ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

ΤΟΜΕΑΣ ΕΠΙΚΟΙΝΩΝΙΩΝ, ΗΛΕΚΤΡΟΝΙΚΗΣ ΚΑΙ ΣΥΣΤΗΜΑΤΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ

Ανάπτυξη οικιακού vCPE με χρήση Linux Container

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

του

ΑΛΕΞΑΝΔΡΟΥ ΚΟΡΠΑ - ΚΑΜΑΡΙΑΝΟΥ

Επιβλέπων : Ευστάθιος Συκάς
Καθηγητής Ε.Μ.Π.

Αθήνα, Ιούνιος 2017

ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ
ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ
ΤΟΜΕΑΣ ΕΠΙΚΟΙΝΩΝΙΩΝ, ΗΛΕΚΤΡΟΝΙΚΗΣ ΚΑΙ ΣΥΣΤΗΜΑΤΩΝ
ΠΛΗΡΟΦΟΡΙΚΗΣ

Ανάπτυξη οικιακού vCPE με χρήση Linux Container

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

του

ΑΛΕΞΑΝΔΡΟΥ ΚΟΡΠΑ - ΚΑΜΑΡΙΑΝΟΥ

Επιβλέπων : Ευστάθιος Συκάς
Καθηγητής Ε.Μ.Π.

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την .....

.....	.....	.....
Ευστάθιος Συκάς Καθηγητής Ε.Μ.Π.	Γεώργιος Στασινόπουλος Καθηγητής Ε.Μ.Π.	Ιωάννα Ρουσσάκη Επίκ. Καθηγήτρια Ε.Μ.Π.

Αθήνα, Ιούνιος 2017

.....

ΑΛΕΞΑΝΔΡΟΣ ΚΟΡΠΑΣ - ΚΑΜΑΡΙΑΝΟΣ

Διπλωματούχος Ηλεκτρολόγος Μηχανικός και Μηχανικός Υπολογιστών Ε.Μ.Π.

Περίληψη

Η μεγάλη ταχύτητα με την οποία το διαδίκτυο εισέρχεται στην καθημερινή μας ζωή, αυξάνει εκθετικά τις απαιτήσεις των ευρυζωνικών δικτύων, απαιτήσεις που προέρχονται από τις εφαρμογές βίντεο, κινητής τηλεφωνίας και διαδικτύου. Ως αποτέλεσμα, οι τηλεπικοινωνιακοί πάροχοι αναζητούν συνεχώς τρόπους επέκτασης και κλιμάκωσης των υπηρεσιών του δικτύου τους, διατηρώντας παράλληλα τα επενδυτικά και λειτουργικά έξοδα χαμηλά. Τα χαρακτηριστικά των παραδοσιακών συσκευών παρουσιάζουν εμπόδια στην απαίτηση αυτή και δημιουργούν πολλούς περιορισμούς που αυξάνουν το κόστος ανάπτυξης και περιορίζουν την επεκτασιμότητα και την αποδοτικότητα λειτουργίας του δικτύου. Αυτή η κατάσταση αναγκάζει τους φορείς εκμετάλλευσης να εξετάσουν εναλλακτικές λύσεις που μπορούν να εξαλείψουν τους περιορισμούς και ταυτόχρονα θα προσφέρουν την δυνατότητα για εξέλιξη των υφιστάμενων υπηρεσιών και ανάπτυξη νέων, με χαμηλότερο κόστος κυκλοφορίας.

Σκοπός της παρούσας διπλωματικής είναι η χρήση της τεχνολογίας των NFV δικτύων, καθώς και της εικονικοποίησης στο επίπεδο του λειτουργικού συστήματος, για την ανάπτυξη εικονικών CPE. Οι συσκευές αυτές, εκμεταλλευόμενες των πλεονεκτημάτων αυτών των τεχνολογιών, προσφέρουν μια εναλλακτική προσέγγιση για την διασύνδεση των οικιακών συσκευών στο διαδίκτυο, στοχεύοντας στην μείωση των εξόδων από μεριάς των τηλεπικοινωνιακών παρόχων και την βελτιστοποίηση του κύκλου κυκλοφορίας νέων υπηρεσιών.

Λέξεις Κλειδιά: εικονικά CPE, NFV δίκτυα, Linux Containers, εικονικοποίηση στο επίπεδο του λειτουργικού, Proxmox VE, Ansible, GRE Tunnel

Abstract

Nowadays, the rapid entry of the Internet in our everyday lives increases exponentially the demands of broadband networks, demands arising from video, mobile and internet applications. Consequently, the telecommunication service providers (TSP) are constantly seeking for new ways to expand and escalate the services provided by their networks and simultaneously preserve both capital expenditure (CapEx) and operational expenditure (OpEx) as low as possible. The characteristics of traditional devices act as barriers to this demand and they conduce to increase of development cost and limitations of the network's scalability and performance. The aforementioned behavior forces operators to consider alternatives, which will eliminate the constraints and at the same time provide the opportunity for progressing existing services and developing new, lower-cost traffic.

Aim of this diploma thesis is to combine the Network Function Virtualization and the OS-level virtualization for the development of virtual CPE devices. These devices, by exploiting the advantages of the mentioned technologies, offer an alternative approach to interconnecting household devices with the Internet, which achieves the ultimate goal of reducing the costs of telecomm operators and optimizing the life cycle of new services.

Keywords: virtual CPE, NFV networks, Linux Containers, OS-level virtualization, Proxmox VE, Ansible, GRE Tunnel

Ευχαριστίες

Η παρούσα διπλωματική αναπτύχθηκε και ολοκληρώθηκε κατά το ακαδημαϊκό έτος 2016-17, ως αποτέλεσμα της συνεργασίας μου με το Τομέα Επικοινωνιών, Ηλεκτρονικής και Συστημάτων Πληροφορικής της Σχολής Ηλεκτρολόγων Μηχανικών και Μηχανικών Ηλεκτρονικών Υπολογιστών του Εθνικού Μετσόβιου Πολυτεχνείου.

Αρχικά, θα ήθελα να ευχαριστήσω τον επιβλέποντα καθηγητή της παρούσας εργασίας, κ. Ευστάθιο Συκά, για την άψογη συνεργασία που είχαμε κατά το διάστημα της συγγραφής της και για την ευκαιρία που μου έδωσε να εργαστώ πάνω σε ένα τόσο ενδιαφέρον θέμα. Επιπλέον, ιδιαίτερες ευχαριστίες θα ήθελα να μεταφέρω στον υποψήφιο διδάκτορα, κ. Πάρη Χαραλάμπου, για την υποστήριξη που μου παρείχε κατά την εκπόνηση της διπλωματικής αυτής. Η διαθεσιμότητα του, ανά πάσα ώρα και στιγμή, στο να μου παρέχει την απαραίτητη βοήθεια ήταν καθοριστικής σημασίας για την ολοκλήρωση της και χωρίς αυτόν το έργο μου θα ήταν σίγουρα δυσκολότερο.

Επιπλέον, θα ήθελα να ευχαριστήσω τους γονείς μου, τα αδέρφια μου και την υπόλοιπη οικογένεια μου, που με στήριξαν όλα αυτά τα χρόνια της φοιτητικής μου, και όχι μόνο, πορείας. Ήταν πάντοτε δίπλα μου, έτοιμοι να με ενθαρρύνουν και να με συμβουλέψουν όποτε χρειάστηκα την βοήθεια τους. Σε αυτούς χρωστάω ότι είμαι, ότι πέτυχα και ότι θα πετύχω.

Τέλος, θα ήταν τεράστια παράλειψη να μην ευχαριστήσω την παρέα μου, τον Γιάννη, τον Λεωνίδα, τον Πέτρο και τον Σπύρο που ομόρφυναν αυτό το κεφάλαιο της ζωής μου με τις ομορφότερες αναμνήσεις. Θα είμαι για πάντα ευγνώμων σε αυτούς για τις στιγμές που ζήσαμε και τους εύχομαι η ζωή τους να είναι γεμάτη επιτυχίες.

Πίνακας περιεχομένων

Πίνακας εικόνων

Πίνακας πινάκων

Πίνακας διαγραμμάτων

1

Εισαγωγή

1.1 Οι περιορισμοί των παραδοσιακών δικτύων

Στο διάστημα των τελευταίων τριών δεκαετιών, η τεχνολογία, και ειδικότερα ο τομέας των τηλεπικοινωνιών, έχει κάνει τεράστια άλματα και αυτή η εκθετική ανάπτυξη την οδήγησε στο να ενταχθεί στην καθημερινή ζωή στο σύνολο του παγκόσμιου πληθυσμού. Ένα από τα σημαντικότερα κομμάτια της, το Internet, έχει μετατραπεί σε απαραίτητο εργαλείο που χρησιμοποιείται αμέτρητες φορές κατά την διάρκεια της ημέρας από τον καθένα μας. Η γρήγορη αυτή κλιμάκωση δημιουργεί μια πρόκληση στους τηλεπικοινωνιακούς παρόχους: την αδιάληπτη παροχή σύνδεσης στο διαδίκτυο με υψηλές ταχύτητες. Είναι αντιληπτό πως όσο η τεχνολογία αναπτύσσεται, οι απαιτήσεις των χρηστών σε ταχύτητες, σε σταθερότητα αλλά κυρίως σε παρεχόμενες υπηρεσίες αυξάνονται. Τα χαρακτηριστικά των παραδοσιακών συσκευών αποτελούν εμπόδιο στην απαίτηση αυτή και περιορίζουν την επεκτασιμότητα, το κόστος ανάπτυξης και την αποδοτικότητα λειτουργίας του δικτύου. Ενδεικτικά, αυτοί οι περιορισμοί αφορούν [1]:

- **Υψηλό επενδυτικό και λειτουργικό κόστος**: Τα επενδυτικά έξοδα (Capital Expenditure – CapEx), όπως η αγορά και η ανανέωση του απαραίτητου εξοπλισμού, τα έξοδα έρευνας και ανάπτυξης κ.λπ., αυξάνονται. Ομοίως, ανάλογη αύξηση παρατηρείται και στα λειτουργικά έξοδα (Operation Expenditure – OpEx), λόγω της αύξησης του κόστους αποθήκευσης, κατανάλωσης και συντήρησης του εξοπλισμού.

- **Περιορισμοί ευελιξίας**: Ο δικτυακός εξοπλισμός των τηλεπικοινωνιακών παρόχων είναι αναπτυγμένος με βάση ένα γενικό σύνολο απαιτήσεων και προσφέρει τη λειτουργικότητα του ως συνδυασμό συγκεκριμένου υλικού και λογισμικού. Το υλικό και το λογισμικό συσκευάζονται ως μονάδα και περιορίζονται στην υλοποίηση του κατασκευαστή. Αυτό περιορίζει τις επιλογές των συνδυασμών χαρακτηριστικών και των δυνατοτήτων υλικού που μπορούν να αναπτυχθούν. Η έλλειψη ευελιξίας και προσαρμογής για την ικανοποίηση των ταχέως μεταβαλλόμενων απαιτήσεων οδηγεί σε αναποτελεσματική χρήση των πόρων.
- **Περιορισμοί κλιμάκωσης**: Οι συσκευές φυσικού δικτύου έχουν περιορισμούς κλιμάκωσης τόσο στο υλικό όσο και στο λογισμικό. Το υλικό απαιτεί ισχύ και χώρο και η έλλειψη αυτών περιορίζει το εν δυνάμει αναπτυσσόμενο υλικό. Από την πλευρά του λογισμικού, οι παραδοσιακές αυτές συσκευές ενδέχεται να μην είναι σε θέση να συμβαδίσουν με την κλιμάκωση των αλλαγών στο δίκτυο, όπως ο αριθμός των πιθανών διαδρομών. Κάθε συσκευή έχει σχεδιαστεί για να χειρίζεται μια περιορισμένη πολυδιάστατη κλίμακα και μόλις αυτό το όριο προσεγγιστεί, ο χειριστής έχει ένα πολύ περιορισμένο σύνολο επιλογών εκτός από την αναβάθμιση της συσκευής.
- **Περιορισμοί χρόνου – αγοράς**: Καθώς οι απαιτήσεις αυξάνονται και αλλάζουν με την πάροδο του χρόνου, ο εξοπλισμός δεν είναι πάντοτε σε θέση να συμβαδίσει γρήγορα με αυτές τις αλλαγές. Ως αποτέλεσμα, οι πάροχοι υπηρεσιών συχνά καθυστερούν να προσφέρουν νέες υπηρεσίες για να ανταποκριθούν στη μετατόπιση των απαιτήσεων της αγοράς, λόγω της αναβάθμισης του εξοπλισμού δικτύωσης που απαιτεί η εφαρμογή νέων υπηρεσιών. Όπως είναι κατανοητό, κάτι τέτοιο είναι μια χρονοβόρος διαδικασία και οδηγεί στην αύξηση του χρονοδιαγράμματος για την προσφορά νέων υπηρεσιών στους πελάτες, με αποτέλεσμα την απώλεια εσόδων.
- **Περιορισμοί αναβάθμισης**: Οι συσκευές και τα δίκτυα πρέπει να αναβαθμιστούν ή να επανεκκινηθούν έπειτα από την πάροδο ενός χρονικού διαστήματος. Κάτι τέτοιο απαιτεί φυσική πρόσβαση στο συγκεκριμένο χώρο για την ανάπτυξη νέου υλικού, την αναμόρφωση της φυσικής σύνδεσης και την αναβάθμιση των εγκαταστάσεων στην περιοχή. Τα επιπλέον έξοδα αλλά και οι φυσικές και τεχνικές δυσκολίες που απαιτεί η διαδικασία αυτή, όπως την προσωρινή μεταφορά της κίνησης, συχνά οδηγούν στην αναστολή της, επιβραδύνοντας περαιτέρω την προσφορά νέων υπηρεσιών.

Στις αρχές της τρέχουσας δεκαετίας, έκανε την εμφάνιση της μια νέα τεχνολογία, η εικονικοποίηση δικτυακών λειτουργιών (Network Function Virtualization – NFV) που έχει ως στόχο την επίλυση των παραπάνω προβλημάτων. Στην συγκεκριμένη τεχνολογία χρησιμοποιούνται οι πρότυπες μέθοδοι εικονικοποίησης υπολογιστικών πόρων για την προσομοίωση δικτυακών λειτουργιών, οι οποίες συμπεριφέρονται αντίστοιχα με τις φυσικές. Τα

πλεονεκτήματα αυτής της νέας τεχνολογίας είναι πολλά και σημαντικά και στην συνέχεια της διπλωματικής εργασίας θα αναλυθούν σε βάθος.

Σκοπός, λοιπόν της παρούσας διπλωματικής εργασίας είναι η ανάπτυξη ενός οικιακού vCPE, το οποίο προσφέρει μια εναλλακτική προσέγγιση σε σχέση με το υπάρχον μοντέλο παροχής σύνδεσης στο διαδίκτυο. Πρόκειται για μια υλοποίηση των δικτύων NFV, ίσως η πρώτη ευρέως αναπτυσσομένη. Με την προσέγγιση αυτή, επαναπροσδιορίζεται το μοντέλο κατανομής του υπολογιστικού βάρους της διασύνδεσης του οικιακού δικτύου στο Internet, το οποίο στην σύγχρονη αρχιτεκτονική συνήθως διαμοιράζεται στους οικιακούς δρομολογητές. Τα οφέλη της συγκεκριμένης τεχνολογίας είναι πολλαπλά και ανάμεσα στους στόχους της εργασίας αυτής είναι η ανάλυση τους. Ενδεικτικά, αναφέρεται η μείωση των επενδυτικών και λειτουργικών εξόδων, όπως και η αύξηση της ταχύτητας κυκλοφορίας νέων υπηρεσιών. Παράλληλα, ένας ακόμη σκοπός της διπλωματικής εργασίας είναι η πλήρης ανάλυση της διαδικασίας ανάπτυξης του σε όλα της τα στάδια, ενώ στο τέλος θα γίνει προσπάθεια αξιολόγησης του μοντέλου που αναπτύχθηκε.

1.2 Οργάνωση κειμένου

Η διπλωματική εργασία αποτελείται από έξι συνολικά κεφάλαια. Στο παρόν κεφάλαιο, το οποίο αποτελεί το κεφάλαιο 1, παρουσιάζεται το πρόβλημα που υφίσταται και τις ανάγκες που συντέλεσαν στη δημιουργία του θέματος της συγκεκριμένης διπλωματικής. Στη συνέχεια παρουσιάζεται επιγραμματικά το αντικείμενο της.

Στο κεφάλαιο 2, καλύπτονται σε βάθος ορισμένες θεωρητικές έννοιες, με στόχο τη δημιουργία του κατάλληλου υποβάθρου για την ευκολότερη κατανόηση του αναγνώστη. Συγκεκριμένα, αναλύονται οι έννοιες της εικονικοποίησης και της εικονικοποίησης δικτυακών λειτουργιών, όπως και αυτές των φυσικών και εικονικών CPE συσκευών.

Στο κεφάλαιο 3, παρουσιάζεται θεωρητικά η αρχιτεκτονική του δικτύου που σχεδιάστηκε. Ειδικότερα, επεξηγείται ο ρόλος και η σημασία των επιμέρους μονάδων που το αποτελούν. Στη συνέχεια, παρουσιάζεται τα εργαλεία που χρησιμοποιήθηκαν για την ενσάρκωση των μονάδων αυτών, όπως η μηχανή εικονικοποίησης Proxmox VE, το εργαλείο αυτοματοποίησης και ενορχήστρωσης Ansible και το πρωτόκολλο GRE Tunnel.

Στο κεφάλαιο 4, γίνεται ανάλυση σε βάθος της διαδικασίας ανάπτυξης του θέματος της διπλωματικής εργασίας, δηλαδή του οικιακού vCPE. Συγκεκριμένα, παρουσιάζεται βήμα προς βήμα η διαδικασία δημιουργίας του με τις απαραίτητες επεξηγήσεις, ενώ στη συνέχεια αναλύεται η αυτοματοποίηση της εν λόγω διαδικασίας.

Στο κεφάλαιο 5, επιχειρείται η αξιολόγηση του ανεπτυγμένου μοντέλου. Αρχικά παρουσιάζονται οι παράμετροι και το σύστημα αξιολόγησης, στη συνέχεια τα αποτελέσματα των πειραμάτων και, τέλος, τα συμπεράσματα που εξάχθηκαν από αυτά.

Στο κεφάλαιο 6, τοποθετείται ο επίλογος της διπλωματικής εργασίας. Σε αυτόν, συνοψίζονται τα συμπεράσματα από το σύνολο της και έπειτα παρουσιάζονται οι προκλήσεις και οι πιθανές επεκτάσεις που προκύπτουν με το πέρας της.

Στο τέλος της παρούσας εργασίας, παρατίθεται η βιβλιογραφία του χρησιμοποιήθηκε για τη συγγραφή της και ακολουθεί το παράρτημα, στο οποίο περιέχεται ο κώδικας που αναπτύχθηκε κατά τη διάρκεια του παρόντος θέματος.

2

Θεωρητικό υπόβαθρο

Σε αυτό το κεφάλαιο θα αναλυθούν οι βασικές τεχνολογίες που χρησιμοποιήθηκαν κατά την εκπόνηση της διπλωματικής, αλλά και θεωρητικές έννοιες που αφορούν την θεματολογία της. Συγκεκριμένα θα επεξηγηθεί η έννοια της εικονικοποίησης, η λειτουργία και η χρησιμότητα των Linux Containers, η τεχνολογία των δικτύων NFV και τα οφέλη της. Τέλος θα παρουσιαστεί η τεχνολογία των CPE συσκευών, αρχικά στην συμβατική τους μορφή και έπειτα σε μια διαφορετική προσέγγιση, που αποτελεί την πρώτη σημαντική υλοποίηση των NFV δικτύων.

2.1 Linux Containers

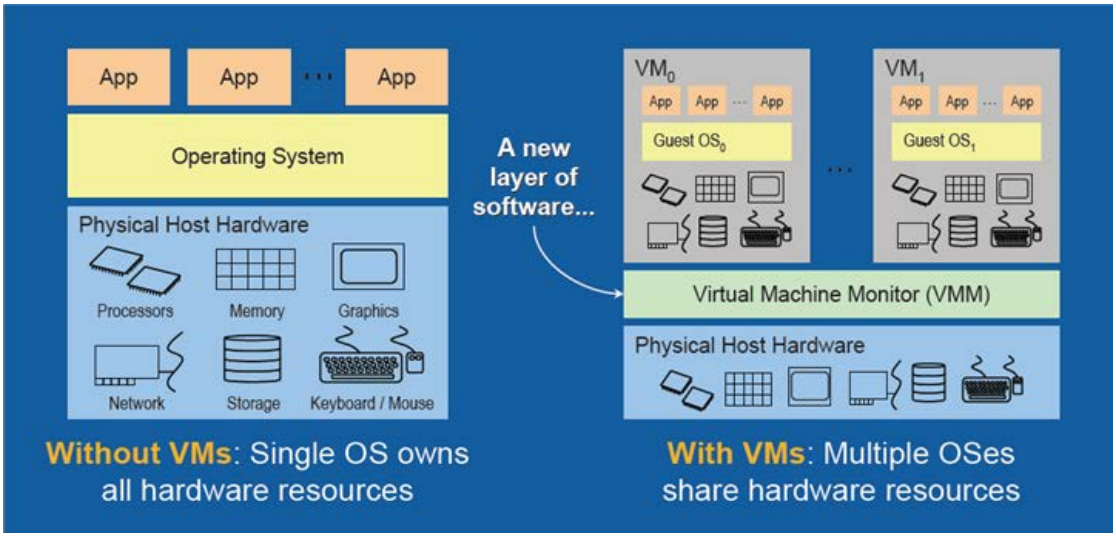
2.1.1 Εικονικοποίηση

Με το όρο της εικονικοποίησης (virtualization) στην πληροφορική, αναφερόμαστε στην δημιουργία «εικονικών» υπολογιστικών πόρων που δρουν ισοδύναμα με τους αντίστοιχους φυσικούς. Πιο συγκεκριμένα, κατά την εικονικοποίηση, ο υπολογιστής χρησιμοποιεί ειδικό λογισμικό, με σκοπό την, είτε πλήρη είτε μερική, προσομοίωση υλικού για έναν ή περισσότερους υπολογιστές, τους οποίους αποκαλούμε συνήθως εικονικές μηχανές (Virtual Machines – VMs). Έτσι, στα εικονικά αυτά συστήματα, ο χρήστης έχει την δυνατότητα να εκτελέσει διεργασίες με πανομοιότυπο τρόπο όπως και σε ένα πραγματικό σύστημα. Το πλήθος των πλεονεκτημάτων που προκύπτουν από την χρήση των εικονικών μηχανών είναι εξαιρετικά μεγάλο. Στην συνέχεια θα παρουσιαστούν κάποια από τα σημαντικότερα.

Το μεγαλύτερο, ενδεχομένως, από τα πλεονεκτήματα της εικονικοποίησης, είναι η μείωση του κόστους μέσω της εξοικονόμησης φυσικών πόρων. Στην βιομηχανία της πληροφορικής, για λόγους απλότητας, συνηθίζεται έντονα κάθε σύστημα να εξυπηρετεί ένα και συγκριμένο σκοπό μέσω μιας εφαρμογής ή ομάδας εφαρμογών [2]. Δεδομένου αυτού λοιπόν, θα ήταν εξαιρετικά δαπανηρή η χρήση φυσικών συστημάτων, καθώς το κόστος αγοράς και συντήρησης αλλά και η κατανάλωση αυτών θα ήταν ιδιαίτερα υψηλή. Έτσι λοιπόν, μέσω της εικονικοποίησης, δίδεται η δυνατότητα χρήσης ενός υπολογιστή (host) ο οποίος θα εξυπηρετεί πολλά εικονικά μηχανήματα (guests) τα οποία θα μπορούν να εκτελούν διαφορετικές εργασίες ή ακόμα και να χρησιμοποιούν διαφορετικό λογισμικό.

Ένα ακόμα σημαντικό πλεονέκτημα είναι η ασφάλεια που προσφέρεται από την απομόνωση είτε μεταξύ του φυσικού μηχανήματος και του εικονικού είτε μεταξύ των εικονικών μηχανών. Συγκεκριμένα, η εικονικοποίηση εγγυάται σε ένα εικονικό μηχάνημα πως οποιαδήποτε αλλαγή συμβεί σε αυτό που μπορεί να προκαλέσει ζημιά στην γενικότερη ομαλή λειτουργεία του (πειραματικές εφαρμογές, κακόβουλα λογισμικά), δεν θα επηρεάσει όσες άλλες εργασίες πραγματοποιούνται στο σύστημα, παρά μόνο το ίδιο.

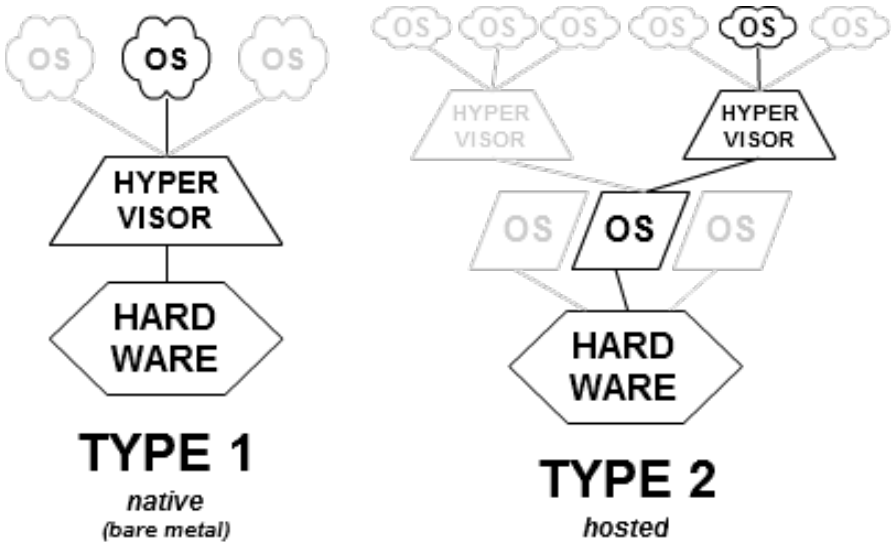
Τέλος, εξαιρετικά σημαντική είναι η απλότητα στην διαχείριση των εικονικών συστημάτων. Μέσω κατάλληλου περιβάλλοντας ή ακόμα και με χρήση εντολών στο κέλυφος του λειτουργικού, ο διαχειριστής έχει την δυνατότητα να δημιουργεί, να διαγράφει ή ακόμα και να τροποποιεί εικονικές μηχανές σε εξαιρετικά μικρό χρονικό διάστημα. Η μικρή αυτή διάρκεια δημιουργίας εξυπηρετεί την γρήγορη αποκατάσταση εικονικών μηχανών σε περίπτωση που το φυσικό μηχάνημα υπέστη κάποια υλική ζημιά. Αυτό επιτυγχάνεται μέσω αντιγράφων ασφαλείας της κατάστασης των εικονικών μηχανών (snapshots), τα οποία βοηθούν ώστε η επαναφορά να γίνεται άμεσα και χωρίς απώλειες για την ομαλή λειτουργία της εικονικής μηχανής. Επίσης, τα snapshots επιτρέπουν στους διαχειριστές του συστήματος να μεταφέρουν γρήγορα και ασφαλώς εικονικές μηχανές από εξυπηρετητές με υψηλό φόρτο εργασίας σε άλλους με χαμηλότερο.



Εικόνα 2-1: Πλήρης Εικονικοποίηση με χρήση VMM

Η πιο συνήθης μορφή εικονικοποίησης είναι η πλήρης εικονικοποίηση (full virtualization). Σε αυτήν, ένα ειδικά σχεδιασμένο λογισμικό, που ονομάζεται hypervisor ή Virtual Machine Monitor (VMM), αναλαμβάνει την προσομοίωση ολόκληρου του υλικού του φυσικού μηχανήματος. Έπειτα, δημιουργεί εικονικές μηχανές που χρησιμοποιούν αυτό το εικονικό υλικό, με αποτέλεσμα κάθε εικονικό μηχάνημα να θεωρεί πως δρα στο δικό του χώρο πυρήνα (kernel-space). Για την πλήρη εικονικοποίηση, απαιτείται η κατάλληλη υποστήριξη της συγκεκριμένης τεχνολογίας από την μεριά του επεξεργαστή.

Οι hypervisors χωρίζονται σε δύο κατηγορίες [3]. Η πρώτη κατηγορία ονομάζεται τύπου Ι ή φυσικοί hypervisors και είναι αυτοί που ελέγχουν απευθείας το υλικό ώστε να φιλοξενούν τα πολλαπλά εικονικά μηχανήματα. Τέτοιοι hypervisors είναι οι Microsoft Hyper-V, VMWare ESXi και Proxmox VE. Η δεύτερη κατηγορία είναι οι φιλοξενούμενοι ή τύπου ΙΙ και περιλαμβάνει hypervisors που είναι εγκατεστημένοι ως προγράμματα σε κάποιο λειτουργικό σύστημα. Τα προγράμματα αυτά χρησιμοποιούν το λειτουργικό για την επικοινωνία τους με το υλικό και τα εικονικά μηχανήματα εκτελούνται ως διεργασίες του. Οι γνωστότεροι hypervisors αυτής της κατηγορίας είναι οι VirtualBox, VMWare Workstation και QEMU. Ωστόσο υπάρχουν περιπτώσεις που όπου ο παραπάνω διαχωρισμός δεν είναι σαφής, όπως στην περίπτωση του KVM των Linux.



Εικόνα 2-2: Οι τύποι των hypervisors

2.1.2 Εικονικοποίηση στο επίπεδο του Λειτουργικού Συστήματος

Μια εναλλακτική μορφή εικονικοποίησης, ιδιαίτερα αναπτυσσόμενη τα τελευταία χρόνια, είναι η εικονικοποίηση στο επίπεδο του λειτουργικού συστήματος (OS-level virtualization) [4]. Συγκεκριμένα, σε αυτή τη μορφή, ο πυρήνας του λειτουργικού συστήματος, με χρήση κατάλληλου λογισμικού, επιτρέπει την δημιουργία πολλαπλών απομονωμένων οντοτήτων περιοχής χρήστη (user-space), τις οποίες συνήθως ονομάζουμε containers. Από την πλευρά του

χρήστη αυτή η μορφή εικονικοποίησης δεν διαφέρει από την πλήρη, καθώς, όπως και στην πλήρη εικονικοποίηση, τα εικονικά μηχανήματα θεωρούν πως είναι αυτόνομα και δεν γνωρίζουν την ύπαρξη άλλων containers. Ωστόσο, από την τεχνική πλευρά έχουν αρκετές διαφορές, με αποτέλεσμα τα containers να πλεονεκτούν έναντι των virtual machines σε κάποιες εφαρμογές.

Στην πλήρη εικονικοποίηση που παρουσιάστηκε παραπάνω, αναφέρθηκε η αναγκαιότητα ύπαρξης κάποιου hypervisor που να εξομοιώνει πλήρως το υλικό της συσκευής. Αντίθετα, στην περίπτωση των container η παρουσία του είναι περιττή, καθώς η εικονικοποίηση επιτυγχάνεται μέσω του λογισμικού εικονικοποίησης (virtualization software). Το λογισμικό αυτό δίνει την δυνατότητα στις εικονικές μηχανές να χρησιμοποιούν απ' ευθείας τις βιβλιοθήκες του λειτουργικού συστήματος του φυσικού μηχανήματος. Έτσι, λόγω αυτής της αρχιτεκτονικής, δεν απαιτείται κάποια ειδική υποστήριξη από την πλευρά του επεξεργαστή. Επιπλέον, η απουσία του hypervisor, καθιστά τα container πιο γρήγορα, καθώς η παρουσία του δημιουργεί μια επιπλέον καθυστέρηση στην λειτουργία των εικονικών μηχανών. Τα παραπάνω έχουν ως αποτέλεσμα την σχεδόν μηδενική διάρκεια δημιουργίας (και διαγραφής αντίστοιχα) ενός container, κάτι που είναι ιδιαίτερα σημαντικό για τις εφαρμογές που απαιτούν την γρήγορη δέσμευση ή αποδέσμευση πόρων, όπως στην περίπτωση της συγκεκριμένης διπλωματικής εργασίας.

Ωστόσο, ένα μειονέκτημα που παρουσιάζουν τα containers έναντι των VMs είναι η μειωμένη ελαστικότητα, όσον αφορά την επιλογή του λειτουργικού συστήματος της εικονικής μηχανής. Ειδικότερα, ένα container πρέπει να έχει λειτουργικό σύστημα της ίδιας οικογένειας με το φυσικό μηχάνημα. Αυτό οφείλεται στο γεγονός πως τα containers, για να πετύχουν τα παραπάνω, χρησιμοποιούν κοινές βιβλιοθήκες με το φυσικό μηχάνημα. Ωστόσο, αυτός ο περιορισμός δεν αποτελεί πρόβλημα στην συγκεκριμένη περίπτωση.

Στην παρούσα διπλωματική εργασία, με γνώμονα την εκμετάλλευση των παραπάνω πλεονεκτημάτων, χρησιμοποιείται μια μέθοδος εικονικοποίησης στο επίπεδο του λειτουργικού συστήματος που ονομάζεται LXC (LinuX Containers) που έχει αναπτυχθεί από πληθώρα προγραμματιστικών ομάδων, μεταξύ των οποίων η IBM και η Google.

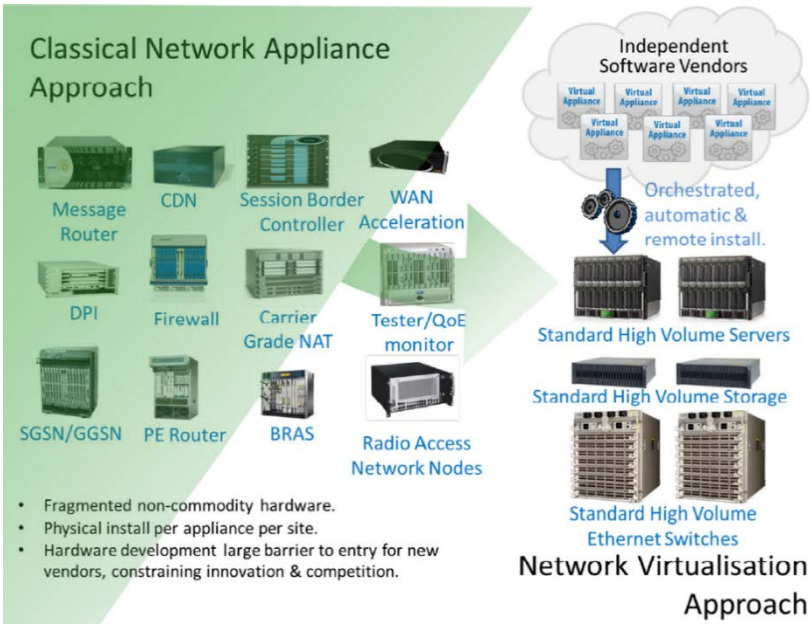
2.2 Εικονικοποίηση δικτυακών λειτουργιών (NFV)

2.2.1 Εισαγωγή

Όπως αναφέρθηκε και στην προηγούμενη υποενότητα, η χρήση φυσικών συστημάτων για την υλοποίηση ενός σκοπού μπορεί να είναι πολύ ασύμφορη. Στον κόσμο των τηλεπικοινωνιών, οι πάροχοι διαθέτουν τεράστιο πλήθος από μεγάλες και πολύπλοκες συσκευές, το οποίο αυξάνεται συνεχώς με την εισαγωγή νέων υπηρεσιών. Αυτό αυξάνει εξαιρετικά τα επενδυτικά έξοδα (CapEx), καθώς πέρα από την αγορά του νέου εξοπλισμού απαιτείται φυσικός χώρος για την τοποθέτηση του, αλλά και τα αντίστοιχα λειτουργικά (OpEx), τα οποία αφορούν τα έξοδα

κατανάλωσης, ψύξης και συντήρησης του. Επίσης, μεγάλο πλήθος του παραπάνω εξοπλισμού, λόγω της ταχύτατης ανάπτυξης του κλάδου, συχνά θεωρείται ξεπερασμένο και ανεπαρκές με αποτέλεσμα να απαιτείται συνεχής αντικατάσταση του. Τέλος, σημαντικός παράγοντας είναι και ο κύκλος ζωής των φυσικών μηχανημάτων, ο οποίος προσθέτει επιπλέον κόστος στους παρόχους.

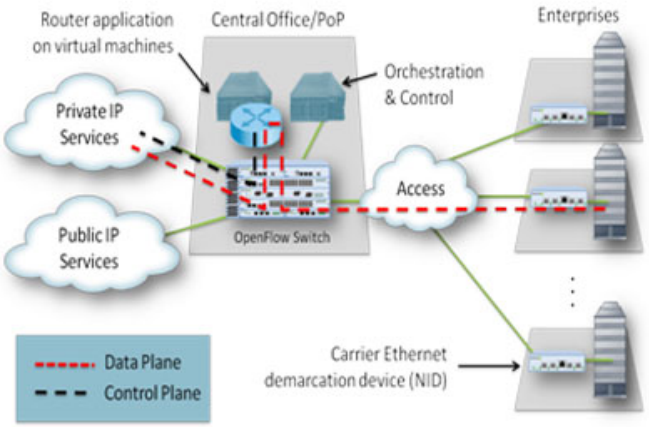
Έτσι, στις αρχές της τρέχουσας δεκαετίας, έκανε την εμφάνιση της μια νέα τεχνολογία, η εικονικοποίηση δικτυακών λειτουργιών (Network Function Virtualization – NFV). Πρόκειται για μια αρχιτεκτονική εικονικών δικτύων η οποία έχει ως στόχο την λύση των παραπάνω προβλημάτων προσφέροντας την δυνατότητα σχεδίασης, υλοποίησης και διαχείρισης διαφόρων δικτυακών υπηρεσιών. Ειδικότερα, χρησιμοποιεί τις τεχνολογίες εικονικοποίησης για να εξομοιώσει τις λειτουργίες δικτυακών κόμβων σε δομικά στοιχεία που μπορούν να συνδεθούν μαζί για να δημιουργήσουν υπηρεσίες τηλεπικοινωνιών. Στόχος της εν λόγω τεχνολογίας είναι να αποσυνδέσει τις λειτουργίες δικτύου από την φυσική συσκευή με σκοπό την εκτέλεση τους από κατάλληλο λογισμικό, όπως την μετάφραση διεύθυνσης IP (Network Address Translation – NAT), του τείχους προστασίας και του διακομιστή DNS.



Εικόνα 2-3: Το όραμα της τεχνολογίας NFV

Τα δίκτυα NFV είναι κατάλληλα σχεδιασμένα έτσι ώστε να εξομοιώνουν πλήρως επιμέρους μονάδες φυσικών δικτύων, όπως εικονικούς εξυπηρετητές, συσκευές αποθήκευσης ή ακόμη και άλλα δίκτυα. Η εξομοίωση αυτή επιτυγχάνεται χρησιμοποιώντας τις πρότυπες τεχνικές εικονικοποίησης υπολογιστών, με μια βασική διαφορά. Μια εικονικοποιημένη δικτυακή λειτουργία (Virtualized Network Function – VNF) προορίζεται να εκτελέσει μια συγκεκριμένη λειτουργία δικτύου, π.χ. δρομολόγηση, μεταγωγή, τείχος προστασίας, εξισορρόπηση φορτίου κ.λπ. και μπορεί να χρειαστεί ένας συνδυασμός αυτών των VNF για την υλοποίηση του πλήρους τμήματος δικτύου που είναι εικονικοποιημένο. Αυτές οι λειτουργίες εκτελούνται πάνω σε

κοινούς εξυπηρετητές, μεταγωγείς και συσκευές αποθήκευσης, ή ακόμα και σε υπολογιστικά νέφη, αντί να χρειάζεται εξειδικευμένες συσκευές για κάθε λειτουργία του δικτύου.



Εικόνα 2-4: Δείγμα υλοποίησης δικτύου NFV για VPN

2.2.2 Πλαίσιο αρχιτεκτονικής δικτύων NFV

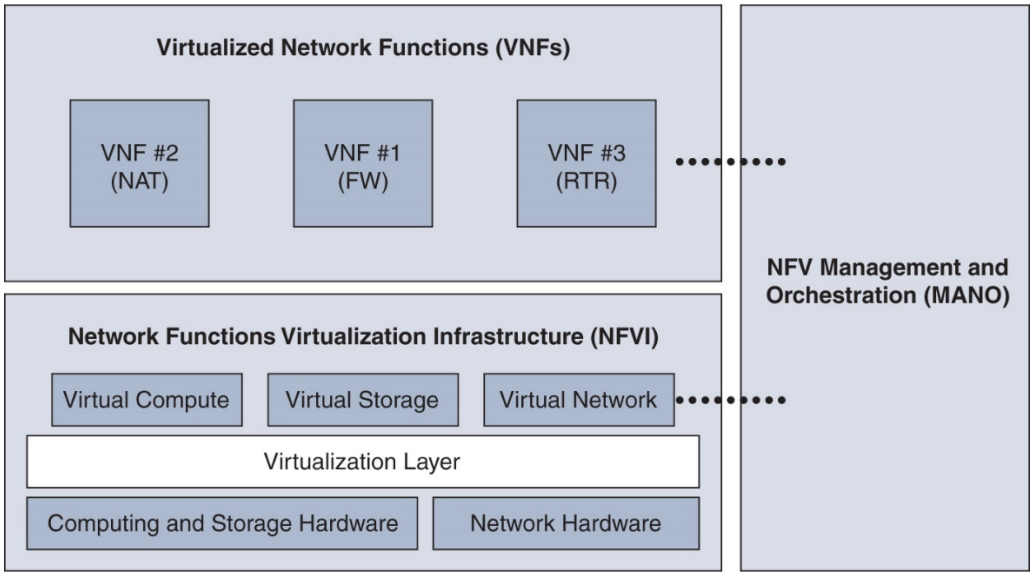
Η αρχιτεκτονική που ορίζει τις παραδοσιακές συσκευές δικτύου είναι αρκετά απλή δεδομένου ότι τόσο το υλικό όσο και το λογισμικό είναι προσαρμοσμένα και στενά ενσωματωμένα. Αντίθετα, η τεχνολογία των NFV επιτρέπει στο λογισμικό να τρέχει σε γενικό κοινόχρηστο υλικό δημιουργώντας πολλαπλά σημεία επαφής για τη διαχείριση. Κάτι τέτοιο έχει ως αποτέλεσμα την αναγκαιότητα ύπαρξης ενός πλαισίου για την τυποποίηση της τεχνολογίας αυτής. Αυτό το πρότυπο πλαίσιο θα πρέπει να διασφαλίζει ότι οι εικονικές λειτουργίες που αναπτύσσονται δεν συνδέονται με συγκεκριμένο υλικό. Με τον τρόπο αυτό, διασφαλίζεται ότι για την υλοποίηση των λειτουργιών αυτών δεν απαιτεί ειδική τροποποίηση στο υλικό του συστήματος.

Το πλαίσιο αρχιτεκτονικής της τεχνολογίας NFV αναπτύχθηκε το 2013 από το Ευρωπαϊκό Ινστιτούτο Τηλεπικοινωνιακών Προτύπων (European Telecommunications Standards Institute – ETSI) και αποτελείται από 3 βασικά μέρη [1] [5]:

1. **Υποδομή εικονικοποίησης δικτυακών λειτουργιών (NFV Infrastructure – NFVI)**: Αποτελεί την βάση της συνολικής αρχιτεκτονικής και περιλαμβάνει το σύνολο των εικονικών και φυσικών πόρων επεξεργασίας, αποθήκευσης και δικτύωσης, καθώς και το λογισμικό εικονικοποίησης. Η υποδομή NFV μπορεί να εκτείνεται σε περισσότερες από μια τοποθεσίες ενώ το δίκτυο που παρέχει συνδεσιμότητα ανάμεσα σε αυτές τις τοποθεσίες θεωρείται μέρος της υποδομής.
2. **Εικονικοποιημένες δικτυακές λειτουργίες (Virtual Network Functions – VNFs)**: Πρόκειται για υλοποιήσεις διάφορων δικτυακών λειτουργιών, στο επίπεδο του

λογισμικού, που εκτελούνται στις εικονικές μηχανές που παρέχονται από την υποδομή NFV.

3. **Πλαίσιο διαχείρισης και ενορχήστρωσης (Management ANd Orchestration - MANO)**: Ορίζεται ως ξεχωριστή μονάδα της αρχιτεκτονικής, που αλληλοεπιδρά με την υποδομή NFV και τα VNFs. Αποτελεί το σύνολο όλων των λειτουργικών μονάδων, των δεδομένων που αυτά χρησιμοποιούν, των σημείων αναφοράς και των διεπαφών μέσω των οποίων αυτές ανταλλάσσουν πληροφορίες με σκοπό τη διαχείριση όλων των πόρων της υποδομής NFV και την ενορχήστρωση των VNFs.

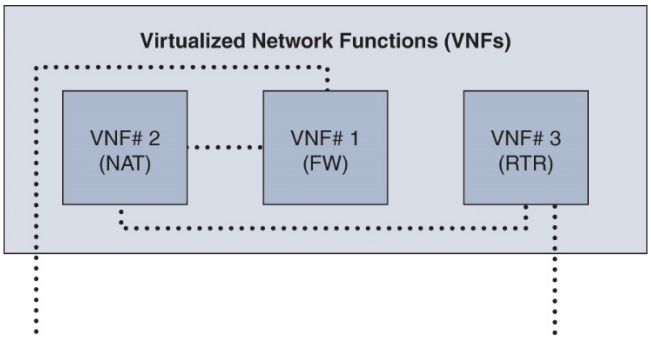


Εικόνα 2-5: Το πλαίσιο αρχιτεκτονικής των NFV σε υψηλό επίπεδο

Ο στόχος του καθορισμού του πλαισίου, και ειδικότερα των επιμέρους λειτουργικών τμημάτων και των σημείων αναφοράς, είναι η προσπάθεια εξάλειψης των προκλήσεων λειτουργικότητας και η τυποποίηση της υλοποίησης. Ο σκοπός και το πεδίο εφαρμογής κάθε ενός από αυτά τα τμήματα είναι σαφώς καθορισμένα στο πλαίσιο. Ομοίως, οι αλληλεπιδράσεις και οι διαδρομές επικοινωνίας ορίζονται μέσω των σημείων αναφοράς και προορίζονται να είναι τυποποιημένες μέθοδοι [1]. Με αυτόν τον τρόπο, χρησιμοποιώντας το παραπάνω πλαίσιο αρχιτεκτονικής, επιτυγχάνεται η πλήρης ευελιξία τους δικτύου. Συγκεκριμένα, το γεγονός πως οι εφαρμογές και το υλικό είναι πλήρως διαχωρισμένες, επιτρέπει την αυτοματοποίηση και την κλιμάκωση της αναπαραγωγής των δικτύων NFV. Επιπλέον, η παραπάνω αρχιτεκτονική επιτρέπει την συνεχή παρακολούθηση των λειτουργικών παραμέτρων του δικτύου, μέσω της διαχειριστικής μονάδας. Το πλαίσιο αρχιτεκτονικής του ETSI και η διαδικασία σκέψης πίσω από τις μονάδες υψηλού επιπέδου που το αποτελούν μπορούν να γίνουν περισσότερο κατανοητά μελετώντας τη διαδικασία οικοδόμησης που οδήγησε σε αυτό το πλαίσιο.

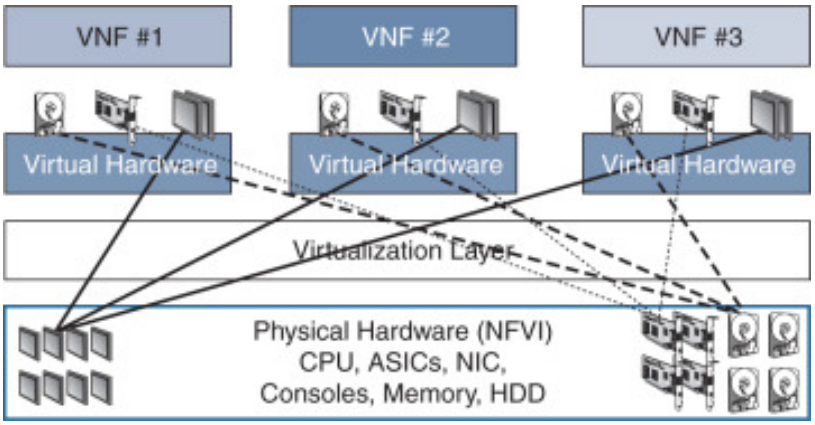
Ξεκινώντας με την θεμελιώδη λειτουργική μονάδα του NFV, τα VNFs μπορούν να αναπτυχθούν είτε ως αυτοτελείς οντότητες είτε ως συνδυασμός πολλαπλών VNFs [1]. Τα

πρωτόκολλα που σχετίζονται με τη εικονικοποιημένη λειτουργία μέσα σε ένα VNF δεν χρειάζεται να γνωρίζουν την συνολική υλοποίηση και επικοινωνούν μεταξύ τους χωρίς να γνωρίζουν ότι δεν είναι σωματικά συνδεδεμένες ή ότι λειτουργούν σε αποκλειστικές φυσικές συσκευές. Επίσης, δεδομένου ότι δεν υπάρχει αποκλειστικό ή προσαρμοσμένο υλικό που έχει σχεδιαστεί για την εκτέλεση αυτών των VNFs, μπορεί να χρησιμοποιηθεί μια εικονική συσκευή με μη εξειδικευμένους πόρους υλικού, όπως επεξεργαστή (CPU), μνήμη, τοπικό δίσκο και διασυνδέσεις δικτύου, για την εκτέλεση αυτών των VNF. Οι τεχνολογίες εικονικοποίησης που δύναται να χρησιμοποιηθούν είναι είτε η πλήρης είτε αυτή στο επίπεδο του λειτουργικού συστήματος, οι οποίες εφαρμόζονται για αρκετά χρόνια σε κέντρα δεδομένων και κρίνονται ως αρκετά ώριμες. Τέλος, εξετάζοντας την σε χαμηλότερο επίπεδο, η μονάδα αυτή περιλαμβάνει επιπλέον την υπομονάδα διαχείρισης (Element –Management – EM), όπου βοηθάει στην υλοποίηση των λειτουργιών διαχείρισης ενός ή περισσότερων VNF.



Εικόνα 2-6: Πολλαπλά VNF που συνεργάζονται για την λειτουργία ενός γενικού VNF

Όπως αναφέρθηκε και παραπάνω τα VNFs εκτελούνται από εικονικές μηχανές, οι οποίες τοποθετούνται στην NFV υποδομή (NFVI). Η υποδομή αυτή χρησιμοποιεί το σύνολο των φυσικών πόρων και αναπτύσσει υποσύνολα αυτών, δημιουργώντας με αυτόν τον τρόπο εικονικές υπολογιστικές, αποθηκευτικές και δικτυακές δεξαμενές που παρέχει στα VNFs για την λειτουργία τους. Για την ομαλή λειτουργία του VNF, ο δημιουργός ορίζει κάποια ελάχιστα χαρακτηριστικά τα οποία οφείλουν να παρέχονται από το NFVI.

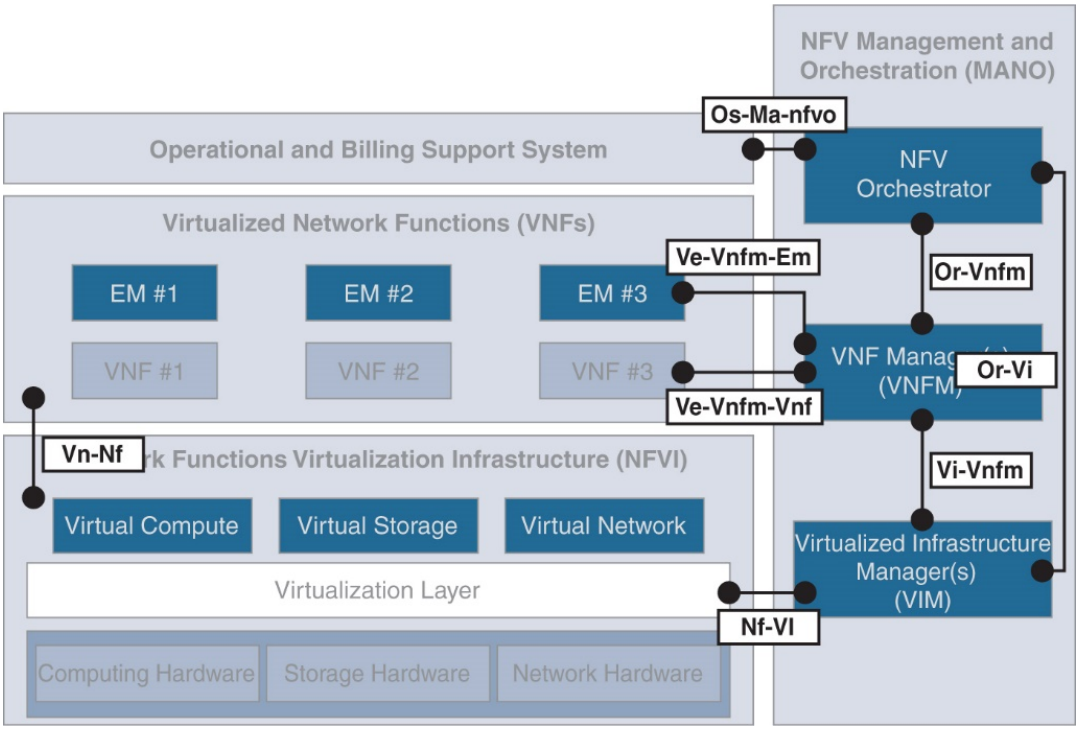


Εικόνα 2-7: Κοινοί φυσικοί πόροι που παρέχονται στα VNFs από την μονάδα NFVI

Σε αντίθεση με τα παραδοσιακά δίκτυα, όπου η διαχείριση τους περιορίζεται στην διαχείριση των επιμέρους φυσικών συσκευών, στα NFV δίκτυα τα σημεία διαχείρισης είναι σαφώς περισσότερα. Αυτό οφείλεται στο γεγονός πως οι συσκευές και οι λειτουργίες τους, όπως έγινε κατανοητό και παραπάνω, έχουν επιμεριστεί σε περισσότερες οντότητες. Ως αποτέλεσμα, η μονάδα διαχείρισης και ενορχήστρωσης (MANO) συγκεντρώνει το μεγαλύτερο ενδιαφέρον, καθώς προορίζεται να έχει πλήρη ορατότητα αυτών των οντοτήτων και είναι υπεύθυνο για την διαχείριση και την εποπτεία τους. Για τον λόγο αυτό, αξίζει να αναλυθεί σε χαμηλότερο επίπεδο. Η μονάδα MANO, λοιπόν, μπορεί να αναλυθεί σε τρία μικρότερα επίπεδα [1] [5]:

1. **Διαχειριστής εικονικής υποδομής (Virtualized Infrastructure Manager – VIM)**: Αποτελεί τη μονάδα διαχείρισης του NFVI και είναι υπεύθυνο για τον έλεγχο και την διαχείριση των υπολογιστικών, αποθηκευτικών και δικτυακών πόρων της NFVI υποδομής, καθώς και του λογισμικού που εφαρμόζει την εικονικοποίηση. Εφόσον η μονάδα VIM διαχειρίζεται άμεσα τους παραπάνω φυσικούς πόρους, πρέπει να διαθέτει πλήρη ορατότητα των τεχνικών και λειτουργικών χαρακτηριστικών τους, όπως η διαθεσιμότητα και η κατανάλωση ισχύς τους. Αξίζει να σημειωθεί πως η συγκεκριμένη μονάδα μπορεί να αποτελείται από πλήθος υπομονάδων που διαχειρίζονται και εποπτεύουν διαφορετικούς φυσικούς πόρους.
2. **Διαχειριστής VNF (VNF Manager - VNFM)**: Ευθύνη της μονάδας αυτής είναι η διαχείριση FCAPS (Fault, Configuration, Accounting, Performance and Security) των VNFs, είτε μέσω άμεσης επικοινωνίας μαζί τους είτε διαμέσου της μονάδας EM. Επιπλέον, αναλαμβάνει την αναβάθμιση των πόρων που χρειάζεται το VNF σε συνεννόηση με την μονάδα VIM. Συγκεκριμένα, όταν το VNF για την λειτουργία του απαιτεί επιπλέον πόρους, όπως επιπλέον μνήμη, η μονάδα αυτή μεταφέρει το αίτημα στο VIM, το οποίο, έχοντας εικόνα των διαθέσιμων πόρων, αποφασίζει η απαίτηση αυτή αν είναι δυνατόν να πραγματοποιηθεί. Αν κάτι τέτοιο είναι εφικτό, ζητά από την υποδομή NFV να τροποποιήσει τους πόρους που διαθέτει στο εικονικό μηχάνημα που εκτελεί την εικονικοποιημένη λειτουργία.
3. **Ενορχηστρωτής VNF (VNF Orchestrator – VNFO)**: Η μονάδα αυτή έχει ως κύρια λειτουργία την επικοινωνία με το σύστημα OSS/BSS (Operations System Support/Business System Support), το οποίο είναι ένα σύστημα που οργανώνει την διαχείριση και την υποστήριξη των παρεχόμενων υπηρεσιών στους πελάτες από τους τηλεπικοινωνιακούς παρόχους. Ρόλος του VNFO είναι, μέσω αυτής της επικοινωνίας, να διαβιβάσει τις απαραίτητες πληροφορίες στα VIM και VNFM για την υλοποίηση των παρεχόμενων υπηρεσιών. Επιπλέον, το VNFO επικοινωνεί με το VIM με σκοπό την πλήρη γνώση και κατανομή των διαθέσιμων πόρων.

Επιπλέον, στο πλαίσιο αρχιτεκτονικής του ETSI ορίζονται σημεία αναφοράς για τον εντοπισμό της επικοινωνίας μεταξύ των μονάδων που το αποτελούν. Ο προσδιορισμός και ο καθορισμός αυτός είναι απαραίτητος για την συνέπεια της ροής πληροφορίας που διακινείται μεταξύ αυτών, δημιουργώντας έναν κοινό τρόπο ανταλλαγής πληροφοριών μεταξύ αυτών. Τα σημεία αναφοράς, λοιπόν, που ορίζονται στο πλαίσιο του ETSI για τα NFV δίκτυα είναι τα εξής [1] [5]:



Εικόνα 2-8: Σημεία αναφοράς πλαισίου αρχιτεκτονικής NFV

- **Os-Ma-nfvo:** Προορίζεται για τον καθορισμό της επικοινωνίας μεταξύ των OSS/BSS και NFVO. Αποτελεί το μοναδικό σημείο αναφοράς μεταξύ αυτών των μονάδων και χρησιμοποιείται για τον πλήρη προσδιορισμό των παρεχόμενων υπηρεσιών.
- **Ve-Vnfm-Vnf:** Εξυπηρετεί στην επικοινωνία μεταξύ των VNFM και VNF. Χρησιμοποιείται από το VNFM για τη διαχείριση του κύκλου ζωής VNF και για την ανταλλαγή ρυθμίσεων και πληροφοριών κατάστασης με το VNF.
- **Ve-Vnfm-Em:** Πραγματοποιείται μεταξύ του VNFM και της μονάδας EM. Σκοπός του είναι η διαχείριση FCAPS, όταν αυτή επιτυγχάνεται μέσω του EM.
- **Nf-Vi:** Αυτό το σημείο αναφοράς ορίζει την ανταλλαγή πληροφοριών μεταξύ του VIM και των λειτουργικών ομάδων στο NFVI. Χρησιμοποιείται για την κατανομή, την διαχείριση και τον έλεγχο των πόρων που διαθέτη το NFVI.
- **Or-Vnfm**: Η επικοινωνία μεταξύ NFVO και VNFM συμβαίνει μέσω αυτού του σημείου αναφοράς και αφορά την εγκατάσταση των VNF και τη ροή πληροφοριών που σχετίζονται με τον κύκλο ζωής τους.

- **Or-Vi:** Το NFVO επικοινωνεί με VIM μέσω αυτού του σημείου αναφοράς για να επηρεάσει τη διαχείριση των πόρων της υποδομής, όπως η δέσμευση πόρων για την προσθήκη VM ή VNF.
- **Vi-Vnfm:** Αυτό το σημείο αναφοράς προορίζεται για τον καθορισμό των προτύπων ανταλλαγής πληροφοριών μεταξύ VIM και VNFM, όπως η αίτηση ενημέρωσης πόρων για VM που εκτελεί VNF.
- **Vn-Nf:** Το μοναδικό σημείο αναφοράς που δεν συμμετέχει για κάποια μονάδα διαχείρισης. Προορίζεται να γνωστοποιήσει τις ανάγκες επίδοσης και φορητότητας του VNF στο μονάδα υποδομής NFV.

Στον παρακάτω πίνακα παρουσιάζονται αναλυτικά τα σημεία αναφοράς, όπως αυτά ορίζονται εκτενώς στο πλαίσιο των NFV που αναπτύχθηκε από το ETSI [1]:

Reference Point	Boundaries	Use Defined in the Framework
Os-Ma-nfvo	OSS/BSS<->NFVO	• Service description and VNF package management. • Network service lifecycle management (instantiation, query, update, scaling, and termination). • VNF life cycle management. • Policy management (access, authorization, etc.) for network service instances. • Querying network service and VNF instances from OSS/BSS. Forwarding events, usage, and performance of network service instances to OSS/BSS.
Ve-Vnfm-vnf	VNFM<->VNF	• Instantiation, instance query, update, scaling up or down, and termination of the VMs. • Configuration and events regarding VNF, from VNFM to VNF. • Configuration and events from VNF to VNFM.
Ve-Vnmf-em	VNFM<->EM	• Instantiation, instance query, update, scaling up or down, and termination of the VMs. • Configuration and events regarding VNF from VNFM to EM. • Configuration and events from EM to VNFM.
Nf-Vi	NFVI<->VIM	• Allocate, update, migrate, terminate VMs. • Create, configure, remove inter-VM connections. • Failure events, usage records, configuration information to the VIM for NFVI resources (physical, software, virtual).
Or-Vnfm	NFVO<->VNFM	• Instantiation, state query, update, scaling, termination and package query of the VNF. • Forwarding VNF events and state information.
Or-Vi	NFVO<->VIM	• NFVI resource reservation, release, and update. • VNF software image allocation, deallocation, and update. • Configuration, usage, events, and results of NFVI to NFVO.
Vi-Vnfm	VIM<->VNFM	• NFVI resource reservation, allocation, and release information. • Events, usage, measurement results, etc. for a NFVI resource used by a VNF.
Vn-Nf	NFVI<->VNF	• Lifecycle, performance, and portability requirements of VNF.

Πίνακας 1: Σημεία αναφοράς του πλαισίου NFV

2.2.3 Τα πλεονεκτήματα των NFV

Κατά το πρώτο κεφάλαιο, παρουσιάστηκαν ενδεικτικά κάποιοι από τους περιορισμούς των παραδοσιακών φυσικών συσκευών δικτύου. Η τεχνολογία των NFV έρχεται να επιλύσει αυτούς τους περιορισμούς, προσφέροντας επιπλέον πολλά και σημαντικά πλεονεκτήματα για την ανάπτυξη του κλάδου των τηλεπικοινωνιών. Τα κυριότερα αυτών είναι τα εξής [1] [5]:

- **Μείωση CapEx και OpEx**: Τα επενδυτικά έξοδα μειώνονται, λόγω του μειωμένου κόστους εξοπλισμού, καθώς δεν απαιτείται κάποιο εξειδικευμένο υλικό για την εκτέλεση των VNF. Επίσης, πτώση παρατηρείται και στις λειτουργικές δαπάνες, όπως της ενεργειακής κατανάλωσης, μέσω της ενοποίησης του εξοπλισμού και διαμοιρασμού του φόρτου εργασίας, καθώς και μέσω της εκμετάλλευσης της ανάπτυξης των σύγχρονων τεχνολογιών διαχείρισης ισχύος σε τυπικούς διακομιστές και συσκευές αποθήκευσης.
- **Ευελιξία υλικού**: Το γεγονός ότι το υλικό και το λογισμικό στην τεχνολογία αυτή είναι πλήρως ανεξάρτητα, παρέχει την ελευθερία στους παρόχους να μην περιορίζονται αποκλειστικά σε έναν προμηθευτή για λόγους συμβατότητας με τον υπόλοιπο εξοπλισμό. Επιπλέον, οι φυσικοί πόροι λόγω της ομοιομορφίας τους δύναται να χρησιμοποιηθούν για την δυναμική επέκταση οποιασδήποτε λειτουργίας.
- **Ταχύτερος κύκλος ζωής υπηρεσιών**: Αύξηση της ταχύτητας του απαιτούμενο χρόνου κυκλοφορίας μιας υπηρεσίας (Time-to-Market), ελαχιστοποιώντας τον τυπικό κύκλο κυκλοφορίας. Τα εμπόδια που προκύπτουν από τα απαιτούμενα επενδυτικά κόστη (αγορά νέου εξοπλισμού) και τους λειτουργικούς περιορισμούς (χρόνος εγκατάστασης του) δεν υφίστανται. Επιπλέον, η δοκιμή και η ολοκλήρωση των υπηρεσιών είναι ταχύτερη, καθώς οι δυνατότητες εκτέλεσης και ελέγχου αυτών είναι περισσότερες και σαφώς αποτελεσματικότερες. Έτσι μειώνονται το κόστος ανάπτυξης και ο χρόνος κυκλοφορίας τους.
- **Ελαστικότητα**: Οι ανάγκες των λειτουργιών για φυσικούς πόρους είναι πιθανό να ποικίλουν, ανάλογα με την χρήση του δικτύου την δεδομένη χρονική στιγμή. Μέσω της δυναμικής δέσμευσης ή αποδέσμευσης των φυσικών πόρων, βελτιστοποιείται η λειτουργία των υπηρεσιών, ενώ εξαλείφονται περιπτώσεις λειτουργιών που έχουν περίσσια πόρων την στιγμή που άλλοι έχουν έλλειψη.
- **Εξειδικευμένες υπηρεσίες:** Οι υπηρεσίες είναι δυνατόν να είναι στοχοθετημένες ανάλογα με την γεωγραφική θέση ή για συγκεκριμένο σύνολο των πελατών. Η κλιμάκωση των υπηρεσιών είναι μεγαλύτερη ανάλογα με τις δεδομένες ανάγκες και η εξυπηρέτηση των πελατών είναι ευκολότερη, δίνοντας λύσεις εξ αποστάσεων, καθώς δεν απαιτείται φυσική παρουσία για την εγκατάσταση νέου υλικού.

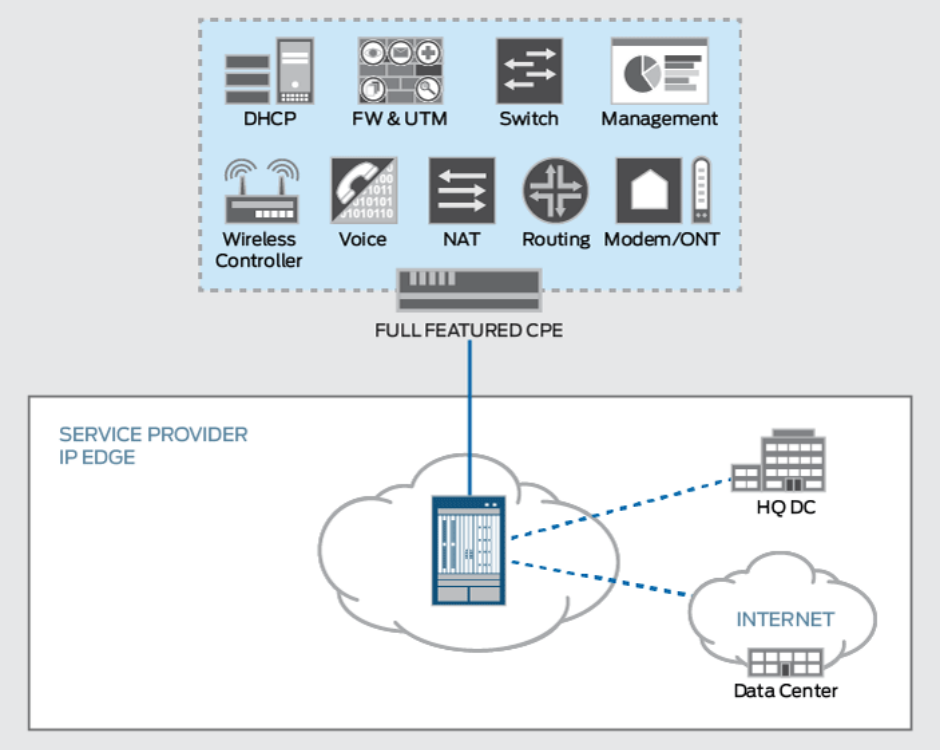
- **Ενθάρρυνση καινοτομίας**: Δίνεται η δυνατότητα σε εταιρίες, ακαδημαϊκούς αλλά και αυτόνομους προγραμματιστές να αναπτύξουν λογισμικά και υπηρεσίες και να τις δοκιμάσουν με μεγαλύτερη ευκολία.
- **Βελτιστοποίηση δικτύου**: Η διαμόρφωσης του δικτύου δύναται να αλλάξει σε πραγματικό χρόνο με βάση τα πραγματικά μοντέλα κίνησης και ζήτησης υπηρεσιών.
- **Γρήγορη αποκατάσταση**: Η εγκατάσταση και η επαναχρησιμοποίηση των εικονικών συσκευών είναι ταχύτερη και οι απώλειες σε περίπτωση βλάβης του φυσικού μηχανήματος μειώνονται δραματικά, λόγω των αντιγράφων κατάστασης των εικονικών συσκευών. Λόγω της ομοιομορφίας των φυσικών συστημάτων οι εξαρτίσεις των εφαρμογών από συγκεκριμένης φύσεως υλικού εξαλείφονται και οι παραπάνω διαδικασίες υλοποιούνται γρήγορα και εύκολα.

2.3 Η τεχνολογία του vCPE

2.3.1 Συμβατικά CPE

Με τον όρο CPE (Customer-Premises Equipment) αναφερόμαστε σε οποιαδήποτε τερματική συσκευή που είναι τοποθετημένη στον χώρο του πελάτη και έχει στόχο την σύνδεση του με τον πάροχο μέσω κατάλληλου τηλεπικοινωνιακού καναλιού. Αν και ο όρος είναι σχετικά γενικός και μπορεί να αναφέρεται ακόμα στις οικιακές τηλεφωνικές συσκευές, στον χώρο των δικτύων, χρησιμοποιείται κυρίως για να περιγράψει τους δρομολογητές που αναλαμβάνουν την σύνδεση του πελάτη με το WAN (Wide Area Network), κοινώς αυτό που αποκαλούμε Internet [6].

Αν και υπάρχουν νέες εξελιγμένες μορφές, που θα γίνουν κατανοητές στην συνέχεια της παρούσας διπλωματικής, κατά κύριο λόγο στην σύγχρονη εποχή χρησιμοποιούνται τα συμβατικά CPE ή αλλιώς Full Featured CPE. Πρόκειται για συσκευές που αναλαμβάνουν εξ ολοκλήρου τις υπηρεσίες που παρέχονται στον πελάτη από τον τηλεπικοινωνιακό πάροχο, καθώς και εσωτερικές εργασίες του δικτύου για την ομαλή λειτουργία του. Επιγραμματικά, στις υποχρεώσεις που έχει ένα κοινό οικιακό CPE περιλαμβάνεται η δρομολόγηση και προώθηση των πακέτων που διακινούνται στο εσωτερικό του δικτύου (ενσύρματων ή ασύρματων), η μετατροπή της ιδιωτικής σε δημόσια διεύθυνση IP και αντιστρόφως (NAT) για την επικοινωνία των οικιακών υπολογιστών με το υπόλοιπο διαδίκτυο, καθώς και ο δυναμικός διαμοιρασμός των οικιακών διευθύνσεων IP ώστε να αποφεύγεται η ταυτόχρονη χρήση διευθύνσεων από πολλαπλούς υπολογιστές (DHCP). Επίσης, προστατεύει το οικιακό δίκτυο από εξωτερικούς κινδύνους και περιορίζει την πρόσβαση σε σελίδες ανεπιθύμητου περιεχομένου μέσω του firewall, ενώ αναλαμβάνει την διαχείριση των υπηρεσιών του παρόχου, όπως υπηρεσίες τηλεφώνου (Voice over IP – VoIP) και τηλεόρασης (Pay TV) [7].



Εικόνα 2-9: Οι λειτουργίες ενός συμβατικού CPE

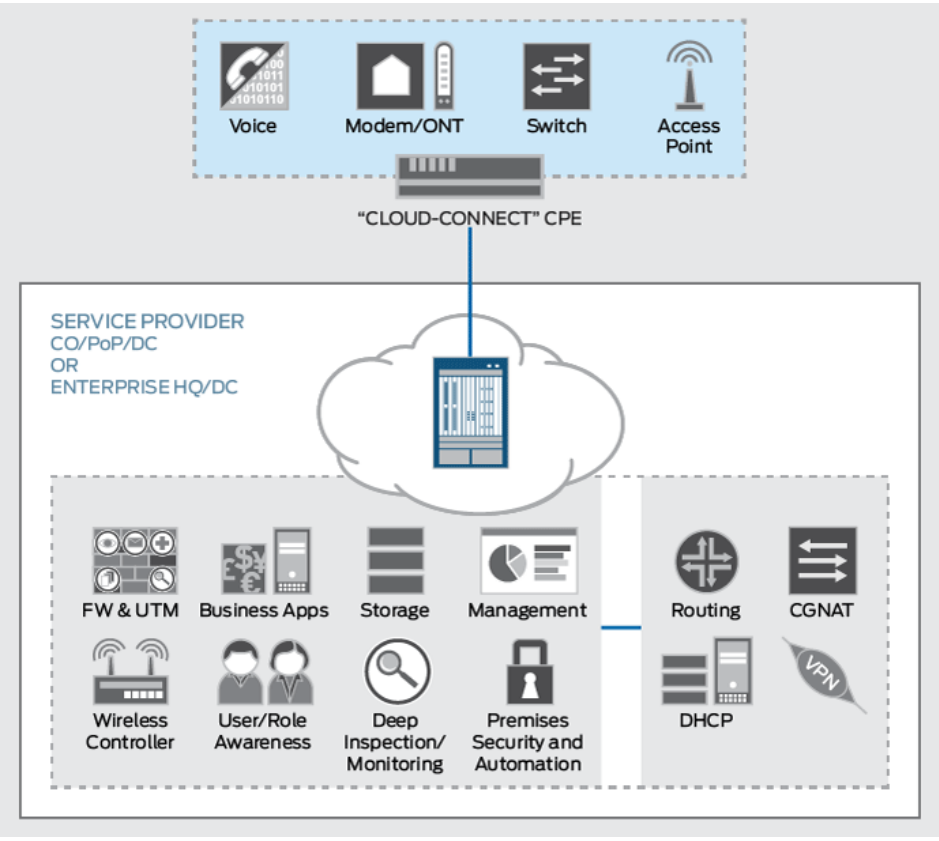
Είναι προφανές πως, με την χρήση των συμβατικών CPE, ο οικιακός δρομολογητής αναλαμβάνει εξ ολοκλήρου το υπολογιστικό βάρος για την διασύνδεση του πελάτη στο διαδίκτυο και των λοιπών υπηρεσιών που του παρέχονται. Από την ιστορική πλευρά, κάτι τέτοιο ήταν απαραίτητο, καθώς κατά την ραγδαία ανάπτυξη του διαδικτύου και γενικότερα της επιστήμης των δικτύων, η υπολογιστική δύναμη που κατείχαν τα υπολογιστικά συστήματα, καθώς και οι τεχνολογικές λύσεις της εποχής, ήταν σαφώς φτωχότερες. Έτσι, προκειμένου οι πάροχοι να προσφέρουν τις παραπάνω υπηρεσίες στους πελάτες τους με ασφάλεια και αξιοπιστία χρησιμοποιώντας κάποιο ενιαίο υπολογιστικό σύστημα, θα έπρεπε να κατέχουν τον ανάλογο σε μέγεθος και δύναμη εξοπλισμό. Κάτι τέτοιο θα αύξανε εξαιρετικά το επενδυτικό, καθώς και το λειτουργικό, κόστος, όπως την κατανάλωση ενέργειας και ψύξης, τον χώρο αποθήκευσης και την συντήρηση του υλικού. Όλα τα παραπάνω συντέλεσαν στην απόφαση των παρόχων να διαμοιράζουν το υπολογιστικό βάρος στον οικιακό δρομολογητή του κάθε πελάτη. Παράλληλα, όμως, το κόστος του παρεχόμενου εξοπλισμού αυξάνεται, λόγω των αυξανόμενων απαιτήσεων στις προδιαγραφές της συσκευής, και, σε συνδυασμό με το πλήθος των πελατών, το συνολικό αυτό κόστος αποκτά πλέον σημαντική διάσταση.

2.3.2 Μια εναλλακτική προσέγγιση

Η ταχεία ανάπτυξη των παρεχόμενων υπηρεσιών στους πελάτες τα τελευταία χρόνια, ωθεί τους τηλεπικοινωνιακούς παρόχους να αναθεωρήσουν το μοντέλο των συμβατικών CPE που περιγράφθηκε προηγουμένως. Συγκεκριμένα, αυτή η ανάπτυξη αυξάνει την πολυπλοκότητα των

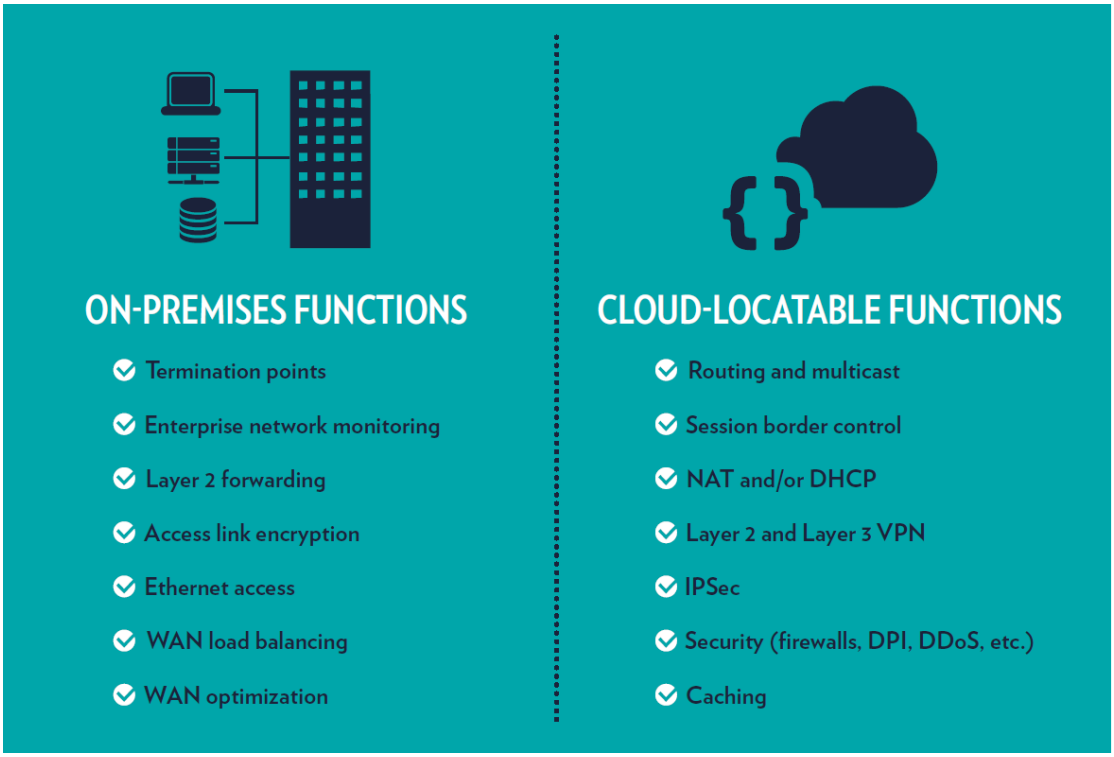
εργασιών που αναλαμβάνει ένας οικιακός δρομολογητής με αποτέλεσμα οι απατήσεις υλικού να είναι μεγαλύτερες και κατά συνέπεια να αυξάνουν το κόστος του. Αυτό αναγκάζει τους παρόχους να αναζητήσουν μοντέλα που να μην βασίζονται στις κεφαλαιακές δαπάνες (CapEx), αλλά στις λειτουργικές δαπάνες. Έτσι, σε συνδυασμό με την ανάπτυξη των εικονικών μηχανών αλλά και τον δικτύων NFV, οι πάροχοι προσανατολίζονται σε εικονικές τεχνολογίες που, λόγω όσων αναλύθηκαν στις προηγούμενες υποενότητες, δεν απαιτούν την αγορά πολλών και εξειδικευμένων συσκευών καθώς οι περισσότερες λειτουργίες γίνονται στο επίπεδο του λογισμικού εντός των εικονικών μηχανών [8].

Αυτό οδήγησε στην δημιουργία των εικονικών CPE (Virtual CPE – vCPE), που μπορούμε να πούμε πως είναι η πρώτη ευρέως διαδεδομένη χρήση της NFV τεχνολογίας. Ειδικότερα, πρόκειται για τα CPE επόμενης γενιάς που αντικαθιστούν τις υπηρεσίες που εκτελούνταν στον οικιακό δρομολογητής με VNFs που εκτελούνται σε μια εικονική μηχανή, τοποθετημένη στην μεριά του παρόχου ή σε κάποιο cloud. Με αυτόν τον τρόπο, οι πάροχοι έχουν τη δυνατότητα να μετατοπίσουν το υπολογιστικό βάρος στους κεντρικούς εξυπηρετητές, χρησιμοποιώντας τους για πολλούς πελάτες ταυτόχρονα, και να ελαχιστοποιήσουν το βάρος που διαμοιράζεται στο δίκτυο. Επίσης, με αυτόν τον τρόπο, η κίνηση διαμοιράζεται μεταξύ των εξυπηρετητών, ώστε να αποφεύγεται η δυσανάλογη κατανομή του φόρτου εργασίας. Στην συνέχεια του κεφαλαίου θα παρουσιαστούν αναλυτικά τα πλεονεκτήματα του μοντέλου αυτού [7].



Εικόνα 2-10: Οι λειτουργίες ενός virtual CPE

Ωστόσο, όπως γίνεται εύκολα αντιληπτό, το φυσικό CPE (physical CPE – pCPE) δεν παύει να υφίσταται στην συγκεκριμένη τεχνολογία, απλώς η πολυπλοκότητα του είναι εξαιρετικά χαμηλότερη [8]. Αυτό συμβαίνει γιατί υπάρχουν λειτουργίες που είναι απαραίτητο να εκτελούνται στην μεριά του πελάτη. Συγκεκριμένα, η παρουσία του pCPE είναι απαραίτητη ώστε να διαχειρίζεται κυρίως την σύνδεση των οικιακών υπολογιστών στο δίκτυο, είτε ενσύρματα είτε ασύρματα, αλλά και την δρομολόγηση των εξωτερικών ή εσωτερικών πακέτων εντός του οικιακού δικτύου. Επιπλέον, η παρουσία του βελτιώνει την απόδοση του οικιακού δικτύου μέσω λειτουργιών όπως το content caching, ενώ παράλληλα διευκολύνει την διερεύνηση και επίλυση τεχνικών προβλημάτων. Τέλος, ίσως ο κρισιμότερος λόγος, δεδομένης της έντονης στροφής κατά τα τελευταία χρόνια προς την ασφάλεια και την προστασία του απορρήτου, είναι η αναγκαιότητα για ορισμένες εφαρμογές και υπηρεσίες να κρυπτογραφηθούν πριν κυκλοφορήσουν στο διαδίκτυο [8].



Εικόνα 2-11: Ο διαμοιρασμός των εργασιών μεταξύ pCPE και vCPE

2.3.3 Τα πλεονεκτήματα των vCPE

Το μοντέλο των vCPE, ως μια τεχνολογία βασισμένη σε αυτή των NFV, προσφέρει και τα αντίστοιχα πλεονεκτήματα που αναλύθηκαν σε προηγούμενη υποενότητα. Παράλληλα όμως, προσφέρει επιπλέον πλεονεκτήματα, αφενός στους τηλεπικοινωνιακούς παρόχους, και αφετέρου στους πελάτες, είτε ιδιώτες είτε επαγγελματίες [8].

Καταρχήν, οι επενδυτικές και λειτουργικές δαπάνες μειώνονται σημαντικά. Ο κύριος λόγος που συντελεί σε αυτό είναι η μείωση των λειτουργικών απαιτήσεων των pCPE συσκευών,

που δίνει την δυνατότητα στους παρόχους να μειώσουν το κόστος κατασκευής τους. Επιπλέον, η χρήση εικονικών μηχανών για την λειτουργία των vCPE οδηγεί στην ενοποίηση του υλικού, που όπως αναλύθηκε στην προηγούμενη ενότητα, προσφέρει σημαντική μείωση στην κατανάλωση ισχύς, ενώ παράλληλα βοηθάει στον διαμοιρασμό του υπολογιστικού βάρους.

Επιπλέον, η ομοιομορφία του εξοπλισμού των παρόχων εξυπηρετεί την ευελιξία του, καθώς τα vCPE μπορούν να επεκτείνονται ανάλογα με τις ανάγκες τους σε πραγματικό χρόνο. Επίσης, μέσω της τεχνολογίας των vCPE οι πάροχοι ανεξαρτητοποιούνται από τους προμηθευτές, καθώς μέχρι πρότινος η αλλαγή προμηθευτή οδηγούσε σε μαζική αλλαγή του εξοπλισμού για λόγους συμβατότητας των λειτουργιών.

Επίσης, μεταξύ των πλεονεκτημάτων των NFV αναφέρθηκε πως η απεξάρτηση των παρόχων από την ανάγκη για αναβάθμιση του εξοπλισμού για την κυκλοφορία νέων υπηρεσιών, εκτός από επιπλέον εξοικονόμηση δαπανών, οδηγεί στην ταχύτερη κυκλοφορίας τους. Έμμεσο αποτέλεσμα αυτού, είναι η προσήλωση των προγραμματιστικών ομάδων στην ανάπτυξη καινοτόμων υπηρεσιών χωρίς να τους απασχολούν οι φυσικοί περιορισμοί που υπάρχουν με τις φυσικές CPE συσκευές.

Από την μεριά των χρηστών, η διεπαφή τους με το διαδίκτυο αναβαθμίζεται σημαντικά, μέσω εξειδικευμένων υπηρεσιών που απευθύνονται ειδικά στις προσωπικές ανάγκες του κάθε πελάτη, ενώ παράλληλα η τεχνική υποστήριξη είναι άμεση, καθώς η συντριπτική πλειοψηφία των διαγνωστικών ελέγχων αλλά και επιδιορθώσεων μπορεί να γίνει εξ αποστάσεως. Επιπλέον, όσον αφορά τις επιχειρήσεις, οι απαραίτητες αναβαθμίσεις στα θέματα ασφαλείας και λογισμικού γίνονται χωρίς την δική τους συμμετοχή, κάτι που κρίνεται ως σημαντικό για την εύρυθμη λειτουργία της.

Τέλος, όπως αναφέρθηκε και στην προηγούμενη υποενότητα η απόδοση, η ευελιξία και η κλιμακωσιμότητα του δικτύου ενισχύεται σημαντικά, τόσο λόγω της αποφόρτισης του pCPE, όσο και μέσω των διαφόρων VNF λειτουργιών που εκτελούνται στην πλευρά του παρόχου.

3

Αρχιτεκτονική δικτύου

Στο κεφάλαιο αυτό, θα γίνει θεωρητική παρουσίαση της αρχιτεκτονικής του δικτύου που σχεδιάστηκε για τις ανάγκες της παρούσας διπλωματικής εργασίας. Ειδικότερα, θα γίνει αναφορά στις επιμέρους μονάδες του δικτύου και θα επεξηγηθεί ο ρόλος τους. Στην συνέχεια του κεφαλαίου, θα παρουσιαστούν θεωρητικά τα εργαλεία που χρησιμοποιήθηκαν για την υλοποίηση αυτών των μονάδων, όπως το Raspberry Pi, το Proxmox VE, το Ansible και το πρωτόκολλο GRE Tunnel. Παράλληλα, θα επεξηγηθούν οι λόγοι που επιλέχθηκαν τα συγκεκριμένα εργαλεία έναντι άλλων, επιχειρώντας μια θεωρητική σύγκριση μεταξύ αυτών.

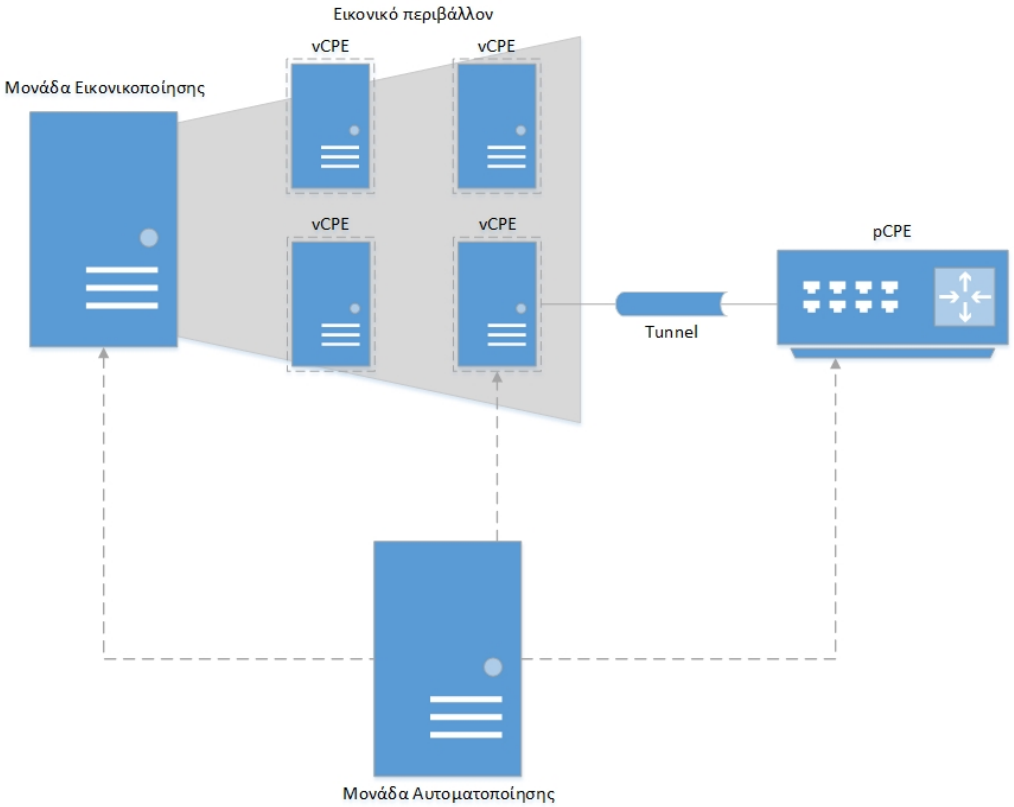
3.1 Θεωρητική περιγραφή δικτύου

Στην παρούσα διπλωματική, κεντρικός στόχος είναι η ανάπτυξη ενός οικιακού vCPE, το οποίο εκτελεί απλές λειτουργίες. Η συσκευή αυτή θα τοποθετείται σε ένα περιβάλλον εικονικοποίησης και θα πρέπει να είναι συνδεδεμένη με ένα pCPE. Για λόγους ευχρηστίας της αναπτυσσόμενης διάταξης, η ύπαρξη μιας μονάδας αυτοματοποίησης της δημιουργίας της κρίνεται απαραίτητη. Αναλυτικότερα, η αρχιτεκτονική του δικτύου περιέχει:

- **Περιβάλλον εικονικοποίησης**: Πρόκειται για ένα εικονικό μηχάνημα το οποίο υποστηρίζει εμφωλευμένη εικονικοποίηση (nested virtualization) μέσω κατάλληλου λογισμικού, ώστε να περιέχει τις εικονικές μηχανές που θα υλοποιούν τις λειτουργίες του vCPE. Με βάση τα όσα παρουσιάστηκαν για το πλαίσιο αρχιτεκτονικής των NFV

δικτύων, η παρούσα μονάδα αποτελεί την υποδομή εικονικοποίησης δικτυακών λειτουργιών NFVI.

- **vCPE**: Το συγκεκριμένο μηχάνημα, όπως ήδη αναφέρθηκε, είναι μια εικονική μηχανή, το οποίο τοποθετείται στο περιβάλλον εικονικοποίησης. Οι λειτουργίες του αφορούν την δρομολόγηση των πακέτων του οικιακού δικτύου προς το διαδίκτυο. Όσον αφορά το πλαίσιο αρχιτεκτονικής των NFV, η συγκεκριμένη εικονική μηχανή αποτελεί μέρος της υποδομής NFVI και εκτελεί τις εικονικές δικτυακές λειτουργίες (VNFs) της αρχιτεκτονικής.
- **pCPE**: Η φυσική αυτή συσκευή αποτελεί τον οικιακό εξοπλισμό του πελάτη. Όπως έχει αναλυθεί σε προηγούμενες ενότητες, είναι απαραίτητη για την διασύνδεση των οικιακών συσκευών στο υπόλοιπο δίκτυο, ενώ ταυτόχρονα αναλαμβάνει την δρομολόγηση των πακέτων που διακινούνται αποκλειστικά μεταξύ των υπολογιστών του οικιακού δικτύου που ανήκει το καθένα.
- **Μονάδα αυτοματοποίησης**: Η συγκεκριμένη μονάδα επιμελείται την ενορχήστρωση των VNFs. Πιο συγκεκριμένα, αποτελεί την μονάδα MANO του πλαισίου λειτουργίας των NFV και αναλαμβάνει την αυτοματοποίηση της παραμετροποίησης των επιμέρους μονάδων της αρχιτεκτονικής.
- **Tunnel**: Το συγκεκριμένο κομμάτι της αρχιτεκτονικής πρόκειται για ένα κανάλι που επιτρέπει την γρήγορη επικοινωνία μεταξύ της φυσικής και εικονικής CPE συσκευής.

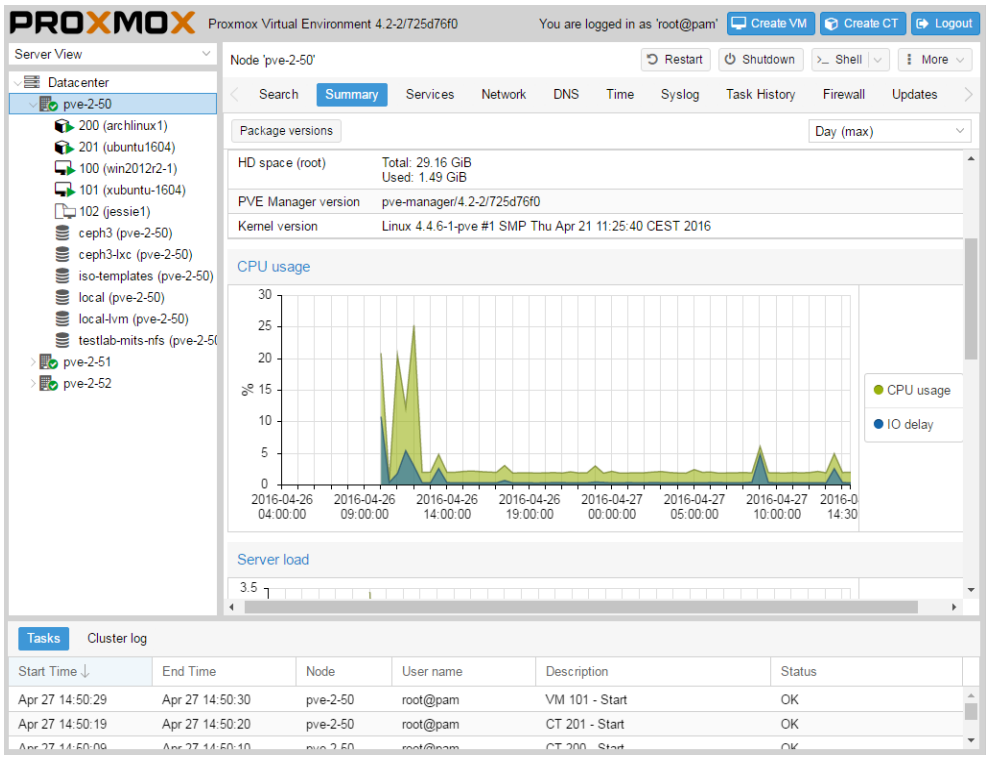


Εικόνα 3-1: Αρχιτεκτονική δικτύου

3.2 *Proxmox Virtual Environment*

Για την δημιουργία των εικονικών μηχανών επιλέχθηκε το Proxmox Virtual Environment, το οποίο είναι ένα περιβάλλον εικονικοποίησης ανοιχτού κώδικα που χρησιμοποιεί τον πυρήνα του Linux και είναι βασισμένο στην διανομή Debian [9] [10]. Το Proxmox VE χρησιμοποιεί την port 8006 για την πρόσβαση στο περιβάλλον διαχείρισης, ενώ παράλληλα διαθέτει REST API για την χρήση του από άλλες εφαρμογές. Επίσης, οι μορφές εικονικοποίησης που υποστηρίζει είναι τα containers (LXC από την έκδοση 4 και έπειτα), καθώς και την πλήρη εικονικοποίηση μέσω του KVM. Όσον αφορά την εγκατάσταση του, υπάρχει η δυνατότητα χειροκίνητης παραμετροποίησης του συστήματος, εγκαθιστώντας τα κατάλληλα πακέτα και δημιουργώντας τα κατάλληλα διαμερίσματα στον δίσκο, αλλά και απευθείας εγκατάσταση του μέσω αρχείου εικόνας .iso στον δίσκο.

Ένα από τα βασικότερα πλεονεκτήματα χρήσης του είναι το εύχρηστο περιβάλλον του. Η απλή και κατανοητή του σχεδίαση προσφέρει στον χρήστη την ευκολία να δημιουργήσει εικονικές μηχανές και containers σε μικρό χρόνο, ενώ παράλληλα μέσω της διαχειριστικής πλατφόρμας έχει την δυνατότητα να συνδεθεί στο τερματικό του κάθε εικονικού μηχανήματος. Χωρίς αυτό το περιβάλλον, η διαδικασία δημιουργίας ενός container, για παράδειγμα, δεν είναι τόσο απλή, καθώς απαιτούνται σχετικές γνώσεις για την δημιουργία ειδικών αρχείων παραμετροποίησης με σκοπό την δημιουργία του με τα επιθυμητά τεχνικά χαρακτηριστικά.



Εικόνα 3-2: Το περιβάλλον του Proxmox PVE

Επιπλέον το REST API που διαθέτει μας δίνει την δυνατότητα, με χρήση των κατάλληλων εργαλείων όπως θα αναλυθεί παρακάτω, να αυτοματοποιήσουμε διαδικασίες, όπως αυτή της δημιουργίας εικονικών μηχανών. Αυτό, όπως θα φανεί και στην συνέχεια, είναι ιδιαίτερα χρήσιμο στην παρούσα διπλωματική. Τέλος, το Proxmox διαθέτει μετρικά εργαλεία για την παρακολούθηση της χρήσης των εικονικών μηχανών, τα οποία παράγουν σχετικές γραφικές παραστάσεις που είναι προσβάσιμες είτε μέσω την διαχειριστικής πλατφόρμας είτε μέσω αντίστοιχων κλήσεων στο REST API.

Παράλληλα, υπάρχουν πολλά ακόμα ισχυρά περιβάλλοντα εικονικοποίησης που είναι ευρέως χρησιμοποιούμενα στην σύγχρονη εποχή. Ένα από τα πιο δημοφιλή είναι το VMM που έχει αναπτύξει η Microsoft, το Hyper-V. Ειδικότερα, ο φυσικός αυτός hypervisor διατίθεται ενσωματωμένος σε ορισμένες σύγχρονες 64-bit εκδόσεις της οικογένειας των Windows. Με την χρήση του είναι εφικτή η πλήρης εικονικοποίηση του υλικού και φιλοξενία εικονικών μηχανών με λειτουργικό σύστημα της ίδιας οικογένειας ή, ανάλογα με την έκδοση λειτουργικού του host μηχανήματος, ορισμένων Linux διανομών, όπως Ubuntu, Debian και CentOS. Ένας ακόμη hypervisor τύπου I είναι το VMWare ESXi που έχει αναπτυχθεί από την VMWare Inc. Υποστηρίζει μόνο την δυνατότητα πλήρους εικονικοποίησης, ενώ το φιλοξενούμενο μηχάνημα μπορεί να είναι σχεδόν οποιουδήποτε λειτουργικού συστήματος.

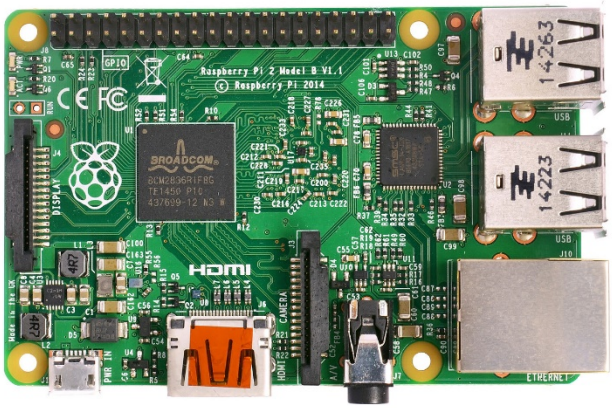
Επιπλέον, κατά τα τελευταία χρόνια, το περιβάλλον εικονικοποίησης Docker μεγαλώνει με ταχύ ρυθμό το μερίδιο του στην αγορά. Ένας σημαντικός λόγος αυτής της ανάπτυξης είναι το γεγονός πως παρέχει την δυνατότητα εικονικοποίησης στο επίπεδο του λογισμικού χρησιμοποιώντας Linux Containers. Επιπλέον διαθέτει ισχυρό REST API και δυνατότητα ενσωμάτωσης με πολλά εργαλεία αυτοματοποίησης και ενορχήστρωσης, μεταξύ των οποίων και τα Ansible, Chef και Puppet που θα αναλυθούν στην συνέχεια του κεφαλαίου. Ένα ακόμη περιβάλλον εικονικοποίησης που υποστηρίζει εικονικοποίηση στο επίπεδο του λειτουργικού συστήματος είναι το νέο, αλλά παράλληλα πολλά υποσχόμενο, Kubernetes που αναπτύσσεται από την Google. Η πρώτη έκδοση του έγινε στα μέσα του 2014 και, εκτός από το γεγονός ότι χρησιμοποιεί LXC για την δημιουργία εικονικών μηχανών, διαθέτει ένα πολύ ικανό REST API.

Συγκρίνοντας τα παραπάνω δημοφιλή περιβάλλοντα εικονικοποίησης, με σκοπό την κατάλληλη επιλογή μεταξύ αυτών για τους σκοπούς της παρούσας διπλωματικής, παρατηρούμε πως το Proxmox VE διαθέτει κάποια χαρακτηριστικά που το καθιστούν καταλληλότερο. Αρχικά, η συγκεκριμένη εφαρμογή απαιτεί εικονικοποίηση στο επίπεδο του λειτουργικού συστήματος και όχι πλήρους, λόγω των πλεονεκτημάτων που αναλύθηκαν παραπάνω σχετικά με την ταχύτητα δημιουργίας και τροποποίησης των containers. Ως αποτέλεσμα, η χρήση των δυο φυσικών hypervisors που παρουσιάστηκαν, Hyper-V και VMWare ESXi, δεν είναι δυνατή λόγω της αδυναμίας τους για εικονικοποίηση αυτής της μορφής. Επιπλέον, σε σχέση με τα υπόλοιπα εργαλεία που υποστηρίζουν αυτό το είδος εικονικοποίησης, δηλαδή το Docker και το Kubernetes,

η επιλογή του Proxmox VE κρίνεται καταλληλότερη. Αφενός, το Docker διαθέτει διάφορους περιορισμούς για τα LXC που δημιουργούνται, όπως για παράδειγμα ότι λαμβάνουν υποχρεωτικά ιδιωτική IP και επικοινωνούν με το διαδίκτυο μέσω του Docker Server και χρήση NAT. Αυτός ο περιορισμός καθιστά αδύνατη την απευθείας επικοινωνία του φυσικού και του εικονικό CPE, καθώς και την άμεση επικοινωνία του container με το διαδίκτυο, και δεν είναι αποδεκτός για την παρούσα διάταξη. Αφετέρου, το Kubernetes, λόγω της πρόσφατης κυκλοφορίας του, κρίνεται ως ανώριμο, σε σχέση με το Proxmox VE, και ως μη κατάλληλο για παραγωγικά συστήματα μεγάλης κλίμακας. Ωστόσο, τα δείγματα που έχει επιδείξει προς το παρόν είναι ιδιαίτερα ενθαρρυντικά θα είχε ενδιαφέρον η μελλοντική επέκταση του παρόντος θέματος με χρήση του συγκεκριμένου περιβάλλοντος εικονικοποίησης.

3.3 Raspberry Pi

Για την αναπαράσταση των φυσικών CPE συσκευών στο δίκτυο, επιλέχθηκε η χρήση των Raspberry Pi [11]. Η συσκευή αυτή είναι ένας μικρός single-board υπολογιστής, ο οποίος αναπτύχθηκε από την βρετανική εταιρία Raspberry Pi Foundation στις αρχές του 2012. Αρχικός σκοπός της εταιρίας ήταν η προώθηση της διδασκαλίας της επιστήμης των υπολογιστών στα σχολεία και σε αναπτυσσόμενες χώρες. Όμως, η ισχυρή υπολογιστική δύναμη που διαθέτει η συσκευή σε συνδυασμό με το εξαιρετικά μικρό της μέγεθος και την εξαιρετικά χαμηλή τιμή της, την κατέστησε σύντομα πολύ δημοφιλή σε πολλών ειδών εφαρμογές σε σχεδόν το σύνολο της πληροφορικής, όπως στον τομέα των τηλεπικοινωνιών, της ρομποτικής και των νευρωνικών δικτύων.



Εικόνα 3-3: Raspberry Pi 2 Model B

Τα βασικά χαρακτηριστικά που περιλαμβάνονται συνήθως στην πλακέτα του Raspberry Pi, η οποία έχει μέγεθος πιστωτικής κάρτας, είναι ο επεξεργαστής τεχνολογίας ARM, η κάρτα γραφικών, η μνήμη RAM, μια θύρα εξόδου εικόνας HDMI και θύρες εισόδου-εξόδου USB (το πλήθος τους ποικίλει ανάλογα με το μοντέλο και την έκδοση του). Επίσης, προσφέρει στον χρήστη την δυνατότητα ανάπτυξης εφαρμογών οι οποίες μπορούν να διασυνδεθούν με custom

τρόπο με εξωτερικές συσκευές, χάρη στους connector pins γενικού σκοπού (General purpose input-output – GPIO) που διαθέτει. Επιπλέον, για την προσθήκη αποθηκευτικού χώρου διαθέτει υποδοχή για εξωτερική κάρτα μνήμης. Τέλος, τα πιο σύγχρονα μοντέλα διαθέτουν ενσωματωμένη κάρτα δικτύου με δυνατότητα ενσύρματης ή ασύρματης διασύνδεσης. Στον παρακάτω πίνακα αναφέρονται αναλυτικά τα τεχνικά χαρακτηριστικά του Raspberry Pi 2 Model B, το οποίο είναι το μοντέλο που χρησιμοποιήθηκε κατά την παρούσα διπλωματική εργασία.

Raspberry Pi 2 Mode B full specifications	
Architecture	ARMv7-A (32-bit)
SoC	Broadcom BCM2836
CPU	900 MHz 32-bit quad-core ARM Cortex-A7
GPU	Broadcom VideoCore IV @ 250 MHz (BCM2837: 3D part of GPU @ 300 MHz, video part of GPU @ 400 MHz) OpenGL ES 2.0 (BCM2835, BCM2836: 24 GFLOPS / BCM2837: 28.8 GFLOPS) MPEG-2 and VC-1 (with license), 1080p30 H.264/MPEG-4 AVC high-profile decoder and encoder (BCM2837: 1080p60)
Memory (SDRAM)	1 GB (shared with GPU)
USB 2.0 ports	4 (via the on-board 5-port USB hub)
Video input	15-pin MIPI camera interface (CSI) connector, used with the Raspberry Pi camera or Raspberry Pi NoIR camera
Video outputs	HDMI (rev 1.3), composite video (3.5 mm TRRS jack), MIPI display interface (DSI) for raw LCD panels
Audio inputs	As of revision 2 boards via I^2S
Audio outputs	Analog via 3.5 mm phone jack; digital via HDMI and, as of revision 2 boards, I^2S
On-board storage	MicroSDHC slot
On-board network	10/100 Mbit/s Ethernet (8P8C) USB adapter on the USB hub
Low-level peripherals	17× GPIO plus the same specific functions, and HAT ID bus
Power ratings	220 mA (1.1 W) average when idle, 820 mA (4.1 W) maximum under stress (monitor, keyboard and mouse connected)
Power source	5 V via MicroUSB or GPIO header
Size	85.60 mm × 56.5 mm (3.370 in × 2.224 in), not including protruding connectors
Weight	45 g
Console	Adding a USB network interface via tethering[71] or a serial cable with optional GPIO power connector

Πίνακας 2: Τα πλήρη χαρακτηριστικά του Raspberry Pi 2 Model B

Όσον αφορά το λειτουργικό σύστημα, το επίσημο λειτουργικό για το Raspberry Pi ονομάζεται Raspbian, το οποίο είναι μια ειδική διανομή Linux βασισμένη στο Debian και βρίσκεται υπό συνεχή ανάπτυξη από την Raspberry Pi Foundation. Ωστόσο, δύναται να λειτουργήσει με οποιοδήποτε λειτουργικό σύστημα είναι συμβατό με επεξεργαστή ARM. Σε αυτή την λίστα βρίσκονται η πλειοψηφία των συστημάτων που διαθέτουν τον πυρήνα των Linux,

καθώς και άλλα που είναι προσαρμοσμένα σε αυτή την αρχιτεκτονική, όπως το FreeBSD και NetBSD.

Οι λόγοι για τους οποίους επιλέχθηκε η συγκεκριμένη συσκευή για την υλοποίηση των φυσικών CPE είναι προφανείς. Πρόκειται για έναν οικονομικό και εξαιρετικά ισχυρό φυσικό υπολογιστή, ο οποίος, με χρήση των δικτυακών λειτουργιών που είναι διαθέσιμες στο λειτουργικό του, μπορεί να προσομοιάσει πλήρως τις λειτουργίες του οικιακού δρομολογητή.

3.4 Ansible

Εικόνα 3-4: Ansible automation and orchestration tool

Το Ansible είναι μια μηχανή αυτοματοποίησης που ανέπτυξε η Red Hat Inc. Χρησιμοποιείται για την αυτόματη παραμετροποίηση συστημάτων, καθώς δίνει την δυνατότητα στον χρήστη να συνδέεται μέσω SSH και να εκτελεί εντολές σε έναν ή περισσότερους απομακρυσμένους υπολογιστές ταυτόχρονα. Επίσης έχει την δυνατότητα να συγγράφει και να εκτελεί ειδικά αρχεία τύπου YAML, τα οποία ονομάζονται playbooks και περιέχουν συγκεκριμένες εργασίες προς εκτέλεση οι οποίες εκτελούνται σειριακά στους ορισμένους υπολογιστές. Στα σημαντικότερα πλεονεκτήματα χρήσης του Ansible, πέρα από το προφανές της εξοικονόμησης χρόνου από την ταυτόχρονη εκτέλεση προκαθορισμένων εντολών σε πολλούς υπολογιστές ταυτόχρονα, είναι η ομαδοποίηση των απομακρυσμένων υπολογιστών. Ειδικότερα, μέσω αρχείων που ονομάζονται inventories, μπορούν να οριστούν πολυεπίπεδες ομάδες από υπολογιστές με σκοπό τον διαχωρισμό των εντολών που θα εκτελεστούν στο καθένα από τα μηχανήματα [12].

Ωστόσο, υπάρχουν πολλά εργαλεία ενορχήστρωσης, εξίσου ισχυρά και δημοφιλή. Ένα από αυτά είναι το Puppet [13]. Το συγκεκριμένο εργαλείο έχει αναπτυχθεί με βάση την γλώσσα Ruby, στην οποία γλώσσα γράφονται τα automation scripts που χρησιμοποιεί σε συνδυασμό με μια άλλη custom γλώσσα, την Puppet DSL (Domain Specific Language). Πρόκειται για ένα από τα πιο ολοκληρωμένα εργαλεία αυτού του είδους καθώς, εκτός από την πληθώρα δυνατοτήτων που παρέχει, διαθέτει επιπλέον ένα αρκετά πλήρες Web UI για την εκτέλεση των scripts και την εποπτεία των λειτουργιών του. Τέλος, χρησιμοποιεί το μοντέλο του agent – master, δηλαδή ενός κόμβου (master) ο οποίος κατέχει τις πληροφορίες παραμετροποίησης και πολλών κόμβων

(agents) οι οποίοι λαμβάνουν τις πληροφορίες αυτές και παραμετροποιούν κατάλληλα το σύστημα στο οποίο εκτελούνται.

Επίσης, ένα ακόμη δημοφιλές εργαλείο αυτοματοποίησης είναι το Chef, το οποίο παρουσιάζει πολλές ομοιότητες με το Puppet. Το εργαλείο αυτό χρησιμοποιεί επίσης την γλώσσα Ruby για την σύνταξη των recipes, δηλαδή των script αυτοματοποίησης που δημιουργεί ο χρήστης. Επιπλέον, ακολουθεί και αυτό το μοντέλο agent – master για την οργάνωση και παραμετροποίηση των κόμβων [13]. Παρόλα αυτά, το Web UI δεν είναι το ίδιο ισχυρό, ενώ το documentation και το community support για το συγκεκριμένο εργαλείο, κατά κοινή ομολογία, δεν είναι ιδιαίτερα πλούσιο [14].

Το σημαντικότερο πλεονέκτημα του Ansible, έναντι των άλλων εργαλείων ενορχήστρωσης που παρουσιάστηκαν, το οποίο είναι και η αιτία της τελικής επιλογής, είναι η απουσία του μοντέλου agent – master στην λειτουργία του [13]. Ειδικότερα, οι κόμβοι στους οποίους εκτελούνται τα playbooks, δεν απαιτείται να έχουν κάποιο ειδικό λογισμικό εγκατεστημένο, εκτός από την γλώσσα Python για κάποιες από τις λειτουργίες του, η οποία μπορεί να εγκατασταθεί εύκολα με την χρήση του ίδιου του εργαλείου. Το γεγονός αυτό είναι προϋπόθεση για την παρούσα διπλωματική, καθώς η παραμετροποίηση που υφίσταται ένας κόμβος γίνεται αμέσως μετά την δημιουργία του και η προεγκατάσταση κάποιου λογισμικού δεν είναι εφικτή.

Όπως ήδη αναφέρθηκε στην πρώτη υποενότητα του συγκεκριμένου κεφαλαίου, το Ansible αποτελεί την μονάδα MANO του αρχιτεκτονικής που σχεδιάστηκε για το παρόν θέμα. Συγκεκριμένα, το Ansible, μέσω κατάλληλων playbook και των κατάλληλων κλήσεων το REST API του Proxmox VE, λειτουργεί ως VIM με στόχο την πλήρη ορατότητα της διαθεσιμότητα των πόρων. Επίσης, αναπτύσσοντας τα απαραίτητα playbooks που εκτελούν τις κατάλληλες κλήσεις, το Ansible λειτουργεί ως VNFM και έχει την δυνατότητα δημιουργίας και τροποποίησης των εικονικών μηχανών. Για τις ανάγκες της διπλωματικής, αναπτύχθηκαν τα playbooks που σχετίζονται με την δημιουργία των vCPE και την παραμετροποίηση αυτών, καθώς και των φυσικών CPE.

3.5 GRE Tunnel

Για την επικοινωνία μεταξύ του φυσικού και εικονικού CPE αποφασίστηκε η εφαρμογή ενός πρωτοκόλλου tunneling. Το πρωτόκολλο αυτό δημιουργεί έναν δίαυλο μεταξύ δύο κόμβων που συνήθως ανήκουν σε απομακρυσμένα δίκτυα και επιτρέπει την απευθείας μεταφορά πακέτων. Συγκεκριμένα, αυτά τα πρωτόκολλα δημιουργούν ένα νέο στρώμα ενθυλάκωσης στο μεταφερόμενο πακέτο. Το στρώμα αναγνωρίζεται από τους επόμενους κόμβους μέσω κατάλληλων επικεφαλίδων, με αποτέλεσμα το πακέτο να δρομολογείται χωρίς την περαιτέρω διάσπαση της ενθυλάκωσης του. Η διαδικασία αυτή, ανάλογα με τον τύπου του πρωτοκόλλου

που χρησιμοποιείται, εξυπηρετεί στην ταχύτητα ή ακόμη και στην ασφάλεια μεταφοράς του πακέτου. Τα δημοφιλέστερα πρωτόκολλα αυτού τύπου είναι το GRE Tunnel και το IPSec.

Το GRE Tunnel (Generic Routing Encapsulation) είναι ένα πρωτόκολλο tunneling που αναπτύχθηκε από την Cisco Systems και ορίστηκε στο πρότυπο RFC 2784. Σκοπός του συγκεκριμένου πρωτοκόλλου είναι η γρήγορη μεταφορά διαφόρων ειδών πακέτων μεταξύ δύο IP δικτύων. Για να επιτύχει αυτό, ο αρχικός κόμβος του GRE Tunnel προσθέτει στο πακέτο ένα εξωτερικό IP header με επιπλέον 4 Bytes, που αποσκοπούν στην αναγνώριση του GRE στρώματος, το οποίο έχει ως παραλήπτη τον τελικό κόμβο του καναλιού, και το προωθεί. Ο επόμενος κόμβος, όπως και κάθε ενδιάμεσος κόμβος μέχρι τον κόμβο – παραλήπτη του GRE, μόλις αναγνωρίζει το GRE header, προωθεί κατάλληλα το πακέτο, με βάση τους κανόνες δρομολόγησης που ακολουθεί, χωρίς περαιτέρω προσπέλαση ή επεξεργασία. Μόλις το πακέτο παραληφθεί από το τον τελικό κόμβο του καναλιού, τότε αυτός αφαιρεί το εξωτερικό IP header και σε περίπτωση που δεν είναι αυτό το τελικός παραλήπτης του εσωτερικού πακέτου, συνεχίζει την προώθηση του. Βασικό πλεονέκτημα αυτής της διαδικασίας είναι η ταχύτητα μεταφοράς του πακέτου πάνω στο κανάλι, λόγω της αμεσότητας που προωθείται το πακέτο από τους ενδιάμεσους κόμβους.

Το IPSec (Internet Protocol Security) είναι ένα πρωτόκολλο tunneling, το οποίο ορίστηκε στο πρότυπο RFC 2406 και αποσκοπεί στην ασφαλή μεταφορά πακέτων IP μεταξύ δύο κόμβων [15]. Συγκεκριμένα, διασφαλίζει την ιδιωτικότητα, την ακεραιότητα και την αυθεντικότητα των πακέτων που μεταφέρονται μεταξύ αυτού του καναλιού. Για αυτό τον σκοπό, πριν την χρήση του για μεταφορά πακέτων, οι δύο κόμβοι που βρίσκονται στα άκρα του ακολουθούν τυποποιημένες διαδικασίες με σκοπό την ανταλλαγή κλειδιών κρυπτογράφησης. Έπειτα, όταν χρειάζεται να μεταφερθεί ένα πακέτο μέσω αυτού, οι κόμβοι χρησιμοποιούν τα κλειδιά αυτά για την κρυπτογράφηση των πακέτων και προσθέτουν κατάλληλα headers που διασφαλίζουν την ακεραιότητα και την αυθεντικότητα του περιεχόμενου. Με αυτό τον τρόπο, οι ενδιάμεσοι κόμβοι δεν μπορούν να διαβάσουν ή να τροποποιήσουν το περιεχόμενο του πακέτου, εκτός από την εξωτερική επικεφαλίδα, την οποία χρησιμοποιούν για την κατάλληλη προώθηση του πακέτου. Μόλις ο κόμβος – παραλήπτης του καναλιού λάβει το πακέτο, αφαιρεί τα πρόσθετα headers, αποκρυπτογραφεί το περιεχόμενο του και, σε περίπτωση που δεν είναι αυτός ο τελικός παραλήπτης, το προωθεί κατάλληλα.

Όσον αφορά την αρχιτεκτονική της παρούσας διπλωματικής εργασίας, το πρωτόκολλο tunneling που αποφασίστηκε να χρησιμοποιηθεί είναι το GRE Tunnel. Η επιλογή αυτή βασίστηκε στις ανάγκες της εν λόγω αρχιτεκτονικής. Συγκεκριμένα, στην προκειμένη περίπτωση η μεταφορά του πακέτου μεταξύ του φυσικού και του εικονικού CPE οφείλει να γίνεται γρήγορα, με τις μικρότερες δυνατές καθυστερήσεις. Αυτό με την χρήση του IPSec δεν είναι εφικτό, καθώς τόσο η εγκατάσταση του όσο και η μεταφορά πακέτων είναι πολύπλοκες και χρονοβόρες

διαδικασίες, λόγω της ανταλλαγής των κρυπτογραφικών κλειδιών και της διαδικασίας κρυπτογράφησης – αποκρυπτογράφησης των μεταφερόμενων πακέτων. Αντίθετα, η εγκατάσταση του GRE Tunnel γίνεται απλά και γρήγορα, καθώς το μόνο που απαιτείται η εκτέλεση των κατάλληλων εντολών στον κάθε κόμβο χωρίς κάποια επιπλέον έγκριση μεταξύ αυτών. Επίσης, εφόσον το δίκτυο στο οποίο λειτουργεί το GRE tunnel είναι στην κατοχή του παρόχου, θεωρείται πως είναι αξιόπιστο και η ασφάλεια των μεταφερόμενων πακέτων είναι διασφαλισμένη. Τέλος, όπως αναλύθηκε και παραπάνω η μεταφορά μέσω του συγκεκριμένου καναλιού είναι από την φύση της ταχύτερη.

4

Ανάπτυξη οικιακού vCPE

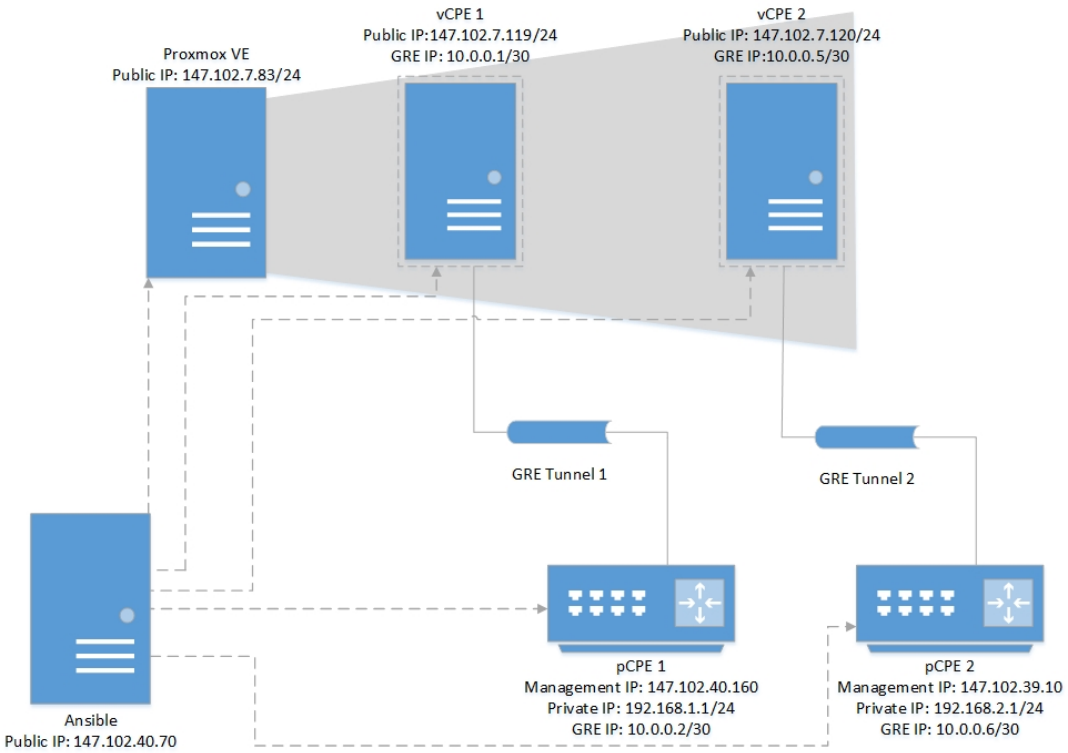
Στο παρόν κεφάλαιο, θα γίνει ανάλυση του πειραματικού μέρους της παρούσας διπλωματικής εργασίας. Ειδικότερα, θα παρουσιαστεί η διάταξη η οποία αναπτύχθηκε, αναλύοντας τις λειτουργίες της και τα τεχνικά χαρακτηριστικά κάθε επιμέρους μονάδας. Έπειτα, θα αναλυθεί ο τρόπος με τον οποίο δημιουργήθηκε η συγκεκριμένη διάταξη με χρήση των τεχνολογιών και των εργαλείων που αναλύθηκαν σε προηγούμενα κεφάλαια. Τέλος, θα παρουσιαστεί η διαδικασία αυτοματοποίησης που αναπτύχθηκε για την γρήγορη αναπαραγωγή της.

4.1 Περιγραφή διάταξης

Για την υλοποίηση σε εργαστηριακό περιβάλλον και πειραματική επιβεβαίωση του στόχου της διπλωματικής, αποφασίστηκε η δημιουργία δύο vCPE συσκευών τοποθετημένων σε κοινό εικονικό περιβάλλον, οι οποίες πρόκειται να συνδεθούν με τους αντίστοιχους οικιακούς δρομολογητές. Ο λόγος για τη χρήση δύο οικιακών vCPE είναι πως με την διάταξη αυτή, αφενός μπορούν να εφαρμοστούν κατάλληλοι έλεγχοι για την σωστή δρομολόγηση των πακέτων των οικιακών δικτύων προς το διαδίκτυο (WAN) και αφετέρου μπορεί επιβεβαιωθεί η απομόνωση των δικτύων, είτε μεταξύ τους είτε με το διαδίκτυο. Προφανώς, η διάταξη αυτή μπορεί να γενικευθεί και σε κάθε εικονικό περιβάλλον να δημιουργηθούν περισσότερα από δύο ζεύγη φυσικών – εικονικών CPE συσκευών. Για την δημιουργία της, παραχωρήθηκε πρόσβαση σε ένα μια εικονική μηχανή με δημόσια IP 147.102.7.83/24, όπου και εγκαταστάθηκε το λειτουργικό του

Proxmox VE, που αναφέρθηκε εκτενώς στην προηγούμενη ενότητα, σκοπός της οποίας είναι να φιλοξενεί τις vCPE συσκευές.

Τον σημαντικότερο ρόλο στην διάταξη έχουν οι εικονικές CPE συσκευές. Κάθε τέτοια συσκευή αναλαμβάνει την διασύνδεση του οικιακού δρομολογητή με το διαδίκτυο. Για να το πετύχει αυτό χρησιμοποιεί την λειτουργία της μετάφρασης διευθύνσεων (Network Address Translation – NAT). Ειδικότερα, το κάθε vCPE θα αντικαθιστά την ιδιωτική διεύθυνση IP του αποστολέα του πακέτου με την δική του δημόσια ώστε το πακέτο να σταλεί στον παραλήπτη, προσθέτοντας την κατάλληλη εγγραφή σε έναν πίνακα που ονομάζεται πίνακας μετάφρασης (Translation Table). Μόλις λάβει την απάντηση για το πακέτο αυτό, τότε αντικαθιστά την δική του δημόσια IP, η οποία τώρα βρίσκεται στην θέση του παραλήπτη, με την αρχική ιδιωτική IP και προωθεί το πακέτο στο pCPE. Οι δύο συσκευές vCPE που δημιουργήθηκαν και χρησιμοποιήθηκαν στην διάταξη διαθέτουν τις δημόσιες IP 147.102.7.119/24 και 147.102.7.120/24 αντίστοιχα.



Εικόνα 4-1: Πειραματική διάταξη

Επιπλέον, η διάταξη περιλαμβάνει και την φυσική CPE συσκευή, η οποία αποτελεί τον οικιακό εξοπλισμό του πελάτη και είναι απαραίτητη για την διασύνδεση των οικιακών συσκευών στο υπόλοιπο δίκτυο, ενώ ταυτόχρονα αναλαμβάνει την δρομολόγηση των πακέτων που διακινούνται αποκλειστικά μεταξύ των υπολογιστών του οικιακού δικτύου. Η συσκευή αυτή, εκτός από την εικονική διεπαφή που αντιστοιχεί στο GRE Tunnel και θα αναλυθεί στην συνέχεια, διαθέτει δύο δικτυακές διεπαφές. Η πρώτη διαθέτει ιδιωτική διεύθυνση IP και ανήκει στο οικιακό

δίκτυο, ενώ η δεύτερη λαμβάνει διεύθυνση IP από το εσωτερικό δίκτυο του παρόχου (στην συγκεκριμένη περίπτωση από το δίκτυο του εργαστηρίου) και αποσκοπεί στην εγκατάσταση του GRE Tunnel και στην απομακρυσμένη διαχείριση του κόμβου. Έτσι λοιπόν, στην παραπάνω διάταξη, οι συσκευές **pCPE 1** και **pCPE 2** διαθέτουν αντιστοίχως τις διευθύνσεις IP 147.102.40.106/24 και 147.102.39.10/24 στο εσωτερικό δίκτυο, ενώ στο οικιακό δίκτυο κατέχουν την ιδιωτική IP 192.168.1.1/24 και 192.168.2.1/24 αντίστοιχα.

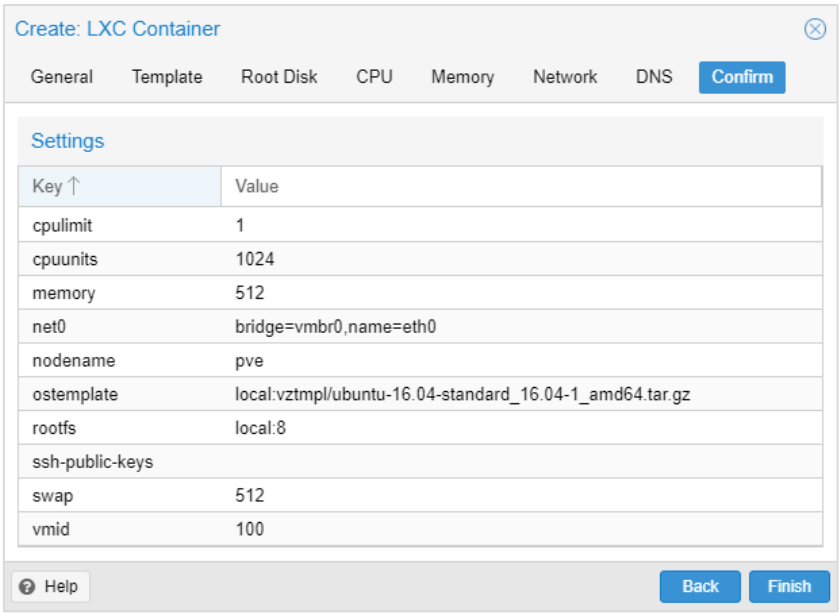
Τέλος, για την ταχύτερη επικοινωνία μεταξύ των φυσικών και εικονικών CPE, αποφασίστηκε η κατασκευή ενός GRE καναλιού, μέσω του οποίου οι συγκεκριμένες συσκευές επικοινωνούν. Με αυτή την εικονική point-to-point διασύνδεση των δύο κόμβων, τα πακέτα που διακινούνται μέσω αυτής δρομολογούνται απευθείας μεταξύ των ενδιάμεσων κόμβων χωρίς την πλήρη διάσπαση της ενθυλάκωσης του πακέτου. Αυτό προσφέρει ταχύτητα στην μεταφορά του, η οποία φαίνεται ως ένα hop. Στην διάταξη, τα vCPE 1 και pCPE 1 διαθέτουν τις ιδιωτικές διευθύνσεις IP 10.0.0.1/30 και 10.0.0.2/30 αντίστοιχα για την μεταξύ τους επικοινωνία, ενώ το ζεύγος vCPE 2 – pCPE 2 διαθέτει τις διευθύνσεις IP 10.0.0.5/30 και 10.0.0.6/30 αντίστοιχα.

4.2 Διαδικασία αναπαραγωγής διάταξης

4.2.1 Δημιουργία των Linux Containers

Το Proxmox VE διαθέτει πλήθος επιλογών για την δημιουργία των Linux Containers. Αρχικά, ο χρήστης μπορεί να δημιουργήσει το LXC μέσω των κατάλληλων εντολών στο κέλυφος του μηχανήματος, οι οποίες παρέχονται προεγκατεστημένες στο λειτουργικό του. Αυτός ο τρόπος, μπορεί να κριθεί ως ο πιο δυσλειτουργικός από άποψη ευκολίας και ταχύτητας, καθώς προϋποθέτει την σύνδεση του χρήστη στο κέλυφος του μηχανήματος μέσω SSH όπως και την γνώση της σύνταξης των κατάλληλων εντολών που απαιτούνται.

Ένας διαφορετικός τρόπος είναι μέσω της σύνδεσης του χρήστη στο γραφικό περιβάλλον του Proxmox VE, το οποίο είναι προσβάσιμο στην port 8006, μέσω οποιουδήποτε περιηγητή. Στο περιβάλλον αυτό, ο χρήστης έχει την δυνατότητα, ανάμεσα σε πλήθος από διαχειριστικά εργαλεία, να χρησιμοποιήσει τον οδηγό δημιουργίας LXC. Έτσι, με εύκολα βήματα μπορεί να επιλέξει τις παραμέτρους της νέας συσκευής, όπως το λειτουργικό, το πλήθος και την ταχύτητα των πυρήνων, το μέγεθος της μνήμης RAM και του αποθηκευτικού χώρου, το πλήθος και την παραμετροποίηση των δικτυακών διεπαφών και πολλά άλλα. Έπειτα, μέσω του συγκεκριμένου περιβάλλοντος έχει την δυνατότητα να συνδεθεί και να εκτελέσει εντολές στο κέλυφος του εικονικού μηχανήματος, να τροποποιήσει τα χαρακτηριστικά του, ενώ μπορεί ακόμα και να παρακολουθήσει σε γραφήματα την χρήση των πόρων του. Πρόκειται για έναν εξαιρετικά γρήγορο και εύχρηστο τρόπο, ο οποίος δεν απαιτεί από τον χρήστη ιδιαίτερες τεχνικές γνώσεις για την δημιουργία και διαχείριση των εικονικών συσκευών.



Εικόνα 4-2: Οδηγός δημιουργίας LXC στο Proxmox VE

Ο τρόπος όμως που συγκεντρώνει το μεγαλύτερο ενδιαφέρον, καθώς μας δίνει την δυνατότητα αυτοματοποίησης της διαδικασίας, είναι μέσω του σχετικού REST API που διαθέτει το Proxmox VE. Αποστέλλοντας το κατάλληλο αίτημα, μεταξύ άλλων λειτουργιών, μπορούμε να δημιουργήσουμε ένα LXC με τις επιθυμητές παραμέτρους, όπως ακριβώς και στο γραφικό περιβάλλον του Proxmox VE. Το API χρησιμοποιεί ως βασική διεύθυνση την *https://your.server:8006/api2/json/*, ενώ οι παράμετροι δηλώνονται είτε στο URL του αιτήματος, είτε μέσω PUT/POST αιτημάτων τοποθετημένα στο σώμα του αιτήματος με χρήση του 'x-www-form-urlencoded' στο content-type header. Η μορφή των απαντήσεων που λαμβάνει ο χρήστης είναι παραμετροποιήσιμη και μπορεί να είναι json, extjs, html ή απλό κείμενο.

Για την διαδικασία αυτή απαιτείται καταρχήν η ταυτοποίηση του χρήστη για λόγους προστασίας [9]. Για αυτό τον λόγο ο χρήστης πρέπει πριν την αποστολή του αιτήματος στο API, να εκτελέσει την εντολή:

```
$ curl -k -d "username=root@pam&password=yourpassword" \
https://147.102.7.83:8006/api2/json/access/ticket
{
  "data":
    {
      "CSRFPreventionToken":"4EEC61E2:lwk7od06fa1+DcPUwBTXCcndyAY",
      "ticket":"PVE:root@pam:4EEC61E2::rsKoApxDTLYPn6H3NNT6iP2mv...",
      "username":"root@pam"
    }
}
```

Όπως φαίνεται στην απάντηση της παραπάνω εντολής, ο χρήστης λαμβάνει ως απάντηση ένα json που περιέχει τα *CSRFPreventionToken* και *ticket*, που είναι απαραίτητα για την

πιστοποίηση του χρήστη. Αυτά έχουν διάρκεια 2 ωρών και μετά το πέρας αυτών ο χρήστης πρέπει να επαναλάβει την διαδικασία ή, πιο απλά, να χρησιμοποιήσει την μέθοδο */access/ticket* του API. Αξίζει να σημειωθεί, πως η παραπάνω απάντηση είναι αρκετά απλουστευμένη, καθώς στην πραγματικότητα περιλαμβάνει πολλές ακόμα παραμέτρους που αφορούν τα δικαιώματα του χρήστη και δεν προβάλλονται στην προκειμένη περίπτωση για λόγους οικονομίας χώρου και απλότητας. Έχοντας στην διάθεση του, λοιπόν, τα παραπάνω, ο χρήστης μπορεί να αποστέλλει αιτήματα στο API του Proxmox VE, τοποθετώντας την παράμετρο *ticket* σε ένα authorization cookie, μαζί με τις υπόλοιπες παραμέτρους (ανάλογα τον τύπο του αιτήματος):

```
$ curl -k -b \ "PVEAuthCookie=PVE:root@pam:4EEC61E2::rsKoApxDTLYPn6H3NNT6iP2mv..."
\
 https://147.102.7.83:8006/api2/json/
```

Επιπλέον, για οποιοδήποτε αίτημα εγγραφής (POST, PUT, DELETE) πρέπει να περιλαμβάνεται η επικεφαλίδα *CSRFPreventionToken*:

```
$ curl -XDELETE -H "CSRFPreventionToken: 4EEC61E2:lwk7od06fa1+DcPUwBTXCcndyAY" ...
```

Για να μπορέσει λοιπόν ο χρήστης να δημιουργήσει ένα LXC μέσω του REST API του Proxmox VE, θα πρέπει να αποστείλει ένα POST αίτημα στην μέθοδο */nodes/{node}/lxc*, όπου η μεταβλητή *node* είναι το όνομα του κόμβου που θα περιέχεται το LXC. Οι παράμετροι της συγκεκριμένης μεθόδου είναι πολλές και ίσως άσκοπο να αναλυθούν όλες, ωστόσο οι σημαντικότερες από αυτές για την παρούσα διπλωματική είναι οι εξής [16]:

Required Parameters			
Name	**Type**	**Format**	**Description**
node	string	<string>	The cluster node name.
ostemplate	string	<string>	The OS template or backup file.
vmid	integer	<integer> (100 - N)	The (unique) ID of the VM.

Πίνακας 3: Υποχρεωτικές μεταβλητές κλήσεως /node/{node}/lxc του Proxmox VE REST API

Optional Parameters			
Name	**Type**	**Format**	**Description**
cpulimit	number	<number> (0 - 128)	Limit of CPU usage. NOTE: If the computer has 2 CPUs, it has total of '2' CPU time. Value '0' indicates no CPU limit.
cpuunits	integer	<integer> (0 - 500000)	CPU weight for a VM. Argument is used in the kernel fair scheduler. The larger the number is, the more CPU time this VM gets. Number is relative to weights of all the other running VMs. NOTE: You can disable fair-scheduler configuration by setting this to 0.

hostname	string	<string>	Set a host name for the container.
memory	integer	<integer> (16 - N)	Amount of RAM for the VM in MB.
net[n]	string	name=<string> [,bridge=<bridge>] [,firewall=<1|0>] [,gw=<GatewayIPv4>] [,gw6=<GatewayIPv6>] [,hwaddr=<XX:XX:XX:XX:XX:XX>] [,ip=<IPv4Format/CIDR>] [,ip6=<IPv6Format/CIDR>] [,mtu=<integer>] [,rate=<mbps>] [,tag=<integer>] [,trunks=<vlanid[;vlanid...]>] [,type=<veth>]	Specifies network interfaces for the container.
password	string	<string>	Sets root password inside container.
rootfs	string	[volume=]<volume> [,acl=<1|0>] [,quota=<1|0>] [,replicate=<1|0>] [,ro=<1|0>] [,shared=<1|0>] [,size=<DiskSize>]	Use volume as container root.
ssh-public-keys	string	<string>	Setup public SSH keys (one key per line, OpenSSH format).
swap	integer	<integer> (0 - N)	Amount of SWAP for the VM in MB.

Πίνακας 4: Προαιρετικές μεταβλητές κλήσεως /node/{node}/lxc του Proxmox VE REST API

Με βάση τα παραπάνω, λοιπόν, τα βήματα για την δημιουργία ένα Linux Container με χρήση του REST API του Proxmox VE είναι τα εξής:

- Αποθήκευση του *authorization cookie* στον τοπικό δίσκο, σε αρχείο που ονομάζεται *cookie*:

```
$ curl --silent --insecure --data "username=root@pam&password=yourpassword" \
 https://147.102.7.83:8006/api2/json/access/ticket \
| jq --raw-output '.data.ticket' | sed 's/^/PVEAuthCookie=/' > cookie
```

- Αποθήκευση του *CSRFPreventionToken* στον τοπικό δίσκο, σε αρχείο που ονομάζεται *csrftoken*:

```
$ curl --silent --insecure --data "username=root@pam&password=yourpassword" \
 https://147.102.7.83:8006/api2/json/access/ticket \
| jq --raw-output '.data.CSRFPreventionToken' | sed 's/^/CSRFPreventionToken:/' >
csrftoken
```

- Δημιουργία ενός Linux Container:

```
$ curl --silent --insecure --cookie "$(<cookie)" --header "$(<csrftoken)" X \
POST \
 --data vmid=100 \
```

```
 --data-urlencode hostname="vcpe1" \
 --data-urlencode ssh-public-keys="$(</dir/to/ssh/public/key)" \
 --data-urlencode ostemplate="local:vztmpl/ubuntu-16.04-standard_16.04-
1_amd64.tar.gz" \
 --data-urlencode rootfs="local-lvm:8" \
 --data-urlencode cpulimit="1" \
 --data-urlencode cpuunits="1024" \
 --data-urlencode memory="512" \
 --data-urlencode swap="512" \
 --data-urlencode net0="bridge=vmbr0,name=eth0,ip=dhcp,ip6=dhcp" \
 https://147.102.7.83:8006/api2/json/nodes/pve/lxc
```

Με αυτόν τον τρόπο έχει δημιουργηθεί ένα LXC το οποίο ονομάζεται *vcpe1*, έχει μοναδικό ID 100 και διαθέτει Ubuntu 16.04 64-bit, 1x1GHz επεξεργαστές, 8GB τοπικό δίσκο, 512MB μνήμη RAM, 512MB μνήμη SWAP και μια δικτυακή διεπαφή eth0 η οποία λαμβάνει διεύθυνση IP μέσω DCHP. Για την σύνδεση σε αυτό το LXC έχει υποτεθεί η ύπαρξη SSH ζευγαριού κλειδιών, όπου στο παραπάνω ερώτημα έχει δηλωθεί η τοποθεσία του δημοσίου κλειδιού. Για την δημιουργία του vcpe2, επαναλαμβάνεται η παραπάνω διαδικασία με τις κατάλληλες τροποποιήσεις στις μεταβλητές *vmid* και *hostname*.

4.2.2 Εγκατάσταση του GRE Tunnel

Όπως αναφέρθηκε στην προηγούμενη υποενότητα, τα φυσικά και τα εικονικά CPE επικοινωνούν μέσω GRE Tunnel. Για τις ανάγκες τις διπλωματικής εργασίας, ως φυσικά CPE έχουν χρησιμοποιηθεί δύο Raspberry Pi. Έτσι, για την εγκατάσταση του GRE Tunnel, εκτελούνται οι εξής εντολές στα φυσικά CPE:

```
@pcpe1:
# modprobe ip_gre
# ip tunnel add gre_cpe1 mode gre remote 147.102.7.119 local 147.102.40.106 \
ttl 255
# ip link set gre_cpe1 up
# ip addr add 10.0.0.2/30 dev gre_cpe1

@pcpe2:
# modprobe ip_gre
# ip tunnel add gre_cpe2 mode gre remote 147.102.7.120 local 147.102.39.10 \
ttl 255
# ip link set gre_cpe2 up
# ip addr add 10.0.0.6/30 dev gre_cpe2
```

Με τις εντολές αυτές, αρχικά ενεργοποιείται το κατάλληλο module. Έπειτα, δημιουργείται εικονική δικτυακή διεπαφή gre_cpe, η οποία είναι της μορφής GRE και έχει ως τοπική διεύθυνση την IP του εκάστοτε pCPE, ενώ ως απομακρυσμένη διεύθυνση την IP του αντίστοιχου vCPE. Τέλος, ενεργοποιείται η συγκεκριμένη διεπαφή και λαμβάνει την κατάλληλη

ιδιωτική IP, όπως έχει παρουσιαστεί στο σχήμα της διάταξης. Ομοίως, εκτελούνται τις παρακάτω εντολές στα εικονικά CPE για να ολοκληρωθεί η εγκατάσταση του GRE Tunnel:

```
@vcpe1:
# modprobe ip_gre
# ip tunnel add gre_cpe1 mode gre remote 147.102.40.106 local 147.102.7.119 \
ttl 255
# ip link set gre_cpe1 up
# ip addr add 10.0.0.1/30 dev gre_cpe1

@vcpe2:
# modprobe ip_gre
# ip tunnel add gre_cpe2 mode gre remote 147.102.39.10 local 147.102.7.120 \
ttl 255
# ip link set gre_cpe2 up
# ip addr add 10.0.0.5/30 dev gre_cpe2
```

4.2.3 Εγγραφές στους πίνακες δρομολόγησης και IP

Η βασική λειτουργία του vCPE στην διάταξη, όπως αναλύθηκε παραπάνω, είναι η μετάφραση διευθύνσεων IP (NAT). Για να επιτευχθεί αυτό, εκτελούνται στα εικονικά CPE οι παρακάτω εντολές:

```
@vcpe1:
# iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE
# iptables -A FORWARD -i eth0 -o gre_cpe1 -m state \
   --state RELATED,ESTABLISHED -j ACCEPT
# iptables -A FORWARD -i gre_cpe1 -o eth0 -j ACCEPT
@vcpe2:
# iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE
# iptables -A FORWARD -i eth0 -o gre_cpe2 -m state \
   --state RELATED,ESTABLISHED -j ACCEPT
# iptables -A FORWARD -i gre_cpe2 -o eth0 -j ACCEPT
```

Με τις εντολές αυτές, ενεργοποιείται το *masquerading*, δηλαδή η αντικατάσταση της ιδιωτικής IP σε δημόσια για τα εξερχόμενα πακέτα του κόμβου και το αντίστροφο για τα εισερχόμενα, καθώς και η προώθηση των πακέτων από και προς το GRE Tunnel. Με αυτό τον τρόπο αντικαθίσταται η ιδιωτική IP στα πακέτα που προέρχονται από το pCPE και προορίζονται για το διαδίκτυο, έπειτα προωθούνται στον παραλήπτη και μόλις το vCPE λαμβάνει τις αντίστοιχες απαντήσεις, επαναφέρει τις αρχικές ιδιωτικές IP και τις προωθεί στο pCPE για να φτάσουν στον πραγματικό παραλήπτη. Για την προώθηση τους μετά από αυτήν την επαναφορά, πρέπει να προσθέσουμε στους πίνακες δρομολόγησης των vCPE τα οικιακά δίκτυα με την παρακάτω εντολή:

```
@vcpe1:
# ip route add 192.168.1.0/24 via 10.0.0.1
```

```
@vcpe2:
# ip route add 192.168.2.0/24 via 10.0.0.5
```

Τέλος, πρέπει να γίνουν οι κατάλληλες εντολές στον πίνακα δρομολόγησης των φυσικών CPE, έτσι ώστε η κίνηση που δεν αφορά το εσωτερικό του οικιακού δικτύου να μεταφέρεται στο αντίστοιχο vCPE και να προωθείται στο διαδίκτυο, όπως περιγράφθηκε παραπάνω. Για αυτό τον σκοπό, εκτελούνται οι παρακάτω εντολές στα φυσικά CPE:

```
@pcpe1:
# ip route replace default via 10.0.0.1
@pcpe2:
# ip route replace default via 10.0.0.5
```

Με την εντολή αυτή, τροποποιούμε την προεπιλεγμένη πύλη, τοποθετώντας την διεύθυνση που αντιστοιχεί στο GRE Tunnel του αντίστοιχου εικονικού CPE. Έτσι οποιαδήποτε εγγραφή στον πίνακα δρομολόγησης δεν αντιστοιχεί με κάποια IP του οικιακού δικτύου, το pCPE θα την προωθεί στο vCPE που είναι συνδεδεμένο ώστε να το διαχειριστεί κατάλληλα.

4.2.4 Παραμετροποιήσεις συστήματος

Για να ολοκληρωθεί η διαδικασία αναπαραγωγής της διάταξης, απαιτούνται κάποιες επιπλέον παραμετροποιήσεις στο λειτουργικό της κάθε συσκευή. Συγκεκριμένα, θα πρέπει να ενεργοποιηθεί η δυνατότητα λειτουργίας ως δρομολογητής στις CPE συσκευές, η οποία είναι από προεπιλογή απενεργοποιημένη στα Linux συστήματα. Αυτό επιτυγχάνεται τροποποιώντας το αρχείο /etc/sysctl.conf, σε όλα τα CPE:

```
/etc/sysctl.conf:
net.ipv4.ip_forward = 1
```

4.2.5 Έλεγχος ορθής λειτουργίας

Για την επιβεβαίωση της σωστής λειτουργίας του GRE Tunnel, επιλέγεται τυχαία ένας κόμβος και εκτελείται η εντολή traceroute, που πέρα από την αναμενόμενη απάντηση, επαληθεύεται και το γεγονός ότι η επικοινωνία μεταξύ των δύο κόμβων πραγματοποιείται με ένα hop. Έτσι, στον κόμβο *pcpe1*:

```
$ traceroute 10.0.0.1
traceroute to 10.0.0.1 (10.0.0.1), 30 hops max, 60 byte packets
 1  10.0.0.1 (10.0.0.1)  0.669 ms  0.727 ms  0.691 ms
```

Στην συνέχεια, με σκοπό την έλεγχο της σωστής προώθησης των πακέτων προς το διαδίκτυο, χρησιμοποιούμε την εντολή traceroute επιχειρώντας να στείλουμε TCP πακέτα σε κάποιο δημόσιο κόμβο. Στην συγκεκριμένη περίπτωση, επιλέχθηκε ο δημόσιος DNS

εξυπηρετητής της Google, ο οποίος διαθέτει την διεύθυνση IP 8.8.8.8. Όπως διακρίνεται παρακάτω, η επικοινωνία έγινε επιτυχώς:

```
$ traceroute 8.8.8.8 -n
traceroute to 8.8.8.8 (8.8.8.8), 30 hops max, 60 byte packets
 1  10.0.0.1  0.750 ms  0.650 ms  0.616 ms
 2  * * *
 3  62.217.96.168  1.486 ms  1.453 ms  1.403 ms
 4  83.97.88.69  1.369 ms  1.407 ms  1.323 ms
 5  62.40.112.215  29.253 ms  29.211 ms  29.180 ms
 6  217.29.66.183  29.125 ms  29.144 ms  29.167 ms
 7  72.14.203.32  29.050 ms  28.904 ms  28.872 ms
 8  * * *
 9  66.249.95.251  29.130 ms 108.170.234.245  29.595 ms 108.170.235.33  29.562 ms
10  8.8.8.8  29.429 ms  29.384 ms  28.997 ms
```

Όσον αφορά την απομόνωση του οικιακού δικτύου από το διαδίκτυο, κάτι τέτοιο είναι προφανές, αφού για την αποστολή των πακέτων, οι υπολογιστές του οικιακού δικτύου χρησιμοποιούν την ιδιωτική IP του δικτύου και το vCPE αναλαμβάνει την μετάφραση διευθύνσεων, πραγματοποιώντας την διαδικασία που αναλύθηκε παραπάνω. Τέλος, τα οικιακά δίκτυα είναι επίσης απομονωμένα, όπως φαίνεται και με την εκτέλεση της παρακάτω εντολής:

```
$ ping 192.168.2.1 -c 4
PING 192.168.2.1 (192.168.2.1) 56(84) bytes of data.

--- 192.168.2.1 ping statistics ---
4 packets transmitted, 0 received, 100% packet loss, time 3024ms
```

4.3 Αυτοματοποίηση

Κατά το προηγούμενο κεφάλαιο, αναλύθηκε η σημασία της ενορχήστρωσης για την ύπαρξη ενός NFV δικτύου. Για τον λόγο αυτό, παραχωρήθηκε πρόσβαση σε μηχάνημα του εργαστηρίου στο οποίο ήταν εγκατεστημένο το Ansible. Στο μηχάνημα αυτό, δημιουργήθηκαν playbooks για την αυτοματοποίηση της διαδικασίας αναπαραγωγής της παραπάνω διάταξης, οι λειτουργίες των οποίων καθώς και οι μεταβλητές που απαιτούνται για την εκτέλεση τους θα αναπτυχθούν πλήρως στην συγκεκριμένη υποενότητα. Παράλληλα, στο παράρτημα της παρούσας εργασίας, βρίσκεται αναλυτικά ο κώδικας των playbook. Η εκτέλεση αυτών γίνεται εντός του συγκεκριμένου μηχανήματος ή οποιουδήποτε άλλου που έχει εγκατεστημένο το Ansible μέσω της εντολής:

```
$ ansible-playbook -i $host, $playbook [--user=USERNAME] \
 [--private-key=PRIVATE_KEY_FILE] [--ask-pass] [--ask-become-pass] \
 [--extra-vars='$var-name1=$var-value1 $var-name2=$var-value2 ...']
```

Στην παραπάνω εντολή, οι μεταβλητή *host* αναπαριστά τον μηχάνημα στο οποίο πρόκειται να εκτελεστεί το *playbook*, η μεταβλητή *playbook* είναι το όνομα του αρχείου, ενώ ο χρήστης και το ιδιωτικό κλειδί SSH δηλώνονται με τις επιλογές *user* και *private-key* αντίστοιχα. Επίσης, για τα playbooks που απαιτούν κωδικό σύνδεσης ή/και κωδικό διαχειριστή είναι απαραίτητες οι επιλογές *ask-pass* ή/και *ask-become-pass*, αντίστοιχα. Τέλος, εντός της επιλογής *extra-vars* τοποθετούνται οι υποχρεωτικές και προαιρετικές μεταβλητές που απαιτούνται για την εκτέλεση του.

4.3.1 Παραμετροποίηση των φυσικών CPE

Για την ολοκληρωμένη παραμετροποίηση των φυσικών CPE δημιουργήθηκαν δύο playbooks, το pcpe_init.yml και το pcpe_set.yml. Ακολουθεί αναλυτική περιγραφή:

4.3.1.1 pcpe_init.yml

Το συγκεκριμένο playbook έχει ως στόχο την αρχική παραμετροποίηση του φυσικού CPE, που αφορά την εγκατάσταση της Python και την εμφύτευση ενός δημοσίου κλειδιού για ευκολότερη μελλοντική σύνδεση. Για την εκτέλεση του απαιτούνται οι επιλογές *ask-pass* και *ask-become-pass*, καθώς και η πρότερη ύπαρξη ενός ζεύγους δημοσίου-ιδιωτικού κλειδιού, ενώ οφείλεται να δηλωθεί ο χρήστης που θα συνδεθεί το Ansible, μέσω της επιλογής *user*. Δεν διαθέτει κάποια υποχρεωτική μεταβλητή, ενώ η μοναδική προαιρετική μεταβλητή με όνομα *ssh_public_key* είναι η τοποθεσία του δημοσίου κλειδιού, με προεπιλεγμένη τιμή την *~/.ssh/diploma.pub*. Οι λειτουργίες που εκτελεί αναλυτικά είναι οι εξής:

- Έλεγχος και εγκατάσταση της τελευταίας έκδοσης *Python*. Το πακέτο αυτό είναι απαραίτητο για την ομαλή εκτέλεση πολλών επιμέρους λειτουργιών που εκτελούνται από το Ansible.
- Εμφύτευση του δημοσίου κλειδιού στον φάκελο ~/.ssh

4.3.1.2 pcpe_set.yml

Με το playbook αυτό επιτυγχάνεται η πλήρης παραμετροποίηση των φυσικών CPE συσκευών όπως αυτή περιγράφθηκε στην προηγούμενη υποενότητα, δηλαδή αναλαμβάνει την εγκατάσταση του GRE Tunnel, την εισαγωγή των απαραίτητων εγγραφών στον πίνακα δρομολόγησης και την παραμετροποίηση του συστήματος. Για την εκτέλεση του είναι απαραίτητη οι επιλογές *ask-become-pass* και *user*, ενώ ο χρήστης χρησιμοποιεί για την σύνδεση του το ιδιωτικό κλειδί που αντιστοιχεί στο δημόσιο που εμφυτεύθηκε με το προηγούμενο playbook. Για την σύνδεση του οι μεταβλητές που χρησιμοποιεί είναι οι εξής:

- local_ip (υποχρεωτική): Η δημόσια IP του pCPE με την οποία θα συνδεθεί στο GRE Tunnel.

- remote_ip (υποχρεωτική): Η δημόσια IP του αντίστοιχου vCPE που είναι συνδεδεμένη στο GRE Tunnel.
- gre_name (προαιρετική): Η ονομασία της εικονικής δικτυακής διεπαφής που δημιουργείται για την εγκατάσταση του GRE Tunnel. Σε περίπτωση που δεν δηλωθεί, η μεταβλητή παίρνει την τιμή "gre_cpe".
- gre_ip (προαιρετική): Η ιδιωτική διεύθυνση IP που αντιστοιχεί στην τοπική διεύθυνση IP του GRE Tunnel. Η προεπιλεγμένη τιμή της είναι 10.0.0.2.
- gateway (προαιρετική): Η απομακρυσμένη διεύθυνση του vCPE που αντιστοιχεί στο GRE Tunnel και θα αποτελέσει την νέα προεπιλεγμένη πύλη του πίνακα δρομολόγησης. Από προεπιλογή, είναι ορισμένη σε 10.0.0.1.

Οι λειτουργίες που εκτελεί το συγκεκριμένο playbook είναι κατά σειρά:

- Έλεγχος για την δήλωση των υποχρεωτικών μεταβλητών. Σε περίπτωση που κάποια από αυτές απουσιάζει, η εκτέλεση του playbook τερματίζει με σφάλμα εμφανίζοντας το κατάλληλο μήνυμα.
- Δημιουργία της εικονικής δικτυακής διεπαφής του GRE Tunnel με το όνομα που έχει οριστεί στην μεταβλητή $gre_name.
- Ενεργοποίηση της συγκεκριμένης διεπαφής.
- Ανάθεση της ιδιωτικής IP $gre_ip στην συγκεκριμένη διεπαφή.
- Ενεργοποίηση της προώθησης IP στο pCPE, έτσι ώστε αυτό να λειτουργεί ως δρομολογητής.
- Τροποποίηση της προεπιλεγμένης πύλης, έτσι ώστε η κίνηση προς το διαδίκτυο να γίνεται μέσω του vCPE.

4.3.2 Παραμετροποίηση των εικονικών CPE

Όσον αφορά την παραμετροποίηση των εικονικών CPE, δημιουργήθηκαν δύο playbooks, με την ονομασία vcpe_create.yml και vcpe_set.yml και τις παρακάτω λειτουργίες:

4.3.2.1 vpce_create.yml

Το συγκεκριμένο playbook εκτελείται στον Proxmox VE και δημιουργεί ένα Linux Container με τα επιθυμητά χαρακτηριστικά, ενώ στο τέλος επιστρέφει μια μεταβλητή περιβάλλοντος LXC_IP όπου περιέχει την δημόσια IP του LXC που δημιουργήθηκε. Η ιδιομορφία του εν λόγω playbook είναι πως επειδή ορισμένες λειτουργίες που εκτελεί, όπως η δημιουργία και η εκκίνηση του LXC γίνονται ασύγχρονα, έχουν τοποθετηθεί στα κατάλληλα σημεία ειδικοί handlers οι οποίοι περιμένουν είτε το σήμα του χρήστη για την συνέχεια της εκτέλεσης τους είτε το να παρέλθει ένα δεδομένο χρονικό διάστημα. Για να επιτευχθεί η σύνδεση

στον Proxmox, απαιτείται η χρήση της επιλογής *private-key*. Κατά τη εκτέλεση του, χρησιμοποιούνται οι εξής μεταβλητές:

- vmid (υποχρεωτική): Η μεταβλητή αυτή είναι το μοναδικό ID που θα διαθέτει το LXC που πρόκειται να δημιουργηθεί. Η τιμή αυτή είναι integer και οφείλει να είναι μεγαλύτερη του 100.
- proxmox_port (προαιρετική): Η πόρτα στην οποία είναι συνδεδεμένο το REST API του Proxmox VE. Η προεπιλεγμένη τιμή είναι *8006* και θα πρέπει να αλλάξει μόνο σε περίπτωση που έχει δηλωθεί διαφορετική κατά την εγκατάσταση του.
- node (προαιρετική): Το όνομα του κόμβου όπου θα περιέχεται το LXC που θα δημιουργηθεί. Η προεπιλεγμένη τιμή στην συγκριμένη περίπτωση είναι *"pve"*.
- hostname (προαιρετική): Το όνομα του νέου LXC. Σε περίπτωση που δεν οριστεί, η τιμή σχηματίζεται αυτόματα συνδυάζοντας το string *"vcpe"* και την μεταβλητή *vmid*.
- ssh_public_keys (προαιρετική): Η τοποθεσία του δημοσίου κλειδιού που θα εμφυτευθεί στο νέο LXC για την απομακρυσμένη σύνδεση. Η προεπιλεγμένη τοποθεσία είναι η *~/.ssh/diploma.pub*.
- ostemplate (προαιρετική): Την διεύθυνση του template που πρόκειται να χρησιμοποιηθεί για την δημιουργία του LXC. Από προεπιλογή, η μεταβλητή περιέχει την τοπική διεύθυνση του template για *Ubuntu 16.04 64-bit*.
- rootfs (προαιρετική): Το μέγεθος του τοπικού δίσκου του νέου LXC. Το προεπιλεγμένο μέγεθος είναι *8GB*.
- cpulimit (προαιρετική): Το πλήθος των πυρήνων του νέου LXC. Σε περίπτωση που δεν δηλωθεί, το νέο μηχάνημα θα κατέχει 1 πυρήνα.
- cpuunits (προαιρετική): Η ταχύτητα σε MHz του κάθε πυρήνα του νέου LXC. Η προεπιλεγμένη ταχύτητα είναι *1GHz*.
- memory (προαιρετική): Το μέγεθος της μνήμης RAM του LXC που θα δημιουργηθεί. Το προεπιλεγμένο μέγεθος της είναι *512MB*.
- swap (προαιρετική): Το μέγεθος της μνήμης SWAP του νέου LXC. Σε περίπτωση που δεν δηλωθεί, το μέγεθός της θα είναι *512MB*.
- net0 (προαιρετική): Η παραμετροποίηση των δικτυακών διεπαφών του νέου LXC. Συγκεκριμένα, σε αυτήν ορίζεται η διεπαφή του node με την οποία θα γίνει bridge η διεπαφή του LXC, το όνομα της και οι διευθύνσεις IPv4 και IPv6 της. Σε περίπτωση που δεν δηλωθεί, το νέο LXC θα γίνει bridge με την διεπαφή *vmbr0* του node και θα ονομάζεται *eth0*, ενώ οι διευθύνσεις IPv4 και IPv6 θα λαμβάνονται μέσω *dhcp*.
- param_file (προαιρετική): Η διαδρομή του αρχείου στο οποίο θα καταγραφεί η διεύθυνση IP του νέου LXC. Η διαδρομή οφείλει να υφίσταται εκ των προτέρων, ενώ σε περίπτωση

που το αρχείο δεν υπάρχει θα δημιουργηθεί. Το προκαθορισμένο αρχείο αυτό είναι το param.conf και τοποθετείται στην διαδρομή που εκτελείται το playbook.

Οι λειτουργίες που εκτελεί το συγκεκριμένο playbook είναι εξής:

- Έλεγχος για την δήλωση των υποχρεωτικών μεταβλητών. Σε περίπτωση που κάποια από αυτές απουσιάζει, η εκτέλεση του playbook τερματίζει με σφάλμα εμφανίζοντας το κατάλληλο μήνυμα.
- Δημιουργία του authorization cookie μέσω κλήσης στο REST API του Proxmox VE, όπως αναλύθηκε στην προηγούμενη υποενότητα, και αποθήκευση του σε τοπική μεταβλητή.
- Δημιουργία του νέου LXC με τα χαρακτηριστικά που έχουν ορισθεί στις παραπάνω μεταβλητές με χρήση του REST API.
- Αναμονή για την δημιουργία του LXC. Σε αυτό το σημείο ο χρήστης καλείται να αναμένει για 35 δευτερόλεπτα.
- Εκκίνηση του LXC μέσω του REST API.
- Αναμονή για εκκίνηση του LXC για 35 δευτερόλεπτα.
- Εύρεση της διεύθυνσης IP του νέου LXC και καταγραφή της στο προκαθορισμένο τοπικό αρχείο. Συγκεκριμένα το Ansible αναζητεί στο εν λόγω αρχείο το όνομα παραμέτρου $LAST_LXC και της αναθέτει ως τιμή την διεύθυνση IP. Αν η παράμετρος δεν υπάρχει θα προστεθεί στο τέλος του αρχείου.

4.3.2.2 vcpe_set.yml

Το συγκεκριμένο playbook εκτελείται εντός του vCPE και η λειτουργία του είναι σε μεγάλο βαθμό με το *pcpe_set.yml*. Συγκεκριμένα, αναλαμβάνει την σύνδεση του vCPE στο GRE Tunnel, καθώς και όλες τις περαιτέρω παραμετροποιήσεις που αναλύθηκαν στην προηγούμενη υποενότητα. Για την σύνδεση Οι μεταβλητές που ορίζονται σε αυτό είναι οι εξής:

- local_ip (υποχρεωτική): Η δημόσια IP του pCPE με την οποία θα συνδεθεί στο GRE Tunnel.
- remote_ip (υποχρεωτική): Η δημόσια IP του αντίστοιχου vCPE που είναι συνδεδεμένη στο GRE Tunnel.
- gre_name (προαιρετική): Η ονομασία της εικονικής δικτυακής διεπαφής που δημιουργείται για την εγκατάσταση του GRE Tunnel. Σε περίπτωση που δεν δηλωθεί, η μεταβλητή παίρνει την τιμή "gre_cpe".
- gre_ip (προαιρετική): Η ιδιωτική διεύθυνση IP, που αντιστοιχεί στην τοπική διεύθυνση IP του GRE Tunnel. Η προεπιλεγμένη τιμή της είναι 10.0.0.2.

- default_if (προαιρετική): Η δικτυακή διεπαφή με την οποία το vCPE είναι συνδεδεμένο στο διαδίκτυο. Σε περίπτωση που οριστεί, η προεπιλεγμένη τιμή που λαμβάνει είναι η *"eth0"*.
- private_net (προαιρετική): Το ιδιωτικό οικιακό δίκτυο, σε μορφή CIDR, με το οποίο είναι συνδεδεμένο το αντίστοιχο pCPE. Για προεπιλεγμένη τιμή, έχει επιλεχθεί το υποδίκτυο *192.168.1.0/24*.

Οι λειτουργίες που εκτελεί το συγκεκριμένο playbook είναι κατά σειρά:

- Έλεγχος για την δήλωση των υποχρεωτικών μεταβλητών. Σε περίπτωση που κάποια από αυτές απουσιάζει, η εκτέλεση του playbook τερματίζει με σφάλμα εμφανίζοντας το κατάλληλο μήνυμα.
- Δημιουργία της εικονικής δικτυακής διεπαφής του GRE Tunnel με το όνομα που έχει οριστεί στην μεταβλητή $gre_name.
- Ενεργοποίηση της συγκεκριμένης διεπαφής.
- Ανάθεση της ιδιωτικής IP $gre_ip στην συγκεκριμένη διεπαφή.
- Ενεργοποίηση της προώθησης IP στο vCPE, έτσι ώστε αυτό να λειτουργεί ως δρομολογητής.
- Εισαγωγή των κατάλληλων εγγραφών στον IPtable του λειτουργικού, ώστε να ενεργοποιηθεί η μετάφραση διευθύνσεων IP (NAT).
- Εισαγωγή των κατάλληλων εγγραφών στον πίνακα δρομολόγησης του vCPE, ώστε να επιτυγχάνεται η επικοινωνία του με το οικιακό δίκτυο.

4.3.3 Αντιστροφή διαδικασίας

Για την αναίρεση της παραπάνω διαδικασίας, αναπτύχθηκαν δύο playbooks, τα οποία ονομάζονται pcpe_unset.yml και vcpe_delete.yml. Οι λειτουργίες αυτών είναι οι εξής:

4.3.3.1 pcpe_unset.yml

Το playbook αυτό εκτελείται εντός του pCPE και σκοπός του είναι η αποσύνδεση του από το GRE Tunnel. Για να λειτουργήσει απαιτείται ο χρήστης να χρησιμοποιήσει τις επιλογές *user* και *ask-become-pass*, καθώς και το απαραίτητο ιδιωτικό κλειδί. Μοναδική μεταβλητή που διαθέτει, η οποία είναι και υποχρεωτική, είναι η μεταβλητή *gateway*, η οποία είναι η προεπιλεγμένη πύλη που θα χρησιμοποιεί η συσκευή μετά την απεγκατάσταση του GRE Tunnel. Αναλυτικότερα, οι λειτουργίες που εκτελεί είναι:

- Έλεγχος για την δήλωση των υποχρεωτικών μεταβλητών. Σε περίπτωση που κάποια από αυτές απουσιάζει, η εκτέλεση του playbook τερματίζει με σφάλμα εμφανίζοντας το κατάλληλο μήνυμα.

- Απενεργοποίηση της εικονικής δικτυακής διεπαφής του GRE Tunnel.
- Διαγραφή της συγκεκριμένης δικτυακής διεπαφής.
- Απενεργοποίηση της προώθησης IP.
- Επαναφορά προεπιλεγμένης πύλης.

4.3.3.2 vcpe_delete.yml

Στόχος αυτού του playbook, το οποίο εκτελείται στον Proxmox, είναι η διαγραφή του της εικονικής CPE συσκευής. Για την εκτέλεση του απαιτείται η επιλογή *private-key*, ώστε να επιτευχθεί η σύνδεση στον Proxmox. Διαθέτει τις εξής μεταβλητές:

- vmid (υποχρεωτική): Η μεταβλητή αυτή είναι το μοναδικό ID που διαθέτει το LXC που πρόκειται να διαγραφεί.
- proxmox_port (προαιρετική): Η πόρτα στην οποία είναι συνδεδεμένο το REST API του Proxmox VE. Η προεπιλεγμένη τιμή είναι *8006* και θα πρέπει να αλλάξει μόνο σε περίπτωση που έχει δηλωθεί διαφορετική κατά την εγκατάσταση του.
- node (προαιρετική): Το όνομα του κόμβου όπου θα περιέχεται το LXC που θα δημιουργηθεί. Η προεπιλεγμένη τιμή στην συγκριμένη περίπτωση είναι *"pve"*.

Οι λειτουργίες που εκτελεί είναι οι εξής:

- Έλεγχος για την δήλωση των υποχρεωτικών μεταβλητών. Σε περίπτωση που κάποια από αυτές απουσιάζει, η εκτέλεση του playbook τερματίζει με σφάλμα εμφανίζοντας το κατάλληλο μήνυμα.
- Δημιουργία του authorization cookie μέσω κλήσης στο REST API του Proxmox VE, όπως αναλύθηκε στην προηγούμενη υποενότητα, και αποθήκευση του σε τοπική μεταβλητή.
- Τερματισμός λειτουργίας του LXC με χρήση του REST API.
- Αναμονή για τον τερματισμό του LXC. Σε αυτό το σημείο ο χρήστης καλείται να αναμένει για 5 δευτερόλεπτα.
- Διαγραφή του LXC μέσω του REST API.

5

Αξιολόγηση

Στο παρόν κεφάλαιο, θα παρουσιαστούν τα αποτελέσματα αξιολόγησης της αναπτυσσόμενης διάταξης που περιγράφθηκε. Αρχικά, θα αναλυθούν οι παράμετροι αξιολόγησης της και η σημασία τους. Έπειτα, θα επεξηγηθεί η πειραματική διαδικασία αναλύοντας τα βήματα που ακολουθήθηκαν, ενώ στην συνέχεια θα παρουσιαστούν τα αποτελέσματα των πειραμάτων, επιχειρώντας παράλληλα την εξαγωγή συμπερασμάτων από αυτά. Τέλος, θα γίνει μια συγκεντρωτική αναφορά των συμπερασμάτων που προέκυψαν από τα πειράματα αξιολόγησης.

5.1 Παράμετροι αξιολόγησης

Εκτός από την ορθότητα λειτουργιάς, σημαντικός παράγοντας για την συνολική αξιολόγηση είναι η απόδοσης της διάταξης που αναπτύχθηκε. Για αυτό τον σκοπό, αποφασίστηκαν οι παρακάτω παράμετροι:

- Ο χρόνος που απαιτείται για την πλήρη δημιουργία και παραμετροποίηση ενός ζευγαριού φυσικού – εικονικού CPE. Ο χρόνος αυτός περιλαμβάνει την δημιουργία του vCPE, την εμφύτευση του δημοσίου κλειδιού στο pCPE, την εγκατάσταση του GRE Tunnel στους δύο κόμβους και την κατάλληλη παραμετροποίηση των πινάκων δρομολόγησης και IP.
- Εκτός από τον συνολικό χρόνο, σημαντική παράμετρος είναι ο απαιτούμενος χρόνος δημιουργίας ενός vCPE, καθώς ο κόμβος αυτός συγκεντρώνει το μεγαλύτερο ενδιαφέρον στην συνολική διάταξη. Επίσης, η διάρκεια αυτής της διαδικασίας αποτελεί μια ένδειξη

για τον χρόνο που απαιτείται για την μετακίνηση ενός υπάρχοντος LXC σε κάποιον άλλον κόμβο, με στόχο την αποφόρτιση του αρχικού κόμβου.

- Ο έλεγχος της μέγιστης ταχύτητας λήψης που δεν παρατηρείται σημαντική απώλεια πακέτων, καθώς και το ποσοστό χρήσης CPU και μνήμης κατά την λήψη.
- Η διάρκεια λήψης της ιστοσελίδας αναφοράς Kepler, που έχει αναπτυχθεί από το Ευρωπαϊκό Ινστιτούτο Τηλεπικοινωνιακών Προτύπων (ETSI), καθώς και άλλων HTTP αρχείων διαφόρων μεγεθών. Παράλληλα μετρήθηκε και το ποσοστό χρήσης CPU και μνήμης κατά την λήψη.

Είναι χρήσιμο να αναφερθεί, πως τα εικονικά CPE που δημιουργήθηκαν κατά την παρούσα διπλωματική διαθέτουν Ubuntu 16.04 64-bit, 1x1GHz επεξεργαστές, 8GB τοπικό δίσκο, 512MB μνήμη RAM και 512MB μνήμη SWAP. Τα συγκεκριμένα χαρακτηριστικά είναι σχετικά αυξημένα σε σχέση με τα ελάχιστα που συστήνει ο διανομέας του συγκεκριμένου λογισμικού, με σκοπό την εγγύηση της ορθής λειτουργίας των vCPE συσκευών.

5.2 Οργάνωση πειραμάτων - Αποτελέσματα

5.2.1 Διάρκεια πλήρους αναπαραγωγής vCPE – pCPE ζεύγους

Για τον υπολογισμό του συνολικού απαιτούμενου χρόνου δημιουργίας ενός ζευγαριού φυσικού – εικονικού CPE, συντάχθηκε κατάλληλο bash script το οποίο εκτελεί τα απαραίτητα playbooks με τις κατάλληλες παραμέτρους. Κατά την εκτέλεση αυτού του script χρησιμοποιείται η εντολή grep για την εύρεση από κατάλληλο αρχείο της διεύθυνση IP του LXC που δημιουργείται από το αντίστοιχο playbook (vcpe_create.yml). Παρακάτω παρουσιάζεται τα bash script που συντάχθηκε:

```
$ cat master.sh
#!/bin/bash
PVE=147.102.7.83
VMID=100
PCPE_IP=147.102.39.10
PCPE_USER=alex
PCPE_GRE=10.0.0.6
HOME_LAN=192.168.2.0/24
VCPE_GRE=10.0.0.5
SSH_PRIVATE=~/.ssh/diploma

ansible-playbook -i $PVE, vcpe_create.yml --private-key=$SSH_PRIVATE \
 --extra-vars="vmid=$VMID"

LAST_LXC=$(grep --only-matching --perl-regex "(?<=LAST_LXC\=).*" params.conf)

ansible-playbook -i $LAST_LXC, vcpe_set.yml --private-key=$SSH_PRIVATE \
 --extra-vars="local_ip=$LAST_LXC remote_ip=$PCPE_IP private_net=$HOME_LAN \
```

```
 gre_ip=$VCPE_GRE"

ansible-playbook -i $PCPE_IP, pcpe_init.yml –-user=$PCPE_USER \
 –-ask-pass –-ask-become-pass

ansible-playbook -i $PCPE_IP, pcpe_set.yml --private-key=$SSH_PRIVATE \
 –-user=$PCPE_USER –-ask-become-pass --extra-vars=""local_ip=$PCPE_IP \
 remote_ip=$LAST_LXC gre_ip=$PCPE_GRE gateway=$VCPE_GRE"
```

Για τον ακριβή υπολογισμό του χρόνου δημιουργίας, χρησιμοποιήθηκε, η εντολή time του Linux. Συγκεκριμένα, εκτελώντας την εντολή αυτή, σε συνδυασμό με το script που συντάχθηκε, δίνεται η δυνατότητα να εξαγωγής του ακριβή χρόνο εκτέλεσης που απαιτείται συνολικά για την εκτέλεση του εν λόγω script. Επιπλέον, για την εξαγωγή ασφαλών συμπερασμάτων, η συγκεκριμένη διαδικασία επαναλήφθηκε 10 φορές και υπολογίσθηκε ο μέσος όρος εκτέλεσης. Τα αποτελέσματα αυτών τον επαναλήψεων παρουσιάζονται αναλυτικά στον παρακάτω πίνακα:

Αριθμός Εκτέλεσης	Διάρκεια (sec)
1η	116,464
2η	105,754
3η	112,487
4η	135,240
5η	104,804
6η	116,971
7η	110,611
8η	113,922
9η	119,186
10η	109,772
Μέσος όρος	**114,521**

Πίνακας 5: Διάρκεια δημιουργίας ζεύγους φυσικού – εικονικού CPE

Παρατηρώντας τις μετρήσεις που καταγράφηκαν, παρατηρούμε πως η διάρκεια δημιουργίας ενός πλήρους ζεύγους εικονικού – φυσικού CPE είναι εξαιρετικά μικρή. Συγκεκριμένα σε λιγότερο από δύο λεπτά, επιτυγχάνεται η διασύνδεση της φυσικής συσκευής με μια νέα εικονική μέσω του GRE Tunnel και η πλήρης παραμετροποίηση τους ώστε να αποκτήσει η πρώτη σύνδεση στο διαδίκτυο μέσω του vCPE. Αξίζει να σημειωθεί πως οι παραπάνω διακυμάνσεις είναι αναμενόμενες, καθώς κατά την εκτέλεση του παραπάνω script ζητώνται οι απαραίτητοι κωδικοί για την σύνδεση με το pCPE, όπως αναλύθηκε και στην παρουσίαση των αντίστοιχων playbook.

5.2.2 Διάρκεια δημιουργίας εικονικού CPE

Για τον υπολογισμό του απαιτούμενου χρόνου για την δημιουργία ενός vCPE, ακολουθήθηκε αντίστοιχη πειραματική διαδικασία με αυτή που περιεγράφηκε κατά το προηγούμενο πείραμα. Συγκεκριμένα, δηλαδή εκτελέσθηκε το κατάλληλο playbook (vcpe_create.yml) και με χρήση της εντολής time, εξάχθηκε ο μέσος όσος του απαιτούμενου χρόνου από 10 διαφορετικές εκτελέσεις. Στον πίνακα που ακολουθεί, παρουσιάζονται οι χρόνοι εκτέλεσης που καταγράφηκαν καθώς και ο μέσος όρος εκτέλεσης που υπολογίσθηκε:

Αριθμός Εκτέλεσης	Διάρκεια (sec)
1η	72,896
2η	72,667
3η	72,605
4η	71,875
5η	72,545
6η	72,473
7η	72,769
8η	72,402
9η	72,778
10η	71,957
Μέσος όρος	**72,497**

Πίνακας 6: Διάρκεια δημιουργίας εικονικού CPE

Με βάση τα παραπάνω αποτελέσματα, συμπεραίνουμε πως η διαδικασία δημιουργίας είναι επίσης εξαιρετικά γρήγορη, με διάρκεια λίγο περισσότερο από ένα λεπτό. Επιπλέον, η διαδικασία αυτή παρουσιάζει πολύ μικρή διακύμανση, κάτι που είναι αναμενόμενο, αφού για δεν απαιτείται η εισαγωγή κωδικού την σύνδεση του Ansible στο Proxmox VE και χρησιμοποιείται το SSH κλειδί που έχει εγκατασταθεί στον κόμβο εκ των προτέρων. Επιπλέον, παρατηρούμε πως η τιμή αυτή είναι πολύ κοντά στο άθροισμα των χρονοκαθυστερήσεων που έχουν προστεθεί στο συγκεκριμένο playbook, λόγω της χρήσης ασύγχρονων κλήσεων προς το REST API του Proxmox VE, οι οποίες έχουν συνολική διάρκεια 65 δευτερόλεπτα. Αυτό μας οδηγεί στο συμπέρασμα πως οι υπόλοιπες διεργασίες που εκτελούνται στο συγκεκριμένο playbook δεν απαιτούν μεγάλο διάρκεια για να ολοκληρωθούν.

5.2.3 Έλεγχος bandwidth

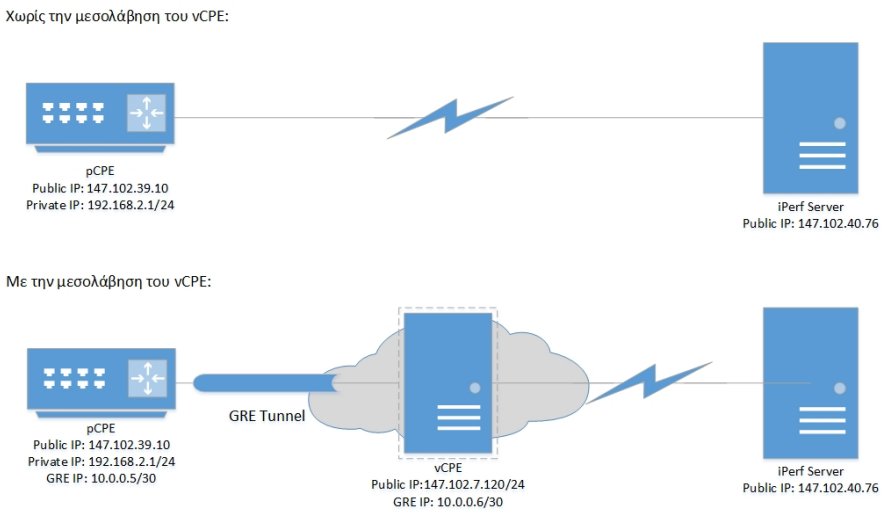
Για να αντλήσουμε τις πληροφορίες που περιεγράφηκαν στην προηγούμενη υποενότητα σχετικά με την ταχύτητα του δικτύου, χρησιμοποιήθηκε το εργαλείο iPerf3. Το εργαλείο αυτό παρέχει την δυνατότητα ελέγχου του ανώτατου ορίου bandwidth που είναι εφικτό να επιτευχθεί σε IP δίκτυα. Επίσης, μεταξύ πολλών άλλων τιμών, ελέγχει την απώλεια πακέτων που παρατηρούνται στο δίκτυο, ενώ για την εξαγωγή των μετρήσεων, υποστηρίζει πλήθος

παραμέτρων, όπως οι χρόνοι, το μέγεθος και το πρωτόκολλο (TCP, UDP, SCTP μέσω IPv4 και IPv6) των πακέτων.

Για την αξιολόγηση της διάταξης που αναπτύχθηκε κατά την παρούσα διπλωματική, εγκαταστάθηκε το iPerf3 σε έναν κόμβο του εργαστηρίου(iPerf Server) με δημόσια IP 147.102.40.76 και σε μια από τις διαθέσιμες φυσικές CPE συσκευές. Δεδομένου ότι το ένα από τα δύο Raspberry Pi που διατέθηκαν για την διπλωματική ανήκει στο ίδιο υποδίκτυο με τον iPerf Server, προτιμήθηκε η χρήση του Raspberry Pi με διεύθυνση IP 147.102.39.10. Έπειτα, με χρήση κατάλληλης εντολής στην φυσική CPE, έγινε αίτημα στον iPerf Server για λήψη πακέτων TCP και UDP με μεταβλητό bandwidth. Στην συνέχεια, παρατηρώντας το πραγματικό bandwidth που επιτεύχθηκε κατά την λήψη TCP πακέτων και αναζητώντας τις τιμές της μεταβλητής ώστε το ποσοστό απώλειας πακέτων UDP να μην ξεπερνά το 1-2%, μπορούμε να αποφανθούμε για το μέγιστο bandwidth που δύναται να επιτευχθεί μέσω της διάταξης. Η σύνταξη της εντολής που εκτελέσθηκε φαίνεται παρακάτω, ενώ η παράμετρος bandwidth αποφασίστηκε να λάβει τις τιμές 1Mbps, 2Mbps, 5Mbps, 10Mbps, 20Mbps, 50Mbps και 100Mbps:

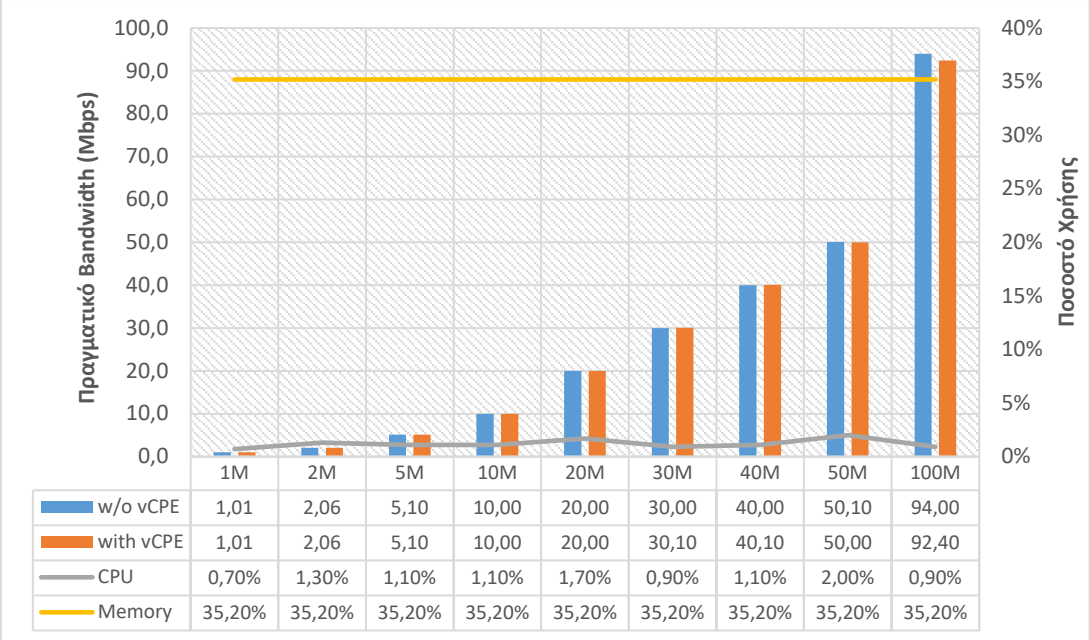
```
$ iperf3 -R -c 147.102.40.76 -b $BANDWIDTH # Αίτημα για λήψη TCP πακέτων
$ iperf3 -R -c $IPERF_SERVER_IP -u -b $BANDWIDTH # Αίτημα για λήψη UDP πακέτων
```

Επιπλέον, για να είναι εφικτή σύγκριση της απόδοσης του vCPE, η παραπάνω διαδικασία ακολουθήθηκε πριν και μετά την σύνδεση της φυσικής CPE συσκευή με την αντίστοιχη εικονική. Στην εικόνα που ακολουθεί, παρουσιάζεται η πειραματική διάταξη:



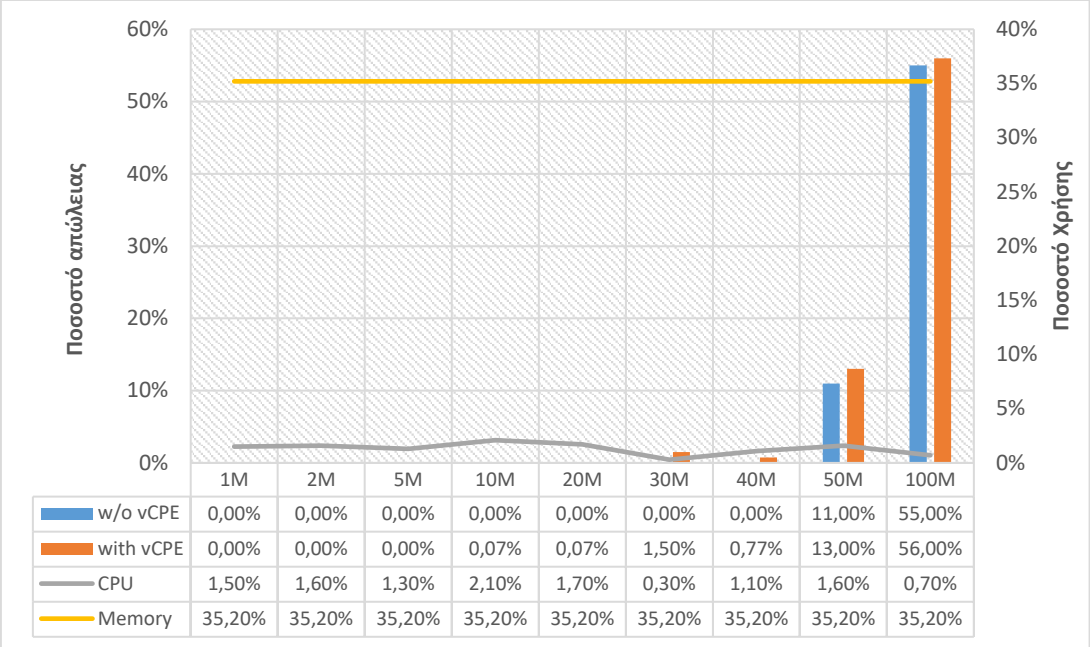
Εικόνα 5-1: Επικοινωνία pCPE - iPerf Server με ή χωρίς την μεσολάβηση vCPE

Στα παρακάτω διαγράμματα φαίνονται αναλυτικά το bandwidth που επιτεύχθηκε κατά την λήψη πακέτων και τα ποσοστά απώλειας πακέτων TCP για τις διάφορες τιμές του ζητούμενου bandwidth καθώς και η μεταβολή του ποσοστού χρήσης (δευτερεύον κατακόρυφος άξονας) CPU και μνήμης στο εικονικό CPE για κάθε αίτημα λήψης πακέτων TCP, τα οποία μετρήθηκαν μέσω της εντολής top.



	1M	2M	5M	10M	20M	30M	40M	50M	100M
w/o vCPE	1,01	2,06	5,10	10,00	20,00	30,00	40,00	50,10	94,00
with vCPE	1,01	2,06	5,10	10,00	20,00	30,10	40,10	50,00	92,40
CPU	0,70%	1,30%	1,10%	1,10%	1,70%	0,90%	1,10%	2,00%	0,90%
Memory	35,20%	35,20%	35,20%	35,20%	35,20%	35,20%	35,20%	35,20%	35,20%

Διάγραμμα 1: Bandwidth και χρήση CPU – μνήμης vCPE κατά την λήψη πακέτων TCP



	1M	2M	5M	10M	20M	30M	40M	50M	100M
w/o vCPE	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%	11,00%	55,00%
with vCPE	0,00%	0,00%	0,00%	0,07%	0,07%	1,50%	0,77%	13,00%	56,00%
CPU	1,50%	1,60%	1,30%	2,10%	1,70%	0,30%	1,10%	1,60%	0,70%
Memory	35,20%	35,20%	35,20%	35,20%	35,20%	35,20%	35,20%	35,20%	35,20%

Διάγραμμα 2: Ποσοστό απωλειών πακέτων και χρήσης CPU – μνήμης vCPE κατά την λήψη πακέτων UDP

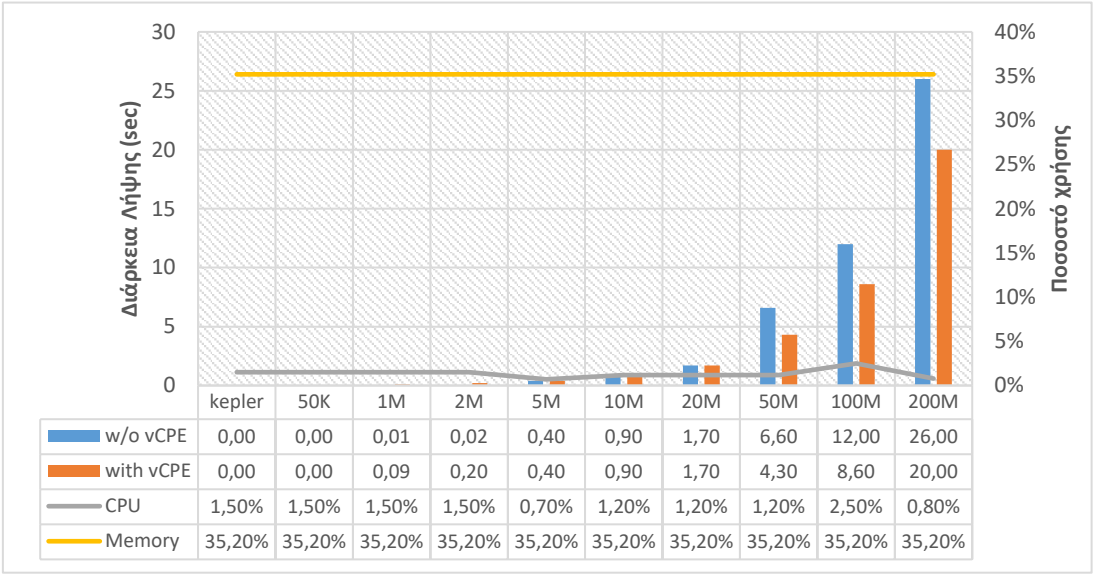
Παρατηρώντας τα παραπάνω διαγράμματα, εξάγονται διάφορα χρήσιμα συμπεράσματα. Καταρχήν, στο διάγραμμα 1, παρατηρούμε πως κατά την λήψη TCP πακέτων το bandwidth που επιτεύχθηκε σε κάθε περίπτωση είναι εξαιρετικά κοντά στο θεωρητικό bandwidth που αιτήθηκε κατά την εκτέλεση της εντολής. Κάτι τέτοιο ήταν σχετικά αναμενόμενο, καθώς το Raspberry Pi που χρησιμοποιείται διαθέτει κάρτα δικτύου με μέγιστη δυνατή ταχύτητα 100 Mbps. Επιπλέον, στο διάγραμμα 2, η απόδοση της δρομολόγησης μέσω vCPE για bandwidth μικρότερο από 10Mbps είναι πανομοιότυπη με την αντίστοιχη που το vCPE απουσιάζει. Στο διάστημα 10Mbps έως 40 Mbps, η διάταξη που χρησιμοποιεί το vCPE παρουσιάζει κάποιες απώλειες, οι οποίες

βρίσκονται εντός του αποδεκτού πλαισίου. Αντίθετα, όταν οι τιμές του bandwidth ξεπεράσουν τα 40 Mbps, παρατηρείται μεγάλη απώλεια πακέτων. Ωστόσο, αντίστοιχη απώλεια παρατηρείται και στην διάταξη πριν την εγκατάσταση του vCPE, επομένως καταλήγουμε στο συμπέρασμα πως το όριο των 40Mbps για την λήψη UDP πακέτων προκύπτει από γενικότερους περιορισμούς που σχετίζονται με την συσκευή και δεν οφείλονται στην δρομολόγηση μέσω του vCPE που αναπτύχθηκε.

Ένα άλλο πολύ σημαντικό συμπέρασμα είναι πως η διαδικασία δρομολόγησης που αναπτύχθηκε δεν επιβαρύνει καθόλου την λειτουργία του εικονικού CPE. Αυτό προκύπτει από το γεγονός ότι οι μεταβολές της CPU είναι τόσο αμελητέες, που κρίνονται ως φυσιολογικές και δικαιολογημένες. Επιπλέον, η μνήμη της συσκευής δεν μεταβάλλεται στο ελάχιστο.

5.2.4 Έλεγχος ταχύτητας λήψης

Στον server που παραχωρήθηκε για την εκτέλεση του προηγούμενου πειράματος, τοποθετήθηκαν, επίσης, η ιστοσελίδα αναφοράς Kepler και διάφορα HTTP αρχεία, με το μέγεθος τους να ξεκινάει από 50KB και να καταλήγει σε 200MB. Έπειτα, με χρήση της εντολής wget στο pCPE ελήφθησαν τα συγκεκριμένα αρχεία και καταγράφηκε ο χρόνος λήψης. Η διαδικασία αυτή, όπως και η προηγούμενη, εκτελέσθηκε δύο φορές: αρχικά πριν την εγκατάσταση του vCPE και έπειτα μετά την εγκατάσταση του. Μέσω την εντολής top καταγράφηκαν τα ποσοστά χρήσης CPU και μνήμης κατά το διάστημα της λήψης. Στα παρακάτω διάγραμμα παρουσιάζονται τα αποτελέσματα:



	kepler	50K	1M	2M	5M	10M	20M	50M	100M	200M
w/o vCPE	0,00	0,00	0,01	0,02	0,40	0,90	1,70	6,60	12,00	26,00
with vCPE	0,00	0,00	0,09	0,20	0,40	0,90	1,70	4,30	8,60	20,00
CPU	1,50%	1,50%	1,50%	1,50%	0,70%	1,20%	1,20%	1,20%	2,50%	0,80%
Memory	35,20%	35,20%	35,20%	35,20%	35,20%	35,20%	35,20%	35,20%	35,20%	35,20%

Διάγραμμα 3: Διάρκεια απωλειών πακέτων και χρήσης CPU – μνήμης vCPE κατά την λήψη αρχείων HTTP

Με βάση τα αποτελέσματα που παρουσιάζονται στο διάγραμμα 3 τα συμπεράσματα που εξάχθηκαν για την θετική απόδοση του vCPE επαληθεύονται πλήρως. Συγκεκριμένα, βλέπουμε πως οι χρόνοι λήψης των HTTP αρχείων στην περίπτωση που η δρομολόγηση επιτυγχάνεται

μέσω του vCPE είναι όμοιοι (και σε μερικές περιπτώσεις καλύτεροι) από την περίπτωση που το vCPE απουσιάζει από την διάταξη. Επιπλέον, το διάγραμμα αυτό επιβεβαιώνει το συμπέρασμα πως η δρομολόγηση που πραγματοποιεί το vCPE δεν επιβαρύνει καθόλου την γενικότερη λειτουργία του, αφού τα ποσοστά χρήσης CPU και μνήμης είναι όμοια με αυτά του προηγούμενου πειράματος.

5.3 Σύνοψη συμπερασμάτων αξιολόγησης

Κρίνοντας από τα αποτελέσματα που παρουσιάστηκαν παραπάνω, ο στόχος της παρούσας διπλωματικής, δηλαδή η δημιουργία ενός οικιακού vCPE που προσομοιάζει τις λειτουργίες του υπάρχοντος φυσικού σε ανάλογη απόδοση, κρίνεται επιτυχημένος. Ειδικότερα, κατά την παρούσας διπλωματική, επιτεύχθηκε η λειτουργία ενός απόλυτα σταθερού και εγγυημένης λειτουργίας vCPE, που μπορεί να δημιουργηθεί σε πολύ μικρό διάστημα. Πιο συγκεκριμένα, για την δημιουργία μιας εικονικής CPE συσκευής απαιτείται κάτι περισσότερο από 1 λεπτό, ενώ για την πλήρη ανάπτυξη ενός ζεύγους εικονικής – φυσικής συσκευής απαιτούνται λιγότερο από 2 λεπτά.

Επιπλέον, η απόδοση του vCPE που αναπτύχθηκε είναι πολύ κοντά στην απόδοση της φυσικής συσκευής. Αρχικά, μέσω του διαγράμματος 1, συμπεραίνουμε πως το μέγιστο bandwidth που μπορεί να επιτευχθεί μέσω του vCPE είναι κοντά σε αυτό που επιτυγχάνεται χωρίς αυτό, δηλαδή κοντά στα 100 Mbps που είναι και το όριο του υλικού. Επιπλέον, βλέποντας το διάγραμμα 2, παρατηρούμε πως για τιμές του bandwidth μικρότερες των 40Mbps το vCPE δεν χάνει σχεδόν καθόλου πακέτα UDP, ενώ για μεγαλύτερες τιμές η απώλειες είναι μεν σημαντικές, ωστόσο αυτές οι απώλειες δεν οφείλονται στην υλοποίηση αφού παρουσιάζονται και χωρίς την παρουσία της εικονικής CPE συσκευής. Αυτό μας οδηγεί στο συμπέρασμα πως η διάταξη είναι ικανή να πραγματοποιήσει λήψεις με ταχύτητες που φτάνουν στα 40Mbps χωρίς κανένα απολύτως πρόβλημα, ενώ η ταχύτητα αυτή μπορεί να φτάσει τα 100Mbps για συνδέσεις tcp. Έτσι η συνολική του απόδοση κρίνεται ως ιδιαίτερα ικανοποιητική.

Η καλή απόδοση της συσκευής που αναπτύχθηκε φαίνεται και από την απαιτούμενη διάρκεια για την λήψη HTTP πακέτων. Η διάρκεια αυτή, που αποτυπώνεται στο διάγραμμα 3, είναι πανομοιότυπη με την διάρκεια που απαιτείται από την φυσική συσκευή, καθώς οι μικρές διαφορές που παρατηρούνται θεωρούνται φυσιολογικές και μπορεί να οφείλονται στην γενικότερη χρήση του δικτύου.

Τέλος, παρατηρούμε πως οι λειτουργίες δρομολόγησης δεν επιβαρύνουν την λειτουργία της εικονικής μηχανής, καθώς σε όλα τα διαγράμματα οι διακυμάνσεις στο ποσοστό χρήσης της CPU είναι τόσο μικρές που χαρακτηρίζονται ως φυσιολογικές στο πλαίσιο λειτουργίας της εικονικής μηχανής, ενώ το ποσοστό χρήσης της μνήμης μένει αμετάβλητο.

6

Επίλογος

6.1 Σύνοψη

Στόχος της παρούσας διπλωματικής ήταν η κατασκευή μιας εικονικής CPE συσκευής, δηλαδή μιας συσκευής που αναλαμβάνει την διασύνδεση του χρήστη στο διαδίκτυο και αποτελεί μια διαφορετική προσέγγιση στο υπάρχον μοντέλο. Στο σύγχρονο μοντέλο, οι οικιακοί δρομολογητές αναλαμβάνουν εξ ολοκλήρου το υπολογιστικό βάρος για τη διασύνδεση του χρήστη στο διαδίκτυο, κάτι που αυξάνει το κόστος της παρεχόμενης συσκευής για τους τηλεπικοινωνιακούς παρόχους. Με το νέο μοντέλο, το μεγαλύτερο μέρος των διεργασιών μετατοπίζεται στην μεριά του παρόχου σε εικονικές συσκευές, επιτρέποντας μεταξύ άλλων την μείωση του κόστους της οικιακής συσκευής CPE. Επιπλέον, η εικονικοποίηση των συσκευών CPE, συντελεί στην συνολική μείωση των επενδυτικών (CapEx) και λειτουργικών εξόδων (OpEx). Ωστόσο, τα οφέλη της νέας τεχνολογίας που πραγματεύεται η συγκεκριμένη διπλωματική δεν περιορίζονται στην μείωση των εξόδων των τηλεπικοινωνιακών παρόχων, αλλά και στην παράλληλη αύξηση των εσόδων. Αυτό επιτυγχάνεται μέσω της αυξημένης ταχύτητας κυκλοφορίας των παρεχόμενων υπηρεσιών, ενώ παράλληλα η τεχνολογία αυτή προσφέρει την δυνατότητα για ανάπτυξη καινοτόμων εφαρμογών.

Κατά την παρούσα διπλωματική, αναπτύχθηκε ένα οικιακό vCPE το οποίο είναι σε θέση να διενεργεί απλές λειτουργίες δρομολόγησης με σκοπό την διασύνδεση του οικιακού δικτύου στο διαδίκτυο, παρέχοντας παράλληλα και την απομόνωση του από αυτό. Αρχικά παρουσιάστηκε

θεωρητικά η αρχιτεκτονική του δικτύου που αναπτύχθηκε για τον σκοπό αυτό, ενώ στην συνέχεια αναλύθηκε λεπτομερώς η διαδικασία αναπαραγωγής του, όπως και αυτή της αυτοματοποίησης του. Τέλος, όπως προέκυψε από τα πειράματα αξιολόγησης που εκτελέσθηκαν, η συσκευή που αναπτύχθηκε έχει αντίστοιχες επιδόσεις με τις φυσικές συσκευές που χρησιμοποιούνται στο σύγχρονο μοντέλο.

6.2 Μελλοντικές επεκτάσεις

Η εικονική συσκευή CPE που αναπτύχθηκε στην παρούσα εργασία, όπως αναλύθηκε και στο κεφάλαιο 4, είναι ικανή να εκτελεί απλές λειτουργίες δρομολόγησης για την επικοινωνία του οικιακού δικτύου με το διαδίκτυο. Είναι σαφές πως οι δυνατότητες μιας τέτοιας τεχνολογίας δεν περιορίζονται σε αυτή την λειτουργία, αντιθέτως το άνω όριο της δεν είναι απόλυτα εμφανές. Ενδεικτικά, θα αναφερθούν στην συνέχεια ορισμένες πιθανές επεκτάσεις της παρούσας διπλωματικής.

Στην παρούσα διάταξη έχει τεθεί ως προϋπόθεση η σύνδεση ενός vCPE με ένα μόνο pCPE. Ωστόσο, είναι δυνατή η τροποποίηση αυτής της υπόθεσης με σκοπό την διασύνδεση περισσότερων φυσικών CPE σε κάθε εικονικό μέσω διαφορετικού GRE Tunnel. Αυτή η επιλογή, θα προσέφερε μεγαλύτερη οικονομία παρεχόμενων πόρων από την μεριά των παρόχων, ενώ θα τους έδινε και την επιλογή να τροποποιούν δυναμικά το εικονικό CPE που αντιστοιχεί κάθε φυσικό έτσι ώστε να υπάρχει αποδοτικότερος διαμοιρασμός της κίνησης. Έτσι, θα αποφεύγεται η ανομοιομορφία του βαθμού λειτουργίας των εικονικών CPE και δεν θα υπάρχει ποσοστό αυτών που υπολειτουργεί, ενώ ταυτόχρονα άλλα υπερλειτουργούν.

Επιπλέον, οι υπηρεσίες του vCPE είναι δυνατόν να εμπλουτιστούν, με σκοπό την βελτίωση εμπειρίας του πελάτη. Μάλιστα, οι υπηρεσίες αυτές είναι δυνατό να προορίζονται για συγκεκριμένους υπολογιστές του οικιακού δικτύου. Αυτό είναι εφικτό λόγω του γεγονότος πως το vCPE γνωρίζει τον τελικό παραλήπτη του κάθε πακέτου, καθώς αυτό είναι που εκτελεί το masquerading. Έτσι, αναθέτοντας στατική διεύθυνση IP σε έναν οικιακό υπολογιστή, το vCPE είναι σε θέση να εκτελεί λειτουργίες που απευθύνονται αποκλειστικά σε αυτόν. Ένα απλό παράδειγμα τέτοιας υπηρεσίας, είναι ο εξειδικευμένος γονικός έλεγχος, δηλαδή ο περιορισμός πρόσβασης σε συγκεκριμένες ιστοσελίδες, ο οποίος μάλιστα θα μπορεί να εφαρμόζεται συγκεκριμένες ώρες την μέρα ανάλογα με την περίπτωση. Με αυτόν τον τρόπο, δίνεται η δυνατότητα σε γονείς να απαγορεύουν στα παιδιά τους την πρόσβαση σε ιστοσελίδες κοινωνικής δικτύωσης κατά την διάρκεια του διαβάσματος ή τον πλήρη αποκλεισμό από σελίδες ακατάλληλου περιεχομένου.

Μια ακόμη σημαντική επέκταση του παρόντος θέματος, θα ήταν η πλήρης μετατροπή όλων των λειτουργιών στο επίπεδο του λογισμικού. Με αυτόν τον τρόπο, οι πάροχοι θα έχουν την δυνατότητα να χρησιμοποιήσουν την πληροφορία που μεταφέρεται εντός του πακέτου είτε για

στατιστικούς λόγους είτε για εμπορικούς λόγους, όπως για παράδειγμα εμφύτευση διαφημίσεων στις ιστοσελίδες που επισκέπτεται ο πελάτης, σε ειδικά πεδία που παρέχουν οι κάτοχοι τους, οι οποίες θα είναι προσαρμοσμένες στο προφίλ του πελάτη.

Τέλος, μεγάλο ενδιαφέρον θα έχει η παρακολούθηση της πορείας ανάπτυξης και σταθερότητας του Kubernetes, το οποίο προβλέπεται πως θα μπορούσε να προσφέρει παραπάνω δυνατότητες και βελτιστοποιήσεις στην ανάπτυξη της παρούσας διπλωματικής.

Βιβλιογραφία

[1] R. Chayapathi, S. F. Hassan και P. Shah, «The Journey to Network Functions Virtualization (NFV) Era,» σε *Network Functions Virtualization (NFV) with a Touch of SDN*, Addison-Wesley Professional, 2016, pp. 1-30.

[2] T. W. Burger, «The Advantages of Using Virtualization Technology in the Enterprise,» 5 Μάρτιος 2012. [Ηλεκτρονικό]. Available: https://software.intel.com/en-us/articles/the-advantages-of-using-virtualization-technology-in-the-enterprise.

[3] Hypervisor, «Wikipedia,» [Ηλεκτρονικό]. Available: https://en.wikipedia.org/wiki/Hypervisor. [Πρόσβαση Ιούλιος 2017].

[4] Operating-system-level virtualization, «Wikipedia,» [Ηλεκτρονικό]. Available: https://en.wikipedia.org/wiki/Operating-system-level_virtualization. [Πρόσβαση Ιούνιος 2017].

[5] ETSI, «Network Functions Virtualisation – Introductory White Paper,» 2012.

[6] Customer-premises equipment, «Wikipedia,» [Ηλεκτρονικό]. Available: https://en.wikipedia.org/wiki/Customer-premises_equipment. [Πρόσβαση Ιούνιος 2017].

[7] Juniper Networks, «Understanding How MX Series Router Cloud CPE Services Virtualize Customer Premises Equipment (CPE) Services,» 31 Μάιος 2016. [Ηλεκτρονικό]. Available: https://www.juniper.net/documentation/en_US/junos/information-products/pathway-pages/subscriber-access/ccpe/cloud-cpe.html.

[8] ADVA Optical Networking SE, «Virtual CPE: Gateway to NFV Success,» [Ηλεκτρονικό]. Available: http://www.advaoptical.com/~/media/Resources/Application%20Notes/Virtual%20CPE%20White%20Paper.ashx.

[9] Proxmox Server Solutions GmbH, «Proxmox Wiki,» [Ηλεκτρονικό]. Available: https://pve.proxmox.com/wiki/. [Πρόσβαση Ιούνιος 2017].

[10] Proxmox Virtual Environment, «Wikipedia,» [Ηλεκτρονικό]. Available: https://en.wikipedia.org/wiki/Proxmox_Virtual_Environment. [Πρόσβαση Ιούνιος 2017].

[11] Raspberry Pi, «Wikipedia,» [Ηλεκτρονικό]. Available: https://en.wikipedia.org/wiki/Raspberry_Pi. [Πρόσβαση Ιούλιος 2017].

[12] Ansible, «Ansible Documentation,» [Ηλεκτρονικό]. Available: http://docs.ansible.com/ansible/latest/index.html.

[13] A. Raza, «Puppet vs. Chef vs. Ansible vs. SaltStack,» 27 Σεπτέμβριος 2016. [Ηλεκτρονικό]. Available: http://www.intigua.com/blog/puppet-vs.-chef-vs.-ansible-vs.-saltstack.

[14] P. Venezia, «Review: Puppet vs. Chef vs. Ansible vs. Salt,» 21 Νοεμβρίου 2013. [Ηλεκτρονικό]. Available: http://www.infoworld.com/article/2609482/data-center/data-center-review-puppet-vs-chef-vs-ansible-vs-salt.html.

[15] L. Phifer, «GRE Tunnels vs IPSec VPN – Differences and More,» TechTarget, Μάιος 2009. [Ηλεκτρονικό]. Available: http://www.ebrahma.com/2012/05/gre-tunnels-vs-ipsec-vpn-differences-and-more/.

[16] Proxmox Server Solutions GmbH, «Proxmox VE REST API Documentation,» [Ηλεκτρονικό]. Available: https://pve.proxmox.com/pve-docs/api-viewer/index.html.

[17] ECI, «The Definitive Guide to vCPE,» [Ηλεκτρονικό]. Available: https://www.ecitele.com/media/1703/white-paper-the-definitive-guide-to-vcpe.pdf.

Παράρτημα

pcpe_init.yml

```
---
- hosts: all
  gather_facts: no

  vars:
     - ssh_public_key: "~/.ssh/diploma.pub"

  pre_tasks:
     - name: 'Install Python'
       raw: apt-get install python -y
       become: yes

  tasks:
     - name: Set authorized key took from file
       authorized_key:
         user: "{{ ansible_user }}"
         state: present
         key: "{{ lookup('file', ssh_public_key) }}"
```

pcpe_set.yml

```
---
- hosts: all

  vars:
     - gre_name: "gre_cpe"
     - gre_ip: "10.0.0.2"
     - gateway: "10.0.0.1"

  tasks:
     - name: Check if local IP is defined
       fail: msg="Variable 'local_ip' is not defined"
       when: local_ip is undefined

     - name: Check if remote IP is defined
       fail: msg="Variable 'remote_ip' is not defined"
       when: remote_ip is undefined

     - name: Create GRE tunnel interface
       command: ip tunnel add {{ gre_name }} mode gre remote {{ remote_ip }} local
{{ local_ip }} ttl 255
       become: yes

     - name: Enable GRE tunnel interface
       command: ip link set {{gre_name }} up
       become: yes

     - name: Setting the IP of GRE tunnel interface
       command: ip addr add {{ gre_ip }}/30 dev {{ gre_name }}
       become: yes

     - name: Enabling IP forwarding
       sysctl:
          name: net.ipv4.ip_forward
          value: 1
          sysctl_set: yes
       become: yes

     - name: Changing default gateway
       command: ip route replace default via {{ gateway }}
       become: yes
```

vcpe_create.yml

```
---
- hosts: all
  remote_user: root

  vars:
    - proxmox_port: "8006"
    - node: "pve"
    - hostname: "cpe{{ vmid }}"
    - ssh_public_keys: "{{ lookup('file', '~/.ssh/diploma.pub') }}"
    - ostemplate: "local:vztmpl/ubuntu-16.04-standard_16.04-1_amd64.tar.gz"
    - rootfs: "local-lvm:8"
    - cpulimit: "1"
    - cpuunits: "1024"
    - memory: "512"
    - swap: "512"
    - net0: "bridge=vmbr0,name=eth0,ip=dhcp,ip6=dhcp"

  tasks:
     - name: Check for required variables
       fail: msg="Variable 'vmid' is not defined"
       when: vmid is undefined

     - name: Creating authentication ticket for Proxmox's REST API
       uri:
         url: https://localhost:{{ proxmox_port }}/api2/json/access/ticket
         method: POST
         body: "username=root@pam&password=d1plomA"
         validate_certs: no
       register: auth

     - name: Creating a new LXC
       uri:
         url: https://localhost:{{ proxmox_port }}/api2/json/nodes/{{ node }}/lxc
         method: POST
         headers:
           CSRFPreventionToken: "{{ auth.json.data.CSRFPreventionToken }}"
           Cookie: "PVEAuthCookie={{ auth.json.data.ticket }}"
         body: vmid={{ vmid | urlencode() }}&hostname={{ hostname | urlencode()
}}&password=d1plomA&ssh-public-keys={{ ssh_public_keys | urlencode()
}}&ostemplate={{ ostemplate | urlencode() }}&rootfs={{ rootfs | urlencode()
}}&cpulimit={{ cpulimit | urlencode() }}&cpuunits={{ cpuunits | urlencode()
}}&memory={{ memory | urlencode() }}&swap={{ swap | urlencode() }}&net0={{ net0 |
urlencode() }}
         validate_certs: no
         return_content: yes

     - name: Wait for LXC to be created
       pause:
         seconds: 35
```

```
    - name: Starting the new LXC
      uri:
        url: https://localhost:{{ proxmox_port }}/api2/json/nodes/{{ node
}}/lxc/{{ vmid }}/status/start
        method: POST
        headers:
          CSRFPreventionToken: "{{ auth.json.data.CSRFPreventionToken }}"
          Cookie: "PVEAuthCookie={{ auth.json.data.ticket }}"
        body: skiplock=1
        validate_certs: no

    - name: Wait for LXC to start
      pause:
        seconds: 10

    - name: Stoping the LXC
      uri:
        url: https://localhost:{{ proxmox_port }}/api2/json/nodes/{{ node
}}/lxc/{{ vmid }}/status/stop
        method: POST
        headers:
          CSRFPreventionToken: "{{ auth.json.data.CSRFPreventionToken }}"
          Cookie: "PVEAuthCookie={{ auth.json.data.ticket }}"
        body: skiplock=1
        validate_certs: no

    - name: Wait for LXC to stop
      pause:
        seconds: 5

    - name: Restarting the LXC
      uri:
        url: https://{{ ansible_default_ipv4.address }}:{{ proxmox_port
}}/api2/json/nodes/{{ node }}/lxc/{{ vmid }}/status/start
        method: POST
        headers:
          CSRFPreventionToken: "{{ auth.json.data.CSRFPreventionToken }}"
          Cookie: "PVEAuthCookie={{ auth.json.data.ticket }}"
        body: skiplock=1
        validate_certs: no

    - name: Wait for LXC to start
      pause:
        seconds: 15

    - name: Retrieving IPv4 of the LXC
      shell: lxc-info -n {{ vmid }} -iH | grep -E '[0-9]{1,3}\.[0-9]{1,3}\.[0-
9]{1,3}\.[0-9]{1,3}'
      register: lxc_ip
```

```
    - name: Exporting LAST_LXC to local file
      local_action:
        module: lineinfile
        path: ~/ansible/params.conf
        regexp: '^LAST_LXC='
        line: 'LAST_LXC={{ lxc_ip.stdout }}'
```

vcpe_set.yml

```
---
- hosts: all
  remote_user: root
  gather_facts: no

  vars:
     - gre_name: "gre_cpe"
     - gre_ip: "10.0.0.1"
     - default_if: "eth0"
     - private_net: "192.168.1.0/24"

  pre_tasks:
     - name: 'Update packages'
       raw: apt-get -y update

     - name: 'Install Python'
       raw: apt-get -y install python-simplejson

  tasks:
     - name: Check if local IP is defined
       fail: msg="Variable 'local_ip' is not defined"
       when: local_ip is undefined

     - name: Check if remote IP is defined
       fail: msg="Variable 'remote_ip' is not defined"
       when: remote_ip is undefined

     - name: Create GRE tunnel interface
       command: ip tunnel add {{ gre_name }} mode gre remote {{ remote_ip }} local
{{ local_ip }} ttl 255

     - name: Enable GRE tunnel interface
       command: ip link set {{gre_name }} up

     - name: Setting the IP of GRE tunnel interface
       command: ip addr add {{ gre_ip }}/30 dev {{ gre_name }}

     - name: Enabling IP forwarding
       sysctl: name=net.ipv4.ip_forward value=1 sysctl_set=yes

     - name: Setting IP and Route Tables
       command: "{{ item }}"
       with_items:
         - iptables -t nat -A POSTROUTING -o {{ default_if }} -j MASQUERADE
         - iptables -A FORWARD -i {{ default_if }} -o {{ gre_name }} -m state --
state RELATED,ESTABLISHED -j ACCEPT
         - iptables -A FORWARD -i {{ gre_name }} -o {{ default_if }} -j ACCEPT
         - ip route add {{ private_net }} via {{ gre_ip }}
```

pcpe_unset.yml

```
---
- hosts: all

  vars:
     - gre_name: "gre_cpe"

  tasks:
     - name: Check if new default gateway is defined
       fail: msg="Variable 'gateway' is not defined"
       when: gateway is undefined

     - name: Disable GRE tunnel interface
       command: ip link set {{ gre_name }} down
       become: yes

     - name: Destroy GRE tunnel interface
       command: ip tunnel del {{ gre_name }}
       become: yes

     - name: Disabling IP forwarding
       sysctl:
          name: net.ipv4.ip_forward
          value: 0
          sysctl_set: yes
       become: yes

     - name: Changing default gateway
       command: ip route replace default via {{ gateway }}
       become: yes
```

vcpe_delete.yml

```
---
- hosts: all
  remote_user: root

  vars:
    - proxmox_port: "8006"
    - node: "pve"

  tasks:
     - name: Check for required variables
       fail: msg="Variable 'vmid' is not defined"
       when: vmid is undefined

     - name: Creating authentication ticket for Proxmox's REST API
       uri:
         url: https://localhost:{{ proxmox_port }}/api2/json/access/ticket
         method: POST
         body: "username=root@pam&password=d1plomA"
         validate_certs: no
       register: auth

     - name: Stoping the {{ hostname }}
       uri:
         url: https://localhost:{{ proxmox_port }}/api2/json/nodes/{{ node
}}/lxc/{{ vmid }}/status/stop
         method: POST
         headers:
           CSRFPreventionToken: "{{ auth.json.data.CSRFPreventionToken }}"
           Cookie: "PVEAuthCookie={{ auth.json.data.ticket }}"
         body: skiplock=1
         validate_certs: no

     - name: Wait for LXC to stop
       pause:
         seconds: 5

     - name: Creating a new LXC with hostname {{ hostname }}
       uri:
         url: https://localhost:{{ proxmox_port }}/api2/json/nodes/{{ node
}}/lxc/{{vmid}}
         method: DELETE
         headers:
           CSRFPreventionToken: "{{ auth.json.data.CSRFPreventionToken }}"
           Cookie: "PVEAuthCookie={{ auth.json.data.ticket }}"
         validate_certs: no
```