καθώς τα CPEs είναι κατανεμημένα σε δεκάδες χιλιάδες πελάτες που αντιστοιχούν σε ένα CO, γίνεται σαφές ότι επιφέρει σημαντικές δυσκολίες, όπως για παράδειγμα αν χρειαστεί να αναβαθμιστεί το υλικό[37].

Σύμφωνα με τη λογική του CORD, που σκοπός είναι να συγκεντρωθούν οι υπηρεσίες δικτύου στο cloud ως εκτελέσιμες εικονικές εκδοχές των αρχικών, το CPE αποκτά την ομότιμη εικονική του εκδοχή που ονομάζεται εικονική πύλη συνδρομητή (Virtual Subscriber Gateway – vSG) η οποία εκτελεί επιλεγμένες λειτουργίες για τον συνδρομητή και εκτελείται σε ένα εικονικό υπολογιστικό στιγμιότυπο (virtual compute instance), όπως ένα VM/container, σε commodity υλικό στο Central Office αντί στην ιδιοκτησία του πελάτη. Ωστόσο παραμένει μία συσκευή στην οικία του πελάτη που συνεχίζουμε να καλούμε CPE, αλλά με περιορισμένη λειτουργία[37].

3.7.1 Σχεδίαση του vSG

Το CORD προσφέρει ένα εύρος εναλλακτικών υλοποιήσεων για vSG, που επικεντρώνονται σε δύο βασικούς σχεδιαστικούς στόχους, με τον έναν να αφορά την λειτουργικότητα και ευελιξία, και τον άλλον να αφορά την απόδοση και την επεκτασιμότητα[37].

Ακολουθεί μια σύντομη περιγραφή αυτών των δύο σχεδιαστικών αποφάσεων.

3.7.1.1 Σχεδίαση ως προς τη λειτουργικότητα

Το CORD παρέχει μία northbound διεπαφή (NBI) που επιτρέπει στους συνδρομητές και στους διαχειριστές (operators) να επιλέγουν και να ελέγχουν ατομικά χαρακτηριστικά (individual features) σε μορφή δέσμης προδιαγραφών συνδρομητή (bundle of subscriber feature), όπου εφαρμόζεται μέσω κάποιας συγκεκριμένης τροποποίησης σε Linux, όπως σε ένα container προορισμένο για αυτόν τον συνδρομητή. Η βασική εφαρμογή αναφοράς (reference implementation) υποστηρίζει συνδεσιμότητα με το διαδίκτυο, καθώς και μία συλλογή παρεμφερών προδιαγραφών όπως είναι[37]:

- Εκκρεμότητα/ Συνέχιση (Suspend/Resume): οι operators μπορούν να διακόπτουν τη σύνδεση του συνδρομητή με το διαδίκτυο, τροποποιώντας το αντίστοιχο container που αφορά τον συνδρομητή ώστε να μην προωθεί κίνηση από αυτόν προς το διαδίκτυο, αλλά με τον πελάτη να μπορεί να προσπελαύνει υπηρεσίες του vSG όπως τα DHCP, DNS.

- Περιορισμένη πρόσβαση (Restricted Access): οι operators μπορούν περιορίσουν την πρόσβαση του συνδρομητή, εφαρμόζοντας iptables έτσι ώστε η HTTP κίνηση να ανακατευθύνεται σε έναν συγκεκριμένο τοπικό web server ο οποίος λειτουργεί ως αποκλειστικός proxy server για το απομακρυσμένο σημείο.

- Γονικός έλεγχος (Parental Control): δίνει τη δυνατότητα στους συνδρομητές να θέσουν φίλτρα γονικού ελέγχου σε επιθυμητές συσκευές στο σπίτι. Αυτό γίνεται ανακατευθύνοντας τις DNS αιτήσεις (requests) της συσκευής (την οποία αναγνωρίζει βάσει της MAC διεύθυνσης) προς ένα τοπικό dnsmasq που τα προωθεί σε μία υπηρεσία γονικού ελέγχου όπως για παράδειγμα το FamilyShield.

- Μέτρηση εύρους ζώνης (Bandwidth Metering): προσφέρει στους operators να διαχειρίζονται το εύρος ζώνης του συνδρομητή (upstream και downstream), καθώς και μέθοδος μέτρησής του.

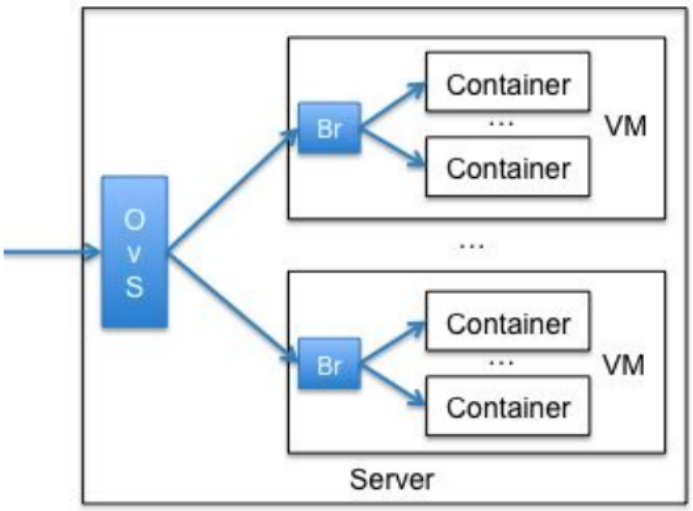
- Πρόσβαση σε διαγνωστικά (Access Diagnostics): παρέχεται η δυνατότητα στους operators να χρησιμοποιούν διαγνωστικά εργαλεία για τη σύνδεση των συνδρομητών, μεσω χρήσης των ping, traceroute, tcpdump.

- Firewall: τόσο οι operators όσο και οι συνδρομητές μπορούν να θέτουν επιθυμητά οικιακά firewalls, τα οποία υλοποιούνται μέσω χρήσης iptables.

3.7.1.2 Σχεδίαση ως προς την απόδοση

Παρέχεται από μία πλειάδα εναλλακτικών τροποποιήσεων τύπου “container σε εικονική μηχανή”, με το container να είναι λογικά συνδεδεμένο με το δίκτυο.

Κάθε συνδρομητής αντιστοιχίζεται με ένα Docker container, και ακολούθως ένα σύνολο από containers τοποθετείται εντός μιας εικονικής μηχανής. Το κάθε container έχει μια LAN-side διεπαφή, διαχειριζομένη από το vOLT, και μία WAN-side διαχειριζομένη από το vRouter[37].



Σχήμα 27: Ανά συνδρομητή Containers εμφωλιασμένα σε εικονικές μηχανές (VMs) [37]

Στη LAN πλευρά (LAN side), γίνεται μία αντιστοίχηση ενός s-tag/c-tag σε κάθε συνδρομητή μέσω του vOLT. Το s-tag υποδηλώνει το VM στο οποίο το OvS θα προωθήσει τα πακέτα, ενώ το c-tag υποδηλώνει το συγκεκριμένο container εντός του VM στο οποίο θα οδηγηθούν τελικά[37].

Στη WAN πλευρά (WAN side), χρησιμοποιείται μία ξεχωριστή ετικέτα VLAN (tag=500), τόσο στα εισερχόμενα όσο και στα εξερχόμενα πακέτα, η οποία υποδηλώνει ότι πρέπει να σταλούν στη WAN διεπαφή του container και να επεξεργαστούν από το vRouter[37].

3.8 Virtual Optical Line Terminal – vOLT

3.8.1 Παραδοσιακά OLT συστήματα

Το OLT (optical line terminal) είναι η τερματική συσκευή ενός οπτικού δικτύου κατανομής (optical distribution network – ODN) στην πλευρά του central office. Πρόκειται δηλαδή για το τερματικό σημείο των τηλεπικοινωνιακών παρόχων, όπου καταφθάνουν οι οπτικές ίνες από τον "έξω κόσμο" και συνδέονται στο central office. Αποτελεί το ένα άκρο ενός PON (passive optical network), με το άλλο να αποτελείται από ένα σύνολο τερματικών σημείων προς την πλευρά των χρηστών, τα οποία καλούνται ONUs (optical network units) ή ONT (optical network terminal).

Βασικές λειτουργίες του OLT αποτελούν η μετατροπή των ηλεκτρικών σημάτων από το central office σε οπτικά σήματα για μεταφορά στο PON και αντίστροφα, καθώς και η δρομολόγηση των εισερχόμενων - εξερχόμενων δεδομένων.

Βασικά συστατικά μέρη του OLT είναι ένας επεξεργαστής (CPU), κάρτες για σύνδεση-επικοινωνία με PON (I/O Blades), καθώς και συστήματα δρομολόγησης όπως gateway router και voice gateway uplink κάρτες.

3.8.2 vOLT - εικονική εκδοχή του OLT

Πρώτος στόχος προς την κατεύθυνση μιας εικονικής έκδοσης OLT, δηλαδή του vOLT, είναι η δημιουργία μιάς κάρτας εισόδου/εξόδου με PON OLT MAC (Media Access Control). Την λύση σε αυτό την έφερε η AT&T που σε συνεργασία με το Open Compute Project ανέπτυξαν ένα τέτοιο I/O Blade, το GPON MAC 1RU "pizza box", το οποίο περιλαμβάνει ένα GPON MAC chip, το οποίο λειτουργεί σύμφωνα με ένα πρόγραμμα απομακρυσμένου ελέγχου το οποίο διευθύνεται μέσω του OpenFlow[38].

Έτσι αντικαθίσταται η κλειστού κώδικα συσκευή που κατέχει το GPON MAC chip με GPON πρωτόκολλο διαχείρισης, VLAN γεφύρωση (bridging) συμβατή με 802.ad, και λειτουργίες Ethernet MAC. Αυτές και άλλες λειτουργίες από την παραδοσιακή συσκευή, αποσυνδέονται και εφαρμόζονται σε λογισμικό[38].

Με άλλα λόγια εδώ χρησιμοποιούμε απλές, κοινές συσκευές για την υλοποίηση των απαραίτητων λειτουργιών σε υλικό (hardware), με το λογικό μέρος από την άλλη να αποσυνδέεται και να τρέχει εντός της cloud υποδομής ως πρόγραμμα ελέγχου, παρεχόμενο από το ONOS, που το ονομάζουμε vOLT και παρέχει τις ολοκληρωμένες λειτουργίες του OLT.

Το vOLT αποτελείται από δύο βασικά μέρη λογισμικού[38]:

- vOLT agent: Εκτελείται εντός ενός container ή εικονικής μηχανής και συνδέει το ONOS με το hardware. Αυτός ο πράκτορας (agent) παρέχει μία OpenFlow northbound διεπαφή έτσι ώστε να ελέγχεται από το ONOS. Έτσι αντιστοιχεί (κάνει map) OpenFlow μηνύματα στα

APIs της συσκευής υλικού και OMCI (ONT Management Control Interface) μηνύματα τα οποία διαχειρίζονται τα PON ONTs.

- Εκτός των άλλων το vOLT agent εκτελεί και λειτουργία bootstrap, όπου ανακαλύπτει το I/O Blade και εγκαθιδρύει μία σύνδεση ελέγχου προς αυτό. Το λογισμικό του vOLT στη συνέχεια παρέχει μία OpenFlow διεπαφή στο ONOS ώστε να προγραμματίσει το vOLT agent. Εν συνεχεία το vOLT agent μετατρέπει αυτά τα OpenFlow μηνύματα σε OMCI για να τα προμηθεύσει στο OLT.

Λειτουργίες διαχείρισης των control-plane λειτουργιών ενός OLT: Περιλαμβάνει λειτουργίες όπως:

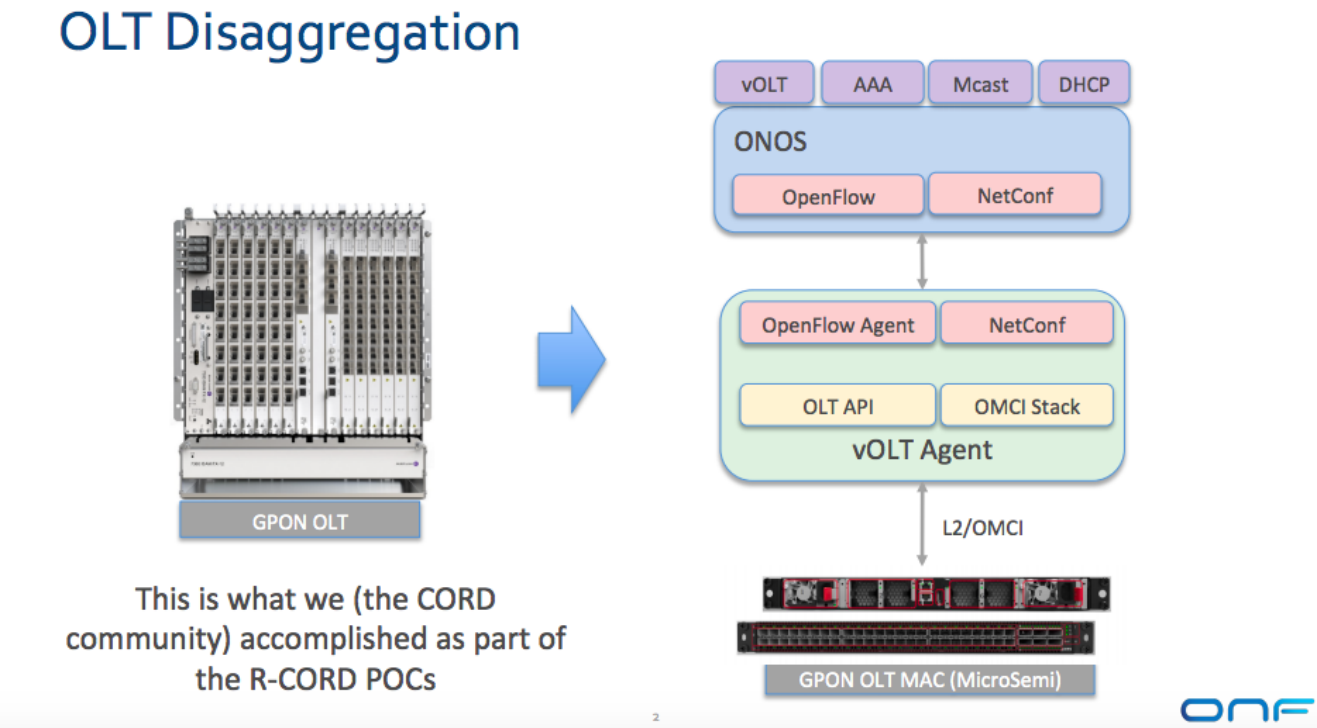
- 802.1x

- IGMP snooping: η διαδικασία να "ακούει" IGMP (Internet Group Management Protocol) κίνηση δικτύου

- VLAN bridging

- OAM(Operation and Maintenance): Λειτουργία και συντήρηση

Αυτές οι λειτουργίες ελέγχου εφαρμόζονται ως εφαρμογές που εκτελούνται στο ONOS, και παρέχουν την σύνδεση των συνδρομητών, πιστοποίηση (authentication – AAA), και εγκαθιστά και διαχειρίζεται VLANs τα οποία συνδέουν τις συσκευές των καταναλωτών και την υποδομή μεταγωγής (switching) του central office σε μία ανά-συνδρομητή βάση (basis), και διαχειρίζεται άλλες control-plane λειτουργίες OLT.

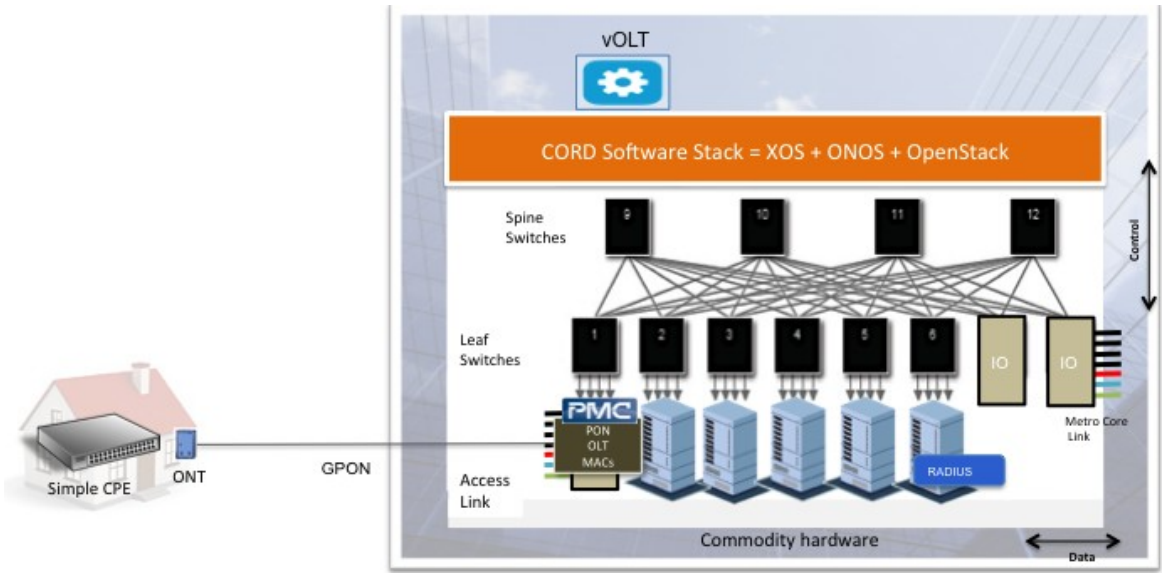


Σχήμα 28: Από το παραδοσιακό, ενσωματωμένο σε συσκευή OLT (αριστερά), μεταφερόμαστε στην εικονική του εκδοχή, το vOLT (δεξιά) [39]

3.8.3 Λειτουργία του vOLT

Ακολουθεί ένα σχήμα που δείχνει περιγραφικά τα όσα αναφέραμε μέχρι τώρα, πως διασυνδέονται μεταξύ τους. Παρατηρούμε στη μεριά του συνδρομητή το απλό CPE, το ONT (Optical Network Terminal), που βρίσκεται κοντά σε αυτόν και αποτελεί τη μία άκρη του PON, που στην περίπτωσή μας είναι συγκεκριμένα η έκδοση GPON. Στο άλλο άκρο βρίσκεται η υποδομή του CORD υλοποιημένη με white boxes – commodity hardware, τύπου data-center η οποία λειτουργεί κατά τα γνωστά κάτω από τη διαχείριση και τον έλεγχο του λογισμικού μέρους του CORD, όπου σχηματίζονται οι διάφορες υπηρεσίες και εφαρμογές, όπως στην περίπτωσή μας το vOLT ύπο τον έλεγχο του ONOS. Βλέπουμε τη σύνδεση της υποδομής του CORD με το GPON μέσω της απλοποιημένης εκδοχής του OLT που αναφέραμε[38].

Ας δούμε τώρα πως εγκαθιδρύεται μία από-άκρο-σε-άκρο (end-to-end) σύνδεση, μεταξύ ενός απομακρυσμένου συνδρομήτη με την CORD υποδομή και κατ' επέκταση με το Internet.



Σχήμα 29: Βασική μορφοποίηση του vOLT με το CORD [38]

Κατά την εκκίνηση του OLT, συνδέεται με μία εφαρμογή vOLT και ξεκινάει το GPON MAC. Το σύστημα ενημερώνεται για τα ONTs που είναι συνδεδεμένα και ενεργά, με το να ενημερώνεται το vOLT μέσω OpenFlow μηνυμάτων για την κατάσταση των Ports (OpenFlow Port status messages), τα οποία περιλαμβάνουν τα IDs των ONT και υποδηλώνουν ότι νέα ports έχουν προσαρτηθεί στο GPON[38].

Από την πλευρά των ONTs, όταν ένα από αυτά προσαρτάται μπορεί να λάβει port up messages για τις διεπαφές του. Έτσι το vOLT εφαρμόζει έναν OpenFlow κανόνα στην διεπαφή, έτσι ώστε να δεχτεί οποιοδήποτε πακέτο 802.1x από το CPE και απορρίπτει την υπόλοιπη κίνηση[38].

Έτσι όταν το CPE του συνδρομητή ξεκινήσει τη λειτουργία του γίνονται οι εξείς λειτουργίες[38]:

- Αρχικά, το CPE δημιουργεί ένα 802.1x πακέτο το οποίο κατέχει ένα πιστοποιητικό (certificate) και το στέλνει προς την OLT κάρτα.

- Με την σειρά του το OLT προωθεί το πακέτο στο ONOS, μέσω του οποίου φτάνει στην vOLT εφαρμογή.

- Εν συνεχεία, το vOLT, στέλνει ερώτημα, βάσει του περιεχομένου του πακέτου, στον τελικό server (RADIUS), ο οποίος κατέχει έναν λογαριασμό καταχωρήσεων (registry account) και ο server ανταποκρίνεται με την πιστοποίηση του συνδρομητή.

Με αυτόν τον τρόπο, στη συνέχεια το vOLT εγκαθιδρύει ένα VLAN, το οποίο συνδέει τον συνδρομητή με το vSG που εκτελείται εντός της CORD υποδομής, ακολουθώντας τις εξείς ενέργειες[38]:

- ενημερώνει το XOS για εκκίνηση ενός vSG container για τον συνδρομητή.

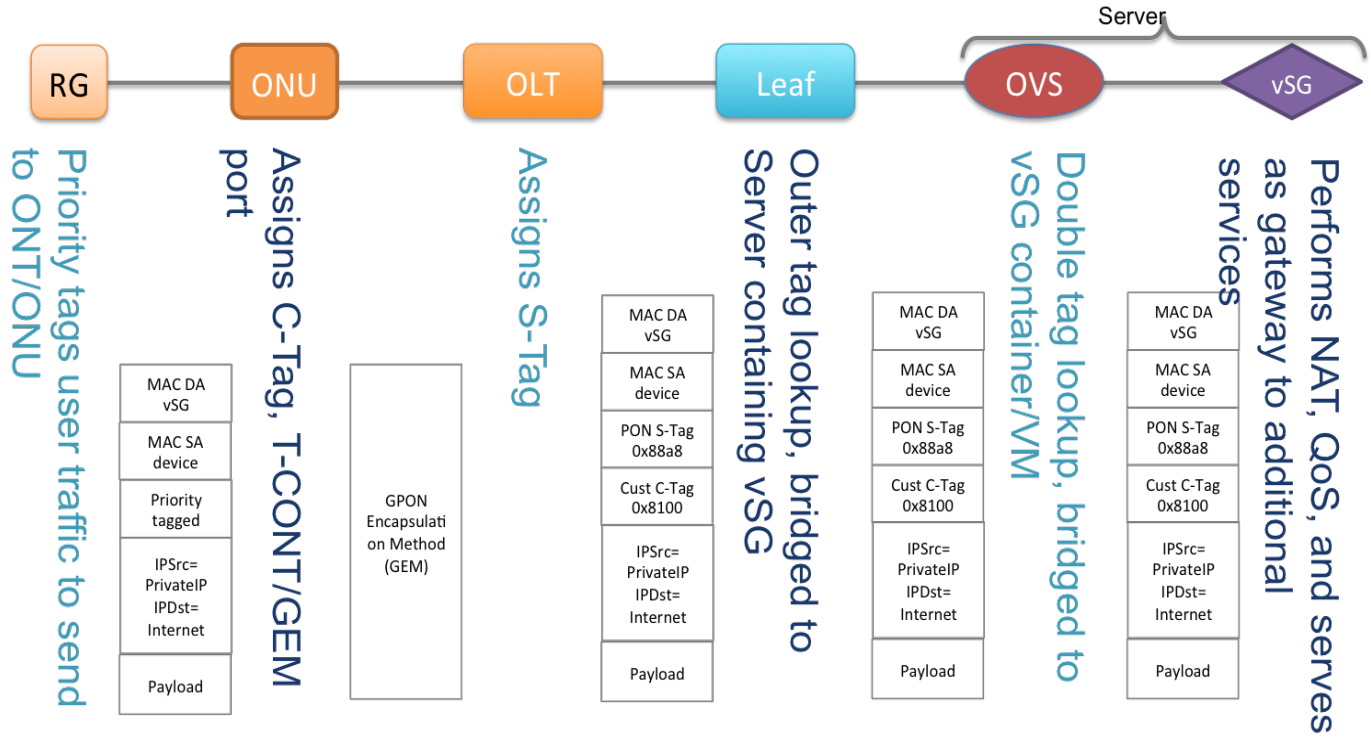
- εκχωρείται ένα VLAN για τον συνδρομητή και ενημερώνεται κατάλληλα το OLT για τη σχετική κίνηση.

- εκχωρούνται πακέτα με καθορισμένες ενδείξεις QoS προς αντίστοιχες ουρές.

- ενημέρωση της ONOS εφαρμογής ελέγχου σχετικά με τη σύνδεση του vSG container με το VLAN.

- τέλος επιστρέφεται στο CPE ένδειξη για την επιτυχή πιστοποίησή του.

Στο σχήμα που ακολουθεί βλέπουμε την επισήμανση (tagging) ενός πακέτου που διέρχεται μέσω VLAN από τον οικιακό εξοπλισμό RG (residential gateway) προς το διαδίκτυο.



Σχήμα 30: Tagging ενός πακέτου εντός VLAN [38]

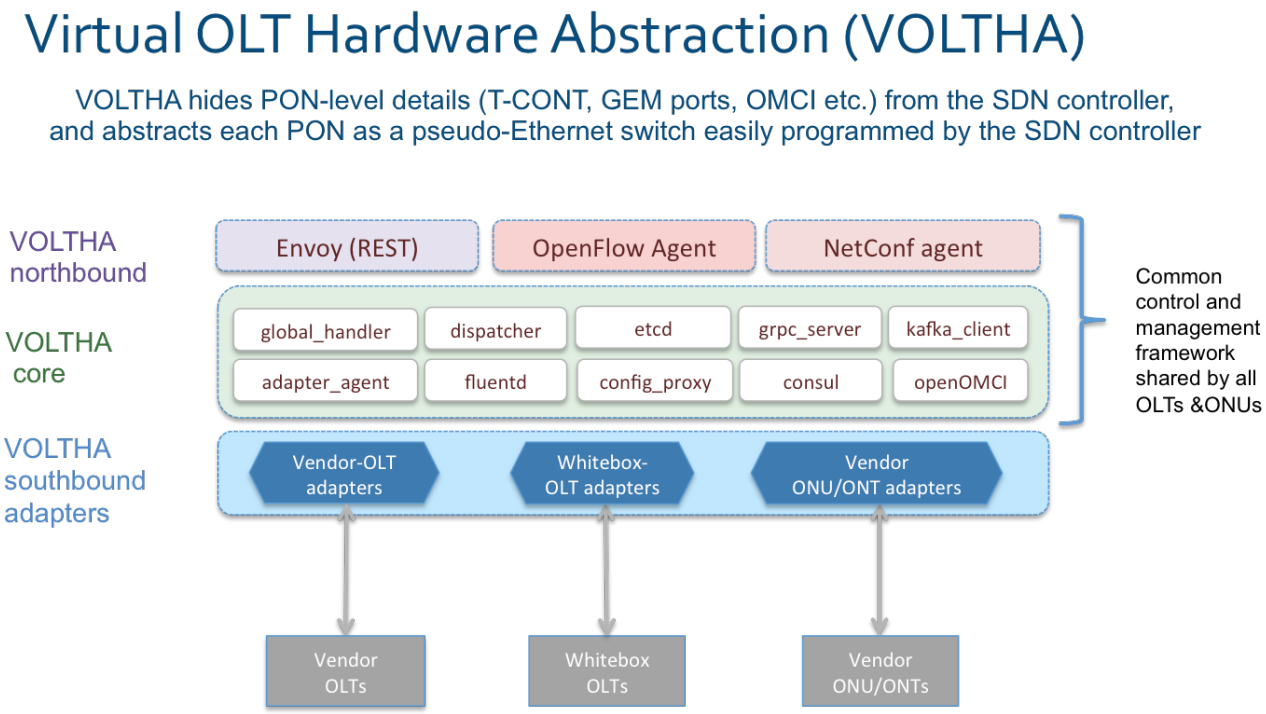
3.8.4 Virtual OLT Hardware Abstraction - VOLTHA

Είδαμε παραπάνω το vOLT, δηλαδή τη σύγχρονη μορφή του OLT, που πλέον εφαρμόζεται πανω σε ανοιχτό, κοινού τύπου (commodity) υλικό, και το λογικό μέρος ως μία εφαρμογή ελέγχου από το ONOS. Τώρα αναφέρουμε το VOTHA, το οποίο αποτελεί το επίσημο πρότζεκτ για την υλοποίηση του vOLT, με αποδοτικό και έγκυρο τρόπο.

Το VOTLHA (Virtual OLT Hardware Abstraction), όπως ανφέρει και το όνομά του αποτελεί μία εικονική αφαίρεση του παραδοσιακού OLT υλικού, δηλαδή του εξοπλισμού ευρυζωνικής πρόσβασης. Πρόκειται για ένα ανοιχτού κώδικα (open source) έργο, το οποίο υποστηρίζει τις έννοιες του "multi-vendor", των αποσυνδεδεμένων (disaggregated) λειτουργιών, και "κάθε ευρυζωνική πρόσβαση ως υπηρεσία (any broadband access as a service)"[43].

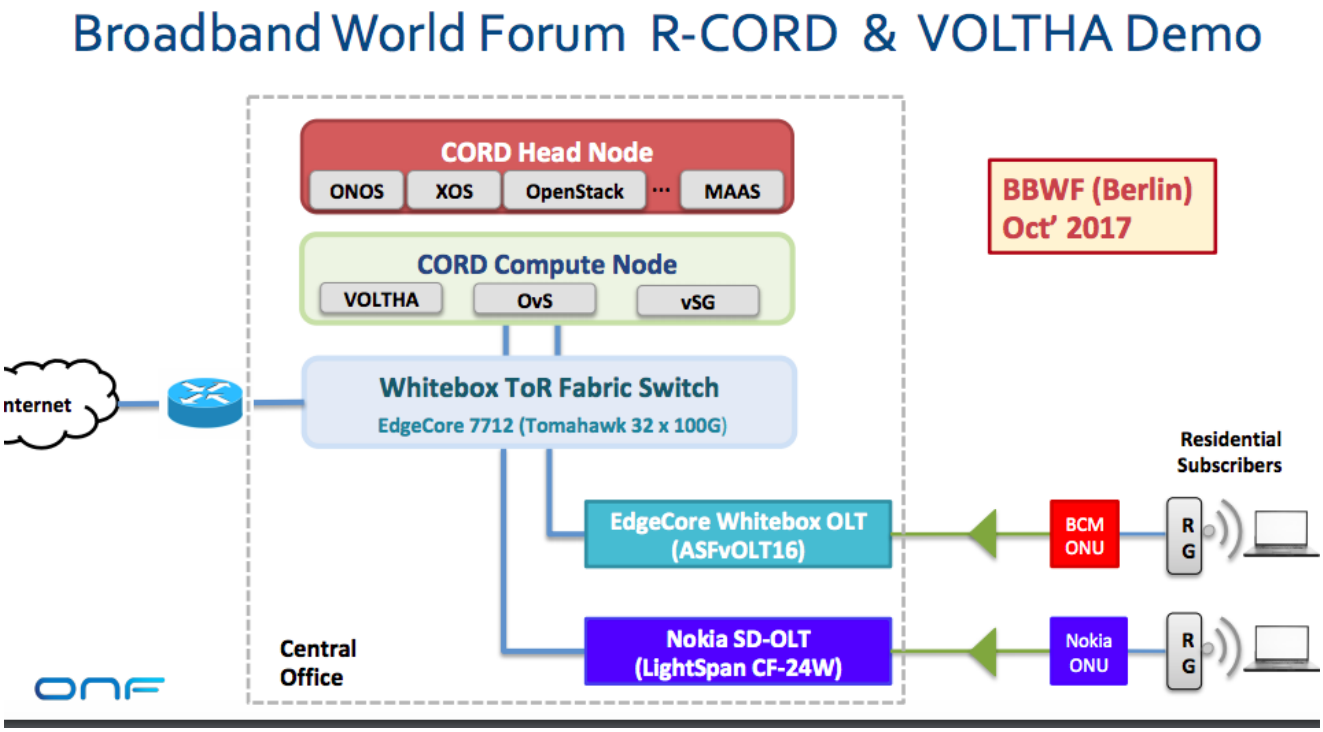
Παρέχει ένα κοινό GPON έλεγχο διαχείρισης συστήματος το οποίο εφαρμόζεται για ένα σύνολο γενικού σκοπού (white boxes) συσκευών, αλλά και για συγκεκριμένου εμπορικού τύπου (vendor specific) PON υλικές συσκευές. Επίσης το VOLTHA ενδέχεται να υποστηρίζει και άλλες τεχνολογίες πρόσβασης όπως EPON, NG-PON και G.Fast[43].

Στη northbound διεπαφή, το VOLTHA παρουσιάζει το PON δίκτυο με αφαιρετικό τρόπο, ως ένα προγραμματίσιμο Ethernet switch σε SDN ελεγκτή. Στην πλευρά του southbound, το VOLTHA επικοινωνεί με OLT συσκευές υλικού, χρησιμοποιώντας vendor-specific πρωτόκολλα, μέσω OLT και ONU προσαρμογέων (adapters)[43].



Σχήμα 31: Αρχιτεκτονική του VOLTHA [44]

Η επόμενη εικόνα δείχνει πως δύο διαφορετικοί συνδρομητές με διαφορετικά ONUs, συνδέονται με διαφορετικά OLTs, ένα υλοποιημένο με white-box και ένα συγκεκριμένου ιδιώτη (εδώ της Nokia), τα οποία και τα δύο ελέγχονται από το VOLTHA.



Σχήμα 32: Σύνδεση συνδρομητών με διαφορετικές τεχνολογίες ONU [39]

3.9 Σύνοψη αρχιτεκτονικής και λειτουργίας του CORD

Όπως είδαμε, για να κατευθυνθούμε από τα παραδοσιακά Central Offices στο CORD, ακολουθήσαμε την τεχνική του virtualization των συσκευών ειδικού σκοπού και των υπηρεσιών που εξυπηρετούν, με το τα αντίστοιχα λογισμικά παραγόμενα της διαδικασίας να εκτελούνται πλεόν σε κοινού σκοπού υλικό το οποίο είναι οργανωμένο σύμφωνα με τα πρότυπα των σύγχρονων data-centers (spine-leaf δομή, κλπ.) . Έτσι απελευθερωνόμαστε από τις δεσμεύσεις που επέφεραν οι παραδοσιακές συσκευές, τόσο στην ανάπτυξη και τη διαχείριση, όσο και στην επεκτασιμότητα (scalability) και οργάνωση του λογισμικού και του υλικού.

Για να επιτευχθεί αυτός ο στόχος της μετατροπής παραδοσιακών συσκευών και υπηρεσιών των τηλεπικοινωνιακών παρόχων σε αντίστοιχες εικονικές εκδόσεις, αρχικά αποδομούμε τη λειτουργικότητα των παραδοσιακών συσκευών, έτσι ώστε να μπορέσουμε να τις ανοικοδομήσουμε με νέους όρους που βασίζονται στον διαχωρισμό του data-plane και control-plane αυτής της λειτουργικότητας.

Ακόμα μία βασική εξέλιξη του CORD από τα παραδοσιακά Central Offices, αφορά αυτό της μορφής μιας γενικής πλατφόρμας εφαρμογής τηλεπικοινωνιακών υπηρεσιών. Αυτό σημαίνει ότι θα

αποτελεί μία βάση για την εφαρμογή των εικονικών υπηρεσιών που δημιουργούμε πάνω σε αυτό, με τρόπο ώστε το συνολικό αποτέλεσμα για την παροχή end-to-end (απ' άκρη σε άκρη) συστημάτων δικτύωσης, να γίνεται με συνοχη, αρμονία και αποτελεσματικότητα. Αυτό είναι εν τέλει που θα αποφέρει τα πλεονεκτήματα της οργάνωσης του CORD, δηλαδή οικονομία, επεκτασιμότητα, και ευελιξία.

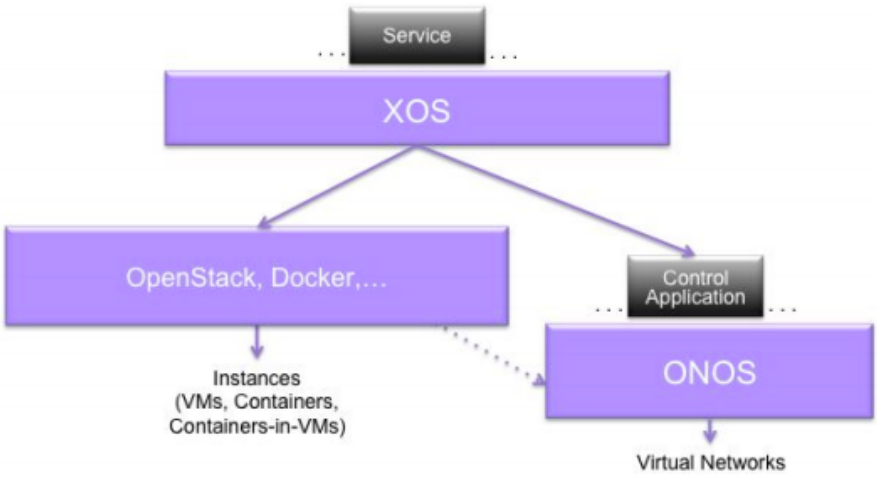
Για την επίτευξη των παραπάνω, ξεκνάμε εφαρμόζοντας τα πρωτεύοντα δομικά στοιχεία – εργαλεία του CORD που είναι τα OpenStack, Docker, ONOS και XOS.

Είδαμε το OpenStack να παίζει τον σημαντικό ρόλο της παροχής των πόρων της υποκείμενης υλικής υποδομής για αποτελεσματική οικοδόμηση εικονικών μηχανών και εικονικών δικτύων. Με άλλα λόγια προάγει τη λογική του "η υποδομή ως υπηρεσία" (Infrastructure-as-a-Service – IaaS).

Το Docker με τη σειρά του προσφέρει στο σύστημα το containerization, δηλαδή το "πακετάρισμα" των υπηρεσιών και των λειτουργιών στα containers, τα οποία αποτελούν πιο ελαφριές μορφές από τις εικονικές μηχανές, ώστε το σύστημα να γίνεται αποδοτικότερο. Για επιπλέον απόδοση και οργάνωση των containers αναφέραμε και τα Kubernetes (K8), τα οποία συνδυάζονται με το Docker για αυτόν τον λόγο.

Το ONOS αποτελεί το λειτουργικό σύστημα του δικτύου για την υποκείμενη υποδομή από white-box switches. Παρέχει ένα σύνολο από εφαρμογές ελέγχου για εφαρμογή διάφορων σημαντικών υπηρεσιών, όπως για παράδειγμα το vOLT, και vRouter που είδαμε παραπάνω. Ακόμα επιτελεί τον ρόλο του SDN ελεγκτή για την υποκείμενη υποδομή μέσω των παραγόμενων εικονικών δικτύων από το Neutron του OpenStack.

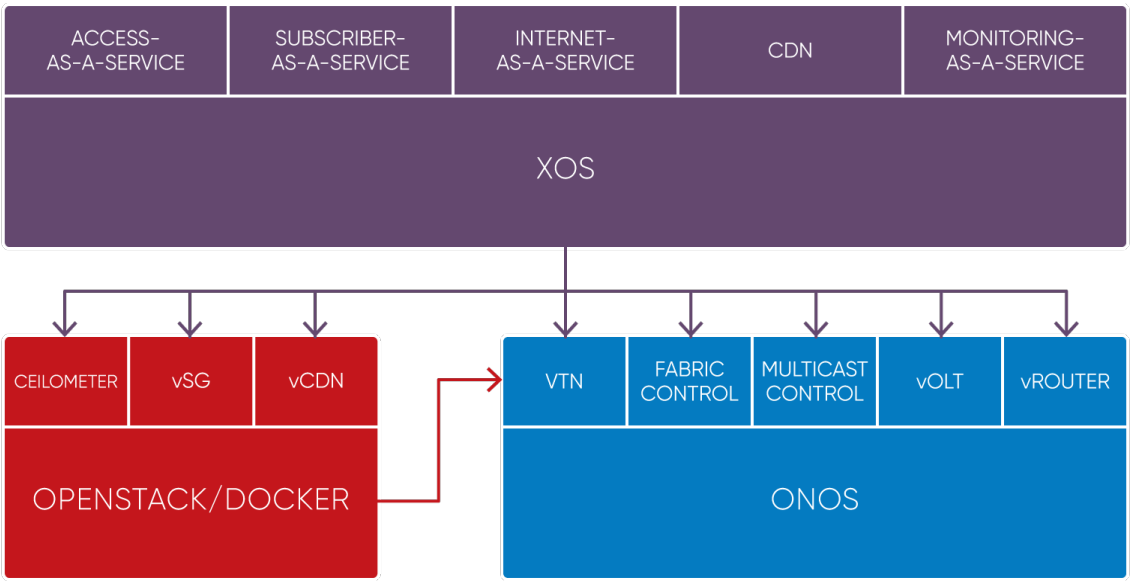
Το XOS βρίσκεται στην κορυφή της ιεραρχίας των συστατικών του CORD καθώς είναι αυτό που δημιουργεί, διαχειρίζεται και ενοποιεί τις τελικές υπηρεσίες (services), τόσο αυτές που αφορούν την υποδομή (infrastructure) και παρέχονται από OpenStack – Docker, όσο και τις υπηρεσίες λογικού ελέγχου (control-plane) που παρέχονται από το ONOS.



Σχήμα 33: Ιεραρχία των δομικών στοιχείων του CORD για την παραγωγή των τελικών υπηρεσιών [50]

Εν συνεχεία παρουσιάστηκαν κάποιες βασικές εφαρμογές και λειτουργίες του CORD. Αναλύθηκε το VTN το οποίο είναι η εφαρμογή που ορίζει τα εικονικά δίκτυα πολλαπλών ενοίκων (multi-tenant), πάνω στο οποίο διαμορφώνονται και διασυνδέονται οι διάφορες υπηρεσίες. Αποτελεί όπως παρατηρήσαμε εφαρμογή "κόμβο" στο CORD, καθώς είναι σημείο συνάντησης των δομικών στοιχείων του CORD και επιτελεί καθοριστικό ρόλο για την διεκπεραίωση της συνολικής λειτουργίας του CORD. Παρέχεται από το ONOS, το οποίο είναι και υπεύθυνο για τη λογική διαχείριση του VTN, αντλεί τους απαραίτητους δικτυακούς πόρους σε μορφή VM/containers από τα OpenStack/Docker ώστε να δημιουργήσει τα επιθυμητά εικονικά δίκτυα (VNs), και τέλος συντονίζεται από το XOS για την εφαρμογή των υπηρεσιών πάνω σε αυτά τα VNs.

Όσον αφορά τις βαικές παρεχόμενες υπηρεσίες του CORD, παρουσιάσαμε τα: vRouter, vOLT/VOLTHA και vSG τα οποία αποτελούν τις εικονικές εκδόσεις των αντίστοιχων παραδοσιακών BNG, OLT και CPE. Είδαμε επίσης πως αυτές οι υπηρεσίες προσφέρουν πρόσβαση των συνδρομητών σε CORD τηλεπικοινωνιακού παρόχου και κα' επέκτασιν στο διαδίκτυο.



Σχήμα 34: Η αλληλεπίδραση των στοιχείων του CORD [72]

3.9.1 Τα οφέλη της εφαρμογής του CORD

Επισυνάπτουμε επιγραμματικά τα οφέλη που επιφέρει η χρήση του CORD[45]:

- Αποτελεί ένα "διάφανο", πλήρως ενσωματωμένο μονοπάτι προς την εφαρμογή της εικονικοποίησης (virtualization) στα central offices.
- Αποφεύγουμε τις προκλήσεις που σχετίζονται με τις παραδοσιακές (legacy) συσκευές.

- Δεν απαιτείται χρήση αφοσιωμένου υλικού (hardware), μειώνοντας έτσι το κόστος.
- Επιφέρει ευελιξία, επεκτασιμότητα και γρήγορες αναβαθμίσεις για να ανταπεξέρχεται στη ζητούμενη προσφορά υπηρεσιών και να ακολουθεί έτσι την ζήτηση.
- Τα central offices μπορούν να ανταπεξέρχονται στους web-scale operators.
- Αποφεύγονται προβλήματα δια-λειτουργικότητας (interoperability) και πρόσβασης.
- Αποτελείται από White-box servers και switches και ανοιχτού κώδικα λογισμικό, συμπεριλαμβανομένων των Docker, ONOS, OpenStack και XOS.
- Μεγάλη ομοιομορφία στις χρεώσεις (bill) των υλικών (materials).
- Ουσιαστική και αρμονική χρήση μεταξύ του NFV και του SDN.
- End-to-end όψη του δικτύου.

4. Το CORD στα κινητά δίκτυα

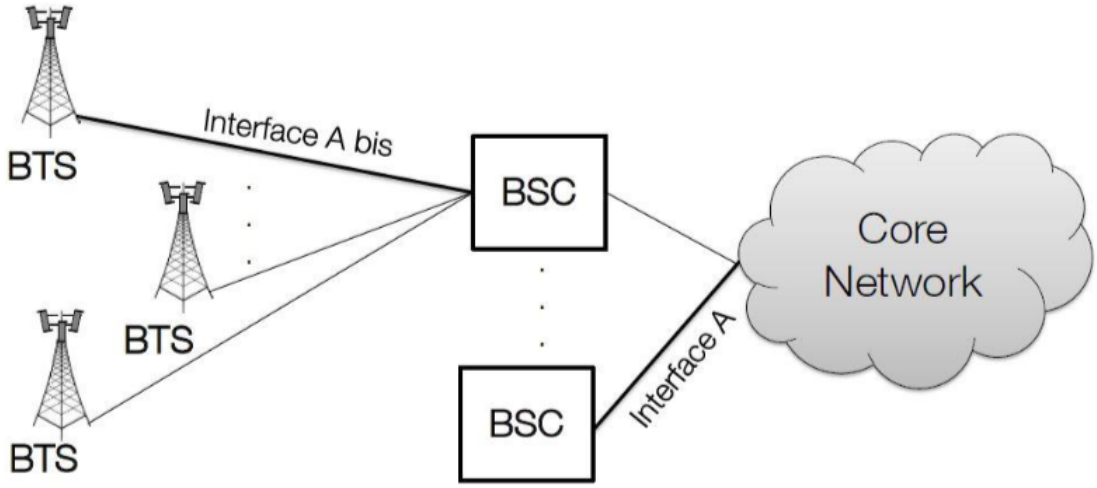
Σε αυτό το κεφάλαιο θα παρουσιαστεί η εφαρμογή του CORD στην κινητή, ασύρματη δικτύωση και την συνδρομή του στην επίτευξη της ανερχόμενης 5G τεχνολογίας. Αρχικά θα γίνει μία σύντομη αναδρομή στις προηγούμενες τεχνολογίες και την εξέλιξη της κινητής δικτύωσης. Ύστερα θα αναφερθούμε στην τρέχουσα κινητή τεχνολογία, το LTE, στην αρχιτεκτονική, και οργάνωσή του, ώστε να αντιληπτό, πως βασισμένοι σε αυτήν την τεχνολογία, θα γίνει το επόμενο βήμα για την επόμενη γενιά κινητής δικτύωσης, όπου το M-CORD βρίσκει εφαρμογή. Ακολούθως θα αναφερθούν εν συντομία κάποιες βασικές αρχές και χαρακτηριστικά του 5G, όπως για παράδειγμα το network slicing. Τέλος το κεφάλαιο ολοκληρώνεται με μία ενότητα, όπου γίνεται περιγραφή του M-CORD, της αρχιτεκτονικής του, καθώς και των εφαρμογών και των βελτιώσεων που επιφέρει στην παροχή κινητής δικτύωσης.

4.1 Αναδρομή στις τεχνολογίες κινητής δικτύωσης

4.1.1 GSM

Η κυψελωτή τηλεπικοινωνία πρώτης γενιάς (1G) βασιζόταν σε αναλογικές διαμορφώσεις σήματος, κάνοντας χρήση διαίρεσης συχνότητας – FDMA (Frequency Division Multiple Access) ως μέσο πολύπλεξης πρόσβασης των συνδρομητών, όπου για κάθε χρήστη αποδιδόταν κάποια συχνότητα, καθιστώντας έτσι αρκετά περιορισμένο τον αριθμό των ταυτόχρονων συνδέσεων. Αυτό το πρόβλημα σε συνδυασμό με την ασυμβατότητα μεταξύ διαφορετικών προτιθέμενων συστημάτων, κατέστησαν το 1G ανεπιτυχές[55].

Έτσι ακολούθησε η επόμενη γενιά κινητών δικτύων (2G), η οποία πρωτοεμφανίστηκε στα μέσα της δεκαετίας του ‘90. Βασική αρχιτεκτονική αυτής της γενιάς ήταν, κυρίως στην Ευρώπη αλλά και αλλού, το GSM (Global System for Mobile communications). Το GSM παρέχει υπηρεσία φωνής σε χρήστες που κινούνται, τόσο μέσα σε μία κυψέλη, όσο και διασχίζοντας πολλαπλές κυψέλες με handover μηχανισμούς, οι οποίοι διαχειρίζονται την ομαλή - διάφανη μετάβαση του χρήστη από μία κυψέλη σε άλλη. Ακόμα εισήγαγε την υπηρεσία σύντομων μηνυμάτων – SMS (Short Message Service), καθώς και μεταφορά δεδομένων χαμηλού ρυθμού. Η πολυπλεξία που χρησιμοποιεί το GSM είναι FDMA μεταξύ των κυψελών και επιπλέον διαίρεση χρόνου – TDMA (Time Division Multiple Access) εσωτερικά των κυψελών. Ο προσφερόμενος ρυθμός μετάδοσης ανά χρήστη είναι 22.8 kbps[55].



Σχήμα 35: Βασική οργάνωση του GSM [55]

Επεξηγούνται τώρα λίγο τα μέρη που αποτελούν την αρχιτεκτονική της υποδομής που φαίνονται στην παραπάνω εικόνα. Ο σταθμός αναμετάδοσης βάσης **BTS** (Base Transceiver Station) περιέχει τα στοιχεία μετάδοσης και λήψης (Transmitter/Receiver Modules – TRX) όπου γίνονται οι λειτουργίες επεξεργασίας των σημάτων. Τα BTS συνδέονται με κάποιο σταθμό βάσης ελέγχου **BSC** (Base Station Controller) ο οποίος είναι υπεύθυνος για τη διαχείριση διάφορων θεμάτων, όπως είναι τα handovers, την RF ισχύ, για τις λειτουργίες συγκέντρωσης–ομογενοποίησης (aggregation/desegregation) της κίνησης, προτού αυτή επικοινωνήσει με το κέντρο κινητής υπηρεσίας μεταγωγής MSC (Mobile service Switching Center) που βρίσκεται στον πυρήνα του δικτύου (Core Network)[55].

Το σύνολο των σταθμών βάσης και των κεραιών που είναι υπεύθυνες για την ραδιο-κάλυψη μίας περιοχής ενός κινητού – ασύρματου δικτύου, καλείται **RAN** (Radio Access Network).

Παρά τη μεγάλη του επιτυχία, το GSM αποδείχτηκε περιοριστικό σε θέματα που αφορούν κυρίως τη χωρητικότητα του δικτύου (ειδικά σε ώρες αιχμής) και στην πρόσβαση των δεδομένων από τον πυρήνα του δικτύου. Ως μία λύση της εποχής ήταν η δημιουργία και η ενσωμάτωση των GPRS και EDGE τεχνολογιών.

Όσον αφορά το **GPRS** (General Packet Radio Service), αποτελεί ένα μεταβατικό στάδιο για υψηλής ταχύτητα μετάδοση δεδομένων πάνω στις προϋπάρχουσες εγκαταστάσεις του GSM. Βασίστηκε στη λειτουργία της μετατροπής των δεδομένων του χρήστη σε πακέτα και στη μεταφορά τους μέσω επίγειου κορμού (backbone network) με χρήση IP, και από εκεί συνδέεται με άλλα PDN (Public Data Network) όπως και το διαδίκτυο.

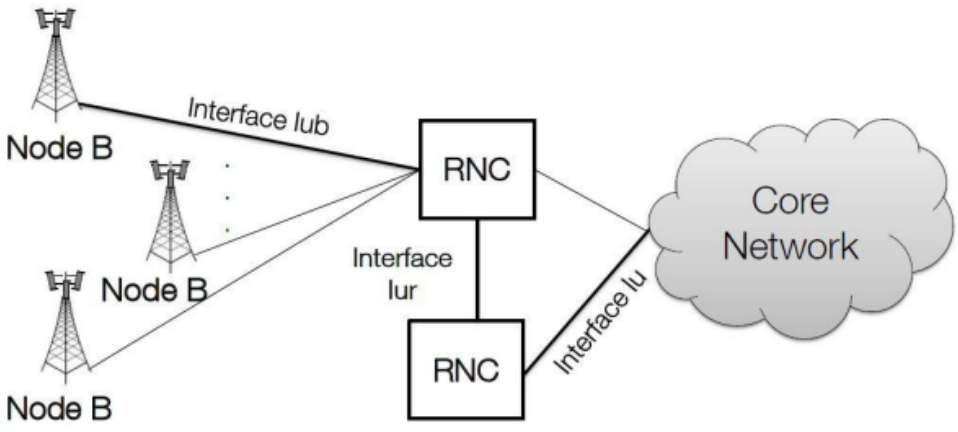
Το **EDGE** (Enhanced Data rates for GSM Evolution), όπως δηλώνει και το όνομά του στόχευσε στην αύξηση των ρυθμών μετάδοσης, καταφέρνοντας ταχύτητα 240 kbps ανά κυψέλη, χάρις τη βελτιωμένη μέθοδο πρόσβασης , χρησιμοποιώντας υψηλού επιπέδου κωδικοποίηση TDMA 200kHz και PSK (phase shift keying) κωδικοποίηση σήματος.

4.1.2 UMTS

Το UMTS (Universal Mobile Telecommunication System) είναι ένα σύστημα κινητής τηλεπικοινωνίας το οποίο αποδόθηκε από το 3GPP (Third Generation Paternship Project), με βασικό στόχο του την αύξηση της χωρητικότητας για υπηρεσίες φωνής, καθώς όμως και τη βελτίωση υπηρεσιών δεδομένων. Το UMTS βασίζεται στο W-CDMA (Wideband Code Division Multiple Access) στα 5 MHz. Όταν δημιουργείται μία κλήση από συνδρομητή, η κλήση αντιστοιχίζεται με έναν συγκεκριμένο κωδικό, ο οποίος είναι γνωστός από τον σταθμό βάσης και το κινητό τερματικό του συνδρομητή, επιτρέποντας έτσι την αναγνώριση και τον διαχωρισμό των κλήσεων πάνω από ένα σήμα φορέα (carrier). Αποδίδει ρυθμούς στα 384 kbps[55].

Στην αρχιτεκτονική του UMTS, ο σταθμός βάσης αναφέρεται ως Node B (Κόμβος Β). Ένα σύνολο από Node B ελέγχεται από κάποιον ραδιοελεγκτή, RNC (Radio Network Controller), μέσω μιας διεπαφής lur η οποία μεταξύ των άλλων χρησιμοποιείται και για την επικοινωνία μεταξύ των RNCs, έτσι ώστε να μπορούν να διαχειρίζονται τα handovers. Τα RNCs με τη σειρά τους συνδέονται με τον πυρήνα του δικτύου μέσω μιας διεπαφής lub[55].

Το UMTS κατάφερε να αυξήσει την ρυθμαπόδοση (throuput) αλλά και να μειώσει την καθυστέρηση (latency) χάρις την εισαγωγή της HSPA (High Speed Packet Access) τεχνολογίας. Η βασική επανάσταση που έφερε αυτή η τεχνολογία είναι ότι η κλήση πλέον βασίζεται στη λογική της μεταγωγής πακέτου και όχι στην μεταγωγή κυκλώματος που γινόταν έως πρότινος. Με αυτόν τον τρόπο εξοικονομούνται σημαντικοί ραδιοπόροι, καθώς η μεταφορά των δεδομένων γίνεται δυναμικά με πακέτα και όχι καταλαμβάνοντας ολόκληρο συχνοτικό δίαυλο για ολόκληρο το χρονικό διάστημα της κλήσης. Ακόμα εφαρμόζονται νέες τεχνικές διαμόρφωσης, τα 16QAM και 64QAM. Επίσης γίνεται χρήση διπλού σήματος φορέα στα 5 MHz, και της τεχνικής MIMO (Multiple In – Multiple Out), πετυχαίνοντας έτσι σημαντική βελτίωση στους ρυθμούς μετάδοσης, φτάνοντας τα 42 Mbps για downlink μετάδοση και 11.5 Mbps για uplink[55].



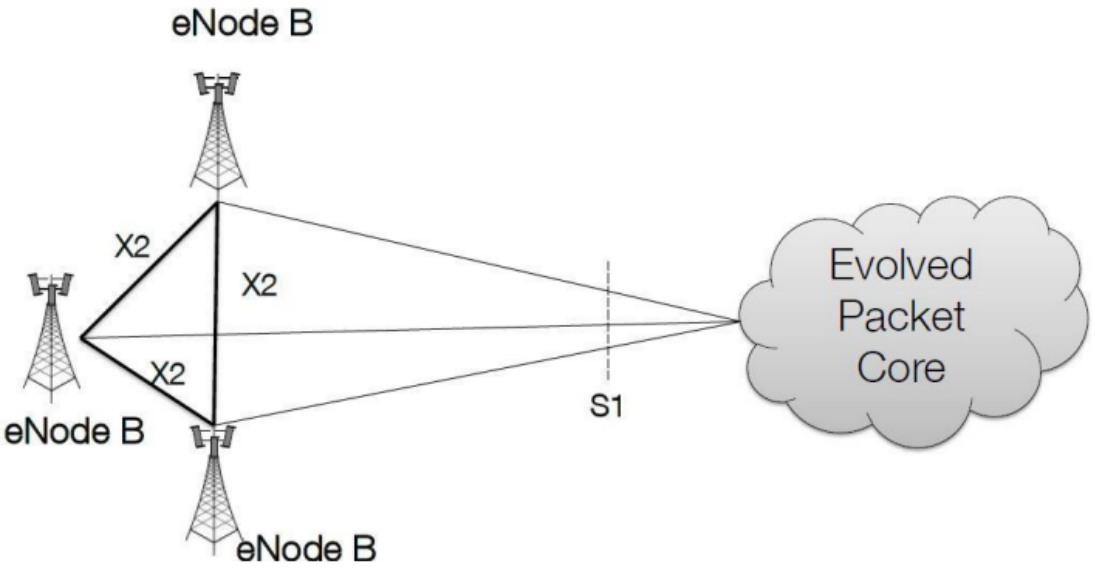
Σχήμα 36: Βασική οργάνωση του UMTS [55]

4.1.3 LTE

Το LTE (Long Term Evolution) είναι στην ουσία η τεχνολογία που εισήγαγε την κινητή δικτύωση στην τεχνολογία τέταρτης γενιά (4G). Αρχικά ορίστηκε ως επέκταση του UMTS, το οποίο βελτιώνει διάφορα σημαντικά χαρακτηριστικά όπως είναι το throughput και η χωρητικότητα κυψέλης, ώστε να καταστήσει τα κινητά δίκτυα σε ισοδύναμη μορφή με αυτά των σταθερών. Σε θεωρητικό επίπεδο το LTE προσφέρει ταχύτητες δεκάδων Mbps τόσο για uplink όσο και για downlink επικοινωνία.

Η βασική τεχνολογία πρόσβασης που χρησιμοποιεί το LTE είναι το OFDM (Orthogonal Frequency Division Multiplexing), επιτρέποντας πιο αποτελεσματική χρήση του φάσματος συχνοτήτων, και πιο ανθεκτικά κανάλια στις παρεμβολές.

Από αρχιτεκτονικής απόψεως, το LTE διαφοροποιείται από το UMTS, παρουσιάζοντας μία πιο ευέλικτη και αποτελεσματική δομή. Τα RNCs παραλείπονται και ως εκ τούτου, επήλθε μείωση στον χρόνο απόκρισης (latency) από άκρο σε άκρο (end-to-end). Η λειτουργία των RNCs μεταφέρεται στους σταθμούς βάσης που σε αυτήν την περίπτωση καλούνται eNodeB. Τα eNodeB συνδέονται στον πυρήνα του δικτύου μιας διεπαφής που καλείται S1. Ο πυρήνας δικτύου εδώ ονομάζεται EPC (Evolved Packet Core). Τα eNodeBs συντονίζονται μεταξύ τους μέσω μιας διεπαφής που καλείται X2. Αυτά τα δύο είδη διεπαφών μαζί συνιστούν το mobile backhaul.



Σχήμα 37: Βασική οργάνωση του LTE [55]

Ο σταθμός βάσης εδώ χωρίζεται σε δύο μέρη. Το ένα είναι υπεύθυνο για την ψηφιακή επεξεργασία σημάτων (radio digital processing), το οποίο ονομάζεται BBU (Bare Band Unit) ή αλλιώς DU (Digital Unit) Το άλλο μέρος είναι υπεύθυνο για τις λειτουργίες που αφορούν τις ραδιοσυχνότητες και ονομάζεται RU (Radio Unit) ή αλλιώς RRH (Remote Radio Head)[55].

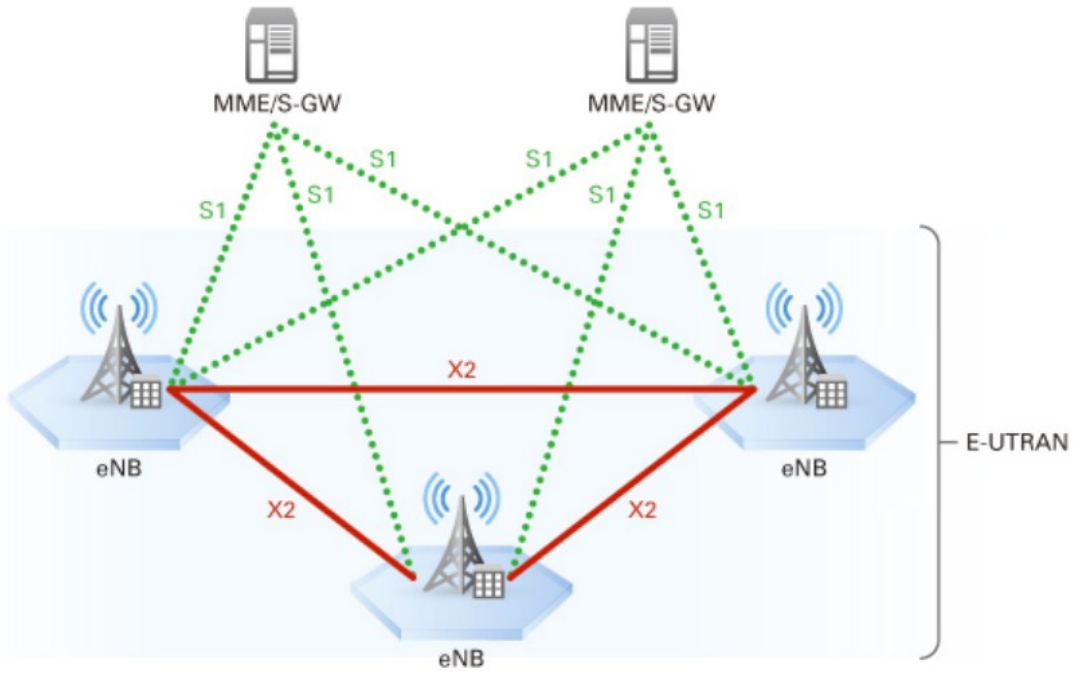
Εδώ το RAN (Radio Access Network) καλείται E-UTRAN (Evolved Universal Terrestrial Radio Access Network) και όπως αναφέραμε κάνει χρήση OFDMA ραδιοπρόσβασης, κεραιών MIMO 4x4 και 2x2.

Στη συνέχεια θα αναπτύξουμε τα βασικά μέρη της αρχιτεκτονικής των κινητών δικτύων που ισχύουν στις σημερινές εγκαταστάσεις, που έχουν κυρίως ως βάση το LTE. Έτσι θα είμαστε σε θέση να κατανοήσουμε τη μετάβαση από τις ισχύουσες τεχνολογίες στις επόμενες βελτιωμένες εκδόσεις τους.

4.1.3.1 Αρχιτεκτονική των LTE κινητών δικτύων

4.1.3.1.1 E-UTRAN Evolved Universal Terrestrial Radio Access Network

Το E-UTRAN αποτελεί το μέρος της αρχιτεκτονικής του LTE για κινητή ραδιοπρόσβαση και περιλαμβάνει τους σταθμούς βάσεις του συστήματος – κεραίες, δηλαδή τα eNodeBs. Με άλλα λόγια πρόκειται για το μέρος του δικτύου που δίνει ασύρματη πρόσβαση στον κινητό χρήστη και αναφερόμαστε σε αυτήν την πρόσβαση ως διεπαφή Uu. Το E-UTRAN μαζί με τον πυρήνα του δικτύου, δηλαδή το EPC, αποτελεί το σύνολο της αρχιτεκτονικής που καλείται EPS. Τα eNodeBs επικοινωνούν με τα γειτονικά τους μέσω μιας διεπαφής X2, ενώ με το EPC επικοινωνούν μέσω μιας διεπαφής S1. Πιο συγκεκριμένα τα eNodeBs συνδέονται με το MME μέσω μίας S1-MME διεπαφής και με το S-GW μέσω μίας S1-U διεπαφής[51].



Σχήμα 38: Η διασύνδεση στο E-UTRAN [76]

Οι βασικές λειτουργίες για του E-UTRAN που αφορούν τη ραδιοεπικοινωνία είναι οι εξείς[51]:

- **Radio Resource Management (RRM)**: Αφορά όλες τις λειτουργίες των ραδιοφορέων, όπως είναι ο έλεγχος ραδιοφορέα (radio bearer control), έλεγχος ραδιο-αποδοχής (radio admission control), χρονοπρογραμματισμός (scheduling) και δυναμική διευθέτηση (dynamic allocation) πόρων για τους χρήστες (UEs) για uplink και downlink.

- **Header compression**: Η χρήση του γίνεται για τη διασφάλιση της αποδοτικής λειτουργίας της ραδιο-διεπαφής. Συμπιέζει τις επικεφαλίδες IP των πακέτων έτσι ώστε να αποφεύγεται το overhead.

- **Security**: Παροχή ασφάλειας μέσω κρυπτογράφησης των δεδομένων που αποστέλλονται μέσω της ραδιο-διεπαφής.

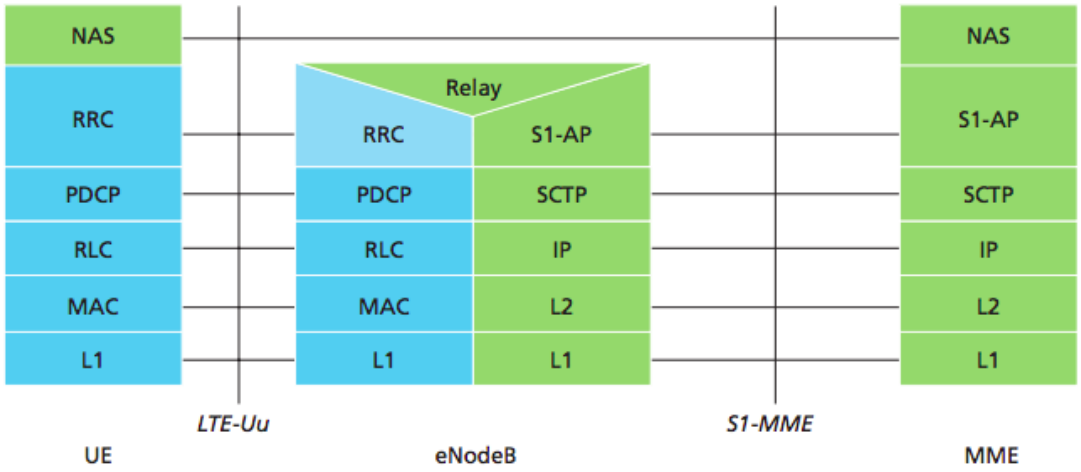
- **Connectivity to the EPC**: Για τη διασύνδεση με το EPC εμπλέκεται σηματοδότηση με το MME και με το S-GW

Το E-UTRAN ενσωματώνει τη radio controller λειτουργία στο eNodeB, επιτυγχάνοντας έτσι την αλληλεπίδραση μεταξύ των διαφορετικών επιπέδων του πρωτοκόλλου[51], και ακόμα μειώνει τον χρόνο ανταπόκρισης και την αποτελεσματικότητα του συστήματος. Με αυτόν τον τρόπο δημιουργείται ένα κατανεμημένο σύστημα ελέγχου και διαχείρισης των handovers, αποφεύγοντας έτσι τη συμφόρηση των κεντρικοποιημένων συστημάτων και τη πιθανή γενική πτώση του συνόλου του συστήματος σε περίπτωση σφάλματος.

Η έλλειψη κεντρκοποιημένου ελέγχου, όμως δημιουργεί το πρόβλημα της ανάγκης για μεταφορά του συνόλου των πληροφοριών του τερματικού χρήστη (UE) από ένα eNodeB σε ένα άλλο, όταν μεταφέρεται από μία κυψέλη σε μία άλλη. Έτσι για τη σωστή διαχείριση αυτής της κατάστασης και για αποφυγή χασίματος πληροφορίας έχουν δημιουργηθεί σχετικοί μηχανισμοί για τις X2 διεπαφές.

4.1.3.1.1.1 E-UTRAN στοίβα πρωτοκόλλου

Η στοίβα πρωτοκόλλων του E-UTRAN αφορά τα πρωτόκολλα του EPS που χρησιμοποιούνται για την επικοινωνία μεταξύ κινητών χρηστών (UE) και των eNodeBs.



Σχήμα 39: Στοίβα πρωτοκόλλων στο E-UTRAN [51]

Οι λειτουργίες κάθε πρωτοκόλλου του E-UTRAN συνοπτικά[54]:

- **Physical layer (L1)**: Το φυσικό επίπεδο, μεταφέρει όλη την πληροφορία από τα MAC κανάλια μεταφοράς μέσω της διεπαφής "αέρα" (air interface) Uu. Αναλαμβάνει την προσαρμογή AMC (Adaptive Coding and Modulation), την αναζήτηση κυψέλης για αρχικό συγχρονισμό και για handover σκοπούς. Επίσης είναι υπεύθυνο και για άλλες μετρικές για το RRC επίπεδο.

- **MAC (Medium Access Control)**: Προσφέρει ένα σύνολο από λογικά κανάλια τα οποία έχουν αντιστοιχιστεί (mapped) με φυσικά κανάλια επιπέδου μεταφοράς. Ακόμα διαχειρίζεται την HARQ διόρθωση σφαλμάτων, την προτεραιότητα των λογικών καναλιών ενός χρήστη (UE), τον χρονοπρογραμματισμό (scheduling) ανάμεσα στα UEs, και άλλα.

- **RLC (Radio Link Control)**: Είναι υπεύθυνο για τη μεταφορά των PDUs (Protocol Data Unit) του PDCP επιπέδου. Λειτουργεί μεταξύ τριών καταστάσεων (modes): TM (transparent Mode), UM (Unacknowledged Mode) και AM (Acknowledged Mode). Ανάλογα με το mode στο οποίο λειτουργεί, μπορεί να παρέχει: ARQ (Automatic Repeat Query) διόρθωση σφαλμάτων, επανακατακερματισμό (re-segmentation) των PDUs, να ξαναζητά RLC δεδομένα σε περιπτώσεις μη σωστής σειράς, καθώς και αναγνώριση διπλοτύπων.

- **PDCP (Packet Data Convergence Protocol)**: Μεταφέρει τα RRC δεδομένα με κρυπτογραφία και προστασία ακεραιότητας. Συμπιέζει και αποσυμπιέζει τα IP δεδομένα , συντηρεί τους αριθμούς ακολουθίας (Sequence Numbers - SN) για τη σωστή σειρά παράδοσης, ανίχνευση και διαγραφή διπλοτυπιών και επαναμετάδοση των SDUs σε περίπτωση handover.

- **RRC (Radio Resources Control)**: Σημαντικές υπηρεσίες που αναλαμβάνει είναι να κάνει broadcast το System Information του NAS (Non Access Stratum) και το System Information του AS (Access Stratum). Επίσης αναλαμβάνει τις λειτουργίες της σελιδοποίησης (paging), εγκαθίδρυση και έκδοση της RRC σύνδεσης μεταξύ του U-ETRAN και του UE, λειτουργίες ασφάλειας όπως η διαχείριση του κλειδιού ασφαλείας (security key), handover, καθώς και διάφορες μετρικές του χρήστη (UE).

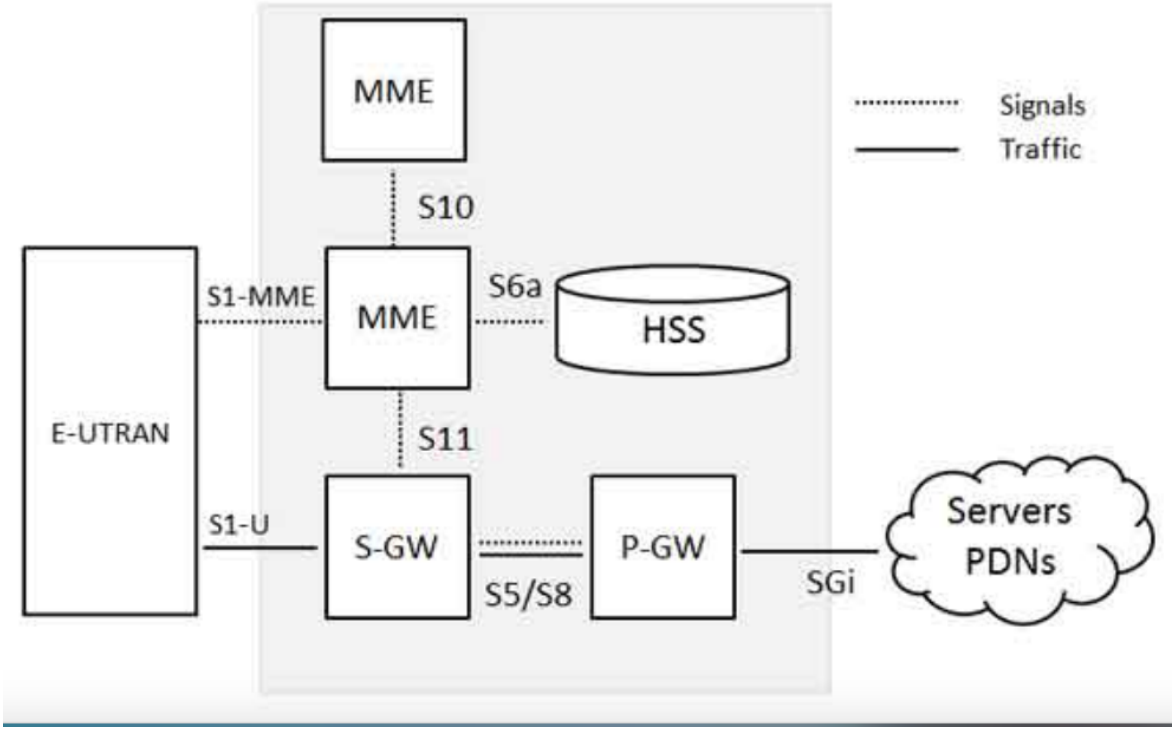
- **NAS (Non Access Stratum)**: Το επίπεδο αυτό αναφέρεται στο υψηλότερο επίπεδο της λίστας πρωτοκόλλων και αφορά το control-plane μεταξύ UE και MME. Στην ουσία διαχειρίζεται την κινητικότητα του χρήστη, και τις διαδικασίες συνόδου για τη σύνδεση του UE με το P-GW.

4.1.3.1.2 Evolved Packet Core - EPC

Το Evolved Packet Core (EPC) αποτελεί τον πυρήνα δικτύου (core network) στην αρχιτεκτονική της 3GPP LTE τεχνολογίας για ασύρματη τεχνολογία. Αποτελεί εξέλιξη των προηγούμενων αρχιτεκτονικών GPRS και UMTS τα οποία βασίζονταν και αυτά στη μεταγωγή πακέτου (packet switching).

Το EPC αποφασίστηκε να έχει μία "επίπεδη αρχιτεκτονική". Ο σκοπός είναι να χειρίζεται αποτελεσματικά, από άποψη απόδοσης και κόστους, το ωφέλιμο φορτίο (payload). Οι εμπλεκόμενοι κόμβοι για το χειρισμό της κίνησης είναι λίγοι και αποφεύγεται η μετατροπή πρωτοκόλλων (protocol conversion).

Στο επόμενο σχήμα απεικονίζεται η αρχιτεκτονική του EPC. Φαίνεται αριστερά πώς το EPC συνδέεται με το E-UTRAN το οποίο αποτελεί το δίκτυο πρόσβασης (Radio Access Network) του LTE και το οποίο αποτελεί το σημείο σύνδεσης του τερματικού χρήστη (UE) και του σταθμού βάσης (eNodeB). Αναφεραμε στο E-UTRAN στην προηγούμενη ενότητα. Το EPC, όπως βλέπουμε απαρτίζεται από τα εξείς τέσσερα δικτυακά στοιχεία: MME (Mobility Management Entity), S-GW (Serving Gateway), P-GW (PDN Gateway) και HSS (Home Subscriber Server). Παρατηρούμε πως το EPC συνδέεται με το E-UTRAN μέσω των στοιχείων MME και S-GW, εών με τα εξωτερικά δίκτυα μέσω του P-GW.



Σχήμα 40: Δομικά στοιχεία και οργάνωση στο EPC [77]

4.1.3.1.2.1 Δομικά Στοιχεία του EPC

- **MME (Mobility Management Entity)**: Το MME είναι ο βασικός κόμβος του EPC, υπεύθυνος για τη διαχείριση του control-plane για το LTE δίκτυο πρόσβασης. Αποτελεί το τερματικό σημείο του NAS. Εκτελεί λειτουργίες ελέγχου και λειτουργίες σηματοδότησης (signaling) για τη διαχείριση της πρόσβασης του UE στις συνδέσεις του δικτύου, την κατανομή των δικτυακών πόρων, καθώς και τη διαχείριση των καταστάσεων κινητικότητας (mobility state), ώστε να διεξάγονται λειτουργίες όπως η ανίχνευση (tracking), η σελιδοποίηση (paging), η περιήγηση (roaming) και τα handovers[57].

  Το MME αποτελεί το ουσιαστικό σημείο που γίνεται η επιλογή των πυλών (gateways), δηλαδή των S-GW και P-GW, του EPC που χρησιμοποιούνται ανά πάσα στιγμή.

- **S-GW (Serving Gateway)**: Το S-GW είναι η πύλη η οποία ενώνει το RAN μέρος μέ το EPC. Είναι υπεύθυνο για τη δρομολόγηση και προώθηση των IP πακέτων από/προς την πλευρά του E-UTRAN και λειτουργεί επίσης ως τοπική άγκυρα κινητικότητας (local mobility anchor), δηλαδή ως σημείο συντονισμού σε περιπτώσεις handover μεταξύ διαφορετικών eNodeBs, αλλά και για αλληλεπίδραση μεταξύ διαφορετικών τεχνολογιών 3GPP, όπως UMTS και GPRS. Σε περίπτωση αδρανούς κατάστασης (idle state) του UE, διατηρεί τις πληροφορίες των φορέων (bearers) και τα downlink δεδομένα, και ενεργοποιεί την διαδικασία της σελιδοποίησης για επανεγκατάσταση των φορέων. Επίσης το S-GW εκτελεί και κάποιες λειτουργίες διαχείρισης όπως είναι η συλλογή πληροφοριών για χρέωση του χρήστη (για παράδειγμα τον όγκο των δεδομένων που έστειλε ή άντλησε ο χρήστης) καθώς και νόμιμη αναχαίτιση[51].

- **P-GW (PDN Gateway):** Το PDN παρέχει συνδεσιμότητα στο UE με εξωτερικά δίκτυα πακέτων δεδομένων (packet data networks – PDN), όντας το σημείο εισόδου και εξόδου της κίνησης του UE. Ένα UE μπορεί να έχει ταυτόχρονη συνδεσιμότητα με περισσότερα του ενός P-GWs για πρόσβαση σε πολλαπλά PDNs. Το P-GW εκτελεί επιβολή πολιτικής (policy enforcement), φιλτράρισμα πεκέτων για κάθε χρήστη, υποστήριξη χρέωσης και νόμιμη αναχαίτιση. Άλλος ένας σημαντικός ρόλος του P-GW είναι να δρα ως άγκυρα κινητικότητας (σημείο συντονισμού), μεταξύ 3-GPP και μη-3GPP τεχνολογιών[59.]

- **HSS (Home Subscriber Server)**: Το HSS είναι μία κεντρική βάση δεδομένων που παρέχει πληροφορίες σχετικές με τον χρήστη και την συνδρομή (subscription). Λειτουργίες του HSS αποτελούν η διαχείριση κινητικότητας (mobility management), υποστήριξη εγκαθίδρυσης κλήσης και συνόδου, πιστοποίηση χρήστη και εξουσιοδότηση πρόσβασης[60].

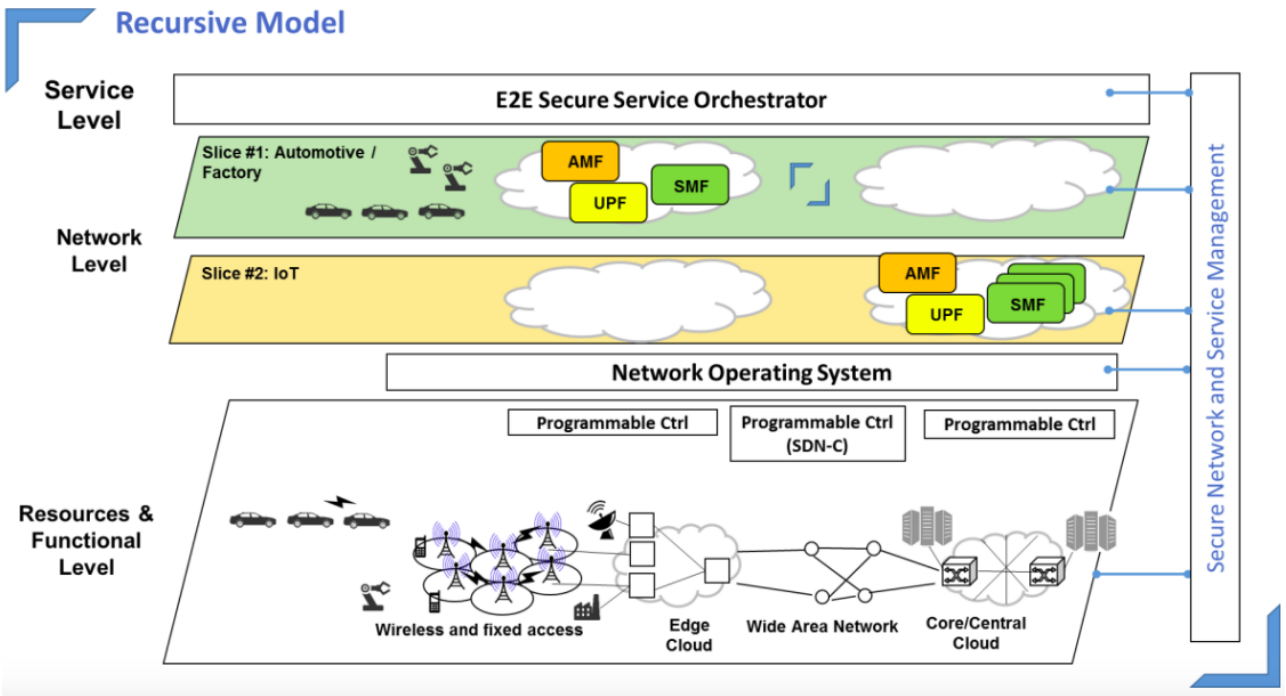
4.1.4 5G

Σε αυτήν την ενότητα θα περιγραφεί συνοπτικά το 5G, και επίσης θα αναφερθούν οι στόχοι του και οι εφαρμογές που αναμένεται να φέρει, ώστε να περιγραφεί στη συνέχεια στο M-CORD που ενδέχεται να αποτελέσει μία πλατφόρμα πάνω στην οποία θα βασιστεί το όραμα του 5G.

4.1.4.1 Εισαγωή

Το 5G στοχεύει στην ανταπόκριση του μεγαλύτερου εύρους υπηρεσιών και εφαρμογών στην ιστορία των κινητών και ασύρματων δικτύων, οι οποίες κατηγοριοποιούνται υπό το βελτιωμένο εύρος ζώνης κινητής δικτύωσης (mobile broadband – eMBB), τη μαζική επικοινωνία μηχανών (massive machine-type communication), και τις υπέρ-αξιόπιστες (ultra-reliable) και χαμηλής καθυστέρησης (low latency) επικοινωνίες (URLLC). Για να ανταποκριθεί σε αυτές τις απαιτήσεις , το 5G σύστημα στοχεύει στην παροχή μιας ευέλικτης (flexible) πλατφόρμας για την ενεργοποίηση νέων επιχειρησιακών περιπτώσεων (business cases) και μοντέλων για ενσωμάτωση βιομηχανικών κλάδων, όπως είναι η αυτοκίνηση, η βιομηχανία και η ψυχαγωγία. Έτσι το 5G εφαρμόζει την τεχνική του network slicing (τμηματοποίηση δικτύου), το οποίο αναδεικνύεται για την προσκόλληση των διαφορετικών τεχνολογικών και επιχειρησιακών αναγκών[61].

Για την επίτευξη αυτού του στόχου το network slicing χρειάζεται να σχεδιαστεί από μία end-to-end σκοπιά, γεφυρώνοντας διαφορετικούς τεχνικούς τομείς (όπως πυρήνα, μεταφορά και δίκτυα πρόσβασης) και διαχειριστικούς (administrative) τομείς (όπως διαφορετικούς operators κινητών δικτύων), συμπεριλαμβανομένων των πεδίων διαχείρισης (management) και του συντονισμού (orchestration). Επιπλέον, στην συνολική αρχιτεκτονική πρέπει να ενσωματωθεί μία αρχιτεκτονική ασφάλειας, για περιπτώσεις όπως η διασφάλιση των προαπαιτούμενων των εφαρμογών και υπηρεσιών που αφορούν κρίσιμες σε ασφάλεια περιπτώσεις χρήσης[61]. Το network slicing περιγράφεται πιο αναλυτικά σε επόμενη ενότητα.



Σχήμα 41: Βασική αρχιτεκτονική του 5G [61]

4.1.4.2 Υπηρεσίες, εφαρμογές και περιπτώσεις χρήσης του 5G

Για να γίνουν καλύτερα κατανοητές οι απαιτήσεις για την 5G δικτυακή υποδομή, παρατίθενται ένας αριθμός από περιπτώσεις χρήσης (use cases) της 5G τεχνολογίας.

Μέσω της αλληλεπίδρασης με την κοινωνία της βιομηχανίας, προκύπτει ένας πρόσθετος αριθμός από από περιπτώσεις χρήσης. Πολλές διαθέσιμες περιπτώσεις χρήσης είναι παραλλαγές από μικρά σύνολα, από βασικές 5G κατηγορίες υπηρεσιών, οι οποίες έχουν συγκεντρωθεί και συμφωνηθεί ως περιεχόμενο του 5G-PPP και διαφορετικών SDOs (service data objects) ως[61]:

- Enhanced Mobile Broadband (eMBB): για περιπτώσεις με αυξημένες ανάγκες σε εύρος ζώνης.
- Ultra-Reliable and Low Latency Communications (URLLC): αναφέρεται στην ανάγκη για αξιοπιστία και μικρό χρόνο καθυστέρησης.
- Massive Machine Type Communications (mMTC): αναφέρεται στην επικοινωνία μεταξύ μεγάλου αριθμού μηχανών.

Επιπρόσθετες περιπτώσεις χρήσης είναι πιθανόν να αναδειχθούν μελλοντικά. Έτσι για μελλοντικά συστήματα 5G είναι απαραίτητο χαρακτηριστικό η ευελιξία, για την προσαρμογή σε νέες περιπτώσεις με μεγάλο εύρος απαιτήσεων[61].

Κάποιες αναδεικνυόμενες σημαντικές ομάδες περιπτώσεων χρήσης και απαιτήσεων για το 5 είναι οι εξείς[61]:

- Dense urban (αναφέρεται στην αυξημένη πυκνότητα κυψελών και τερματικών συσκευών σε αστικό επίπεδο)
- Broadband (50+ Mbps) everywhere (απαίτηση για εύρη ζώνης μεγαλύτερα των 50Mbps παντού)
- Connected vehicles (συνδεσιμότητα με οχήματα – μέρος της ιδέας του Internet of Things)
- Future smart offices (έξυπνα γραφεία)
- Low bandwidth IoT (αναφέρεται στην εξυπηρέτηση των συνδεδεμένων συσκευών με το IoT οι οποίες δεν απαιτούν μεγάλο εύρος ζώνης)
- Tactile internet / automation (αυτοματισμοί παρεχόμενοι μέσω του διαδικτύου)

Όσον αφορά τις περιπτώσεις χρήσης και τους σχετικούς σημαντικούς δείκτες απόδοσης (key performance indicators – KPIs), τα οποία έχουν να κάνουν με την δικτυακή εγκατάσταση και λειτουργία, έχουμε τις επόμενες περιπτώσεις[61]:

- Network slicing, το οποίο αφορά την ικανότητα δημιουργίας end-to-end τμημάτων (slices) πάνω στην ίδια υποδομή, για ετερογενείς υπηρεσίες.

- Multi-tenancy, το οποίο αφορά την ικανότητα για προσφορά υπηρεσιών συνδεσιμότητας σε πολλαπλούς ένοικους και να συνδυάζει πόρους από διαφορετικούς operators.

- Flexibility, δηλαδή ευελιξία, που αφορά την δυνατότητα για δυναμική τροποποίηση των δικτύων στο χρόνο και στον χώρο, εξαρτημένο σε προβλεπόμενα και μη, γεγονότα.

Ακόμα αξίζει να αναφερθούν κάποια σημαντικά χαρακτηριστικά που απαιτούνται από την τεχνολογία, ώστε να καταστεί εμπορικά επιτυχής[61].

- Χρόνος εγκατάστασης υπηρεσίας (service deployment time): ορίζεται ως η διάρκεια που απαιτείται για εγκατάσταση (setup) των end-to-end logical network slices, που χαρακτηρίζονται από ανάλογες εγγυήσεις επιπέδου δικτύου.

- Όγκος δεδομένων (data volume): η ποσότητα πληροφορίας που μεταφέρεται σε ένα χρονικό διάστημα πάνω σε μία αφοσιωμένη περιοχή.

- Αυτονομία (autonomy): δηλαδή η χρονική διάρκεια για ένα συστατικό στοιχείο (component) να γίνει λειτουργικό, χωρίς παροχή ισχύος. Σχετίζεται με τον χρόνο ζωής της μπαταρίας, τη χωρητικότητα μπαταρίας, και την απόδοση (energy efficient).

- Ασφάλεια (security): ορίζεται ως ένα χαρακτηριστικό του συστήματος για διασφάλιση της προστασίας των πόρων και την κάλυψη διάφορων διαστάσεων, όπως μεταξύ άλλων, πιστοποίηση (authentication), εμπιστευτικότητα δεδομένων, ακεραιότητα δεδομένων, έλεγχος πρόσβασης και μη-απόρριψη (non repudiation).

4.1.4.3 Network Slicing

Το 5G δίκτυο αναμένεται να εμπλέκει την ενσωμάτωση διάφορων δικτύων, από διαφορετικούς τομείς , και τα 5G υποσυστήματα θα επιτρέπουν λογικά τμήματα δικτύου (logical network slices) μεταξύ πολλαπλών τομέων και τεχνολογιών για τη δημιουργία υπηρεσιών προσανατολισμένων σε ένοικους ή σε υπηρεσίες. Το network slicing πραγματοποιεί την end-to-end (E2E) ιδέα, ξεκινώντας με το mobile edge, έπειτα με το mobile transport, συμπεριλαμβανομένων των fronthaul (FH) και backhaul (BH) τμημάτων, έως τον πυρήνα του δικτύου (core network – CN). Αυτό θα επιτρέψει στους operators να παρέχουν δίκτυα βάσει της αρχής “as-a-Servive”, και να να ανταποκριθούν στο μεγάλο εύρος από περιπτώσεις χρήσης (use cases) που αναμένεται να αναδειχθούν[61].

Ενώ τα παραδοσιακά (legacy) συστήματα κινητής δικτύωσης όπως το 4G, φιλοξενούν πολλαπλές τηλεπικοινωνιακές υπηρεσίες (όπως τα MBB, ήχος, SMS) στην ίδια αρχιτεκτονική (LTE/EPC), το network slicing στοχεύει στην οικοδόμηση αφοσιωμένων λογικών δικτύων, τα οποία παρουσιάζουν διάφορες λειτουργικές τροποποιήσεις, ανάλογα την τηλεπικοινωνιακή υπηρεσία (eMMB, V2X, URLLC, mMTC). Ακόμα τα παραδοσιακά συστήματα χαρακτηρίζονται από μονολιθικά δικτυακά στοιχεία, που συνδέονται στενά με υλικό, λογισμικό και λειτουργικότητα. Αντίθετα στην 5G αρχιτεκτονική, αποσυνδέονται οι δικτυακές λειτουργίες που βασίζονται στο λογισμικό, από τους πόρους της υποκείμενης υποδομής, με χρήση διαφορετικών τεχνολογιών

αφαίρεσης πόρων (different resource abstraction technologies). Δηλαδή πραγματοποιείται μεταφέροντας αυτές τις λειτουργίες στο λογισμικό (“softwarisation”), μέσω των NVF και SDN. Λειτουργίες multitasking και multiplexing επιτρέπουν τον διαμοιρασμό της υλικής υποδομής που δεν είναι virtualised[61].

Έτσι με την χρήση του NVF και του SDN, δημιουργούνται πολλαπλά end-to-end εικονικά – λογικά δίκτυα ανεξάρτητα από το υλικό, και απομονωμένα μεταξύ τους, τα οποία τρέχουν πάνω σε μία κοινή, διαμοιραζόμενη υποδομή. Αυτό μας επιτρέπει να εφαρμόζουμε αυτά τα εικονικά δίκτυα, προσαρμοσμένα στις εκάστοτε ανάγκες του κάθε χρήστη, προσφέροντάς του καλύτερο QoE (quality of experience), καθώς επίσης την δυνατότητα στους διαχειριστές δικτύου για καλύτερη διαχείριση των δικτυακών πόρων και υπηρεσιών.

Το network slicing είναι ένας συνδυασμός από επαρκώς ρυθμισμένες (configured) λειτουργίες δικτύου, δικτυακές εφαρμογές και της υποκείμενης cloud υποδομής (από φυσικούς, εικονικούς ή ακόμα και εξομοιωμένους πόρους, πόρους RAN κλπ.) τα οποία συγκεντρώνονται μαζί ώστε να πληρούνται οι απαιτήσεις για συγκεκριμένη περίπτωση χρήσης (specific use case), όπως για παράδειγμα συγκεκριμένες απαιτήσεις σε εύρος ζώνης, καθυστέρησης (latency), επεξεργασίας (processing) και ανθεκτικότητα (resiliency) για συγκεκριμένο επιχειρησιακό σκοπό[61].

Με λίγα λόγια, το network slicing είναι μία end-to-end ιδέα που καλύπτει όλα τα δικτυακά τμήματα συμπεριλαμβανομένων των radio networks, wire access, core, και edge networks[61].

4.2 M-CORD

4.2.1 Εισαγωγή

Τα τελευταία χρόνια παρατηρείται τα κινητά δίκτυα να “επιβαρύνονται” με μία διαρκώς αυξανόμενη απαίτηση από τους χρήστες, μέσω των έξυπνων κινητών συσκευών τους όπως είναι τα smartphones, για υπηρεσίες που προσφέρουν υψηλής ποιότητας video, streaming, online – gaming και άλλα. Αυτό όπως είναι σαφές πολλαπλασιάζει την κίνηση και απαιτεί από τους παρόχους των κινητών τηλεπικοινωνιών την ανάγκη για κάλυψη αυτών των απαιτήσεων υπηρεσιών, όπως είναι η παροχή αυξημένου εύρους ζώνης και μικρής καθυστέρησης. Επιπρόσθετα οι ενεργές έξυπνες κινητές συσκευές, όπως τα smartphones συνεχώς αυξάνονται, και πλέον προστίθενται και νέου είδους συσκευές (οικιακές ηλεκτρικές συσκευές, αυτοκίνητα, έξυπνα σπίτια και έξυπνες πόλεις) οι οποίες ακολουθούν την ιδέα του Internet of Things (IoT) και συνδέονται με τα ασύρματα δίκτυα, καθιστώντας το πρόβλημα ακόμα μεγαλύτερο.

Επίσης η σημερινή κινητή τηλεπικοινωνιακή υποδομή, που βασίζεται στο LTE, έχει οικοδομηθεί με ιδιόκτητο κλειστό εξοπλισμό – συσκευές, οι οποίες δεν επιτρέπουν την αποδοτική χρήση των δικτυακών πόρων. Ακόμα είναι δύσκολο να τροποποιηθεί το ισχύον δίκτυο για διαφορετικές ανάγκες πελατών ή τοποθεσιών και η αρχιτεκτονική του δεν παρέχεται για τη δημιουργία των αναδεικνυομένων υπηρεσιών[65].

Το 5G, η τεχνολογία επόμενης γενιάς, έχει προταθεί για την παροχή ενός αποδοτικού και ευέλικτου πλαισίου (framework) για διαχείριση των πόρων και οργάνωση δικτύου, για εφαρμογές και υπηρεσίες κινητών δικτύων, όπως αυτές που αναφέραμε παραπάνω. Ακόμα σκοπός του 5G είναι να επεκτείνει τον ρόλο των κινητών δικτύων, ώστε να ενσωματώσουν την ιδέα του Internet of Things, καθώς και επιχειρησιακών απαιτήσεων. Ένα από τα σημαντικά χαρακτηριστικά του 5G είναι το network slicing και η διαχείριση των slices, πράγμα που συνεισφέρει στην καλύτερη διαχείριση των δικτυακών πόρων και απόδοση υπηρεσιών βάσει του απαιτούμενου QoS (Quality of Service)[63].

Από τα παραπάνω, παρατηρείται η ανάγκη για μία πλατφόρμα αναφοράς για το 5G, η οποία θα είναι ανοιχτού κώδικα και προγραμματίσιμη για γρήγορη εξέλιξη, καινοτομία, επέκταση και πειραματισμό. Η πλατφόρμα ατή πρέπει να ικανοποιεί τα εξείς προαπαιτούμενα[65]:

- Προγραμματισιμότητα (Programmability): Θα πρέπει οι προγραμματιστές να έχουν την δυνατότητα να αναπτύσσουν νέες δυνατότητες υποδομής, καθώς και νέες υπηρεσίες για περιπτώσεις χρήσης, των οποίων οι απαιτήσεις σε δίκτυο διαφέρουν.

- Ενορχήστρωση (Orchestrability): Η αρχιτεκτονική θα πρέπει να παρέχει μία ανοιχτή πλατφόρμα ενορχήστρωσης για τον λειτουργό (mobile operator), πάνω στην οποία
  οι καινοτομίες για νέες υπηρεσίες, η προμήθευση υπηρεσιών και η ενορχήστρωση, είναι δυναμικά εφικτές.

- Επεκτασιμότητα (Scalability): Η πλατφόρμα πρέπει να είναι επεκτάσιμη, τόσο όσον αφορά το υλικό όσο και το λογισμικό, όπως γίνεται στα σύγχρονα cloud συστήματα.

- Χρήση εμπορικού υλικού (Use of Merchant Silicon): Η πλατφόρμα θα πρέπει να εκμεταλλεύεται την διάσπαση (disaggregation) των παραδοσιακών συσκευών και τη μέγιστη δυνατή χρήση εμπορικού κοινού υλικού για ουσιαστική μείωση του CAPEX/OPEX και την αποφυγή εξάρτησης από ιδιώτη.

- Χρήση λογισμικού ανοιχτού κώδικα (Use of Open Source Software): Για την κάλυψη των αναγκών για πειραματισμό και ανάπτυξη από το ευρύ προγραμματιστικό κοινό.

- Προγραμματιστικός έλεγχος κλειστής επανάληψης για αυτό-οργανώμενα δίκτυα (Programmatic Closed Loop Control for Self-Organized Networks): Η αρχιτεκτονική των 5G δικτύων απαιτεί δυνατότητα παρατήρησης (observability) σε πραγματικό χρόνο (real time), και ανοιχτά ορισμένες αφαιρέσεις για αυτό-οργανώμενα δίκτυα και έξυπνες εφαρμογές.

Καθώς έχει παρουσιαστεί η γενική τεχνολογία του CORD, γίνεται φανερό ότι η λύση για κάλυψη αυτών των αναγκών δίνεται με τον καλύτερο τρόπο από το CORD, το οποίο όπως πλέον έχει γίνει σαφές, αποτελεί μία πλατφόρμα όπου συνδυάζει τις τεχνολογίες NFV, SDN και cloud, με ανοιχτό κώδικα και πάνω σε "ανοιχτό" κοινής χρήσης υλικό (white boxes). Αναφέρθηκε επίσης ότι το CORD κατακερματίζει τις δικτυακές λειτουργίες και τις υλοποιεί μέσω επεκτάσιμων υπηρεσιών[4]. Για την περίπτωση των κινητών δικτύων έχει αναπτυχθεί η αντίστοιχη mobile έκδοση του CORD, δηλαδή το M-CORD, το οποίο θα αναλυθεί στη συνέχεια, καθώς και κάποιες σημαντικές εφαρμογές του, οι οποίες δίνουν λύση σε βασικές ανάγκες για την αποδοτική υλοποίηση του 5G.

4.2.2 Το M-CORD και η αρχιτεκτονική του

Το M-CORD είναι το κομμάτι του γενικού CORD project το οποίο πραγματεύεται με το μέρος των κινητών – ασύρματων δικτύων. Έτσι λοιπόν συμπεριλαμβάνει τα βασικά συστατικά και τη φιλοσοφία του CORD για την παραγωγή και τη διαχείριση υπηρεσιών κινητής δικτύωσης.

Έτσι χρησιμοποιεί κατά τα γνωστά[65]:

- OpenStack: πλατφόρμα διαχείρισης για cloud – datacenter για παροχή "υποδομής ως υπηρεσία" (IaaS).
- Docker: η πλατφόρμα λογισμικού που επιτρέπει την εφαρμογή και διασύνδεση υπηρεσιών μέσα σε containers.
- ONOS: το οποίο ελέγχει την υποκείμενη white-box switch υποδομή, οργανωμένη σε spine-leaf τοπολογία, και φιλοξενεί μία συλλογή από εφαρμογές ελέγχου, οι οποίες εγκαθιστούν τις απαραίτητες υπηρεσίες, ελεγχόμενες από το ONOS, και ενσωματωμένα εικονικά δίκτυα στην υποκείμενη υποδομή για υπηρεσίες, ελεγχόμενες από το OpenStack.
- XOS: το οποίο είναι το λειτουργικό σύστημα που αποδίδει την έννοια "όλα ως υπηρεσία" (XaaS), όντας υπεύθυνη για τη συγκρότηση και τη σύνθεση των υπηρεσιών.

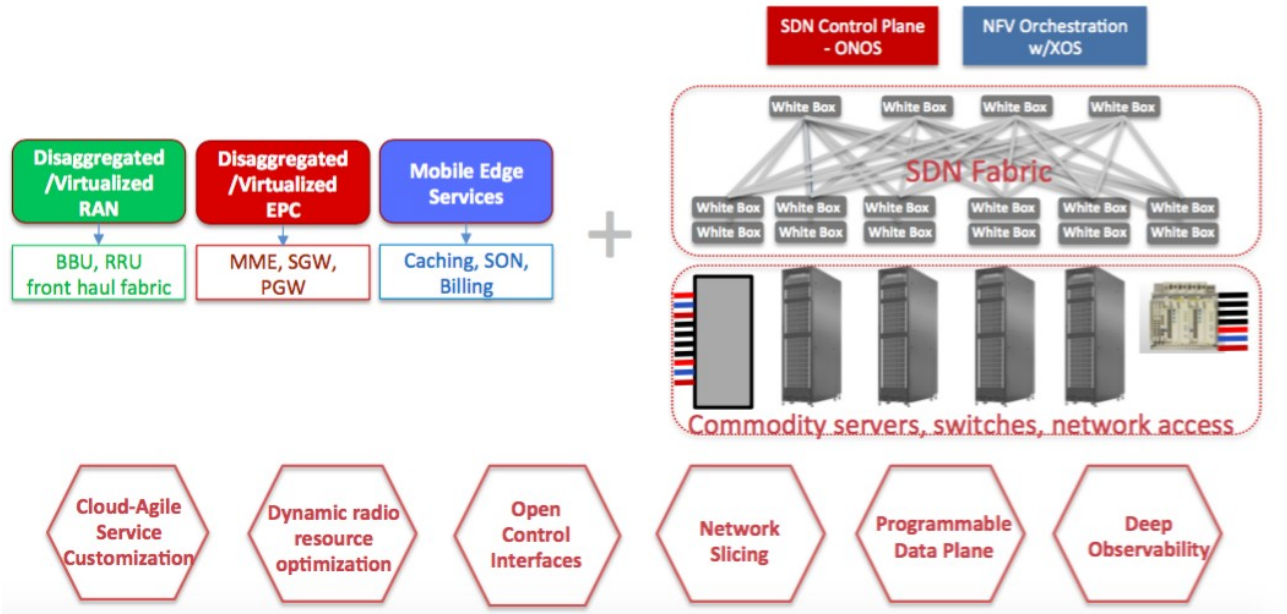
Το M-CORD είναι επηρεασμένο από το 5G και τις αναδεικνυόμενες περιπτώσεις χρήσης του και είναι προγραμματιστικά εφαρμόσιμο σε ένα εύρος από εκτελεστικούς στόχους (performance targets) πάνω στην ίδια πλατφόρμα[4]. Μεταμορφώνει το κινητό δίκτυο έτσι ώστε τα SDN control plane και data plane να διαχωρίζονται, με το SDN control plane να είναι λογικά κεντρικοποιημένο. Οι λειτουργίες κυψελωτού δικτύου καθώς και συγκεκριμένες υπηρεσίες διαχειριστή (operator specific services) είναι διασπασμένες (disaggregated) και εικονικοποιημένες (virtualised). Οι εικονικές λειτουργίες και υπηρεσίες με τη σειρά τους συνθέτονται ως επεκτάσιμες υπηρεσίες και το συνολικό κυψελωτό δίκτυο είναι οργανωμένο, ώστε σύνολα υπηρεσιών που είναι προσανατολισμένα σε συγκεκριμένες περιπτώσεις χρήσης, να είναι έτοιμα και δυναμικά επεκτάσιμα (dynamically scaled)[65].

Όπως αναφέρθηκε το M-CORD είναι δομημένο με ανοιχτού κώδικα λογισμικό και πάνω σε κοινού εμπορικού τύπου υλικό. Αυτό επιτρέπει στην αρχιτεκτονική του M-CORD να είναι πολύ οικονομική και μειώνει το κόστος στησίματος (setup) των data-centers. Επίσης αναφέραμε ότι δομείται συνδυάζοντας τις βασικές έννοιες του CORD, που είναι το SDN, το NFV και το Cloud, μειώνοντας έτσι το CAPEX και OPEX κόστος και βελτιώνει την ευελιξία και την επεκτασιμότητα των εφαρμογών και των υπηρεσιών[62].

Το M-CORD στοχεύει στο να παρέχει βελτιωμένη χρήση πόρων, παρέχοντας διαχείριση πόρων σε πραγματικό χρόνο, καταγραφή (monitoring) του framework και εκμετάλλευση της χρήσης πολλαπλών τεχνολογιών ραδιο-πρόσβασης RAT (Radio Access Technologies). Μπορεί να χρησιμοποιηθεί για παροχή προσαρμοσμένης (customized) σύνθεσης υπηρεσιών και διαφοροποίηση του QoE που βασίζεται πάνω στις διαφορετικές απαιτήσεις υπηρεσιών ώστε να παρέχεται καλύτερη προσαρμοσμένη υπηρεσία και υψηλότερη ποιότητα εμπειρίας των πελατών[62].

Η αρχιτεκτονική του M-CORD συνδυάζει, αποδομημένο και εικονικοποιημένο Radio Access Network - RAN (disaggregated /virtualised RAN), αποδομημένο και εικονικοποιημένο

Evolved Packet Core - EPC (disaggregated /virtualised EPC) και τις Mobile Edge υπηρεσίες. Η CORD αρχιτεκτονική δημιουργεί μία πλατφόρμα που παρέχει cloud ευέλικτες, προσαρμοσμένες υπηρεσίες με αποδοτικό network slicing και ενδελεχή παρατηρησιμότητα (observability). Διαθέτει δυναμική βελτιστοποίηση ραδιο-πόρων και προγραμματίσιμο data-plane, καθιστώντας το, ισχυρό και ευέλικτο[62].



Σχήμα 42: Το M-CORD ως συνδυασμός της σύγχρονης κινητής δικτύωσης και του CORD [78]

4.2.2.1 Disaggregated/virtualised RAN

Το Radio Access Network, σύμφωνα με την τρέχουσα αρχιτεκτονική του LTE περιλαμβάνει τους σταθμούς βάσεις που καλούνται eNodeBs. Το κάθε eNodeB όπως αναφέρθηκε, αποτελείται από δύο τμήματα, το BBU (Baseband Unit) και το RRU (Remote Radio Unit). Το RRU (ή αλλιώς RRH) είναι το υπεύθυνο μέρος του σταθμού βάσης, για λειτουργίες που αφορούν τις ραδιοσυχνότητες, δηλαδή αποτελούν κατά κάποιον τρόπο τις κεραίες του σταθμού βάσης. Το BBU από την άλλη, είναι το μέρος που είναι υπεύθυνο για την ψηφιακή επεξεργασία των σημάτων. Στην υλοποίηση του M-CORD, όπου εφαρμόζεται κεντρικοποιημένη εικονική αρχιτεκτονική RAN, τα BBUs μεταφέρονται ως εικονικές εκδοχές (vBBU) στην πλευρά του EPC.

Μία πρόσφατη τάση στο περιβάλλον του 5G επιδιώκει προσεγγίσεις πολλαπλών τμηματοποιήσεων για τους σταθμούς βάσης, κατάλληλες για μικρές κυψέλες που επιφέρουν μείωση στις δαπάνες που αφορούν το setup και τη συντήρηση. Οι λύσεις εικονικού RAN αποτελούν ένα αναπόσπαστο κομμάτι της εξέλιξης της τρέχουσας αρχιτεκτονικής δικτύων προς την αρχιτεκτονική του 5G. Αυτή η προσέγγιση μας παρέχει ικανότητα για διαχείριση και καθοδήγηση της χωρητικότητας (capacity) του δικτύου για περιοχές αυξημένης ζήτησης σε δυναμικό περιβάλλον. Επιπλέον, αυτό αποτελεί καλύτερο σενάριο από την παραδοσιακή αρχιτεκτονική, όπου το δίκτυο στήνεται έτσι ώστε να πληρεί τη μέγιστη αναμενόμενη ζήτηση, σε κάθε κόμβο.

Συμπερασματικά με τη νέα τροποποίηση του RAN, παρατηρείται ότι επιτυγχάνεται υψηλή ευελιξία και επεκτασιμότητα (scalability), κεντρικοποιημένος συντονισμός, βελτιστοποίηση της χρήσης του φάσματος και μείωση του κόστους.

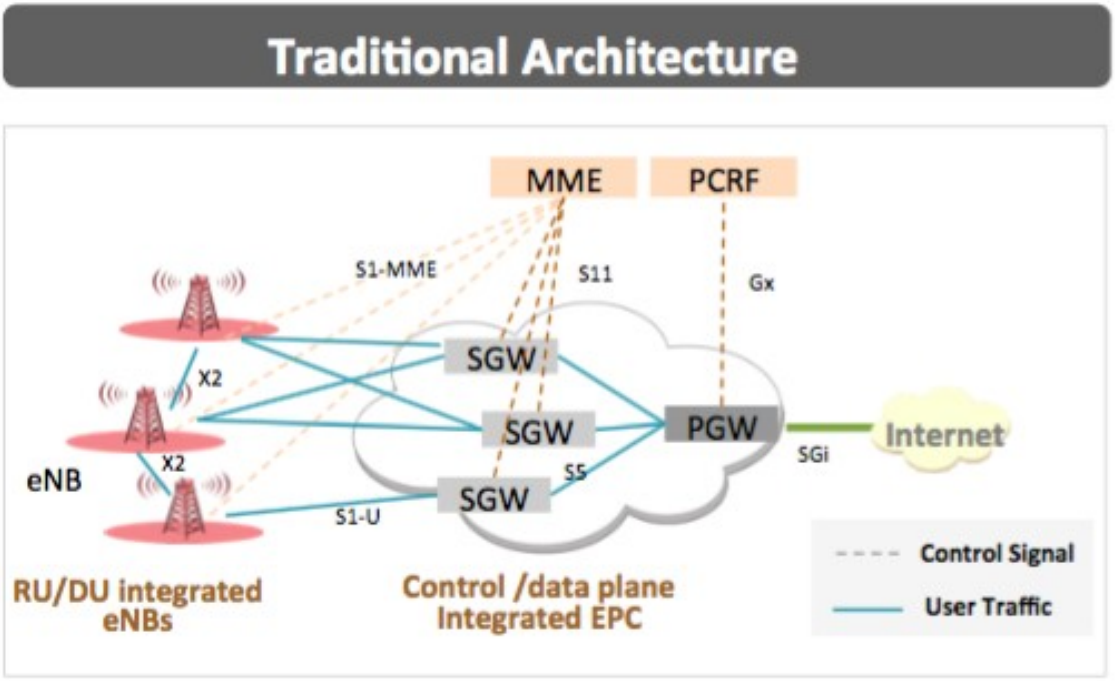
4.2.2.2 Disaggregated/virtualised EPC

Όσον αφορά το EPC, όπως προαναφέρθηκε, ενοποιεί τον ήχο και την κίνηση δεδομένων και χρησιμοποιεί υπηρεσίες του Internet Protocol (IP) για τη μεταφορά πακέτων. Περιλαμβάνει το 1) MME (Mobile Management Entity) για εκτέλεση των καταστάσεων συνόδου, της πιστοποίησης μαζί με ανίχνευση (tracking) χρήστη, 2) SGW (Serving Gateway) το οποίο εκτελεί τη δρομολόγηση των πακέτων δεδομένων μέσα από ολόκληρο το δίκτυο πρόσβασης, 3) το PGW (PDN Gateway) το οποίο παρέχει συνδεσιμότητα στο UE με εξωτερικά δίκτυα πακέτων δεδομένων (packet data networks – PDN) και ακόμα προάγει το QoS παρέχοντας μία διεπαφή μεταξύ του LTE και των προηγούμενων τεχνολογιών, και 4) PCRF (Policy and Rules Function) το οποίο παρέχει ανίχνευση ροής δεδομένων και επιβάλλει πολιτικές (policies) και κανόνες χρέωσης[62].

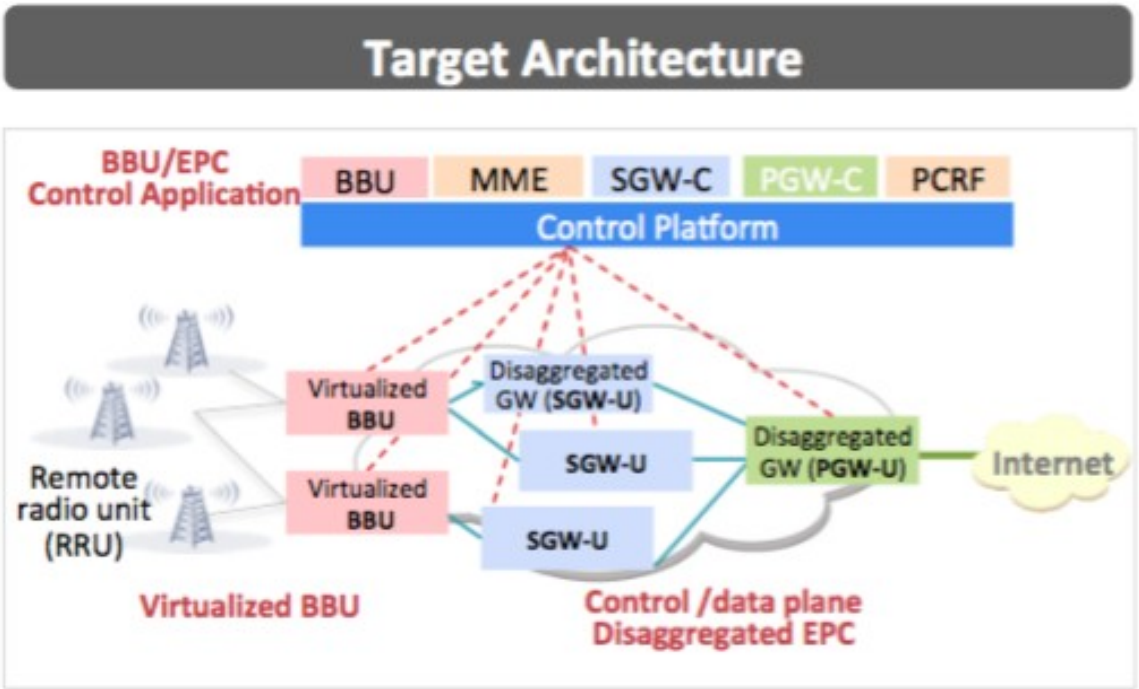
Προχωρώντας στην εικονική εκδοχή του EPC, που επιβάλλει η 5G αρχιτεκτονική, τα παραπάνω στοιχεία αποδομούνται και εικονικοποιούνται (disaggregate/virtualise), οπότε οι νέες εικονικές εκδοχές που προκύπτουν μπορούν να εκτελούνται πάνω σε κοινού σκοπού servers – white boxes και με κεντρικοποιημένο έλεγχο σύμφωνα με το CORD (υπό τον έλεγχο του ONOS). Αναφέρουμε επίσης εδώ ότι οι πύλες SGW και PGW διαιρούνται σε δύο μέρη το καθένα, ένα που αφορά το contol plane και αναφέρονται ως SGW-C και PGW-C αντίστοιχα, και το άλλο αφορά το user plane και καλούνται SGW-U και PGW-U.

Με τη νέα αρχιτεκτονική των αποδομημένων (disaggregated) λειτουργιών – μονάδων του EPC, τα οποία τώρα εκτελούνται σε commodity υλικό, και τα λογικά (control plane) μέρη τους ελέγχονται από μία κεντρική πλατφόρμα ελέγχου, μας επιτρέπει την ανεξάρτητη επεκτασιμότητα, τον κεντρικό ενιαίο έλεγχο, την επιλογή του υλικού που ταιριάζει καλύτερα στο SLA (Service Level Agreement), καθώς και την αναμενόμενη μείωση του κόστους.

Ακολουθούν δύο διαγράμματα, με το πρώτο να απεικονίζει τη γενική αρχιτεκτονική των παραδοσιακών RAN και EPC, και το δεύτερο να απεικονίζει τη νέα αρχιτεκτονική που προκύπτει από τις αποδομημένες/εικονικοποιημένες εκδοχές τους που αναφέραμε.



Σχήμα 43: Παραδοσιακή LTE Αρχιτεκτονική [78]



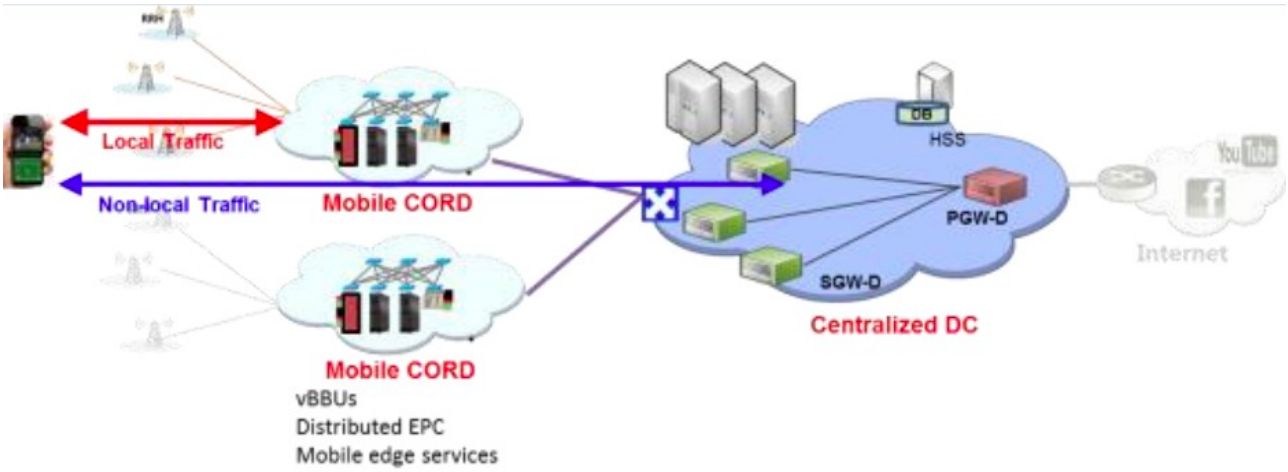
Σχήμα 44: Νέα επιθυμητή αρχιτεκτονική κινητής δικτύωσης [78]

4.2.2.3 Mobile Edge

Με τον ολοένα και αυξανόμενο αριθμό συσκευών, εφαρμογών και υπηρεσιών που συνδέονται στην κινητή δικτύωση, όπως είναι ο ερχομός του IoT, των υπηρεσιών εικονικής πραγματικότητας, το on-line gaming, που όπως αναφέρθηκε και προηγουμένως, απαιτούν ελάχιστες καθυστερήσεις και αυξημένο εύρος ζώνης, αυξάνουν πολύ τον κυκλοφοριακό φόρτο δεδομένων στην επικοινωνία με τον πυρήνα του δικτύου.

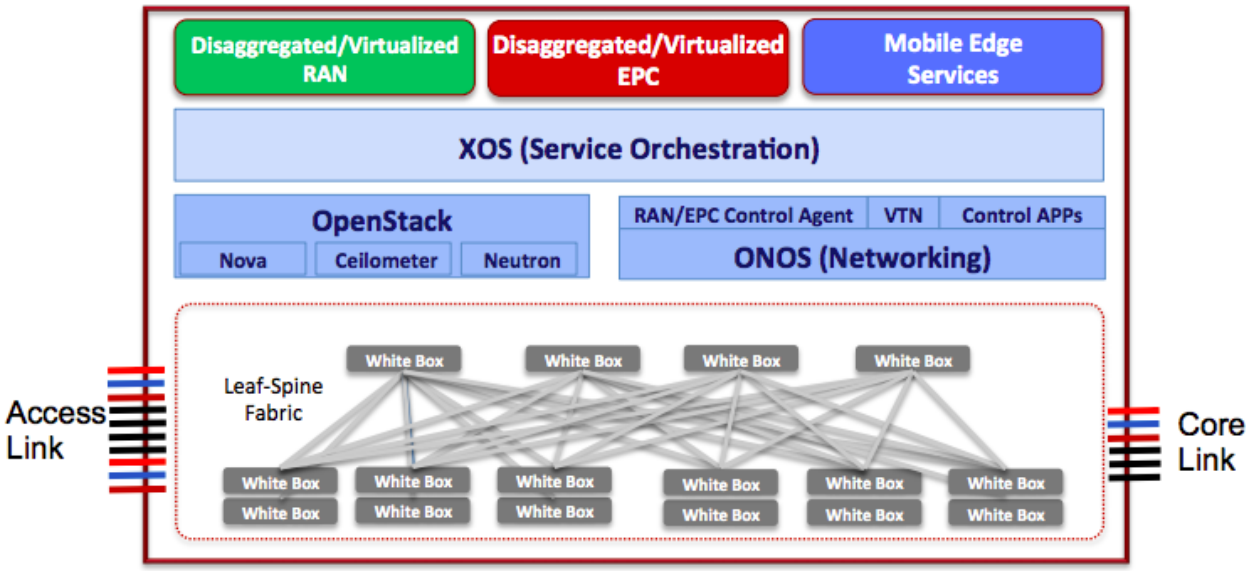
Έτσι η λύση η οποία προτείνεται είναι το mobile edge, που αναφέρεται σε πολλαπλά αποκεντρωμένα συστήματα πυρήνα δικτύου, υπεύθυνο το καθένα για μερικούς σταθμούς βάσεις και αντίστοιχα κυψελών, καθώς και για τις λειτουργίες EPC/vBBU προς αυτές. Με αυτόν τον τρόπο επιμερίζεται ο φόρτος εργασίας και κίνησης από ένα κεντρικό σημείο, σε αρκετά μικρότερα κατανεμημένα σημεία. Ακόμα γίνεται καλύτερη διαχείριση πόρων και επιμέρους υπηρεσιών, μικρότερες καθυστερήσεις και αυξημένη ρυθμαπόδοση (throughput), λόγω της τοπικότητας. Επιπρόσθετα γίνεται δυνατή η εγκατάσταση μικρότερων κυψελών (micro-cells) συμβάλλοντας με τη σειρά τους στην αύξηση της πυκνότητας/χωρητικότητας των χρηστών, στον ακριβέστερο προσδιορισμό της τοποθεσίας και των ίδιων των τερματικών, επιτρέποντας την κατάλληλη διαμόρφωση των υπηρεσιών σύμφωνα με τις συγκεκριμένες ανάγκες σε επίπεδο χρήστη ή τοπικό επίπεδο, επιτυγχάνοντας έτσι καλύτερη εξυπηρέτηση – εμπειρία για τον χρήστη (QoE).

Σύμφωνα με τα παραπάνω το M-CORD αποτελεί μία αποτελεσματική και αποδοτική λύση για το mobile edge. Προς αυτήν την κατεύθυνση κάθε mobile edge, δηλαδή κάθε αποκεντρωμένος – επιμέρους πυρήνας δικτύου, ακολουθεί την τεχνολογία του M-CORD. Η σχεδίαση του mobile edge του M-CORD, περιλαμβάνει: 1) έναν SDN control-plane που χρησιμοποιεί το ONOS για έλεγχο των εικονικών RAN και EPC στοιχείων, 2) πόρους βασισμένους σε SDN/NFV οι οποίες μπορούν να οικοδομηθούν και να τεμαχιστούν (sliced), ώστε να παρέχουν κινητές υπηρεσίες με την αναγκαία απόδοση. Αυτά μαζί χρησιμοποιούνται ώστε να παράξουν διάφορες δικτυακές υπηρεσίες στο mobile edge όπως, εφαρμογές αυτο-οργανώμενου δικτύου (Self-Organising Network – SON) και τοπική αποθήκευση (caching)[64].



Σχήμα 45: Το M-CORD στην υλοποίηση του mobile edge [65]

Σύμφωνα με όλα τα παραπάνω αποτυπώνεται η συνολική εικόνα του M-CORD, το οποίο επιτυγχάνει τις παραπάνω προσεγγίσεις προς τη νέα σύγχρονη κινητή δικτύωση. Ακολουθεί μία εικόνα που παρουσιάζει τη γενική αρχιτεκτονική του M-CORD.

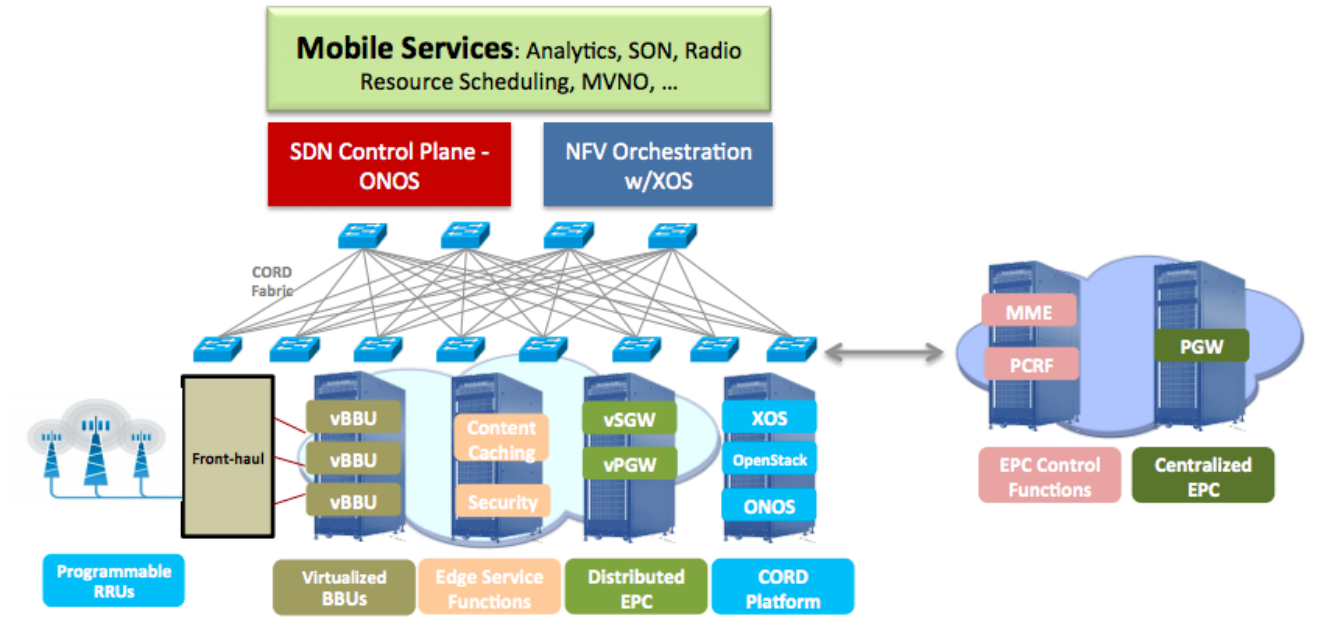


Σχήμα 46: Αρχιτεκτονική του M-CORD [78]

Κατά τα γνωστά, από την σχεδίαση του CORD, το OpenStack χρησιμοποιείται για να παράξει εικονικές μηχανές, βάσει της υποκείμενης υποδομής, με τις οποίες στη συνέχεια θα υλοποιηθούν τα containers από το Docker, μέσα στα οποία θα εκτελεστούν οι υπηρεσίες. Το ONOS επιτελεί τον λογικό έλεγχο των υπηρεσιών, δηλαδή διαχειρίζεται το control plane, καθώς επίσης παρέχει εφαρμογές ελέγχου του συστήματος, και άλλες σημαντικές εφαρμογές όπως το VTN που συνεισφέρει στη σύνθεση και διασύνδεση των υπηρεσιών. Το XOS βρίσκεται στην κορυφή ως το σύστημα της παραγωγής των υπηρεσιών, της διαχείρισής τους, καθώς και της ενοποίησης και ενορχήστρωσης του συνολικού συστήματος.

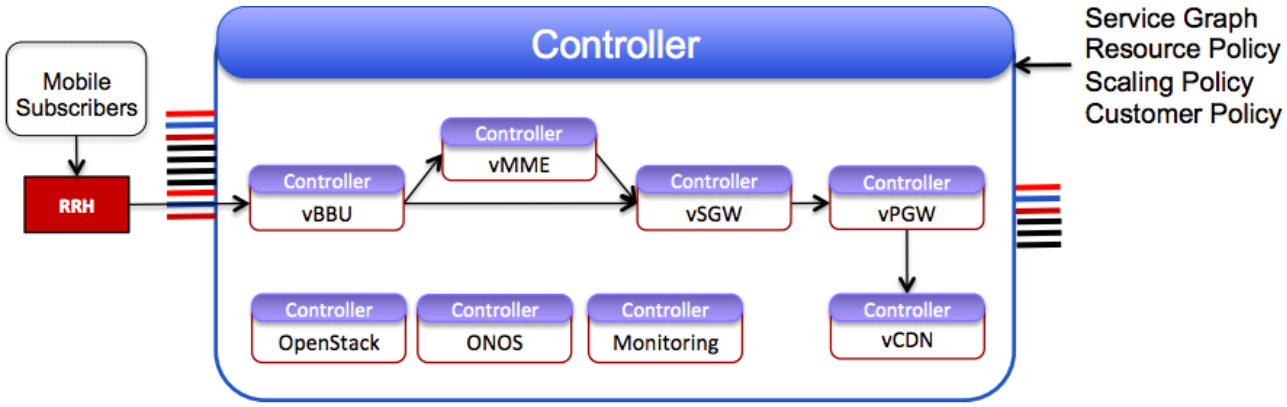
Έτσι παράγονται οι εικονικές υπηρεσίες κινητής δικτύωσης, που αντιστοιχούν στις λειτουργίες και οντότητες των παραδοσιακών RAN και EPC. Δηλαδή παράγουμε, από τη μεριά του RAN, την αντίστοιχη εικονική έκδοση του BBU (Baseband Unit), που την αναφέρουμε ως vBBU και πλέον μεταφέρεται στη μεριά του EPC, μέσα στο M-CORD κεντρικό σύστημα. Και από την πλευρά του EPC, έχουμε τις εικονικές λειτουργίες - υπηρεσίες: vSGW, vPGW, vMME, αντίστοιχες των παραδοσιακών μονάδων που έχουμε αναφέρει. Με αυτόν τον τρόπο οι δικτυακές λειτουργίες πλέον εφαρμόζονται ως εικονικές υπηρεσίες στην υποδομή του M-CORD που είναι υλοποιημένη σε commodity hardware.

Ακολουθεί ένα σχήμα που απεικονίζει την εφαρμογή (implementation) του M-CORD ως mobile edge.



Σχήμα 47: Εφαρμογή του M-CORD [78]

Στο επόμενο σχήμα απεικονίζεται ένα γράφημα υπηρεσιών (service graph) που παρουσιάζει την εσωτερική λειτουργία του M-CORD, ως αλληλουχία των βασικών υπηρεσιών του.



Σχήμα 48: Εσωτερική όψη της λειτουργίας του M-CORD ως μία σειρά εκτελούμενων υπηρεσιών [78]

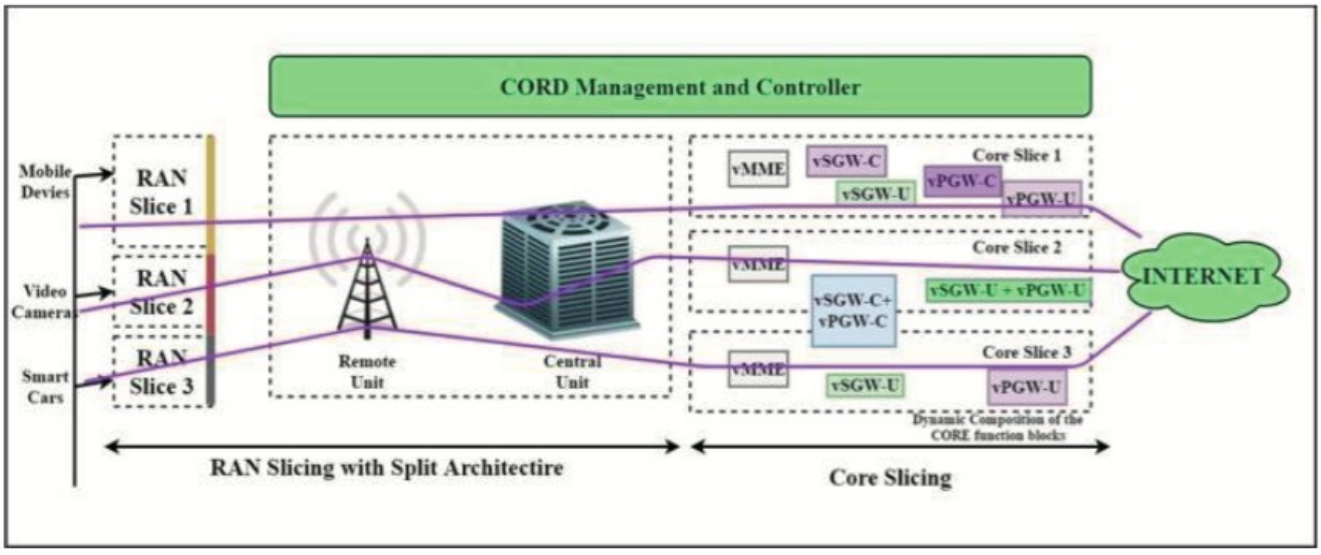
4.2.3 Εφαρμογές του M-CORD

Συνεχίζοντας, ολοκληρώνεται το κεφάλαιο με μία σύντομη περιγραφή βασικών εφαρμογών και καινοτομιών που προσφέρονται από το M-CORD και βελτιώνουν σημαντικά την κινητή δικτύωση.

4.2.3.1 End-to-End Slicing

Καθώς ένας από τους στόχους του 5G είναι η παροχή πολλαπλών ασύρματων υπηρεσιών στοχευμένων σε διαφορετικές περιπτώσεις με διαφορετικές απαιτήσεις και quality of service, απαιτείται μία πλατφόρμα που θα παρέχει με δυναμικό τρόπο εικονικά δίκτυα με διαφορετικά χαρακτηριστικά ώστε να πληρούν αυτές τις επιμέρους απαιτήσεις. Για αυτόν τον λόγο, χρησιμοποιείται σαν λύση το network slicing, το οποίο παρέχεται ως μία εφαρμογή από το M-CORD πάνω στα υπάρχοντα LTE δίκτυα.

Το network slicing χρησιμοποιεί χαρακτηριστικά – κλειδιά του M-CORD όπως είναι η αποδόμηση και η εικονικοποίηση του RAN και του EPC, ανοιχτού κώδικα δομικά στοιχεία (open source building blocks) για το RAN και το EPC, τα οποία επιτρέπουν την προσαρμογή και διαμόρφωση, και ακολουθεί πλήρως την λογισμικά ορισμένη προσέγγισή του. Το M-CORD παρέχει την απαραίτητη λειτουργικότητα για τον προγραμματισμό των ορισμάτων (definitions) των sclices και συνδέει κατάλληλα τα RAN και CORE slices. Στην πλευρά του RAN, η διαδικασία της τμηματοποίησης (slicing) περιλαμβάνει την εικονικοποίηση μπλοκ πόρων, και διευθέτηση των υποσυνόλων αυτών των μπλοκ πόρων σε διαφορετικά slices και στις αντίστοιχες λειτουργίες ελέγχου, όπως είναι ο χρονοπρογραμματισμός (scheduling), το handoff, και ο έλεγχος αποδοχής. Στην πλευρά του πυρήνα (CORE) του δικτύου, το slicing περιλαμβάνει συσχετισμένα, αποδομημένα εικονικά EPC στοιχεία σε κάθε slice[65].



Σχήμα 49: End-to-end slicing [62]

4.2.3.2 Εφαρμογές στον Πυρήνα (CORE)

4.2.3.2.1 Βελτιστοποιημένος πυρήνας για στατικές IoT συσκευές

Καθώς οι τύποι και ο αριθμός των συσκευών, που συνδέονται με το internet διαρκώς αυξάνεται όσο προχωράμε στην τάση του Internet of Things – IoT, και σε συνδυασμό με τον αυξανόμενο αριθμό των προσφερόμενων υπηρεσιών και εφαρμογών στις παραδοσιακές έξυπνες κινητές συσκευές, όπως τα smartphones, τα κινητά δίκτυα δέχονται ένα μεγάλο overhead που προέρχεται από την control plane σηματοδότηση.

Το M-CORD δίνει λύση στο πρόβλημα ενσωματώνοντας τις MME αποσυνδεμένες λειτουργίες, την δυνατότητα εφαρμογής slicing στον πυρήνα, την υπηρεσία επιλογής για slicing, και μία νέα connectionless πύλη (gateway) για την εξάλειψη του παραδοσιακού GTP tunneling overhead για IoT κίνηση[65].

Ο βελτιστοποιημένος πυρήνας εφαρμόζει ξεχωριστά slices για τις συνηθισμένες κινητές συσκευές και τις σταθερές IoT συσκευές. Το ανοιχτού κώδικα MME του M-CORD, βελτιώνεται με τη λειτουργικότητα να διαχωρίζει μία συνηθισμένη κινητή συσκευή από μία στατική IoT, βασισμένο σε κριτήρια όπως το IMSI (International Mobile Subscriber Identity) ή το APN (Access Point Name) όνομα. Στην περίπτωση των συνηθισμένων LTE κινητών, η διαχείριση κινητικότητας (mobility management) λαμβάνει τα δεδομένα του χρήστη για διεπαφή με παραδοσιακά SGW και PGW. Ωστόσο, το βελτιωμένο MME συνδέει τις IoT συσκευές απευθείας σε ασύνδεσμη πύλη (connectionless gateway), έναν συνδυασμό SGW και PGW, και έτσι αφαιρεί κάθε περιοδικό σήμα ή σήμα κινητικότητας που σχετίζονται με κίνηση παραδοσιακού LTE.

4.2.3.2.2 Προγραμματίσιμος και επεκτάσιμος ασύνδετος πυρήνας

Οι στατικές συσκευές που συνδέονται με το internet ακολουθώντας τη λογική του IoT που αναφέρθηκε συνήθως εκτελούν απλές λειτουργίες και δεν απαιτούν πολύπλοκες online λειτουργίες, σαν τις παραδοσιακές συσκευές όπως τα smartphones, που εκτελούν μια πληθώρα από εφαρμογές και υπηρεσίες με αυξημένες απαιτήσεις σε δικτυακούς πόρους. Όμως το υπάρχον LTE δίκτυο είναι σχεδιασμένο για την εξυπηρέτηση των συνηθισμένων κινητών συσκευών, και έτσι δεν υπάρχει προβλεπόμενη σχετική ρύθμιση για μειωμένη δαπάνη πόρων σε στατικές υπηρεσίες, ώστε να γίνεται οικονομία στους δικτυακούς πόρους και να χειρίζεται έναν αριθμό από στατικές συσκευές.

Ο σχεδιασμός του M-CORD από την άλλη, επιτρέπει την υποστήριξη τόσο κινητών συσκευών όσο και μεγάλου αριθμού στατικών. Αυτό επιτυγχάνεται μέσω των εξείς σημαντικών μεθόδων[65]:

- Σημαντική μείωση της σηματοδότησης και του overhead για το στατικό IoT, ταξινομώντας την κίνηση στο RAN και δρομολογώντας ασύνδετη (connectionless) μη-ενθυλακωμένης (non-encapsulated) IoT κίνηση απευθείας στο S/P-GW, αλλά επίσης υποστηρίζει την κίνηση παραδοσιακών κινητών συσκευών με GTP tunnels και κανονική σηματοδότηση.

- Χρήση ενός επεκτάσιμου δικτύου πυρήνα βασισμένο σε SDN, με διαχωρισμένα τα control και user planes για τα S/P-GWs το οποίο επιτρέπει ανεξάρτητη οριζόντια επέκταση χρηστών και τα control planes να χειρίζονται την αντίστοιχη κίνηση.

- Επίτευξη υψηλής απόδοσης στο user plane μέσω χρήσης DPDK (data plane development kit) τεχνολογίας.

4.2.3.3 Αυξημένη δημόσια ασφάλεια ως υπηρεσία

Τα τηλεπικοινωνιακά δίκτυα, παραδοσιακά παρέχουν υπηρεσίες κινδύνου και ασφάλειας, όπως οι κλασικές τηλεφωνικές γραμμές προς αστυνομία, την πυροσβεστική και το ΕΚΑΒ, οι οποίες παρέχονται από τους παρόχους άμεσα και χωρίς χρέωση. Στα σημερινά δίκτυα όμως, υπάρχει η ανάγκη για παροχή υπηρεσιών έκτακτης ανάγκης, με περισσότερα είδη πληροφορίας από τις απλές κλήσης φωνής, δηλαδή δύναται να περιέχουν video, τοποθεσία και άλλες χρήσιμες σχετικές πληροφορίες. Έτσι θα πρέπει το δίκτυο να μπορεί να αναγνωρίζει αυτές τις κλήσεις για σχετικές υπηρεσίες ανάγκης, παρέχοντάς τους την αναγκαία κίνηση και ποιότητα υπηρεσίας QoS, χωρίς να διακόπτει την λειτουργία της λοιπής κοινής παρεχόμενης κίνησης.

Για τον σκοπό αυτόν, το M-CORD έχει ενσωματώσει μία υπηρεσία network cookies, βασισμένο σε ένα ανοιχτό και ασφαλές API, για να ταιριάζει (map) την κίνηση, βασισμένο στο network cookie για διαφορετικά SLA (service level agreement) στο δίκτυο, ακόμα και όταν αυτή η κίνηση είναι κρυπτογραφημένη, ή έρχεται από πολλαπλά CDNs (content delivery networks) και δημόσια data-centers. Εξοπλισμένη με το δικό της network cookie, η κίνηση από μία δημόσια ασφαλή εφαρμογή μπορεί να αντιμετωπιστεί ως ένα ειδικό SLA το οποίο αποφεύγει τους χρεωστικούς ελέγχους για τον χρήστη σε έκτακτη ανάγκη, και του παρέχει το αναγκαίο εύρος ζώνης. Η υπηρεσία του network cookie μπορεί να εφαρμοστεί και σε άλλες περιπτώσεις.[65]

4.2.3.4 Προσαρμοστική διαγνωστική υπηρεσία

Η προσαρμοστική διαγνωστική υπηρεσία (Adaptive Analytics Service) επωφελείται από την ικανότητα της M-CORD πλατφόρμας να εκκινεί test και monitoring agents χρησιμοποιώντας εργαλεία σύνθεσης υπηρεσιών σύμφωνα με το μοντέλο του CORD. Η παρακολούθηση ως υπηρεσία (monitoring as a service) του CORD μπορεί να χρησιμοποιηθεί εντός των σχεδιάσεων δικτυακών υπηρεσιών, ώστε να επιτραπεί λειτουργική ετοιμότητα. Το μοντέλο υπηρεσίας που προκύπτει, συνδυάζει έναν ακριβή ορισμό της λειτουργίας της υπηρεσίας και μετρικές ποιότητας που χρειάζονται για να δοκιμαστεί και να διασφαλιστεί η υπηρεσία. Αυτή η καινοτομία συνδυάζει τη χρήση δύο κύριων ιδεών για testing: 1) Παρακολούθηση παθητικής δικτυακής κίνησης η οποία γίνεται από μία υπηρεσία που καλείται Assurance as a Service (AaaS), η οποία συλλέγει μόνο σχετική δικτυακή κίνηση, δημιουργεί τις μετρικές ποιότητας υπηρεσίας που έχουν ορισθεί στο μοντέλο υπηρεσίας, και τα δημοσιεύει μέσω μιας ανοιχτής τροφοδοσίας σε εξωτερικές διαγνωστικές λειτουργίες. 2) Ενεργή μέτρηση ως υπηρεσία, ή αλλιώς Test as a Service (TaaS), η οποία χρησιμοποιεί σύνθετη κίνηση για τροφοδοσία της παρακολούθησης της απόδοσης του μονοπατιού υπηρεσίας (service path performance monitoring) που χρειάζεται από εφαρμογές διάγνωσης, όπως root cause segmentation μαζί με το service path.[65]

5. Συμπεράσματα

Ο τομέας των δικτύων και των τηλεπικοινωνιών βρίσκεται συνεχώς σε ραγδαία ανάπτυξη και εξέλιξη, δημιουργώντας διαρκώς την ανάγκη για αναβάθμιση της υποδομής. Ακόμα οι χρήστες απαιτούν ολοένα και περισσότερες εφαρμογές με αυξημένη ανάγκη σε πόρους και μικρές ανοχές σε καθυστερήσεις. Οι πάροχοι τηλεπικοινωνιών αντιλαμβανόμενοι τις επιτακτικές ανάγκες και απαιτήσεις που ανακύπτουν, για να ανταπεξέλθουν με οικονομικό και αποδοτικό τρόπο στρέφονται στην εξυπηρέτηση των συνδρομητών τους, βασιζόμενοι στην τεχνολογία του cloud και των σημερινών Datacenters. Έτσι ακολουθείται η λογική της εφαρμογής των δικτυακών παροχών και εφαρμογών με τη μορφή εικονικών εκδοχών – υπηρεσιών που εκτελούνται πάνω σε απλές – κοινές εμπορικές συσκευές γενικού σκοπού με λογισμικό ανοιχτού κώδικα και με κεντρικό έλεγχο και διαχείριση.

Για την επίτευξη αυτής της ιδέας, ακολουθείται η κατεύθυνση της εφαρμογής των τεχνολογιών του Software Defined Networking - SDN, του Network Functions Virtualisation - NFV και του Cloud Networking, πάνω στις υποδομές των σύγχρονων multi-tenant Datacenters.

Το CORD αποτελεί μία πλατφόρμα ανοιχτού κώδικα που βρίσκεται σε ανάπτυξη, για την παροχή δικτυακών υπηρεσιών από τους τηλεπικοινωνιακούς παρόχους και αποτελεί ιδανική λύση προς την κατεύθυνση αυτή, καθώς συνδυάζει τις παραπάνω τεχνολογίες, χρησιμοποιώντας κατά βάση τα εξείς εργαλεία:

- OpenStack: Δίνει τη δυνατότητα δημιουργίας εικονικών μηχανών και δικτύων, βασισμένα στην υποκείμενη υλική υποδομή του δικτύου – Datacenter.

- Docker: Παρέχει την τεχνολογία του containerization, δηλαδή την κατασκευή containers, τα οποία αποτελούν πιο ελαφριά εκδοχή των εικονικών μηχανών, μέσα στα οποία εκτελούνται τα στιγμιότυπα των υπηρεσιών.

- ONOS: Αποτελεί τον SDN ελεγκτή του συστήματος, προσφέροντας την SDN τεχνολογία, τον κεντρικό έλεγχο, αλλά και την εφαρμογή διάφορων δικτυακών εφαρμογών.

- XOS: Είναι το σύστημα που είναι υπεύθυνο για τη δημιουργία υπηρεσιών, την διαχείρισή τους, καθώς και για τον συντονισμό όλου του συστήματος του CORD.

Έτσι το CORD καθίσταται ως μία πλατφόρμα αναφοράς για την υλοποίηση (διεκπεραίωση) της ιδέας "τα πάντα ως υπηρεσία" επιφέροντας σημαντικά πλεονεκτήματα όπως είναι ευελιξία στην ανάπτυξη εφαρμογών και υπηρεσιών, άμεσα τροποποιήσιμα παρεχόμενα δίκτυα με κεντρικό έλεγχο, επεκτασιμότητα τόσο σε επίπεδο εφαρμογών και χρηστών όσο και σε επίπεδο υλικής υποδομής, μείωση κόστους CAPEX/OPEX, ανοιχτό στο ευρύ κοινό για έρευνα και συμβολή στην ανάπτυξη καθώς αποτελεί πλατφόρμα ανοιχτού κώδικα.

Ακόμα η γενικότητα του προτύπου του CORD το καθιστά τροποποιήσιμο για εκδοχές του που εστιάζουν σε διαφορετικής φύσης δίκτυα όπως είναι το M-CORD. Το M-CORD το οποίο αποτέλεσε και τον απώτερο σκοπό μελέτης αυτής της διπλωματικής, αποτελεί, την εκδοχή του CORD που αφορά τα δίκτυα κινητών επικοινωνιών και αποσκοπεί στο να αποτελέσει μία πλατφόρμα βάσης για την υλοποίηση του επερχόμενου 5G. Καθώς όπως παρουσιάστηκε, η φύση της 5G τεχνολογίας κινητής δικτύωσης ταιριάζει απόλυτα με τα χαρακτηριστικά που αποδίδει το CORD, όπως είναι η απόδοση των λειτουργιών δικτύου ως υπηρεσίες και το network slicing, το M-CORD αποτελεί ίσως το ιδανικό σημείο αναφοράς για τη νέα γενιά των κινητών επικοινωνιών.

6. Μελλοντική Έρευνα

Το CORD είναι ένα πολλά υποσχόμενο έργο για την δικτύωση που βρίσκεται ακόμα σε ανάπτυξη, και εξέλιξη. Είναι γνωστό ότι οι εμπλεκόμενοι σε αυτό, στρέφονται στη δημιουργία προτύπων για την πλαισιοποίηση των επιμέρους εφαρμογών ως αυτόνομες οντότητες για την καλύτερη διαχείριση στην συνολική ανάπτυξη του CORD και τη λειτουργία του. Παρουσιάστηκε το Trellis που αποσκοπεί στην ανάπτυξη και διαχείριση της κεντρικής υποδομής του CORD. Ακόμα αναφέρθηκε το VOLTHA που αφορά το πρότζεκτ για το μέρος του CORD που αφορά την υλοποίηση του vOLT με τον πλέον αποδοτικό τρόπο. Άλλο ένα παράδειγμα είναι το SEBA το οποίο "πακετάρει" λειτουργίες που αφορούν το "οικιακό" (residential) CORD. Αυτή η τάση φαίνεται να βρίσκει μεγάλο αντίκτυπο στην έρευνα για την ανάπτυξη του CORD και έτσι παρατηρείται συνεχώς να προκύπτουν και να εξελίσσονται νέα πρότζεκτ αυτής της μορφής.

Καθώς το CORD αποτελεί μία δομή που βασίζεται στον κεντρικό SDN έλεγχο (ONOS), θα πρέπει να κατέχει μηχανισμούς ώστε να διασφαλιστεί η παρεχόμενη ασφάλεια της λειτουργίας του στο μέγιστο, δηλαδή θα πρέπει να διασφαλίζεται η εύρυθμη λειτουργία ελέγχου τόσο προς τις εφαρμογές όσο και προς την υποκείμενη υποδομή.

Όπως αναφέρθηκε στην ενότητα του υποστρώματος υποδομής στο Trellis, γίνεται χρήση παραδοσιακών πρωτοκόλλων όπως είναι τα ARP, IP/TCP τα οποία ίσως να αρχίσουν να θεωρούνται παροχημένα και επιπλέον ιστορικά έχουν δείξει ότι είναι ιδιαίτερα επιρρεπή σε επιθέσεις (π.χ. ARP spoofing). Έτσι, είναι επιθυμητό να διεξαχθεί έρευνα, για την δημιουργία νέων αντίστοιχων πρωτοκόλλων δρομολόγησης, ώστε να ανταποκρίνονται στα σύγχρονα υπολογιστικά δίκτυα και να ανταπεξέρχονται σε κενά ασφάλειας.

Όσον αφορά το M-CORD, όπως έγινε σαφές αποτελεί μία ισχυρή δομή για την εφαρμογή του 5G, αλλά ακόμα βρίσκεται υπό ανάπτυξη. Για παράδειγμα το ποσοστό του RAN που μπορεί να μεταφερθεί στον πυρήνα και να εκτελεστεί με εικονικά τρόπο, βάσει υπηρεσιών, δεν είναι ακόμα σαφώς προσδιορισμένο. Σε αυτό το σημείο γίνεται αντιληπτή η ανάγκη για έρευνα που θα στοχεύει στη μεταφορά του, όσο δυνατόν, μεγαλύτερου μέρους του RAN στο εσωτερικό του CORD για τη διασφάλιση της βέλτιστης λειτουργίας της κινητής δικτύωσης.

Επίσης, η ερευνητική κοινότητα που ασχολείται με το M-CORD, αναμένεται να μελετήσει και να παρουσιάσει επιπλέον εφαρμογές που θα απορρέουν απ' ευθείας από την υποδομή του CORD, με στόχο την καλύτερη δικτυακή παροχή αλλά και ικανοποίηση των χρηστών των ασύρματων συσκευών.

Βιβλιογραφία

[1] Rouven Krebs, Christof Momm, Samuel Kounev "*Architectural Concerns in Multi-Tenant SaaS Applications*", 2013

[2] Mohammad Al-Fares, Alexander Loukissas, Amin Vahdat "*A Scalable, Commodity Data Center Network Architecture*", 2008

[3] Teemu Koponen, Keith Amidon, Peter Balland, Martín Casado, Anupam Chanda, Bryan Fulton, Igor Ganichev, Jesse Gross, Natasha Gude, Paul Ingram, Ethan Jackson, Andrew Lambeth, Romain Lenglet, Shih-Hao Li, Amar Padmanabhan, Justin Pettit, Ben Pfaff, Rajiv Ramanathan, Scott Shenker, Alan Shieh, Jeremy Stribling, Pankaj Thakkar, Dan Wendlandt, Alexander Yip, Ronghua Zhang, "*Network Virtualization in Multi-tenant Datacenters*", 2014

[4] Ben Pfaff, Justin Pettit, Teemu Koponen, Ethan J. Jackson, Andy Zhou, Jarno Rajahalme, Jesse Gross, Alex Wang, Jonathan Stringer, Pravin Shelar, Keith Amidon, Mart´ın Casado, "T*he Design and Implementation of Open vSwitch*", 2015

[5] Multi-tenancy definition: https://whatis.techtarget.com/definition/multi-tenancy

[6] Microservices definition: https://searchmicroservices.techtarget.com/definition/microservices

[7] Open Networking Foundation White Paper, "*Software-Defined Networking: The New Norm for Networks*", April 13, 2012

[8] Software Defined Networking: https://searchnetworking.techtarget.com/definition/software-defined-networking-SDN

[9] Open Networking Foundation, SDN definition: https://www.opennetworking.org/sdn-definition/

[10] Open Networking Foundation, "*SDN Architecture Overview*", Version 1.0 December 12, 2013

[11] https://wiki.opencord.org/display/CORD/VTN+Design+Note

[12] https://en.wikipedia.org/wiki/Network_function_virtualization

[13] https://searchnetworking.techtarget.com/definition/network-functions-virtualization-NFV

[14] Network Functions Virtualisation White Paper, "*An Introduction, Benefits, Enablers, Challenges & Call for Action*", 2012

[15] European Telecommunications Standards Institute (ETSI), Network Functions Virtualisation (NFV): https://www.etsi.org/technologies/nfv

[16] https://www.openstack.org/software/

[17] https://en.wikipedia.org/wiki/OpenStack

[18] https://en.wikipedia.org/wiki/Docker_(software)

[19] Carl Boettiger, Center for Stock Assessment Research, "*An introduction to Docker for reproducible research, with examples from the R environment*", 2 October 2014

[20] Sumologic, Kubernetes: https://www.sumologic.com/devops/what-is-kubernetes

[21] Redhat, Kubernetes: https://www.redhat.com/en/topics/containers/what-is-kubernetes

[22] Open Networking Lab White Paper "*Introducing ONOS - a SDN network operating system for Service Providers*", 2014

[23]ONOS wiki: https://wiki.onosproject.org/display/ONOS/Wiki+Home

[24] Open Networking Foundation, ONOS: https://www.opennetworking.org/onos/

[25] https://wiki.onosproject.org/display/ONOS/System+Components

[26] Larry Peterson, Scott Baker, Marc De Leenheer, Open Networking Lab, Andy Bavier, Sapan Bhatia, Jude Nelson, Mike Wawrzoniak, Princeton University, John Hartman, University of Arizona, "*XOS: An Extensible Cloud Operating System*", 2015

[27] Open Networking Foundation, Trellis: https://www.opennetworking.org/trellis/

[28] Sapan Bhatia, Murtaza Motiwala, Wolfgang M¨uhlbauer, Yogesh Mundada, Vytautas Valancius, Andy Bavier, Nick Feamster, Larry Peterson, Jennifer Rexford, Princeton University, Georgia Tech, T-Labs/TU Berlin, "*Trellis: A Platform for Building Flexible, Fast Virtual Networks on Commodity Hardware*", 2008

[29] https://wiki.opencord.org/display/CORD/Trellis%3A+CORD+Network+Infrastructure

[30] https://wiki.opendaylight.org/view/OpenDaylight_Virtual_Tenant_Network_(VTN):Overview

[31] https://wiki.opencord.org/pages/viewpage.action?pageId=1278093

[32] https://wiki.opencord.org/display/CORD/Trellis+Underlay+Fabric

[33] https://wiki.opencord.org/display/CORD/Virtual+Network+Overlay

[34] https://wiki.opencord.org/pages/viewpage.action?pageId=1278093

[35] https://wiki.opendaylight.org/view/OpenDaylight_Virtual_Tenant_Network_(VTN):Overview

[36] https://wiki.opencord.org/display/CORD/Service+Composition+and+the+Role+of+VTN

[37] Scott Baker, Andy Bavier, Ali Al-Shabibi and Larry Peterson, "*Virtual Subscriber Gateway (vSG)*", Open Networking Lab, February 16, 2016

[38] CORD wiki, vOLT: https://wiki.opencord.org/pages/viewpage.action?pageId=1278086

[39] Open Networking Foundation, Jonathan Hart, "*VOLTHA Overview and Roadmap*", March 27, 2018: https://opencord.org/wp-content/uploads/2018/03/Project-VOLTHA-Overview-_-Roadmap-ONF-Vision-Workshop.pdf

[40] https://searchnetworking.techtarget.com/definition/Optical-line-terminal-OLT

[41] https://searchnetworking.techtarget.com/definition/passive-optical-network-PON

[42] https://en.wikipedia.org/wiki/Optical_line_termination

[43] https://www.opennetworking.org/voltha/

[44] https://wiki.opencord.org/display/CORD/VOLTHA

[45] "How CORD will impact your central office", CommScope, 2018

[50] CORD White Paper "*Central Office Re−architected as a Datacenter*", March 14, 2016

[51]Alcatel-Lucent, Strategic White Paper, "*The LTE Network Architecture*", 2013

[52] http://www.3gpp.org/technologies/keywords-acronyms/98-lte

[53] http://ecee.colorado.edu/~ecen4242/LTE/e_utran.html

[54] https://www.tutorialspoint.com/lte/lte_protocol_stack_layers.htm

[55] Zakaria Tayq, Université de Limoges, Thèse de doctorat, "*Fronthaul integration and monitoring in 5G networks*", Submitted on 13 Feb 2018 on HAL archives-ouvertes.fr

[56] http://www.3gpp.org/technologies/keywords-acronyms/100-the-evolved-packet-core

[57] Alcatel-Lucent, Strategic White Paper, "*Introduction to Evolved Packet Core*"

[58] Nisha Chauhan, Vivek Kumar, "*A Detail Review on Multiprotocol Label Switching (MPLS)*", International Journal of Advanced Research in Computer Engineering & Technology (IJARCET) Volume 4 Issue 4, April 2015

[59] https://en.wikipedia.org/wiki/System_Architecture_Evolution

[60] http://www.3gpp.org/technologies/keywords-acronyms/100-the-evolved-packet-core

[61] 5G PPP Architecture Working Group, "View on 5G Architecture (Version 2.0)", 2017

[62] Kathiravan Srinivasan, Nitesh Kumar Agrawal, "*A Study on M-CORD based Architecture in Traffic Offloading for 5G-enabled Multiaccess Edge Computing Networks*", IEEE International Conference on Applied System Innovation 2018

[63] Muhammad Tahir Abbas, Talha Ahmed Khan, Asif Mahmood, Javier Jose Diaz Rivera and Wang-Cheol SONG, "*Introducing Network Slice Management inside M-CORD-Based-5G Framework*", NOMS 2018 - 2018 IEEE/IFIP Network Operations and Management Symposium

[64] Mingeun Yoon, Tom Tofigh, Guru Parulkar, "Rethinking the Mobile Edge Network with CORD (Mobile-CORD)", IEEE Software Defined Networks, March 2016

[65] CORD Project | Whitepaper, "M-CORD as an Open Reference Solution for 5G Enablement", 2017

[66] Arjun Singh, Joon Ong, Amit Agarwal, Glen Anderson, Ashby Armistead, Roy Bannon, Seb Boving, Gaurav Desai, Bob Felderman, Paulie Germano, Anand Kanagala, Jeff Provost, Jason Simmons, Eiichi Tanda, Jim Wanderer, Urs Hölzle, Stephen Stuart, and Amin Vahdat, "*Jupiter Rising: A Decade of Clos Topologies and Centralized Control in Google's Datacenter Network*", 2015

[67] Nick McKeown, Tom Anderson, Hari Balakrishnan, Guru Parulkar, Larry Peterson, Jennifer Rexford, Scott Shenker, Jonathan Turner, "*OpenFlow: Enabling Innovation in Campus Networks*", March 14, 2008

[68] Pankaj Berde, Matteo Gerola, Jonathan Hart, Yuta Higuchi, Masayoshi Kobayashi, Toshio Koide, Bob Lantz, Brian O'Connor, Pavlin Radoslavov, William Snow, Guru Parulkar, "*ONOS: Towards an Open, Distributed SDN OS*", 2014

[69] Wes Felter, Alexandre Ferreira, Ram Rajamony, Juan Rubio, "*An Updated Performance Comparison of Virtual Machines and Linux Containers*", IBM Research Report, July 21, 2014

[70] European Telecommunications Standards Institute (ETSI), Introductory White Paper, "*Network Functions Virtualisation, An Introduction, Benefits, Enablers, Challenges & Call for Action*", October 22-24, 2012, "SDN and OpenFlow World Congress", Darmstadt-Germany

[71] A Whitepaper by ON.LAB, AT&T and ONOS "CORD: The Central Office Re-architected as a Datacenter" 2015: http://onosproject.org/wp-content/uploads/2015/06/PoC_CORD.pdf

[72] https://www.opennetworking.org/cord/

[73] CORD: Central Office Re-architected as a Datacenter, Larry Peterson, Open Networking Lab: http://opencord.org/wp-content/uploads/2016/10/BBWF-CORD.pdf

[74] On data center scale, OpenFlow, and SDN, Brad Hedlund: http://bradhedlund.com/2011/04/21/data-center-scale-openflow-sdn/

[75]https://www.researchgate.net/figure/Unified-SDN-for-MCC-Energy-Optimisation-Concept_fig3_306048144

[76] http://ecee.colorado.edu/~ecen4242/LTE/e_utran.html

[77] https://www.tutorialspoint.com/lte/lte_network_architecture.htm

[78] "*M-CORD: Mobile CORD - Enable 5G on CORD*", March 2016: http://opencord.org/wp-content/uploads/2016/03/M-CORD-March-2016.pdf

[79] Ankit Singla, Balakrishnan Chandrasekaran, P. Brighten Godfrey, Bruce Maggs, "*Towards a Speed of Light Internet*", 13 May 2015

[80] https://www.accton.com/Technology-Brief/cord-fundamentals-with-openstack/

[81] Babak Bashari Rad, Harrison John Bhatti, Mohammad Ahmadi, "*An Introduction to Docker and Analysis of its Performance*", International Journal of Computer Science and Network Security, VOL.17 No.3, March 2017

[82] Fetia Bannour, Sami Souihi and Abdelhamid Mellouk, "*Adaptive State Consistency for Distributed ONOS Controllers*", The 2018 IEEE Global Communications Conference, December 2018

[83] Mohammed Sameer, Bhargavi Goswami, "*Experimenting with ONOS Scalability on Software Defined Network*", Journal of Advanced Research in Dynamical and Control Systems, January 2019

[84] Christos Bouras, Anastasia Kollia, Andreas Papazois, "*Exploring SDN & NFV in 5G Using ONOS & POX Controllers*", International Journal of Interdisciplinary Telecommunications and Networking, Volume 10, Issue 4, October-December 2018