Πανεπιστήμιο Πατρών

Τμήμα Μηχανικών Ηλεκτρονικών Υπολογιστών και Πληροφορικής

Διπλωματική Εργασία

SDN & Northbound Interfaces

Παναγιώτης Ζ. Μαυρούκας ΑΜ : 5180

Υπεύθυνος Καθηγητής : Μπούρας Ι. Χρήστος

Επιβλέπων : Στάμος Κώστας

Πάτρα 2019

Ευχαριστίες

Αρχικά θα ήθελα να ευχαριστήσω τον υπεύθυνο Καθηγητή της διπλωματικής και Καθηγητή του Τμήματος Μηχανικών Ηλεκτρονικών Υπολογιστών και Πληροφορικής της Πολυτεχνικής σχολής του Πανεπιστημίου Πατρών κ. Χρήστο Ι. Μπούρα, για τη δυνατότητα που μου έδωσε να ασχοληθώ με ένα τόσο ενδιαφέρον θέμα.

Ευχαριστώ ακόμα τον κ. Κωνσταντίνο Στάμο για την συνεργασία και τις πολύ χρήσιμες υποδείξεις του, κατά την εκπόνηση της παρούσας διπλωματικής εργασίας.

Τέλος ευχαριστώ θερμά όλη την οικογένειά μου και ιδιαίτερα τον πατέρα μου για τη βοήθεια και την αμέριστη υποστήριξή του, καθ' όλη τη διάρκεια των σπουδών μου και κατά την εκπόνηση της παρούσας εργασίας.

Πάτρα, Σεπτέμβριος 2019

Παναγιώτης Ζ. Μαυρούκας

Περίληψη

Στη εποχή μας, η επιστήμη των υπολογιστών και οι εφαρμογές της έχουν σημειώσει αλματώδη προόδο σε πολλούς και διαφορετικούς τομείς της κοινωνικής, οικονομικής, πολιτισμικής ζωής, συμβάλλοντας ουσιαστικά στην οικονομική ανάπτυξη και πρόοδο, όπως και στη βελτίωση του βιωτικού επιπέδου των ανθρώπων. Ένας από τους σημαντικότερους τομείς της επιστήμης των υπολογιστών, ο οποίος παρουσιάζει μεγάλη ανάπτυξη και έχει εξαιρετικό οικονομικό ενδιαφέρον, είναι τα υπολογιστικά δίκτυα. Οι ραγδαίες εξελίξεις στις νέες τεχνολογίες και οι συνακόλουθες νέες προκλήσεις που προκύπτουν, αυξάνουν συνεχώς την πολυπλοκότητά των δικτύων, προκειμένου αυτά να ανταποκριθούν στις τεχνολογικές αλλαγές, τις νέες εφαρμογές και τη χρήση νέων πρωτοκόλλων.

Τα παραδοσιακά δίκτυα δεν ήταν πλέον σε θέση να αντιμετωπίσουν τις προκλήσεις της νέας εποχής και εξ' αυτού του λόγου η ανάπτυξη μιας νέας τεχνολογίας στο σχεδιασμό και τη λειτουργία των δικτύων προέκυψε περισσότερο ως ανάγκη, παρά ως απαίτηση σ΄αυτά. Η νέα προσέγγιση στη δημιουργία σύγχρονων δικτύων σχετίζεται με την κεντρική και απλοποιημένη διαχείρισή τους μέσω ευφυούς προγραμματισμού και ως εκ τούτου με την αύξηση της ευελιξίας, της προγραματισιμότητας και της ικανότητάς τους να ικανοποιούν τις αυξημένες ανάγκες των χρηστών τους. Η νέα αυτή προσέγγιση στη αρχιτεκτονική της δόμησης και λειτουργίας των δικτύων, αναπτύχθηκε την τελευταία δεκαετία και είναι γνωστή ως Δικτύωση Καθοριζόμενη από Λογισμικό (Software Defined Networking-SDN).

Η βασική ιδέα της SDN αρχιτεκτονικής είναι διαχωρισμός του επιπέδου ελέγχου του δικτύου από το επίπεδο δεδομένων/προώθησης, με αποτέλεσμα το επίπεδο ελέγου να είναι σε θέση να διαχειρίζεται το σύνολο των δικτυακών συσκευών και να έχει την ευθύνη για το που θα προωθηθεί μια πληροφορία. Το επίπεδο ελέγχου υλοποιείται σε έναν ή περισσότερους ελεγκτές (SDN controllers), οι οποίοι δημιουργούν και ελέγχουν την πλήρη τοπολογία του δικτύου. Το πιο σημαντικό χαρακτηριστικό ενός SDN ελεγκτή είναι η δυνατότητά του να διαχειρίζεται το δίκτυο από ένα και μόνο σημείο, χωρίς να χρειάζεται να έχει πρόσβαση σε όλες τις συσκευές του δικτύου. Αυτό επίσης δίνει τη δυνατότητα εύκολης πρόσβασης και δοκιμών σε νέες υπηρεσίες στην τοπολογία του δικτύου, χωρίς να απαιτείται επαναπρογραμματισμός των συσκευών του και χωρίς να τίθενται σε κίνδυνο οι υπάρχουσες υπηρεσίες. Η επικοινωνία μεταξύ του κεντρικοποιημένου επιπέδου ελέγχου και του επιπέδου δεδομένων πραγματοποιείται μέσω της Νότιας Διεπαφής (SBI) και ενός εξειδικευμένου πρωτοκόλλου (Openflow). Η Βόρεια Διεπαφή (NBI) είναι μια Διεπαφή Προγραμματισμού Εφαρμογής (API) για την επικοινωνία μεταξύ του επιπέδου ελέγχου και της στιβάδας εφαρμογών.

Η παρούσα εργασία επικεντρώθηκε στο οικοσύστημα SDN, με έμφαση στην αρχιτε-κτονική του (Κεφάλαιο 2), στις κύριες πτυχές των ελεγκτών SDN (Κεφάλαιο3), το ρόλο και τις λειτουργίες της Βόρειας Διεπαφής (Κεφάλαιο 4), στην παρουσίαση των τριών πλέον δημοφιλών ελεγκτών ανοιχτού κώδικα του OpenDaylight, του ONOS και του Ryu (Κεφάλαια 5, 6 & 7) και τέλος παρουσιάζεται μια σύγκριση της ωριμότητας και των δυνατοτήτων των τριών προαναφερθέντων ελεγκτών (Κεφάλαιο 8).

Executive Summary

In our time, computer science and its applications have made a tremendous progress in many and different areas of economic, social and cultural life, contributing substantially to economic growth, as well as to improving the standard of living of people. One of the most important domains of computer science, which is highly developed and of great economic interest, is associated with computer networks. With the rapid pace of progress of new technologies and the consequent challenges brought forward, the complexity of networks increased significantly, due to the needs to support new services, application and the use of different protocols.

Traditional networks are no longer capable to cope with the challenges and demands of the new era, thus the need to develop a new technology in the design and operation of networks became more than a demand, it became a necessity. The new approach to create modern networks is about centralization and simplification of their management, through intelligent programming, thereby increasing their flexibility, programmability and ability to meet the increased demands from the users.

This new approach to the network building and operating architecture, emerged and developed over a decade now and it is known as Software Defined Networking (SDN). The main concept of the SDN architecture is the separation of the control plane from the data/forwarding plane, thus the control plane manages all network devices and therefore has full responsibility where all information should be sent. This level (control plane) is implemented on one or more controllers, which create and control the complete network topology. The most important feature of these controllers, is the ability to manage the whole network through a single point, without having access to all network equipment. It also provides easy access and test new services in a network topology, without requiring re-programming of its devices and jeopardizing the existing services. The communication between centralized control plane and data plane is realized through Southbound Interface (SBI) and a specific protocol (Openflow). The Northbound Interface (NBI) is the Application Programming Interface (API) for the communication between control plane and application layer. This most important interface provides abstraction to the network application developers, making possible to implement the desire functionalities without concerns regarding the low level details of the underlying infrastructure.

This thesis focuses of SDN ecosystem, mainly on the network's architecture (Chapter 2), the main aspects of SDN controllers (Chapter 3), analysis of the role and functionalities the Northbound Interfaces (Chapter 4), presentation of the three most popular open-source SDN controllers, namely OpenDayLight, ONOS and Ryu (Chapters 5, 6 & 7), and finally a comparison of the maturity and capabilities of the above mentioned controllers is presented (Chapter 8).

Περιεχόμενα

Λίστα Σχημάτων

Λίστα Πινάκων

Λίστα Ακρωνυμίων

SDN: *Software-Defined Network*

ONF: *Open Network Foundation*

ONOS: *Open Network Operating System*

ODL: *OpenDaylight Project*

API: *Application Programming Interface*

NBI: *Northbound Interface*

SBI: *Southbound Interface*

EBI: *Eastbound Interface*

WBI: *Westbound Interface*

NFV: *Network Function Vitrualization*

LAN: *Local Area Network*

VLAN: *Virtual Local Area Network*

VPN: *Virtual Private Network*

MAN: *Metropolitan Area Network*

WAN: *Wide Area Network*

OSI: *Open Systems Interconnection*

QoS: *Quality of Service*

ACL: *Access Control Lists*

MAC: *Media Access Control*

DAL: *Device & Resource Abstraction Layer*

CAL: *Control Abstraction Layer*

MAL: *Management Abstraction Layer*

NSAL: *Network Service Abstraction Layer*

LLDP: *Link Layer Discovery Protocol*

BGP: *Border Gateway Protocol*

XMPP: *Extensible Messaging & Presence Protocol*

PCE: *Path Computation Element*

MPLS-TP:*Multiprotocol Label Switching - Transport Profile*

IoT: *Internet of Things*

TLS: *Transport Layer Security*

VM: *Virtual Machine*

IXP: *Internet Exchange Point*

OF: *OpenFlow Protocol*

POF: *Protocol Oblivious Forwarding*

OvSDB: *Open vSwitch Database Management Protocol*

SNMP: *Simple Network Management Protocol*

DHCP: *Dynamic Host Configuration Protocol*

REST: *Representational State Transfer*

On.Lab: *Open Networking Lab*

OSGi: *Open Service Gateway Initiative*

IEFT: *Internet Endineering Task Force*

ForCes: *Forwarding & Control Element Separation*

FRP: *Function Reactive Programming*

1. Εισαγωγή

Η δεκαετία του ’80 χαρακτηρίστηκε από την εντυπωσιακή εισβολή των ηλεκτρονικών υπολογιστών στην καθημερινότητά μας, ενώ η δεκαετία του ’90 σημαδεύτηκε από την μεγάλη εξάπλωση των δικτύων υπολογιστών και του Internet, καθώς και εξελιγμένων εφαρμογών διαδικτύου.

Στα χρόνια που ακολούθησαν μετά το έτος 2000, γίναμε μάρτυρες της ανάπτυξης και χρήσης μιας πλειάδας καινοτόμων υπηρεσιών τηλεπληροφορικής, οι οποίες άλλαξαν ριζικά τον τρόπο της ζωής μας. Με τη σύγκλιση και σύνδεση της πληροφορικής και των τηλεπικοινωνιών, αναπτύχθηκαν νέα συστήματα και υπηρεσίες που βελτιώνουν τη λειτουργία του δημόσιου και ιδιωτικού τομέα, αυξάνοντας τις επιδόσεις και την παραγωγικότητά τους, ενισχύοντας παράλληλα και την ανταγωνιστικότητά τους στο περιβάλλον της παγκοσμιοποιημένης οικονομίας.

Οι δικτυακές υποδομές, προσέφεραν το κατάλληλο περιβάλλον για την ανάπτυξη της τηλεφωνίας πάνω σε ψηφιακά δίκτυα, για την παροχή μεγάλου αριθμού τηλεϋπηρεσιών που σχετίζονται με την επεξεργασία και αναμετάδοση πληροφοριών σε διάφορες μορφές (π.χ κείμενο, ήχος, εικόνα, βίντεο, ραδιόφωνο κλπ) και φυσικά τα Δίκτυα Υπολογιστών που προσφέρουν μια ενιαία πλατφόρμα στην οποία αναπτύσσονται καινοτόμες εφαρμογές και λειτουργίες παροχής πρωτοποριακών υπηρεσιών.

Σήμερα πάνω από περίπου 4 δισεκατομμύρια χρήστες σε παγκόσμιο επίπεδο συνδέονται σήμερα στο Διαδίκτυο, μέσω 500 χιλιάδων Αυτόνομων Συστημάτων (Autonomous Systems – ASes) και ο αριθμός αυτός βαίνει συνεχώς αυξανόμενος με ταχείς ρυθμούς [1] & [2]. Κάθε αυτόμονο σύστημα χρειάζεται μιά σειρά από εφαρμογές για τη διαχείριση του/των δικτύου/ων του. Η υλοποίηση αυτών των διαφορετικής εμβέλειας εφαρμογών, καθίσταται εξαιρετικά δύσκολη με τη χρήση παραδοσιακών δικτυακών στοιχείων. Αυτά τα στοιχεία είναι συνήθως βασισμένα σε Ολοκληρωμένα Κυκλώματα Καθορισμένης Εφαρμογής (Application Specific Integrated Circuits-ASICs), που καθορίζονται από τους κατασκευαστές και για να λειτουργήσουν απαιτούν την ενσωμάτωση Λειτουργικών Συστημάτων (Operating Systems-OS) με εκατοντάδες γραμμές κώδικα σε γλώσσες χαμηλού επιπέδου.

Η διαμόρφωση και η εφαρμογή των πολιτικών αυτών των συσκευών, δεν είναι μόνο μια χρονοβόρα διαδικασία, αλλά και δύσκολη. Επιπλέον τα δίκτυα γίνονται δύσκαμπτα, λόγω της ειδικής φύσης των εφαρμογών αυτών που ενσωματώνονται στις συσκευές, περιορίζοντας σημαντικά τις δυνατότητες βελτιστοποίησης της λειτουργίας και διαχείρισης τους.

Εκτός από την εκρηκτική αύξηση των κινητών (ψηφιακών) συσκευών, καθώς και των δεδομένων που διακινούνται από και προς αυτές μέσω διαδικτύου, όπως προαναφέρθηκε, η εικονοποίηση (Virtualization), καθώς και οι αναδυόμενες υπηρεσίες νέφους (Cloud), είναι ορισμένες από τις εφαρμογές, που τα τελευταία χρόνια, έχουν υποχρεώσει τον τομέα των υπολογιστικών δικτύων σε αναστοχασμό, σχετικά με τις μέχρι σήμερα (παραδοσιακές) αρχιτεκτονικές σχεδιασμού τους. Έτσι λοιπόν τα δίκτυα είναι πλέον υποχρεωμένα να εξελίσσονται συνεχώς, με στόχο να ανταποκριθούν τις υψηλότερες απαιτήσεις και ταχύτερες αλλαγές της ζήτησης, γεγονός που οδηγεί σε πολυπλοκότερες και μεγαλύτερες δικτυακές υποδομές, καθώς προστίθενται σ' αυτές μεγαλύτερος αριθμός διακινούμενων πληροφοριών και εξυπηρετούμενων συσκευών.

Λαμβάνοντας υπόψη ότι, περισσότερες έξυπνες κινητές συσκευές (π.χ tablets, notebooks, smartphones) χρησιμοποιούνται σήμερα για πρόσβαση στα δίκτυα, οι απαιτήσεις αυξάνονται προκειμένου όχι μόνο να εξυπηρετείται η διακίνηση (traffic) των δεδομένων αλλά και να διασφαλίζεται η ποιότητα όλων των παραμέτρων επικοινωνίας (π.χ ταχύτητα, ασφάλεια). Επιπλέον, η συνεχώς αυξανόμενη υιοθέτηση υπηρεσιών νέφους (Cloud) από δημόσιους και ιδιωτικούς φορείς, έχουν οδηγήσει σε μια απρόβλεπτα μεγάλη ανάπτυξη των υπηρεσιών αυτών, ενώ η διαχείριση "μεγάλων δεδομένων" (Big Data) απαιτεί μια εκτεταμένη παράλληλη διαδικασία με τη χρήση μεγάλου αριθμού servers, απαιτώντας πρόσθετη υπολογιστική χωρητικότητα. Τελικά η ευελιξία, η υψηλή διαθεσιμότητα, η επεκτασιμότητα και η ασφάλεια είναι τα γνωρίσματα που πρέπει να καθορίζουν τη δομή των νέων δικτύων.

Οι δικτυακές επικοινωνίες σήμερα, συγκροτούνται από διαφορετικά σύνολα πρωτοκόλλων, τα οποία έχουν σχεδιαστεί, ώστε να συνδέουν κόμβους εξ αποστάσεως, με διαφορετικές ταχύτητες, τοπολογίες και προδιαγραφές υπηρεσιών, γεγονός που δημιουργεί βασικούς περιορισμούς στα παραδοσιακά δίκτυα. Έτσι πρέπει να έχουν καθοριστεί αυστηρά οποιεσδήποτε προσθήκες ή αφαιρέσεις

συσκευών, μεταγωγέων (Switches), δρομολογητών (Routers), τείχους προστασίας (Firewall), αυθεντικοποίησης πυλών στα δίκτυα (Network Authentication Portals) κ.α. Επιπλέον πρέπει να ενημερωθούν όλες οι σχετικές παράμετροι των δικτυακών πρωτοκόλλων π.χ. Access Control Lists, Virtual Local Area Networks, QoS κ.α. Το ίδιο ισχύει και στις περιπτώσεις οποιασδήποτε αλλαγής ή ενημέρωσης στη διαμόρφωση που απαιτείται να εφαρμοστεί απευθείας σε κάθε συσκευή του δικτύου. Επιπροσθέτως θα πρέπει να λαμβάνεται υπόψη η δικτυακή τοπολογία, οι κατασκευαστές των μεταγωγέων και οι εκδόσεις του λογισμικού το οποίο χρησιμοποιείται.

Το κύριο χαρακτηριστικό της παραδοσιακής αρχιτεκτονικής δόμησης των δικτύων, είναι η συνδυασμένη (ενιαία) λειτουργία του επιπέδου ελέγχου (Control Plane) με το επίπεδο δεδομένων (Data Plane) του δικτύου. Οι κύριες αρμοδιότητες του επιπέδου ελέγχου αφορούν στη διαμόρφωση των κόμβων του δικτύου, καθώς και στον προγραμματισμό των διαδρομών ροής. Στη συνέχεια, με τον προσδιορισμό των συγκεκριμένων διαδρομών ροής, αυτές προωθούνται στο επίπεδο μετάδοσης των δεδομένων (Data Plane).

Οι μεταγωγείς Ethernet είναι ένα χαρακτηριστικό παράδειγμα δικτυακών κόμβων, το όποιο λειτουργεί με την παραπάνω παραδοσιακή αρχιτεκτονική. Ένας μεταγωγέας Ethernet λειτουργεί πάνω στο επίπεδο σύνδεσης δεδομένων του μοντέλου OSI (Open Systems Interconnection model) και η δημιουργία του βασίζεται τόσο στο επίπεδο ελέγχου όσο και στο επίπεδο δεδομένων. Το σημαντικότερο κομμάτι της λογικής ελέγχου ενός μεταγωγέα σαν αυτόν, είναι ο πίνακας προώθησης (forwarding table), στον οποίο εμπεριέχεται μια λίστα MAC διευθύνσεων που είναι συζευγμένες με τις αντίστοιχες θύρες εξόδου. Στην πλειονότητα των περιπτώσεων οι μεταγωγείς και οι λοιπές δικτυακές συσκευές συνδυάζονται για να σχηματίσουν μια κατανεμημένη δικτυακή αρχιτεκτονική, η οποία συγκριτικά με ένα κεντρικοποιημένο αρχιτεκτονικό μοντέλο παρέχει βελτιωμένη επεκτασιμότητα και εφεδρικότητα (Redundancy). Σε αυτό το σημείο αξίζει να σημειωθεί ότι οι αλλαγές διαμόρφωσης και οι ενημερώσεις πρέπει να γίνοται απευθείας για κάθε συσκευή του δικτύου ξεχωριστά. Αναπόφευκτα λοιπόν δίκτυα μεγάλης κλίμακας/ή πολύ μεγάλης κλίμακας που ακολουθούν αυτή την αρχιτεκτονική σχεδισμού, χρειάζονται αρκετό χρόνο και πόρους για να ενημερωθούν. Επιπροσθέτως, στις παραδοσιακές αρχιτεκτονικές δικτύων οι πόροι και οι έλεγχοι των πολιτικών, ενημερώνονται κάθε φορά που ενημερώνονται και οι απαιτήσεις των

εξωτερικών εφαρμογών. Τέλος οι εξωτερικές εφαρμογές δεν λαμβάνουν καμία πληροφόριση σχετικά με την κατάσταση του δικτύου.

Ο σχεδιασμός ενός δικτύου με την παραδοσιακή αρχιτεκτονική (π.χ Ethernet η οποία είναι η πιο διαδεδομένη μέχρι τις μέρες μας) είναι κατάλληλος για μικρού μεγέθους τοπικά δίκτυα (π.χ δίκτυα μικρών και μεσαίων επιχειρήσεων, οικιακά δίκτυα κλπ), παρ' ότι υπάρχουν θέματα ασφαλειας (πρόσβαση από εξωτερικούς χρήστες), τα οποία όμως μπορούν να αντιμετωπιστούν με τη χρήση VLAN (Virtual Local Area Network), που βοηθά στην προστασία του δικτύου. Υπάρχουν όμως και πολυ πιο εξελιγμένες εφαρμογές σχεδιασμού βασισμένες στα πρότυπα Ethernet, που χρησιμοποιούνται στα Μητροπολιτικά (MAN) και Ευρείας Περιοχής (WAN) Δικτύα για τη σύνδεση συνδρομητών και επιχειρήσεων για τη παροχή ενός μεγαλύτερου πλέγματος υπηρεσιών ή το Διαδίκτυο. Η χρήση της παραδοσιακής αρχιτεκτονικής σήμερα για την δημιουργία δικτύων, παραμένει προσφιλής χάρις στα σημαντικά πλεονεκτήματα που παρουσιάζει, όπως η απλότητα, το χαμηλό κόστος των απαιτούμενων υποδομών και της λειτουργίας τους, η μεγάλη διάρκεια ζωής και η αξιόλογη ευελιξία και επεκτασιμότητα .

Η μέχρι σήμερα χρησιμοποιούμενες (παραδοσιακές) αρχιτεκτονικές στη δόμηση των υπολογιστικών δικτύων, δεν εχουν σχεδιαστεί για να ικανοποιούν τις συνεχώς αυξανόμενες απαιτήσεις και ανάγκες των χρηστών, των επιχειρήσεων και των παρόχων με όρους διακίνησης, διαθεσιμότητας και επεκτασιμότητας. Ήταν λοιπόν προφανής η ανάγκη ανάπτυξης μιας νέας τεχνολογίας και μιας καινούργιας προσέγγισης στην αρχιτεκτονική των δικτύων μέσω πρόγραμματισμού (ευφυή προγραμματιζόμενα δίκτυα), η οποία θα αντιμετώπιζε με επιτυχία τους περιορισμούς του παραδοσιακού σχεδιασμού τους.

Η ιδέα της δημιουργίας ευφυών προγραμματιζόμενων δικτύων, και οι προσπάθειες ξεκινούν από τα μέσα της δεκαετίας του '90 [3] & [4]. Το OPENSIG (Open Signaling – 1995), το GSMP (General Switch Management Protocol, 1999-2002), το Active Networking (1996), το Tempest (1998), το PCE (Path Computation Element – 2004), το 4D project (2004-2011), το NetConf (Network Configuration Protocol), το ForCES (Forwarding and Control Element Separation) που ολοκληρώθηκε το 2015, το Ethane (2007) και το ONF (Open Networking Foundation), αποτέλεσαν προσπάθειες ομάδων εργασίας ειδικών που εμπλέκονταν στην ανάπτυξη προγραμματιζόμενων δικτύων

και των σχετικών προτύπων (πρωτοκόλλων), που διέπουν την λειτουργία των επί μέρους στοιχείων τους. Σύντομη περιγραφή των ανωτέρω παρουσιάζεται στο ***Παράρτημα Ι***.

Μετά το 2011 αρχίζει να αναπτύσσεται η τεχνολογία της νέας αρχιτεκτονικής δικτύων, γνωστής ως Software Defined Networking - SDN (Δικτύωση Καθοριζόμενη από Λογισμικό), η οποία στηριζόμενη στις εμπειρείες και τα αποτελεσματα των εργασιών μέχρι τότε, κεντρικοποιεί και απλουστεύει τη διαχείριση του δικτύου, εισάγοντας τη δυνατότητα προγραμματισμού του και αυξάνοντας με τον τρόπο αυτό την ευελιξία του και τις δυνατότητές του.

Σε ένα SDN δίκτυο, ο μηχανικός ή ο διαχειριστής του είναι σε θέση να διαμορφώνει την κυκλοφορία των δεδομένων, μέσω μιας κεντρικής κονσόλας ελέγχου χωρίς προηγουμένως να απαιτείται να ρυθμίσει οποιονδήποτε μεμονωμένο μεταγωγέα (Switch) του δικτύου. Το κεντρικό σύστημα ελέγχου δικτύου SDN **(**SDN Controller) κατευθύνει τους μεταγωγείς για την παροχή συγκεκριμένων υπηρεσιών δικτύου, όποτε αυτές χρειάζονται, ανεξάρτητα από τις συγκεκριμένες συνδέσεις μεταξύ ενός διακομιστή (Server) και των συσκευών (Devices).

Ένα τυπικό δίκτυο SDN συγκροτείται από τρεις στιβάδες (Layers), τη Στιβάδα των Εφαρμογών (Application Layer), τη Στιβάδα του Ελέγχου (Control Layer) και τη Στιβάδα της Υποδομής (Infrastructure Layer). Κάθε μια από τις ανωτέρω Στιβάδες, είναι συγκεντρωμένες (ομαδοποιημένες) συγκεκριμένες εφαρμογές και λειτουργίες του δικτύου. Ειδικότερα :

- Στη Στιβάδα των Εφαρμογών βρίσκονται όλες οι εφαρμογές του δικτύου, που αξιοποιούν τις πληροφορίες που λαμβάνουν από τις άλλες Στιβάδες για την αποτελεσματικότερη διαχείριση του δικτύου ή και να αναπτύξουν νέες καινοτόμες εφαρμογές για τη βελτίωση της λειτουργίας του και την επέκταση των δυνατοτήτων του.
- Στη Στιβάδα του Ελέχγου βρίσκεται εκγαταστημένη η ευφυής λογική του δικτύου, η οποία διαχειρίζεται κεντρικά όλο το δίκτυο. Αφ’ ενός παρακολουθεί, προγραμματίζει και ελέγχει την υποδομή (Infrastructure) του δικτύου, αφ΄ετέρου δε ενημερώνει και παρακολουθεί τις εφαρμογές που πραγματοποιούνται στην Στιβάδα των Εφαρμογών.

- Τέλος στη Στιβάδα των Υποδομών (Infrastructure Layer) βρίσκονται οι διάφοροι εξοπλισμοί δικτύωσης (μεταγωγείς, δρομολογητές, κλπ) είτε σε φυσική είτε σε εικονική (Virtual) μορφή, που υλοποιούν την κυκλοφορία των δεδομένων.

Οι τρεις ανωτέρω Στιβάδες, επικοινωνούν μέσω του κεντρικού ελεγκτή, χρησιμοποιώντας δύο διεπαφές προγραμματισμού εφαρμογών (Application Programming Interface - API) την προς βορρά (Northbound APIs) και την προς νότο (Southbound APIs), οι οποίες επιτελούν απόλυτα διαφορετικές λειτουργίες μεταξύ των Στιβάδων του δικτύου χρησιμοποιώντας και διαφορετικά πρωτόκολλα. Οι επικοινωνία μεταξύ των Ελεγκτών (servers) που βρίσκονται εγκαταστημένοι στη Στιβάδα Ελέγχου γίνεται με των διεπαφών προς ανατολάς και προς δυσμάς (East/Westbound APIs).

Στα επόμενα Κεφάλαια παρουσιάζονται αναλυτικότερα τα δικτύα SDN και τα επιμέρους δομικά στοιχεία που τα συγκροτούν, καθώς και οι χρησιμοποιούμενες διεπαφές για την επικοινωνία τους. Ιδιαίτερη έμφαση δίνεται στην παρουσίαση των εφαρμογών της προς Βορράν Διεπαφής (Northbound Interface API) και των πλατφορμών ONOS, OpenDaylight και Ryu οι οποίες χρησιμοποιούνται από τους Ελεγκτές των SDN δικτύων.

2. Software-Defined Networks

2.1 Ορισμός και Βασικές Αρχές SDN

Υπάρχουν αρκετοί ορισμοί για την Δικτύωση Καθοριζόμενη από Λογισμικό (Software-Defined Networking) [5] & [6]. Περιγραφικά, η Δικτύωση Καθοριζόμενη από Λογισμικό/Δικτύωση Προγραμματιζόμενης Λογικής, είναι μια αναδυόμενη αρχιτεκτονική στο σχεδιασμό των σύγχρονων δικτύων, η οποία εδράζεται στην αποσύνδεση της λειτουργίας του επιπέδου δεδομένων από αυτή του επιπέδου ελέγχου του δικτύου, το οποίο επίπεδο ελέγχου είναι πλέον προγραμματίσιμο μέσω ενός κεντρικού ελεγκτή (centralized controller). Με το διαχωρισμό αυτό οι μεταγωγείς καθίστανται απλές συσκευές προώθησης, ενώ η λήψη των αποφάσεων μεταφέρεται στον ελεγκτή ο οποίος πλέον διαθέτει την συνολική εικόνα του δικτύου όπως και του προγραμματισμού των αφαιρέσεων. Ο στόχος του συγκεκριμένου σχεδιασμού είναι να καταστήσει τα δίκτυα περισσότερο δυναμικά, ευκολότερα στη διαχείριση τους, οικονομικώς αποδοτικότερα, προσαρμόσιμα και ευέλικτα επιτρέποντας στις επιχειρήσεις και τους παρόχους υπηρεσιών να ανταποκρίνονται γρήγορα στις ταχέως μεταβαλλόμενες επιχειρηματικές απαιτήσεις και ανάγκες[7].Η νέα αυτή αρχιτεκτονική προσέγγιση καθιστά τα SDN δίκτυα :

- Άμεσα προγραμματίσιμα (Directly Programmable), επειδή ο έλεγχός τους αποσυνδέεται από τις λειτουργίες προώθησης.
- Ευέλικτα (Agile), επειδή η αφαίρεση του ελέγχου από τη διαβίβαση επιτρέπει στους διαχειριστές να προσαρμόζουν δυναμικά τη ροή κυκλοφορίας σε όλο το δίκτυο, ώστε να ανταποκρίνονται στις μεταβαλλόμενες ανάγκες.
- Κεντρικά διαχειρίσιμα (Centrally Managed), αφού η νοημοσύνη δικτύου είναι (λογικά) συγκεντρωμένη σε ελεγκτές SDN, οι οποίοι βασίζονται σε λογισμικό και διατηρούν μια συνολική εικόνα του δικτύου, η οποία εμφανίζεται σε εφαρμογές και μηχανισμούς πολιτικών ως ένας και μόνος λογικός μεταγωγέας.
- Προγραμματικά διαμορφώσιμα (Programmaticaly Configured), δεδομένου ότι επιτρέπουν στο διαχειριστή του SDN, να ρυθμίζει, να διασφαλίζει και να βελτιστοποιεί πολύ γρήγορα τους πόρους του δικτύου, μέσω δυναμικών

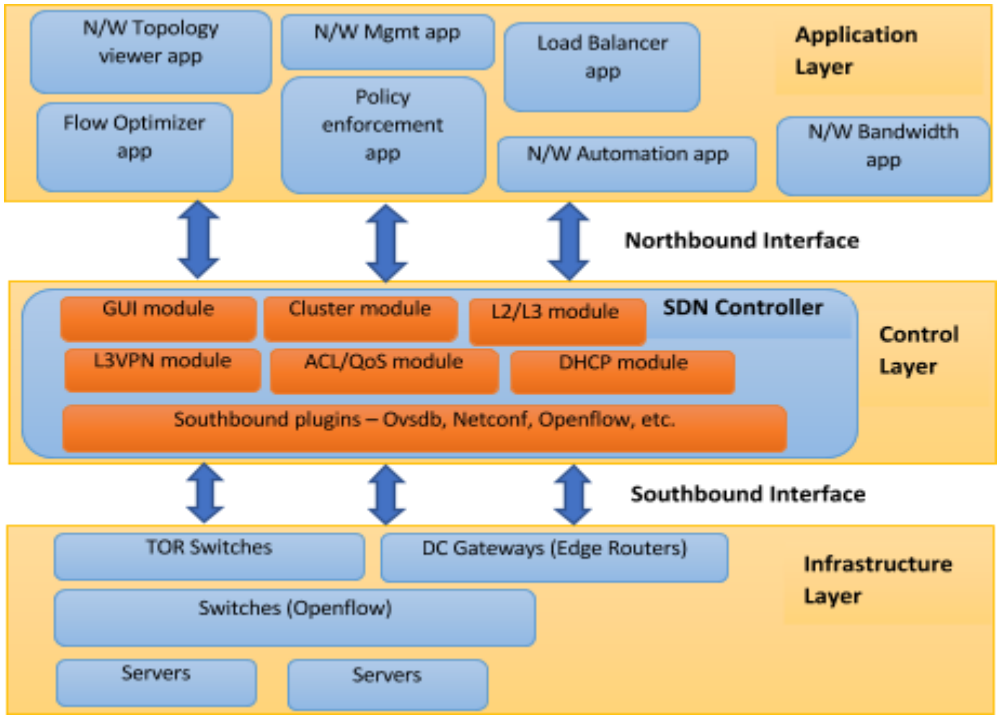
αυτοματοποιημένων προγραμμάτων, που μπορεί ο ίδιος να "γράψει", αφού αυτά δεν εξαρτώνται από ιδιόκτητα λογισμικά.

- Εφαρμόσιμα μέσω της χρήσης ανοιχτών πρωτοκόλλων και ανεξάρτητα από προμηθευτές (Open Standards-Based and Ventor Neutral), γεγονός που απλοποιεί το σχεδιασμό και τη λειτουργία του SDN δικτύου, επειδή οι οδηγίες παρέχονται από τους ελεγκτές SDN, αντί για πολλαπλές συσκευές και πρωτόκολλα συγκεκριμένων προμηθευτών.

2.2 Αρχιτεκτονική SDN Δικτύων

2.2.1 Στιβάδες και Επίπεδα SDN

Όπως ήδη αναφέρθηκε στην Εισαγωγή ένα Δίκτυο Καθοριζόμενο από Λογισμικό (SDN) συγκροτείται από τρεις Στιβάδες (Υποδομών, Ελέγχου και Εφαρμογών) στις οποίες βρίσκονται συγκεντρωμένες όλες οι λειτουργίες σχετικά με τις υπηρεσίες που παρέχει, καθώς και οι εφαρμογές που χρησιμοποιεί για την υλοποίησή τους. Στο *Σχήμα 1* που ακολουθεί παρουσιάζονται διαγραμματικά οι Στιβάδες που συγκροτούν ένα τυπικό δίκτυο SDN. Ειδικότερα:



Σχήμα 1 : Αρχιτεκτονική SDN δικτύων

1. ***Η Στιβάδα των Υποδομών (Infrastructure Layer)***, η οποία περιλαμβάνει τους πάσης φύσεως εξοπλισμούς της δικτύωσης, οι οποίοι συγκροτούν το υποκείμενο δίκτυο για την προώθηση της κυκλοφορίας των δεδομένων. Μπορεί να είναι ένα σύνολο μεταγωγέων (Switches) και δρομολογητών (Routers) σε ένα κέντρο δεδομένων (Data Center). Η Στοιβάδα αυτή μπορεί να έχει φυσική μορφή (συσκευές) πάνω από την οποία μπορεί να εγκατασταθεί η εικονοποίηση (Virtualization) του δικτύου, μέσω της Στιβάδας Ελέγχου του Δικτύου, όπου οι ελεγκτές που βρίσκονται εγκαταστημένη σ' αυτήν και διαχειρίζονται το υποκείμενο δίκτυο.

2. ***Η Στιβάδα Ελέγχου (Control Layer)*** είναι η περιοχή του δικτύου, όπου είναι εγκαταστημένη η ευφυής λογική των Ελεγκτών (Controllers), για την λειτουργία της υποδομής του δικτύου. Σ' αυτή τη στιβάδα εγγράφεται πολύς επιχειρησιακός προγραμματισμός στον Ελεγκτή, για τη συλλογή και διατήρηση διαφόρων τύπων πληροφοριών και ενημερώσεων που αφορούν στο δίκτυο, στοιχείων για την κατάστασή του, την τοπολογία του, στατιστικά δεδομένα και άλλα πολλά. Με δεδομένο τον κεντρικό ρόλο του ελεγκτή στη διαχείριση και λειτουργία του δικτύου, αυτός θα πρέπει να είναι προγραμματισμένος για αντιμετωπίζει πραγματικές καταστάσεις χρήσης του συστήματος, όπως η μεταγωγή, η δρομολόγηση, η L2 VPN, η L3 VPN, κανόνες που αφορούν στη ασφάλεια του τείχους προστασίας, SDN, DHCP, και η δημιουργία συστάδων (Clustering). Από τη στιγμή που τα προαναφερθέντα εφαρμόζονται στο δίκτυο, αυτές οι υπηρεσίες κοινοποιούνται στη Στιβάδα Εφαρμογών, μέσω των αντίστοιχων διεπαφών προγραμματισμού εφαρμογών τους (APIs) που συνήθως στηρίζονται στην αρχιτεκτονική REST (Representational State Tranfer). Έτσι το έργο των διαχειριστών του δικτύου καθίσταται εύκολο σε ότι αφορά στη διαμόρφωση, διαχείριση και παρακολούθηση του υποκείμενου δικτύου, με τη χρήση από τους ελεγκτές εφαρμογών που βρίσκονται στο υπερκείμενο δίκτυο. Με δεδομένο ότι η στιβάδα ελέγχου βρίσκεται στο μέσον της δομής του δικτύου, η επικοινωνία με τις άλλες στιβάδες πραγματοποιείται μέσω δύο τύπων διεπαφής της Βόρειας και της Νότιας. Η Βόρεια Διεπαφή (Northbound Interface - NBI), προορίζεται για την επικοινωνία με την υπερκείμενη (ανώτερη) στιβάδα και εφαρμόζεται μέσω API REST των ελεγκτών του

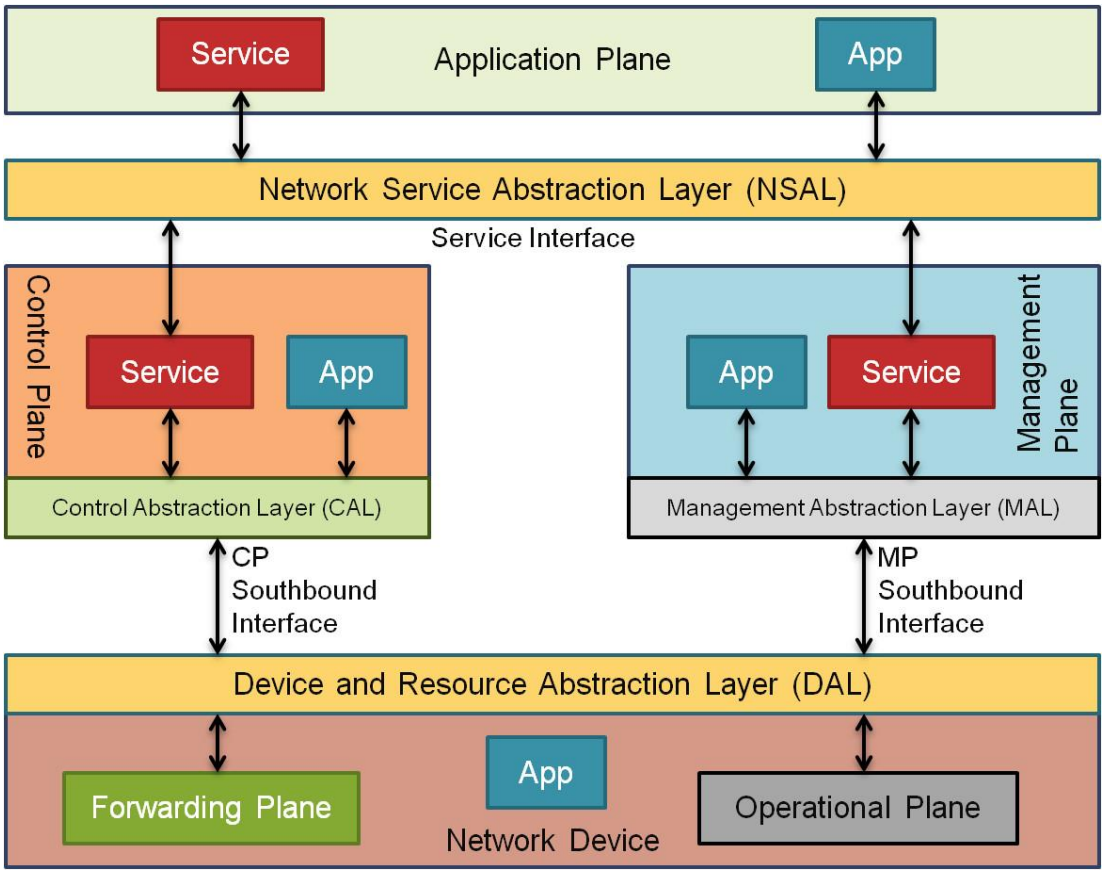
δικτύου. Η Νότια Διεπαφή (Southbound Interface - SBI) αφορά στην επικοινωνία με την υποκείμενη στιβάδα της υποδομής (Infrastructure Layer), η οποία γενικά υλοποιείται μέσω SBI πρωτοκόλλων όπως το OpenFlow, το NetConf, το Ovsdb κλπ. Στη Στιβάδα Ελέγχου είναι εγκαταστημένες δύο επιπλέον διεπαφές, η ανατολική διεπαφή (Eastbound Interface - EBI) και η Δυτική Διεπαφή (Westbound Interface - WBI) οι οποίες υπηρετούν την επικοινωνία μεταξύ των διακομιστών (Servers) που βρίσκονται εγκαταστημένοι στη στιβάδα ελέγχου. Με τις Διεπαφές αυτές θα ασχοληθούμε ιδιαίτερα σε επόμενο κεφάλαιο.

3. ***Η Στιβάδα Εφαρμογών (Application Layer)*** είναι η περιοχή του δικτύου, όπου αναπτύσσονται οι πλέον καινοτόμες εφαρμογές του, αξιοποιώντας όλες τις διαθέσιμες πληροφορίες και στοιχεία του δικτύου, όπως η τοπολογία, η κατάστασή του, τα στατιστικά δεδομένα που συγκεντρώνονται, κλπ, οι οποίες προέρχονται από τη στιβάδα ελέγχου. Ειδικότερα αναπτύσσονται εφαρμογές που σχετίζονται με τον αυτοματισμό του δικτύου, τη διαμόρφωση και τη διαχείρισή του, την παρακολούθηση της λειτουργίας του, την αντιμετώπιση προβλημάτων που εμφανίζονται, τις πολιτικές και την ασφάλειά του. Τέτοιες εφαρμογές μπορούν να παρέχουν καθολικές λύσεις (End-to-end solutions) σε επιχειρήσεις και κέντρα δεδομένων. Οι προμηθευτές των δικτύων, αναπτύσσουν και προωθούν δικές τους λύσεις για τις SDN εφαρμογές όπως βελτιστοποιητές ροής (Flow optimizers), εικονικούς δρομολογητές (Virtual routers), οπτικοποιητές δικτύων (Network visualizers), ισοσταθμιστές φορτίου (Load balancers) κλπ.

Οι παραπάνω Στιβάδες (Layers) συγκροτούνται από σειρά Επιπέδων (Planes) καθένα από τα οποία έχει συγκεκριμένο ρόλο και λειτουργία στο δίκτυο SDN. Το τυποποιητικό έγγραφο RFC7426 [8] προτείνει μια πιο σύγχρονη αρχιτεκτονική προσέγγιση για τα SDN δίκτυα, η οποία επικεντρώνεται στις συσκευές (devices) του δικτύου. Οι συσκευές, είναι απλοί ή σύνθετοι πόροι (resources) και μπορούν να εφαρμόζονται είτε ως λογισμικό είτε ως υλισμικό. Και τούτο γιατί ο όρος "συσκευή" χρησιμοποιείται υπό τη γενική του έννοια, ανεξάρτητα από πραγματικό αντικείμενο/εφαρμογή που μπορεί να έχει φυσική ή εικονική υπόσταση. Η γενική χρήση του όρου πόρος, καθιστά το αρχιτεκτονικό πρότυπο RFC7426 επίσης

εφαρμόσιμο στα πεδία (domains) της εικονοποίησης των δικτυακών λειτουργιών (Network Function Virtualization – NFV) και του ελέγχου Αρχείων Συστήματος (System File Checker – SFC).

Η τυπική απεικόνιση ενός σύγχρονου SDN δικτύου σύμφωνα με το RFC7426, αποτελείται από συγκεκριμένες στοιβάδες αφαίρεσης (Abstraction Layers), από διεπαφές (Interfaces) και διακριτά επίπεδα (Planes). Τα Επίπεδα περιλαμβάνουν μια ομάδα λειτουργιών/διαδικασιών και πόρων που αναφέρονται στην ίδια λειτουργία/διαδικασία, όπως για παράδειγμα το Επίπεδο Ελέγχου ή το Επίπεδο Διαχείρισης. Οι Στιβάδες Αφαίρεσης παραπέμπουν στην αφαίρεση πόρων συγκεκριμένων Επιπέδων, ενώ οι διεπαφές (APIs) αναφέρονται στην επικοινωνία μεταξύ των επιπέδων του δικτύου. Στο *Σχήμα 2* παρουσιάζεται σχηματικά το SDN δίκτυο σύμφωνα με το RFC7426. Σύμφωνα με αυτό, το δίκτυο περιλαμβάνει πέντε (5) επίπεδα :



Σχήμα 2 : Αρχιτεκτονική SDN σύμφωνα με το πρότυπο RFC7426

***Α. Το Επίπεδο Προώθησης (Forwarding Plane/Data Plane)*:** Έχει την ευθύνη για τον χειρισμό πακέτων δεδομένων με βάση τις οδηγίες που λαμβάνονται από το *Επιπεδο Ελέγχου.* Στις δραστηριότητες του Επιπέδου περιλαμβάνονται η χωρίς περιορισμούς μεταβίβαση, απόρριψη, ή και αλλαγή των πακέτων δεδομένων. Παραδείγματα των χρησιμοποιούμενων πόρων προώθησης είναι οι ταξινομητές (classifiers), οι μετρητές (meters) κλπ. Το Επίπεδο Προώθησης συχνά αναφέρεται και ως Επίπεδο Δεδομένων (Data Plane), ή ως Διαδρομή Δεδομένων (Data Path) και εμπεριέχει στοιχεία και λειτουργίες που σχετίζονται με το φιλτράρισμα, τις ρυθμίσεις (buffering), τη διαμέτρηση των πακέτων κ.ά.

***Β. Το Επίπεδο Λειτουργίας (Operational Plane)*:** Είναι υπεύθυνο για τη λειτουργική κατάσταση μιας συσκευής του δικτύου, π.χ αν είναι ενεργή, αν είναι ανενεργή, για το αριθμό των διαθέσιμων θυρών, την κατάσταση κάθε θύρας κ.ο.κ. Ως παραδείγματα επιχειρησιακών πόρων του Επιπέδου, αναφέρονται οι θύρες, η μνήμη.

***Γ. Το Επίπεδο Ελέγχου ή Ελεγκτής (Control Plane or Controller)*:** Αποφασίζει σχετικά με τον τρόπο προώθησης των πακέτων δεδομένων, από μία ή περισσσότερες συσκευές του δικτύου και προωθεί τις συγκεκριμένες αποφάσεις (εντολές) στις συσκευές του δικτύου προς εκτέλεση. Το κεντρικό έργο αυτού του Επιπέδου είναι η τελειοποίηση των πινάκων προώθησης (forwarding tables), οι οποίοι είναι εγκαταστημένοι στο Επίπεδο Προώθησης βάσει της τοπολογίας του δικτύου, ή εξωτερικών αιτημάτων εξυπηρέτησης. Στις κύριες αρμοδιότητές του περιλαμβάνονται ακόμα, ο εντοπισμός τοπολογικών αλλαγών, οι κανόνες εγκατάστασης και προώθησης, η παροχή υπηρεσιών κ.ά. Στο Επίπεδο αυτό είναι εγκαταστημένο και το Λειτουργικό Σύστημα του Δικτύου (Network Operating System - NOS), το οποίο συχνά αναφέρεται και ως Ελεγκτής (Controller), το οποίο εφαρμόζει τους κανόνες λειτουργίας στις συσκευές του υποκείμενου επιπέδου λειτουργίας (Data Plane). Αυτοί οι κανόνες και οι πολιτικές σχεδιάζονται στο επόμενο Επίπεδο (Management Plane) του δικτύου.

***Δ. Το Επίπεδο Διαχείρισης (Management Plane)*:** Έχει την ευθύνη της παρακολούθησης, της διαμόρφωσης και της συντήρησης των συσκευών του δικτύου. Το Επίπεδο Διαχείρισης μπορεί να χρησιμοποιηθεί για τη διαμόρφωση του Επιπέδου Προώθησης, αλλά αυτό συμβάνει εξαιρετικά σπάνια, και γίνεται

μάλλον πολύ γενικά, σε σχέση με το αν αυτή η διαμόρφωση λάμβανε χώρα στο Επίπεδο Ελέγχου. Το Επίπεδο Διαχείρισης συχνά θεωρείται υποσύνολο του Επιπέδου Ελέγχου.

***Ε. Το Επίπεδο Εφαρμογών (Application Plane)*:** Στο Επίπεδο αυτό εντοπίζονται όλες οι υπηρεσίες και οι εφαρμογές που καθορίζουν την συμπεριφορά του δικτύου. Δεν θεωρούνται μέρος του συγκεκριμένου Επιπέδου εφαρμογές που υποστηρίζουν πρωτίστως, ή άμεσα το Επίπεδο Προώθησης (π.χ οι διεργασίες δρομολόγησης εντός του Επιπέδου Ελέγχου).

Στα SDN δίκτυα το Επίπεδο Ελέγχου αποτελεί το κέντρο, το οποίο ελέγχει ένα κατανεμειμένο Επίπεδο Προώθησης και υλοποιείται πλήρως μέσω λογισμικού το οποίο είναι εγκαταστημένο στο υλισμικό (Hardware) του δικτύου. Ως εκ τούτου το SDN μπορεί να χαρακτηριστεί ως προγραμματιζόμενο δίκτυο, όπου το χρησιμοποιούμενο λογισμικό καθορίζει και χειρίζεται τη συμπεριφορά των συσκευών του. Η αποσύνδεση του Επιπέδου Ελέγχου από το Επίπεδο Δεδομένων καθιστά το πρώτο προγραμματίσιμο και ως εκ τούτου του παρέχει τη δυνατότητα αφαίρεσης συσκευών του δικτύου από τα επίπεδα εφαρμογών και υπηρεσιών, τα οποία πλέον μπορεί να διαχειρίζεται ως εικονικές οντότητες.

Η αρχιτεκτονική ενός δικτύου SDN εκτός από τα προαναφερθέντα επίπεδα, περιλαμβάνει και τις ακόλουθες στιβάδες αφαίρεσης (abstraction layers)[1]:

- *Στιβάδα Αφαίρεσης Συσκευής και Πόρου (Device & Resource Abstraction Layer-DAL)*: Αφαιρεί τους πόρους των Επιπέδων Προώθησης και Λειτουργίας μιας συσκευής και τα προωθεί στα Επίπεδα Ελέγχου και Διαχείρισης. Η DAL είναι μια εκ των πλέον σημαντικών στιβάδων αφαίρεσης, αφού οι υπηρεσίες που παρέχουν τα υπόλοιπα επίπεδα εξαρτώνται από τον πλούτο και την ευελιξία της στην περιγραφή των πόρων.

[1] Αφαίρεση (abstraction) : ονομάζεται η διαδικασία απομάκρυνσης/αφαίρεσης χαρακτηριστικών από κάποιο αντικείμενο, προκειμένου να περιγραφεί μόνο με τα βασικά του χαρακτηριστικά. Στον αντικειμενοστραφή προγραμματισμό (Object Oriented Programming) η αφαίρεση είναι μιά εκ των τριών βασικών αρχών του μαζί με την ενθυλάκωση (encapsulation) και τη μεταβίβαση (inheritance). Μέσω της διαδικασίας αφαίρεσης, ο προγραμματιστής, είναι σε θέση να κρύψει/αφαιρέσει δεδομένα που καθορίζουν μια οντότητα, *πλήν των βασικών*, προκειμένου να μειώσει την πολυπλοκότητα και να αυξήσει την αποτελεσματικότητα του προγραμματισμού. Το αποτέλεσμα της αφαίρεσης, είναι μια οντότητα με επιλεγμένα χαρακτηριστικά και συγκεκριμένη συμπεριφορά, για συγκεκριμένη χρήση της αρχικής (πρωτότυπης) οντότητας.
https://computersciencewiki.org/index.php/Abstraction

- *Στιβάδα Αφαίρεσης Ελέγχου (Control Abstraction Layer)*: Αφαιρεί την Νότια Διεπαφή (SBI) του Επιπέδου Ελέγχου και της DAL από τις εφαρμογές και τις υπηρεσίες του Επιπέδου Ελέγχου.
- *Στιβάδα Αφαίρεσης Διαχείρισης (Management Abstraction Layer)*: Αφαιρεί τη Νότια Διεπαφή (SBI) του Επιπέδου Διαχείρισης και της DAL από τις εφαρμογές και τις υπηρεσίες του Επιπέδου Διαχείρισης.
- *Στιβάδα Αφαίρεσης Υπηρεσιών Δικτύου (Network Service Abstraction Layer)*: Παρέχει πρόσβαση στις υπηρεσίες των Επιπέδων Ελέγχου, Διαχείρισης και Εφαρμογών σε άλλες υπηρεσίες και εφαρμογές.

Η επικοινωνία μεταξύ των ανωτέρω περιγραφέντων Επιπέδων (Planes) και Στιβάδων (Layers) του Δικτύου SDN γίνεται μέσω συγκεκριμένων και καλώς καθορισμένων Προγραμματιζόμενων Διεπαφών Εφαρμογών (Application Programmable Interfaces - APIs). Οι διεπαφές αυτές χωρίζονται : στις Νότιες Διεπαφές *(Southbound APIs)* για την επικοινωνία των επιπέδων ελέγχου και δεδομένων, στις Βόρειες Διεπαφές *(Northbound APIs)* για την επικοινωνία των επιπεδών ελέγχου και εφαρμογών και στις περιπτώσεις κατανεμημένων ελεγκτών (Distributed Controllers) στις Ανατολικές/Δυτικές Διεπαφές *(East/Westbound APIs).*

Εκτός απο τη δικτυακή αφαίρεση, η αρχιτεκτονική SDN δίνει τη δυνατότητα χρήσης ενός πλήθους Διεπαφών Προγραμματισμού Εφαρμογών (Application Programming Interfaces - API's), οι οποίες απλοποιούν την εφαρμογή κοινών δικτυακών υπηρεσιών, όπως η δρομολόγηση (Traffic) , η πολυεκπομπή (Multicast), η ασφάλεια (Security), η πρόσβαση ελέγχου, η διαχείριση εύρους ζώνης, η Ποιότητα των Υπηρεσιών (Quality of Services - QoS), η μηχανική της κυκλοφορίας των δεδομένων (Traffic Engineering), η ενεργειακή απόδοση (Energy Efficiency), καθώς και διάφορες μορφές πολιτικών διαχείρισης. Ο διαχωρισμός αυτός δίνει επίσης τη δυνατότητα μεγαλύτερης ευελιξίας, προγραμματισμού, ελευθερίας (ανεξαρτησίας) επιλογής προμηθευτή (vendor-agnostic), καθώς και της οικονομικής αποδοτικότητας και της καινοτομικότητας στη αρχιτεκτονική των δικτύων.

2.2.2 Δομικά Στοιχεία SDN

Με βάση τα όσα έχουν ήδη αναφερθεί για την δικτύωση καθοριζόμενη από λογισμικό, προκύπτει ότι τα βασικά δομικά της στοιχεία είναι οι Μεταγωγείς (Switches), οι Ελεγκτές (Controllers), οι Διεπαφές (Interfaces) και οι Εφαρμογές Δικτύου (Network Applications). Συνοπτικά παρουσιάζονται στη συνέχεια τα βασικά δομικά στοιχεία της αρχιτεκτονικής SDN :

- ***Μεταγωγέας (Switch)***

Πρόκειται για ένα λογισμικό, ή μια δικτυακή συσκευή (υλισμικό), που συνδέει τις συσκευές ενός υπολογιστικού δικτύου, εν προκειμένω του SDN, με τη χρήση μεταγωγής πακέτων δεδομένων, τα οποία λαμβάνει, επεξεργάζεται και προωθεί σε μια συσκευή προορισμού [9]. Οι μεταγωγείς βρίσκονται εγκαταστημένοι και λειτουργούν στο Επίπεδο Προώθησης/Δεδομένων (υποκείμενο δίκτυο SDN). Ο μεταγωγέας πρέπει να υποστηρίζει SDN πρωτόκολλα, τα οποία καθορίζουν τους κανόνες και τις συνθήκες για την επικοινωνία μεταξύ των δικτυακών συσκευών.

Το πλέον γνωστό και ευρέως χρησιμοποιούμενο πρωτόκολλο στους SDN μεταγωγείς είναι το OpenFlow, γι’ αυτό και συχνά οι μεταγωγείς καλούνται και μεταγωγείς OpenFlow [10]. Υπάρχουν επίσης και ορισμένα άλλα πρωτόκολλα τα οποία αναπτύχθηκαν από κατασκευαστές για τους SDN μεταγωγείς, όπως το Cisco OpFlex, το NetConf, το BGP (Border Gateway Protocol), το XMPP (Extensible Messaging and Presence Protocol) κ.ά, τα οποία περιγράφονται συνοπτικά στο επόμενο Κεφάλαιο.

Ανεξάρτητα αν πρόκειται για φυσικό ή εικονικό μεταγωγέα, αυτός εμπεριέχει ενσωματωμένες μόνο λειτουργίες του Επιπέδου Προώθησης/Δεδομένων για την προώθηση των πακέτων. Το Επίπεδο Ελέγχου στο οποίο περιέχεται η υψηλού επιπέδου δρομολόγηση των δεδομένων, όπως ήδη έχει αναφερθεί, έχει αποσυνδεθεί από το υλισμικό των μεταγωγέων και υλοποιείται στον Ελεγκτή SDN, μέσω μιας εφαρμογής που είναι εγκαταστημένη σ’ αυτόν ή κάπου αλλού. Κάθε μεταγωγέας στο SDN δίκτυο είναι προγραμματίσιμος από τον SDN Ελεγκτή, μέσω συγκεκριμένου πρωτοκόλλου και οι επικοινωνίες μεταξύ των συσκευών και των εφαρμογών γίνονται μέσω αυτού.

Ένας μεταγωγέας αποτελείται από θύρες (Ports) και πίνακες (Tables). Τα πακέτα δεδομένων καταφθάνουν και διαβιβάζονται μέσω των θυρών αυτών. Οι πίνακες αποτελούνται από σειρές που περιέχουν ταξινομητές (classifiers) και ενέργειες. Όταν ένας μεταγωγέας λάβει ένα πακέτο δεδομένων, το οποίο δεν ταυτίζεται με κάποια σειρά του πίνακα, αυτό αποστέλλεται στον Ελεγκτή με το ερώτημα σχετικά με την τύχη του συγκεκριμένου πακέτου. Ο Ελεγκτής με τη σειρά του μπορεί να προωθήσει (download) μια σειρά η οποία να περιλαμβάνει τον πρώτο καλύτερο ταξινομητή, ο οποίος ταυτίζεται καλύτερα με το συγκεκριμένο πακέτο δεδομένων, μαζί με τις σχετικές ενέργειες. Αυτές οι τελευταίες διαχειρίζονται στη συνέχεια την συμπεριφορά του πακέτου, που μπορεί να είναι η προώθησή του σε μια θύρα/ή θύρες, η ενθυλάκωση και προώθηση στον Ελεγκτή, η απόρριψη του πακέτου, ή η προώθησή του κανονικά στον αγωγό επεξεργασίας. Η συγκεκριμένη αυτοματοποιημένη διαδικασία καθιστά ευκολότερη την ενσωμάτωση και διαχείριση διαφορετικών εφαρμογών.

Σε ορισμένες περιπτώσεις τίθεται το ζήτημα της χρήσης παλαιού τύπου μεταγωγέων (legacy), οι οποίοι δεν είναι κατάλληλοι για περιβάλλον SDN, προκειμένου να αξιοποιούνται τα ευεργετήματα που παρέχει η δικτύωση καθοριζόμενη από λογισμικό. Έτσι στο ερώτημα αν ένας μεταγωγέας παλαιότερης τεχνολογίας μπορεί να χρησιμοποιηθεί ως SDN μεταγωγέας, η απάντηση είναι ότι αυτό είναι εφικτό μέσω της χρήσης της Διευθυνσιοδοτούμενης Μνήμης Τριπλού Περιεχομένου (Ternary Content Addressable Memory – TCAM) με OpenFlow v.1.0 [11]. Η TCAM επιτρέπει σε ένα πακέτο δεδομένων να αξιολογηθεί με βάση μια ολόκληρη λίστα πρόσβασης σε ένα μοναδικό πίνακα αναζήτησης. Η TCAM ενωματωμένη στους δρομολογητές χρησιμοποιείται για ταχύτερες αναζητήσεις διεύθυνσης, παρέχοντας δυνατότητα ταχύτερης δρομολόγησης. Παρ' όλα αυτά η προαναφερθείσα λύση δεν είναι και η πιο αποδοτική, δεδομένου ότι παρουσιάζει μειονεκτήματα, ως προς την κατανάλωση ενέργειας, το κόστος, το χώρο που καταλαμβάνει σε μνήμη κλπ.

Το κύριο πλεονέκτημα που παρουσιάζουν οι μεταγωγείς SDN, είναι η ευκολία του ελέγχου ροής και η διαμόρφωσή τους. Δεν απαιτείται να γίνουν άμεσες (απευθείας) ρυθμίσεις στον μεταγωγέα (π.χ μετάβαση στις τοποθεσίες μεταγωγέων, είσοδος στη γραμμή εντολών και διαμόρφωση). Όλα αυτά μπορούν να πραγματοποιηθούν με έλεγχο εξ αποστάσεως, με προγραμματισμό πολλαπλών μεταγωγέων μέσω ενός μοναδικού ελεγκτή, με τη χρήση ενός SDN πρωτοκόλλου και της παροχής μιας

εφαρμογής (API) για τους μεταγωγείς. Άλλα πλεονεκτήματα είναι η ευκολία της ρύθμισης φορτίου (load-balancing) ακόμα και σε υψηλούς ρυθμούς μεταβίβασης δεδομένων, καθώς και το γεγονός της απομόνωσης της κίνησης, χωρίς την ανάγκη για Εικονικά Τοπικά Δίκτυα (Vitrual Local Networks - VLANs), αφού οι λειτουργίες του Ελεγκτή δεν επιτρέπουν ορισμένες συνδέσεις.

- ***Ελεγκτής (Controller)***

Ο Ελεγκτής SDN αποτελεί βασικό δομικό στοιχείο στην αρχιετκτονική ενός Δικτύου Καθοριζόμενο από Λογισμικό και είναι εγκατεστημένος στο Επίπεδο Ελέγχου του συστήματος [12]. Ο κύριος ρόλος του είναι η διαχείριση της κίνησης των δεδομένων στα στοιχεία (συσκευές) του υποκείμενου δικτύου χρησιμοποιώντας σειρά οδηγιών, οι οποίες ορίζονται ως ροές (flows).

Ο Ελεγκτής Δικτύου Καθοριζόμενο από Λογισμικό (SDN Controller), είναι ουσιαστικά μια εφαρμογή, η οποία σχετίζεται με τον έλεγχο των ροών, προκειμένου να βελτιώσει τη διαχείριση του δικτύου και να αυξήσει τις επιδόσεις των εφαρμογών του. Η πλατφόρμα του Ελεγκτή SDN, τυπικά "τρέχει" σε ένα διακομιστή (Server), χρησιμοποιώντας πρωτόκολλα για να κατευθύνει τους μεταγωγείς να διαβιβάσουν τα πακέτα δεδομένων σε μια συσκευή προορισμού. Οι Ελεγκτές κατευθύνουν την κίνηση των πακέτων δεδομένων, σύμφωνα με τις πολιτικές προώθησης που ορίζονται από τον χειριστή του δικτύου, μειώνοντας με τον τρόπο αυτό τις διαμορφώσεις σε κάθε μια συσκευή του δικτύου. Έτσι αφαιρώντας το Επίπεδο Ελέγχου από το υλισμικό του υποκείμενου δικτύου (Επίπεδο Προώθησης/Δεδομένων) και υλοποιώντας το ως λογισμικό, ο κεντρικοποιημένος ελεγκτής διευκολύνει την αυτοματοποιημένη λειτουργία του δικτύου και καθιστά ευκολότερη την ενοποίηση και διαχείριση των επιχειρησιακών εφαρμογών.

Οι ελεγκτές μπορεί να παρουσιάζουν ορισμένες διαφοροποιημένες ιδιότητες μεταξύ τους, όμως ο πυρήνας των βασικών λειτουργικών χαρακτηριστικών τους είναι παρόμοιος, όπως π.χ. η συλλογή πληροφοριών και παρακολούθηση σχετικά με την τοπολογία του δικτύου, η συγκέντρωση και επεξεργασία στατιστικών δεδομένων, οι κοινοποιήσεις, η απογραφή και η διαχείριση των συσκευών του (υποκείμενου) δικτύου κ.ά. Για την εκτέλεση των προαναφερθεισών λειτουργιών/εφαρμογών , κάθε ελεγκτής χρησιμοποιεί διεπαφές (Interfaces), για να συνδεθεί με τα Επίπεδα του δικτύου. Η Νότια Διεπαφή (Southbound Interface), χρησιμοποιείται για την

επικοινωνία και την υλοποίηση της προώθησης (forwarding) των δεδομένων που βρίσκονται στο υποκείμενο δίκτυο και η Βόρεια Διεπαφή (Northbound Interface) για την επικοινωνία με το υπερκείμενο δίκτυο των εφαρμογών (applications). Για τις περιπτώσεις επικοινωνίας εφαρμογών στο Επίπεδο του Ελέγχου, χρησιμοποιούνται η ανατολική/δυτική διεπαφή (Eastbound/Westbound).

Οι Ελεγκτές μπορούν να κατηγοριοποιηθούν με βάση συγκεκριμένα χαρακτηριστικά της λειτουργίας τους, ένα στοιχείο κλειδί που καθορίζει την αρχιτεκτονική του τρόπου υλοποίησης της αποστολής τους [13]. Έτσι υπάρχουν οι κεντροποιημένοι ελεγκτές (centralized controllers), που αποτελούνται από μια μοναδική οντότητα που έχει την ευθύνη της διαχείρισης των συσκευών του δικτύου. Οι κατανεμημένοι ελεγκτές (distributed controllers), συγκροτούνται από ένα αριθμό οντοτήτων οι οποίες συνεργάζονται μεταξύ τους για τη διαχείριση των στοιχείων του υποκείμενου δικτύου. Οι διεπαφές προς ανατολάς και προς δυσμάς (Eastbound & Westbound) είναι προσδιοριστικοί παράγοντες στη λειτουργία των κατανεμημένων ελεγκτών. Ορισμένοι Ελεγκτές όπως οι NOX & POX έχουν σχεδιαστεί με κεντροποιημένη αρχιτεκτονική δομή, ενώ άλλοι όπως ο FloodLight, ο OpenDaylight είναι κατανεμημένης μορφής. Στον Πίνακα Ι παρουσιάζεται μια σύνοψη σχετικά με τους Ελεγκτές SDN.

ΠΙΝΑΚΑΣ Ι

Συνοπτική Παρουσίαση των Ελεγκτών SDN

Ελεγκτής	Γλώσσα Προγρ/σμού	Αρχιτεκτονική (*)	SBIs	NBIs	EWBIs
NOX	C++	C	OpenFlow 1.0	ad-hoc	-
POX	Python	C	OpenFlow 1.0	ad-hoc	-
FloodLight	Java	C	OpenFlow 1.0-1.3	REST,JAVA, RPC Quantem	-
OpenDaylight	Java	D	OpenFlow 1.0-1.3 OvSDB, SNMP	REST, RESTCONF, XMPP, NETCONF	SDNi
Kandoo	C, C++, Python	D	OpenFlow 1.0-1.2	JAVA, RPC	Messaging Channel
DISCO	Java	D	OpenFlow 1.0	REST	AMQP
Beacon	Java	C	OpenFlow 1.0	ad-hoc	-
ONOS	Java	D	OpenFlow 1.0-1.3	REST, Neutron	Raft
Onix	C++	D	OpenFlow 1.0 OvSDB	Onix API	Zookeeper
PANE	Haskell	D	OpenFlow 1.0	PANE API	Zookeeper
HyperFlow	C++	D	OpenFlow 1.0	-	WheelFS

(*) C=Centralized Controller = Κεντροποιημένος Ελεγκτής , D=Distributed Controller =Κατανεμημένος Ελεγκτής.

Οι περισσότεροι ελεγκτές υποστηρίζουν το πρωτόκολλο OpenFlow για τη νότια διεπαφή, όμως και το OpenDaylight υποστηρίζει ένα ευρύ φάσμα νότιων διεπαφών (π.χ OpenFlow, OvSDB, SNMP, NetConf).

- ***Διεπαφές (Interfaces)***

Ως διαπεφή στην επιστήμη των υπολογιστών ορίζεται το κοινό όριο δια μέσου του οποίου δύο ή περισσότερα χωριστά στοιχεία ενός υπολογιστικού συστήματος ανταλλάσουν πληροφορίες. Η ανταλλαγή αυτή μπορεί να είναι μεταξύ λογισμικού, υλισμικού, περιφερειακών συσκευών, ανθρώπων, ή συνδυασμού των προηγουμένων.

Σε ένα δίκτυο καθοριζόμενο από λογισμικό (SDN), όπου οι λειτουργίες ελέγχου έχουν διαχωριστεί από τις λειτουργίες προώθησης και εφαρμογών, ο ρόλος των διεπαφών είναι η διασύνδεση των επιπέδων προώθησης και εφαρμογών με το επίπεδο ελέγχου που αποτελεί και το κεντρικό σημείο αναφοράς για τη λειτουργία του δικτύου. Στα SDN δίκτυα οι διεπαφές αποτελούν δομικό στοιχείο τους και περιλαμβάνουν, τη Νότια Διεπαφή (Southbound Interface - SBI) για τη διασύνδεση του Ελεγκή με τη προώθηση των δεδομένων, τη Βόρεια Διεπαφή (Northbound Interface - NBI) για τη διασύνδεση του ελεγκτή με τις εφαρμογές και την Ανατολική/Δυτική Διεπαφή (East/Westbound Interface - E/WI), στις περιπτώσεις που χρησιμοποιούνται κατανεμημένοι ελεγκτές στο πεδίο ελέχγου.

Νότια Διεπαφή (Southbound Interface - SBI)

Τα Southbound API's διευκολύνουν τον αποτελεσματικό έλεγχο του υποκείμενου δικτύου και επιτρέπουν στον ελεγκτή να κάνει δυναμικές αλλαγές σε πραγματικό χρόνο, σύμφωνα με τις απαιτήσεις και ανάγκες που υπάρχουν, για την προώθηση των δεδομένων. Το κύριο έργο των συγκεκριμένων διεπαφών είναι η προώθηση των κοινοποιήσεων που διαβιβάζονται από τον Ελεγκτή στις συσκευές του επιπέδου δεδομένων και η παροχή στοιχείων και δεδομένων από τις συσκευές αυτές προς τον Ελεγκτή του δικτύου. Τα δεδομένα αυτά σχετίζονται με την τοπολογία του υποκείμενου δικτύου, τον καθορισμό των ροών, και την υλοποίηση των αιτημάτων τα οποία αποστέλλονται από το επίπεδο της διαχείρισης.

Μερικές από τις πιο γνωστές southbound διεπαφές είναι οι: OpenFlow (OF) [14], ForCES (FORwarding & Control Element Separation) [15], Open virtual Switch Database (OvSDB) [16], Protocol Oblivious Forwarding (POF) [17], OpFlex, [18], OpenState [19] κλπ.

Το OpenFlow είναι η ευρύτερα χρησιμοποιούμενη διεπαφή (API) και θεωρείται πλέον μια πρότυπη Southbound Interface (SBI) στα SDN Δίκτυα. Οι περισσότερες προτάσεις αποτελούν είτε επεκτάσεις της συγκεκριμένης διεπαφής (OF) ή με κάποιο τρόπο συνδέονται με αυτή. Δύο προτάσεις για SBIs, όπως για παράδειγμα η OvSDB και OFConfig λειτουργούν συνεργατικά με την OpenFlow, προκειμένου να παρέχουν μεγαλύτερες δυνατότητες διαμόρφωσης. Οι υπόλοιπες διεπαφές OpFlex, ForCES και NetConf είναι εντελώς ανεξάρτητες από το OpenFlow. Στη συνέχεια θα γίνει συνοπτική αναφορά στη βιβλιογραφία που αναφέρεται στο OpenFlow, όπως και στις SBIs που αφορούν στα δικτύων αισθητήρων (Sensor Networks) και στο Διαδίκτυο των Πραγμάτων (Internet of Things - IoT).

OpenFlow Protocol

Το πρωτόκολλο OpenFlow το οποίο δημιουργήθηκε και καθιερώθηκε από τον ONF (Open Network Foundation) υποστηρίζει την αλληλεπίδραση ελεγκτή-μεταγωγέα, δεδομένου ότι καθορίζει την επικοινωνία μεταξύ του υλισμικού μεταγωγέων και του ελεγκτή δικτύου για την εκτέλεση εργασιών προώθησης δεδομένων. Σε αυτό το σημείο αξίζει να σημειωθεί ότι παρότι μία από τις βασικές ιδέες του SDN είναι η αποφυγή της αξιοποίησης ενός και μόνο προμηθευτή και της εξάρτησης από ένα πρωτόκολλο, το OpenFlow δεν εξυπηρετεί αυτό το σκοπό.

Ο OpenFlow μεταγωγέας είναι το βασικότερο στοιχείο προώθησης που είναι προσβάσιμο μέσω του πρωτοκόλλου αλλά και της διεπαφής OpenFlow. Αν και με μία πρώτη εντύπωση αυτή η ρύθμιση φαίνεται να απλοποιεί το υλισμικό των μεταγωγέων, οι flow-based SDN αρχιτεκτονικές όπως το OpenFlow μπορεί να χρειάζονται πρόσθετες καταχωρήσεις στους πίνακες προώθησης (forwarding tables), επιπλέον χώρο buffer, μετρητές στατιστικών κ.τ.λ. Ένας μεταγωγέας OpenFlow αποτελείται από ένα πίνακα ροής με τον οποίο εκτελούνται οι αναζητήσεις και προωθήσεις των πακέτων. Κάθε πίνακας ροής περιέχει μια σειρά από καταχωρήσεις ροής, με αυτές να αποτελούνται από πεδία επικεφαλίδων ή πεδία αντιστοίχισης αλλά και από μετρητές που χρησιμοποιούνται για τη συλλογή στατιστικών, όπως για

παράδειγμα το πλήθος των ληφθέντων πακέτων, το πλήθος των bytes τους, τη συνολική διάρκεια της ροής (flow) κ.α. Όταν ένα πακέτο καταφθάνει στον μεταγωγέα τα πεδία των επικεφαλίδων εξάγονται και αντιστοιχίζονται με τις καταχωρήσεις ροής που είναι εγκατεστημένες σε αυτόν. Εάν υπάρχει αντιστοίχιση τότε εκτελείται η κατάλληλη ενέργεια στην συγκεκριμένη ροή. Στην περίπτωση που δεν υπάρχει αντιστοίχιση με τον πίνακα ροής εκτελούνται κάποιες προεπιλεγμένες ενέργειες για τα πακέτα.

Ένα ελεγκτής χρησιμοποιεί ένα ασφαλές κανάλι για να επικοινωνήσει με κάποιο μεταγωγέα και τα OpenFlow πακέτα στέλνονται μέσω αυτού του καναλιού. Για λόγους ασφαλείας η έκδοση OpenFlow 1.3.0 παρέχει προαιρετική υποστήριξη για την κρυπτογραφημένη TLS (Transport Layer Security) επικοινωνία και ανταλλαγή πιστοποιητικών μεταξύ των ελεγκτών και των μεταγωγέων. Οι κατηγορίες μηνυμάτων που υποστηρίζει το OpenFlow πρωτόκολλο είναι οι παρακάτω :

- Controller-to-switch : Η κατηγορία αυτή αναφέρεται στα μηνύματα που αποστέλλονται από τον ελεγκτή και τα οποία σε αρκετές περιπτώσεις απαιτούν απόκριση (response) από τον αντίστοιχο μεταγωγέα. Η συγκεκριμένη κατηγορία μηνυμάτων καθιστά ικανό τον ελεγκτή στο να διαχειριστεί τις λογικές καταστάσεις (logical state) του μεταγωγέα συμπεριλαμβανόμενης και της διαμόρφωσης αυτού, τις πληροφορίες ροής του και τις ομαδοποιημένες καταχωρήσεις πινάκων.
- Symmetric : Στην κατηγορία αυτή συμπεριλαμβάνονται τα μηνύματα που αποστέλονται από κάποιον ελεγκτή ή μεταγωγέα. Εδώ συμπεριλαμβάνονται τα τυπικά μηνύματα που ανταλλάσονται μεταξύ ενός ελεγκτή και ενός μεταγωγέα όταν πρωτοκαθιερώνεται μία σύνδεση μεταξύ τους. Συμπεριλαμβάνονται επίσης τα μηνύματα echo και reply με τα οποία μπορεί να υπολογιστεί η καθυστέρηση (latency), το εύρος ζώνης (bandwidth) μιας σύνδεσης ελεγκτή-μεταγωγέα.
- Asynchronous : Η κατηγορία αυτή περιλαμβάνει διάφορα ασύχρονα μηνύματα κατάστασης (status messeges), τα οποία αποστέλονται στον εκάστοτε ελεγκτή.

Βόρεια Διεπαφή (Northbound Interface - NBI)

Οι Βόρειες Διεπαφές Προγραμματισμού Εφαρμογών (Northbound APIs) αναφέρονται στην επικοινωνία μεταξύ του επιπέδου επιχειρησιακών εφαρμογών (Application Plane) και του επιπέδου ελέγχου (Control Plane). Μέχρι σήμερα θεωρείται το λιγότερο ερευνημένο και τυποποιημένο κομμάτι του SDN. Τα Northbound API's χρησιμοποιούνται για την υλοποίηση και ανάπτυξη εφαρμογών διαχείρισης και επίβλεψης δικτύων, οι οποίες είναι ανεξάρτητες από τους προμηθευτές. Ένα μεγάλο πλεονέκτημα των Northbound API's είναι το γεγονός ότι είναι εύκολα τροποποιήσιμα με τη χρήση γλωσσών προγραμματισμού υψηλού επιπέδου όπως Java, Python, C++ κτλ. Σε αυτό το σημείο επισημαίνεται ότι μέχρι στιγμής δεν έχει θεσπιστεί κάποιο πρωτόκολλο ως πρότυπο για το Northbound κομμάτι των δικτύων. Οι τρέχουσες APIs υλοποιούνται για συγκεκριμένες εφαρμογές με γνώμονα τη λογική των Ad-hoc δικτύων. Αυτό συμβαίνει διότι τα συστήματα εξωτερικής διαχείρισης και οι δικτυακές υπηρεσίες επιθυμούν την εξαγωγή πληροφοριών σχετικά με τα υποκείμενα δίκτυα ή τον έλεγχο μιας πτυχής της συμπεριφοράς/πολιτικής του δικτύου.

Το Northbound Interface επίσης ορίζεται αποκλειστικά σε λογισμικό, ενώ οι αλληλεπιδράσεις μεταξύ ελεγκτών-μεταγωγέων βασίζονται σε υλοποιήσεις υλισμικού. Παρότι μέχρι τώρα έχουν σχεδιαστεί αρκετοί ελεγκτές, οι διεπαφές των εφαρμογών τους βρίσκονται ακόμα σε πρώιμα στάδια, καθώς είναι ανεξάρτητοι και ασύμβατοι μεταξύ τους. Με βάση τα παραπάνω οι SDN εφαρμογές θα συνεχίσουν να αναπτύσσονται με βάση τα Ad-hoc δίκτυα μέχρις ότου να καθοριστεί ένα συγκεκριμένο πρότυπο για τα Northbound Interfaces. Αναλυτικότερα οι Βόρειες Διεπαφές (NBIs) παρουσιάζονται στο Κεφάλαιο 4.

Ανατολική/Δυτική Διεπαφή (East/Westbound Interface – W/EBI)

Ο κεντρικός έλεγχος επί του δικτύου είναι χαρακτηριστικό του SDN, με τον ελεγκτή να έχει τη δυνατότητα παρακολούθησης ενός περιορισμένου αριθμού μεταγωγέων. Εξαιτίας της εκθετικής αύξησης των συσκευών στο υποκείμενο δίκτυο, αλλά και κατασκευής μεγάλης κλίμακας δικτύων, οι καταναμημένοι ελεγκτές (Distributed Controllers) αποτελούν πλέον βασικό προαπαιτούμενο για την απρόσκοπτη λειτουργία τους. Σε μιας τέτοιας μορφής κατανομή, ο κάθε ελεγκτής έχει το δικό του πεδίο συσκευών στο υποκείμενο δίκτυο. Ακόμα, οι ελεγκτές έχουν ανάγκη να

μοιράζονται τις πληροφορίες για τα σχετικά πεδία τους, προκειμένου να έχουν τη συνολική εικόνα του δικτύου. Οι προς ανατολάς και προς δυσμάς διεπαφές (East/WestBound Interfaces) χρησιμοποιούνται για την εισαγωγή και εξαγωγή πληροφοριών μεταξύ των κατανεμημένων ελεγκτών για τους λόγους που προαναφέρθηκαν. Οι συγκεκριμένες διεπαφές επιτρέπουν την επικοινωνία μεταξύ των υπαρχουσών συσκευών του δικτύου (π.χ δρομολογητές κλπ) με τους ελεγκτές.

- ***Δικτυακές Εφαρμογές***

Οι δικτυακές εφαρμογές αποτελούν προγράμματα που επικοινωνούν με τον SDN ελεγκτή μέσω διαφόρων API's. Οι εφαρμογές αυτές μπορούν να κατασκευάσουν μια αφηρημένη οπτική του δικτύου με τη συλλογή πληροφοριών από τον ελεγκτή με σκοπό τη λήψη αποφάσεων. Ακόμη αυτές οι εφαρμογές μπορούν να περιλαμβάνουν εφαρμογές διαχείρισης, ανάλυσης και επιχειρηματικότητας του δικτύου. Παραδείγματος χάριν μια εφαρμογή ανάλυσης μπορεί να κατασκευασθεί με τέτοιο τρόπο ώστε να αναγνωρίζει την οποιαδήποτε δικτυακή δραστηριότητα με σκοπό την αναβάθμιση της ασφάλειας αυτής. Σε ένα SDN δίκτυο οι πάροχοι υπηρεσιών μπορούν να δημιουργήσουν διάφορες εφαρμογές που να εστιάζουν στην ελάττωση κόστους, στην βελτίωση της εμπειρίας του πελάτη κ.α. Όπως είναι αναμενόμενο δεν είναι αποκλειστικά όλες οι SDN εφαρμογές καινούργιες, καθώς ένα μεγάλο πλήθος από αυτές αντιγράφουν και βελτιώνουν τις ήδη υπάρχουσες εφαρμογές που τρέχουν επί του παρόντος στους δρομολογητές και στους μεταγωγείς. Οι εφαρμογές δρομολόγησης φερειπείν θα λειτουργήσουν βασιζόμενες στις γνώσεις και τα χαρακτηριστικά του επιπέδου εφαρμογής (application level). Επιπλέον όταν χρησιμοποιούνται εφαρμογές SDN, η δρομολόγηση περιεχομένου μπορεί να σχεδιαστεί έτσι ώστε να εφαρμόζονται έλεγχοι διαθεσιμότητας υπηρεσιών πριν την τροφοδότηση της ροής στους μεταγωγείς του δικτύου. Τέλος ενώ θεωρείται ότι η τεχνολογία SDN εστιάζει κυρίως στα control και data plane, οι δικτακές εφαρμογές είναι επίσης υπεύθυνες για την παροχή βελτιώσεων τόσο στους φορείς εκμετάλλευσης, όσο και στους χρήστες.

2.3 Πεδία Εφαρμογής SDN

Η τεχνολογία SDN υπόσχεται πολλές νέες δυνατότητες σε διάφορους τομείς που σχετίζονται με τη δικτύωση των υπολογιστών. Ο διαχώρισμος του επιπέδου ελέγχου από το επίπεδο δεδομένων, καθώς και άλλες αρχές του SDN θέτουν τις βάσεις για την χρήση αυτού σε πολλές διαφορετικές περιπτώσεις αξιοποιώντας όλα τα οφέλη που προσφέρει. Σε αυτό το υποκεφάλαιο παρουσιάζονται περιπτώσεις στις οποίες έχουν προταθεί λύσεις ή έχουν υλοποιηθεί ήδη, που βασίζονται στα SDN δίκτυα.

- **Κέντρα Δεδομένων (Data Centers)**

Τα Κέντρα Δεδομένων εξελίσσονται συνεχώς με εντυπωσιακούς ρυθμούς τα τελευταία χρόνια, σε μια προσπάθεια να ανταποκριθούν στις συνεχώς αυξανόμενες απαιτήσεις και ταχύτατες αλλαγές της αγοράς (επιχειρήσεις, οργανισμοί, μεγάλοι χρήστες κλπ). Τα κλασσικά κέντρα δεδομένων χρησιμοποιούν δρομολογητές (routers) για τη σύνδεση του πυρήνα τους με το διαδίκτυο, καθώς και μεταγωγείς για τη συνδεσή τους με εξυπηρετητές (servers) και άλλους μεταγωγείς. Η προσεκτική διαχείριση της κυκλοφορίας των δεδομένων και η εφαρμογή των πολιτικών που πρέπει να εφαρμοστούν είναι κρίσιμα ζητούμενα για τα κέντρα δεδομένων μεγάλης κλίμακας, ειδικά όταν μια ανατρεπτική υπηρεσία (disruptive service) ή επιπλέον καθυστέρηση μπορεί να οδηγήσει είτε σε μαζική αύξηση της παραγωγικότητας του Κέντρου, είτε σε απώλεια κερδών [20]. Η διαχείριση των κέντρων δεδομένων μεταφέρεται όλο και περισσότερο σε εικονικές μηχανές (Virtual Machines) με βάση τις αλλαγές των συμπεριφορών στην αγορά και των προτύπων διαμοίρασης των πληροφοριών. Τα σύγχρονα κέντρα δεδομένων έχουν πολλές σχεδιαστικές απαιτήσεις, όπως την ευκολία μετάβασης τους σε εικονικές μηχανές, την αποτελεσματική επικοινωνία μεταξύ των εξυπηρετητών, καθώς και την ελαχιστοποίηση της διαμόρφωσης των μεταγωγέων και των υποδοχέων (hosts) [21]. Λόγω προκλήσεων και της πολυπλοκότητας που έχει να αντιμετωπίσει η μηχανική των συγκεκριμένων δικτύων μεγάλης κλίμακας, συχνά πρέπει να είναι προετοιμασμένα να ικανοποιήσουν υψηλή ζήτηση. Εξ αυτού του λόγου τα Κέντρα Δεδομένων λειτουργούν συνήθως κάτω από τις πραγματικές τους δυνατότητες, προκειμένου να είναι σε θέση να αντιμετωπίσουν υπερβάλλουσα ζήτηση εργασιών (φορτίων), όποτε αυτό απαιτηθεί. Για τους παραπάνω λόγους συνιστάται η χρήση αρχιτεκτονικής SDN στη δόμησή τους. Επιπλέον ένας άλλος κρίσιμος παράγοντας στη λειτουργία τους, που αποκτά συνεχώς

και μεγαλύτερη σημασία, είναι η κατανάλωση ενέργειας του δικτύου, το κόστος της οποίας δεν είναι καθόλου ευκαταφρόνητο στα μεγάλης κλίμακας Κέντρα [22]. Από τα σημαντικότερα λοιπόν ζητήματα σε λειτουργικό και ερευνητικό επίπεδο των Κέντρων Δεδομένων είναι, η βελτίωση των διακομιστών και η ψύξη τους μέσω καλύτερης διαχείρισης του υλισμικού και του λογισμικού, όπως και η κατανάλωση ενέργειας της υποδομής του δικτύου, που αντιπροσωπεύει το 20% της συνολικής κατανάλωσής τους. Για την αντιμετώπιση του θέματος, προτάθηκε λοιπόν η χρήση Elastic Tree - ενός ισχυρού δικτυακού διαχειριστή ενέργειας - που χρησιμοποιεί SDN για να προσδιορίσει το ελάχιστο απαιτούμενο ενεργειακό subset. Η λύση Honeyguide [23] είναι μια άλλη επιλογή βελτιστοποίησης της κατανάλωσης που δεν στηρίζεται σε SDN, και η οποία χρησιμοποιεί εικονικές μηχανές μετάβασης, προκειμένου να απενεργοποιεί περισσότερες μηχανές και μεταγωγείς του συστήματος όταν δεν χρειάζονται για την εκτέλεση των εργασιών του.

- **Backbone Networks**

Με τον όρο Backbone (ραχοκοκαλιά) ορίζεται ένα μέρος ενός υπολογιστικού δικτύου το οποίο διασυνδέει διάφορα τμήματα αυτού, παρέχοντας μια διαδρομή για την ανταλλαγή πληροφοριών μεταξύ διαφορετικών LANs (Local Area Network) ή υποδικτύων. Ένα Backbone Network μπορεί να συνδέσει στο ίδιο κτίριο ποικίλα δίκτυα τα οποία βρίσκονται σε διαφορετικά κτίρια όπως πχ. δίκτυα σε πανεπιστημιακό περιβάλλον (Campus) και δίκτυα ευρείας περιοχής (Wide Area Networks). Συνήθως η χωρητικότητα ενός δικτύου ραχοκοκαλιάς (backbone network) είναι μεγαλύτερη από τα δίκτυα που είναι συνδεδεμένα σε αυτό. Ένα αντιπροσωπευτικό παράδειγμα ενός τέτοιου δικτύου αποτελεί το Internet Backbone, που μπορεί να οριστεί από τις κύριες διαδρομές δεδομένων μεταξύ μεγάλων, στρατηγικά διασυνδεδεμένων δικτύων υπολογιστών και δρομολογητών πυρήνα στο Διαδίκτυο. Η SDN αρχιτεκτονική μπορεί να χρησιμοποιηθεί στις περιπτώσεις δικτύων ραχοκοκαλιάς μεγάλης κλίμακας (large-scale Backbone Networks), ώστε να επιτευχθεί προγραμματισιμότητα και υψηλή διαθεσιμότητα, με την εταιρία Google να αποτελεί το χαρακτηριστικότερο παράδειγμα λειτουργίας ενός τέτοιου δικτύου. Η στρατηγική της Google όσων αφορά την SDN τεχνολογία αποτελείται από τους εξής τρείς πυλώνες :

- Jupiter: Η Google βασιζόμενη στις αρχές των SDN δικτύων κατασκεύασε το Jupiter, μια διασύνδεση κέντρων δεδομένων ικανή να υποστηρίξει περισσότερους από 100000 εξυπηρετητές. Από το 2013 και έπειτα η τεχνολογία αυτή υποστηρίζει περισσότερα από 1 Petabytes ανά δευτερόλεπτο (1Pb/s) του συνολικού εύρους ζώνης για να εξυπηρετεί τις υπηρεσίες της.
- B4 WAN interconnect : Η Google δημιούργησε το B4 [24] για να συνδέσει τα κέντρα δεδομένων μεταξύ τους και για να αναπαράγει με τον τρόπο αυτό δεδομένα σε πραγματικό χρόνο μεταξύ μεμονωμένων πανεπιστημιουπόλεων.
- Andromeda : Η Andromeda της Google είναι μια στοίβα network function virtualization (NFV) που της επιτρέπει να παρέχει τις ίδιες δυνατότητες που διαθέτει στις εγγενείς εφαρμογές της, μέχρι και τις εικονικές μηχανές που εκτελούνται στην πλατφόρμα Google Cloud.

- **Internet Exchange Points (IXP)**

Ένα Internet Exchange Point αποτελεί τη φυσική υποδομή μέσω της οποίας οι πάροχοι των υπηρεσιών του Διαδικτύου (Internet Service Providers - ISPs) και τα δίκτυα παροχής περιεχομένων (Content Delivery Networks - CDN) ανταλλάσουν Internet traffic μεταξύ των δικτύων τους. Σήμερα τα Internet Exchange Points (IXPs) χρησιμοποιούν το Border Gateway Protocol (BGP) ως το βασικό πρωτόκολλο δρομολόγησης μεταξύ των τομέων τους, το οποίο όμως εμπεριέχει από κάποιους περιορισμούς, καθώς η δρομολόγηση της διαμοίρασης πληροφοριών βασίζεται μόνο στο πρόθεμα της IP διεύθυνσης προορισμού. Η ανάπτυξη του Software Defined Networking στα IXP’s υπόσχεται πολλές ευκαιρίες όπως την απελευθέρωσή τους από τους εκάστοτε περιορισμούς διαδικτυακών πρωτοκόλλων, την απαλλαγή από την προηγμένη εξισορρόπηση φορτίου (load balancing) κ.α.

- **Wireless Access Networks**

Πλέον ένα μεγάλο πλήθος προσπαθειών έχουν επικεντρωθεί στη συνδεσιμότητα στο πλαίσιο δικτύων ασύρματης πρόσβασης, τα οποία βασίζονται σε υποδομές όπως π.χ. τα κυψελοειδή και Wi-Fi δίκτυα. Η συντριπτική πλειοψηφία των αιτήσεων τελικών χρηστών (End-user) για υπηρεσίες εφαρμογών προέρχεται από τις κινητές συσκευές, οι οποίες είναι συνδεδεμένες στο δίκτυο μέσω Wi-Fi. Είναι λοιπόν σημαντικό να εξεταστεί ο ρόλος της SDN τεχνολογίας στα σημερινά ασύρματα LANs και να

καθοριστεί η επερχόμενη εξέλιξη του στα χρόνια που θα ακολουθήσουν. Για παράδειγμα το Project OpenRoads οραματίζεται ένα κόσμο στον οποίο οι χρήστες μπορούν να μετακινούνται ανάμεσα σε διαφορετικές ασύρματες υποδομές, διαχειριζόμενες από διάφορους παρόχους. Για αυτόν ακριβώς το σκοπό προτάθηκε η ανάπτυξη ενός SDN δικτύου βασιζόμενο σε ασύρματες αρχιτεκτονικές, το οποίο θα είναι backwards-compatible, αλλά θα είναι παράλληλα ανοιχτό και κοινόχρηστο ανάμεσα στους διαφορετικούς παρόχους υπηρεσιών. Το παραπάνω εγχείρημα αποτέλεσε έμπνευση για μεταγενέστερες έρευνες στην αντιμετώπιση συγκεκριμένων απαιτήσεων και προκλήσεων στην ανάπτυξη ενός κυψελοειδούς δικτύου καθοριζόμενου από λογισμικό.

- **Optical Networks**

Η διαχείριση της κυκλοφορίας των δεδομένων ως πλήθος ροών, δίνει στα SDN και OpenFlow δίκτυα τη δυνατότητα να υποστηρίξουν και να ενσωματώσουν πολλαπλές δικτυακές τεχνολογίες. Έτσι είναι δυνατόν να παρέχεται επίσης κάποιος ενοποιημένος έλεγχος αγνωστικής-τεχνολογίας για τα οπτικά δίκτυα μεταφοράς, αλλά και για τη διευκόλυνση αλληλεπίδρασης μεταξύ τόσο των δικτύων μεταγωγής πακέτων όσο και των κυκλωμάτων. Σύμφωνα με το Optical Transport Working Group (OTWG), το οποίο δημιουργήθηκε από τον ONF το 2013, τα πλεονεκτήματα από την εφαρμογή των προτύπων SDN και OpenFlow ειδικότερα στα δίκτυα οπτικών μεταφορών περιλαμβάνουν : βελτιωμένο έλεγχο του οπτικού δικτύου μεταφορών και περισσότερη ευελιξία στη διαχειρισή αυτού, την υλοποίηση συστημάτων διαχείρισης και ελέγχου τρίτων, καθώς και την ανάπτυξη νέων υπηρεσιών με την αξιοποίηση της εικονοποιήσης (virtualization) και της γενικότερης τεχνολογίας SDN.

- **Home and Small Business**

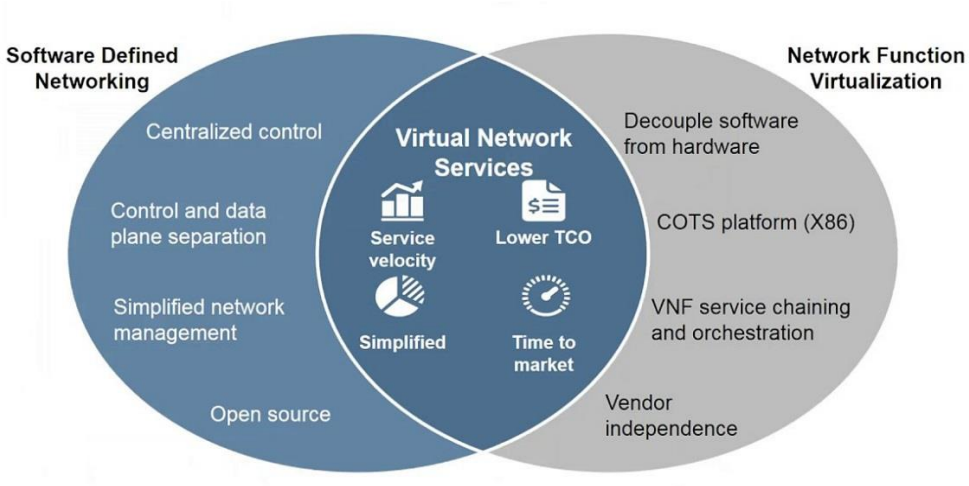
Ένα μεγάλο πλήθος ερευνών έχουν εξετάσει και έχουν καταλήξει στο γεγονός ότι η SDN τεχνολογία μπορεί να χρησιμοποιηθεί σε ακόμη μικρότερα δίκτυα, όπως για παράδειγμα τα δίκτυα που συναντάμε σε κατοικίες και σε μικρές επιχειρήσεις. Καθώς αυτά τα περιβάλλοντα παρουσιάζουν μια διαρκή αύξηση στην πολυπλοκότητά τους, το ίδιο σημαντικά αυξάνεται η ανάγκη για την πιο προσεκτική διαχείριση και ασφάλεια του εκάστοτε δικτύου. Δυστυχώς όμως δεν είναι καθόλου πρακτικό να υπάρχει ένας ειδικός διαχειριστής δικτύου σε κάθε κατοικία ή γραφείο. Ο Feamster [25] προτείνει για αυτής της κατηγορίας τα δίκτυα, την λειτουργία με βάση τη λογική

“plug in and forget”, δίνοντας τη διαχείρισή τους σε εξωτερικούς συνεργάτες (outsourcing), γεγονός που θα μπορούσε να επιτευχθεί μέσω τηλεχειρισμού προγραμματιζόμενων μεταγωγέων και με εφαρμογές κατανεμημένης παρακολούθησης του δικτύου με τη χρήση αλγορίθμων συμπεράσματος για την ανίχνευση θεμάτων ασφάλειας που τυχόν θα προκύψουν.

2.4 SDN και Network Function Virtualization

Η Εικονοποίηση Λειτουργιών Δικτύου (Network Function Virtualization - NFV) είναι μια αρχιτεκτονική δικτύου, που χρησιμοποιεί τις IT τεχνολογίες για την εικονοποίηση των κόμβων ενός δικτύου σε δομικά στοιχεία, τα οποία με τη σειρά τους μπορούν να συνδεθούν μεταξύ τους και να αποτελέσουν τη βάση για την παροχή υπηρεσιών επικοινωνίας. Το NFV μπορεί παραδείγματος χάριν να αναφέρεται σε κινητές υπηρεσίες, όπως η ισοστάθμιση φορτίου (load balancing) και προστασία (firewalling), μακρυά από καθορισμένο για το σκοπό αυτό υλισμικό, σε ένα εικονοποιημένο περιβάλλον. Οι λύσεις NFV έχουν αναπτυχθεί κυρίως στα πλαίσια των κέντρων δεδομένων (data centers) για πλατφόρμες νέφους (cloud platforms) χρησιμοποιούμενες τόσο από τις επιχειρήσεις, όσο και από τους παρόχους υπηρεσιών. Το NFV και το SDN αποτελούν δύο στενά συνδεδεμένες τεχνολογίες που σε αρκετές περιπτώσεις δρουν συμπληρωματικά μεταξύ τους. Στη σημερινή εποχή, η πλειοψηφία των NFV πλατφόρμων περιέχουν ελεγκτές SDN. Από την άλλη πλευρά παρότι η τεχνολογία NFV μπορεί να πετύχει τους σκοπούς της χωρίς τη χρήση SDN μηχανισμών, αν χρησιμοποιηθούν οι αναφερθείσες αρχές, τότε μπορεί να ενισχυθεί η απόδοση, να απλοποιηθεί η συμβατότητα των υπαρχουσών εφαρμογών και διευκολυνθεί το σύνολο των διαδικασιών λειτουργίας και συντήρησης του δικτύου. Επιπροσθέτως το NFV μπορεί να υποστηρίξει την τεχνολογία SDN παρέχοντας της τις κατάλληλες υποδομές, ώστε να εκτελεσθεί το SDN λογισμικό πάνω σε αυτές. Περαιτέρω το NFV εστιάζει και αυτό στους στόχους του SDN, αναφορικά με τη χρήση ευρέως χρησιμοποιούμενων διακομιστών και μεταγωγέων. Έτσι λοιπόν όλες αυτές οι λύσεις ενδέχεται να οδηγηθούν σε συγχώνευση των συστημάτων ενορχήστρωσης, όπως οι πλατφόρμες διαχείρισης νέφους (cloud management platforms), ή πλατφόρμες ενορχηστρώσεων υπηρεσίων δικτύου.

Την τρέχουσα χρονική περίοδο πολλοί προμηθευτές δικτύων, επενδύουν τόσο στη NFV όσο και στη SDN τεχνολογία. Οι οδηγοί (drivers) και τα οφέλη που προκύπτουν από τις προαναφερθείσες τεχνολογίες συγκλίνουν, κάτι που δεν εκπλήσσει καθώς το NFV συχνά βασίζεται στη χρήση ενός SDN ελεγκτή για να πετύχει τα επιθυμητά αποτελέσματα. Συνεπώς η συνεργασία όλων των παραπάνω απλοποιεί τη διαχείριση των δικτύων, επιταχύνει τη παροχή νέων υπηρεσιών και ενδεχομένως μειώνει το κόστος που απαιτούνταν μέχρι πρότινος [26]. Στο *Σχήμα 3* που ακολουθεί παρουσιάζονται οι δυνατότητες και τα οφέλη της συνεργασίας των τεχνολογιών NFV και SDN.



Σχήμα 3 : Συνεργασία των τεχνολογιών NFV και SDN

2.5 Προκλήσεις για τα SDN Δίκτυα

Παρά το γεγονός ότι η έννοια της Δικτύωσης Καθοριζόμενης από Λογισμικό (SDN) δεν είναι πλέον καινούργια, μόλις τα τελευταία χρόνια γίνεται μία συστηματική προσπάθεια για την τυποποίηση, αλλά και την εφαρμογή της σε διάφορες δικτυακές τοπολογίες. Όπως είναι λοιπόν αναμενόμενο υπάρχουν ακόμα αρκετές πτυχές της τεχνολογίας αυτής που πρέπει να ερευνηθούν, όπως επίσης υπάρχουν αρκετές προκλήσεις που πρέπει να υπερκεραστούν.Αναφορικά με τα χαρακτηριστικά του SDN έχουν εκφραστεί ανησυχίες σχετικά με την ασφάλεια, την προγραμματισιμότητα και την κλιμακοδιαθεσιμότητα (scalability) του δικτύου. Αυτό αφορά κυρίως τους ελεγκτές, οι οποίοι είναι υπεύθυνοι για την εξασφάλιση των

παραπάνω χαρακτηριστικών γνωρισμάτων. Πέραν των προσπαθειών βελτίωσης των προαναφερθέντων χαρακτηριστικών, πρέπει ακόμα να ληφθούν υπόψη και οι τομείς της έρευνας στο σχεδιασμό των ελεγκτών, των μεταγωγέων και του Northbound Interface.

Ειδικότερα για τη Βόρεια Διεπαφή (Northbound Interface) επί του παρόντος δεν έχουν ορισθεί πολλά πρωτόκολλα για την αλληλεπίδραση μεταξύ ελεγκτών και υπηρεσιών. Ο βασικός στόχος φαίνεται να είναι μία πιο τυποποιημένη διεπαφή -όπως συμβαίνει στην περίπτωση της Νότιας Διεπαφής (Southbound Inteface) - μέσω της οποίας οι εφαρμογές θα έχουν πρόσβαση στο υλισμικό και θα επικοινωνούν με άλλες εφαρμογές. Η υπάρχουσα βιβλιογραφία (Procera [27] και Pyretic [28]) αναφερόμενη στην Βόρεια Διεπαφή προτείνει τη δημιουργία μιας απλοποιημένης και επαναχρησιμοποιήσιμης αφαίρεσης, η οποία θα χρησιμοποιείται για τον προγραμματισμό των ελεγκτών.

Περαιτέρω οι δικτυακές συσκευές οφείλουν να προχωρήσουν σε πολλές βελτιώσεις σε ότι αφορά στα πρωτόκολλα, στα επιχειρηματικά μοντέλα και τις εφαρμογές. Οι εργασίες που γίνονται σ’ αυτό το πεδίο εστιάζονται στο να καταστήσουν ευκολότερη τη σύνθεση τμημάτων λογισμικού (components) για τον έλεγχο και την ευκολότερη εξυπηρέτηση των διαδικασιών αποσφαλμάτωσης (debugging) και δοκιμών των εφαρμογών.

Η ασφάλεια είναι άλλη μία μεγάλη πρόκληση για τα SDN δίκτυα. Μέχρι σήμερα η επιχειρηματική και ακαδημαϊκή κοινότητα έχουν να επιδείξουν περιορισμένη συζήτηση γύρω από το συγκεκριμένο πεδίο. Συνεπώς για να γίνει η SDN τεχνολογία ευρύτερα αποδεκτή στη χρήση της απαιτείται μεγαλύτερη εστίαση στο κομμάτι της ασφάλειας και για αυτό το λόγο ο οργανισμός ONF έχει σχηματίσει μια αντίστοιχη ομάδα για να συντονίσει και να μελετήσει όλα τα ζητήματα που σχετίζονται με αυτήν.Το πιο σημαντικό ζήτημα ασφάλειας αφορά στον κεντρικό έλεγχο του δικτύου, ο οποίος μπορεί να εκτεθεί σε επιθέσεις, δεδομένης της αξίας του για το όλο σύστημα, την ενσωμάτωση των πρωτοκόλλων παλαιού τύπου, τις συνδέσεις μεταξύ τομέων κ.α. Σε γενικές γραμμές στις SDN πλατφόρμες υπάρχουν τρωτά σημεία. Για παράδειγμα ερωτήματα ασφάλειας έχουν τεθεί, σχετικά με τους μηχανισμούς ταυτοποίησης και εξουσιοδότησης στο επίπεδο ελεγκτών-εφαρμογών, με το αν έχουν

τη δυνατότητα να επιτρέψουν την πρόσβαση πολλαπλών οργανισμών σε πόρους δικτύου, παρεχοντάς τους παράλληλα την κατάλληλη προστασία.

Μια ακόμα σημαντική πρόκληση που αντιμετωπίζουν τα SDN δίκτυα είναι η δυνατότητα κλιμάκωσης/κλιμακοδιαθεσιμότητας (scalability). Σε αντίθεση με τα σημερινά ιεραρχικά δίκτυα (Hierachical Networks) που έχουν μεγάλες δυνατότητες κλιμάκωσης, εκφράζονται ανησυχίες κατά πόσο ένας μόνο ελεγκτής μπορεί να κλιμακωθεί σε μεγάλα αυτόνομα συστήματα και κατά πόσο είναι ικανός να διαχειριστεί μεγάλους όγκους κυκλοφορίας. Το ζήτημα αυτό μπορεί να διαχωριστεί χαλαρά σε δύο μέρη, την κλιμακοδιαθεσιμότητα του ελεγκτή και την κλιμακο-διαθεσιμότητα των κόμβων δικτύου. Όσων αφορά την κλιμάκωση του ελεγκτή εντοπίζονται οι ακόλουθες προκλήσεις. Η πρώτη εστιάζει στο χρόνο καθυστέρησης (latency) που εισάγεται με την ανταλλαγή πληροφοριών δικτύου μεταξύ πολλαπλών κόμβων και ενός μονάχα ελεγκτή. Η δεύτερη πρόκληση ασχολείται με τον τρόπο επικοινωνίας μεταξύ των ελεγκτών με τη χρήση East και Westbound API's. Τέλος εξετάζεται το μέγεθος και η λειτουργία της back-end βάσης δεδομένων του SDN ελεγκτή.

Συνοψίζοντας, τα SDN δίκτυα παρέχουν ευελιξία, κεντρικοποιημένο έλεγχο και ανοιχτές διεπαφές μεταξύ των κόμβων οδηγώντας έτσι σε ένα πιο αποτελεσματική δικτύωση. Παρόλα αυτά όμως για επιτευχθούν τα προαναφερθέντα και για να στραφούμε στα SDN δίκτυα, αφήνοντας πίσω μας τις παραδοσιακές δικτυακές αρχιτεκτονικές, εκκρεμεί η επίλυση ενός σημαντικού πλήθους προκλήσεων.

3. Software-Defined Network Controllers

Οι ελεγκτές αποτελούν τον πυρήνα της SDN αρχιτεκτονικής, καθώς οι ίδιοι συγκροτούν το κεντρικό σημείο του δικτύου, το οποίο είναι υπεύθυνο για την ενεργοποιήση και ενορχήστρωση της επικοινωνίας μεταξύ των εφαρμογών που αντιπροσωπεύουν τις συσκευές δικτύου και την επιχειρηματική λογική. Η επικοινωνία αυτή είναι εφικτή μέσω των Northbound και Southbound API's, στα οποία οφειλεί όλες τις απλοποιήσεις και αυτοματοποιήσεις που υφίστανται στο δίκτυο. Ιδανικά οι ελεγκτές συμβάλλουν στην ευφυΐα-λογική, στη ευελιξία, στην κλιμάκωση, και στο αποτελεσματικό κόστος του συνόλου της SDN υποδομής. Συνεπώς ο στρατηγικός τους ρόλος, οδήγησε στη συνεχή ανάπτυξη τους τόσο σε επίπεδο έρευνας, όσο και σε επίπεδο εμπορίου. Τα τελευταία χρόνια σημειώθηκε μεγάλη προόδος στον τομέα των ελεγκτών, αφού κατασκευάστηκαν αρκετά νέα μοντέλα με αποκλειστικό στόχο την βελτίωση των δικτυακών υποδομών που χρησιμοποιούσαν μέχρι πρότινος όλες οι μεγάλες εταιρίες. Αυτό το κεφάλαιο αποτελεί μια επισκόπηση των SDN ελεγκτών παρέχοντας αρχικά μια σύντομη ιστορική αναδρομή πάνω στο θέμα αυτό.

3.1 Σύντομη Αναδρομή

Όπως είναι αναμενόμενο η εξέλιξη των ελεγκτών ξεκινά μαζί με τη σύλληψη της ίδιας της ιδέας των SDN δικτύων. Ο πρώτος SDN ελεγκτής ήταν ο NOX και κατασκευάστηκε αρχικά από την εταιρία Nicira Networks το 2009. Την ίδια χρονική περίοδο κυκλοφόρησε και η πρώτη έκδοση του πρωτόκολλου OpenFlow. Καθώς ο ελεγκτής NOX αποτελούσε ένα πρόγραμμα ανοιχτού κώδικα (open source), ο ίδιος έγινε δωρεά στην SDN κοινότητα και σύντομα έθεσε τις βάσεις για την επίλυση πολλών και διαφόρων ζητημάτων. Έκτοτε κυκλοφόρησαν πολλαπλές εκδόσεις του NOX όπως για παράδειγμα η έκδοση POX το 2011 [29], η οποία βασίζεται σε Python και εμπεριέχει το πρωτόκολλο OpenvSwitch Database (OvSDB). Στη σημερινή εποχή οι NOX ελεγκτές δεν χρησιμοποιούνται σχεδόν καθόλου, ενώ οι POX ελεγκτές χρησιμοποιούνται αποκλειστικά και μόνο από την ερευνητική κοινότητα. Το επόμενο άξιο αναφοράς βήμα στο χρονοδιάγραμμα των SDN ελεγκτών θα μπορούσε να είναι η ανάπτυξη της πλατφορμας ONIX. Το ONIX ήταν μία κατανεμημένη πλατφόρμα

ελέγχου για παραγωγικά δίκτυα μεγάλης κλίμακας, η οποία συν-αναπτύχθηκε από τις Nicira, Google και NTT. Έθεσε επίσης τις βάσεις για τον VMware ελεγκτή που τον καθιστά μια από τις κορυφαίες εφαρμογές στην αγορά του σήμερα. Το 2010 εισήχθη στην αγορά ο Beacon controller [30], μια λύση ανοιχτύ κώδικα που αναπτύχθηκε στο Πανεπιστήμιο του Stanford. Ο Beacon, μια μετεξέλιξη του NOX controller, αποτελεί ένα Openflow ελεγκτή που βασίζεται σε Java, ο οποίος έγινε αρκετά δημοφιλής χάρη στην ευκολία που παρουσίαζε στην ανάπτυξη και εφαρμογή του. Οι ελεγκτές που ακολούθησαν μετά τον Beacon εμπνεύστηκαν σημαντικά από αυτόν, σε θέματα επιλογών σχεδιασμού. Ο διάδοχος του Beacon ονομάστηκε Floodlight [31] και αναπτύχθηκε από την εταιρία Big Switch Networks. Πολλά ζητήματα του Beacon ελεγκτή επιλύθηκαν μέσω του ελεγκτή Floodlight, καθιστώντας τον δεύτερο ως τον πιο πλούσιο ελεγκτή από άποψη χαρακτηριστικών. Μέσω της αντίστοιχης προσθήκης (plug-in) OpenStack, ο Floodlight μπορούσε να χρησιμοποιηθεί για τον έλεγχο μεγάλων ομάδων δικτυακών, υπολογιστικών και αποθηκευτικών πόρων. Αξίζει να σημειωθεί ότι η πλειοψηφία των σημερινών ελεγκτών παρέχουν OpenStack υποστήριξη. Πλέον υπάρχει πληθώρα ελεγκτών ανοιχτού κώδικα όπως ο Ryu, ο FlowER, ο LOOM, ο Trema, ο OpenMUL κ.ά . Περισσότερα σχετικά τους ελεγκτές αυτούς βρίσκονται στα επόμενα κεφάλαια. Πολλοί ελεγκτές αναπτύχθηκαν για εμπορική χρήση από διάφορες εταιρίες τελευταία χρόνια όπως πχ τη Cisco, τη HP, τη IBM, τη VMware και τη Juniper. Στις 8 Απριλίου 2013 ανακοινώθηκε ο open-source οργανισμός OpenDaylight, που αποτελεί και κομμάτι του οργανισμού Linux. Ο ελεγκτής OpenDaylight στηρίζεται και αυτός στο σχεδιασμό Beacon που προϋπήρχε και όπως είναι λογικό η γλώσσα στην οποία βασίζεται είναι η Java. Ο ODL υποστηρίζει επίσης το πρωτόκολλο OpenFlow, καθώς και άλλες εφαρμογές της νότιας διεπαφής (Southbound API's), ενώ περιλαμβάνει σημαντικά χαρακτηριστικά, όπως η δημιουργία συστάδων (clustering) και υψηλή διαθεσιμότητα (high availability) κλπ. Ως αντίπαλο δέος και ανταγωνιστικός προς τον ODL ελεγκτή το 2013, ο μη κερδοσκοπικός οργανισμός OnLab κατασκευάζει τον ελεγκτή ONOS (Open Network Operating Systems). Ο ελεγκτής αυτός υποστηρίζεται μέχρι σήμερα από ένα μεγάλο πλήθος εταιριών με τις πιο γνωστές να είναι η Microsoft, AT&T, HP, Ericsson, NTT, Ciena και Extreme Networks κ.ά. Περισσότερα στοιχεία για τους ελεγκτές ODL, ONOS και Ryu παρατίθενται σε ειδικά Κεφάλαια 5,6 & 7 της παρούσας εργασίας .

3.2 Χαρακτηριστικά και Δυνατότητες

Ήδη έχει αναφερθεί προηγουμένως ότι οι κύριοι σκοποί του SDN είναι η διαχείριση και ο προγραμματισμός δικτύων μέσω του διαχωρισμού του Επιπέδου Δεδομένων από το Επίπεδο Ελέγχου. Για την επίτευξή τους χρειάζεται ο σχεδιασμός ενός κεντρικοποιημένου (centralized) μοντέλου, το οποίο καθοδηγείται από τις δυνατότητες και τα χαρακτηριστικά του εκάστοτε ελεγκτή, ο οποίος αποτελεί και την "καρδιά" του συστήματος. Με τη πάροδο του χρόνου ο ρόλος του ελεγκτή εξελίχθηκε και βελτιώθηκαν σημαντικά τα βασικά του χαρακτηριστικά και δυνατότητες, προκειμένου αντιμετωπίσουν τις νέες απαιτήσεις και εξελίξεις της αγοράς και να παράσχουν καλύτερες και πιο αποτελεσματικές λύσεις στους οργανισμούς που χρησιμοποιούν SDN. Στη συνέχεια αναφέρονται οι σημαντικότερες δυνατότητες που διαθέτει ένας SDN ελεγκτής :

- **Αποδοτικότητα (Efficiency)**

Η αποδοτικότητα είναι ένας όρος που χρησιμοποιείται σε αυτή την εργασία για να περιγράψει την επίδοση (performance), τις δυνατότητες κλιμάκωσης/κλιμακοδιαθεσιμότητας (scalability) και την ασφάλεια (security) ενός δικτύου. Το ζητούμενο είναι ένας SDN ελεγκτής να ικανοποιεί αυτά τα τρία χαρακτηριστικά με το βέλτιστο τρόπο. Σύμφωνα με την τρέχουσα βιβλιογραφία η επίδοση και οι δυνατότητες κλιμάκωσης, χρησιμοποιούνται ώστε να περιγραφεί χρόνος απόκρισης και ο αριθμός ροών τα οποία μπορεί να διαχειριστεί ένας ελεγκτής. Αυτά είναι πολύ σημαντικά χαρακτηριστικά ανεξαρτήτως της χρήσης του ελεγκτή. Η ασφάλεια επίσης, μπορεί να αναφέρεται σε διάφορες λειτουργίες που πρέπει να εκτελεί ο ελεγκτής για να είναι συμβατός με τον συνεχώς αυξανόμενο αριθμό των αντίστοιχων απαιτήσεων. Καθώς ο αριθμός των εκδόσεων και των εφαρμογών των ελεγκτών αυξάνεται, προκύπτει η ανάγκη για συγκριτικές μελέτες σχετικά με την αποτελεσματικότητά τους. Ενδεικτικά αναφέρονται αποτελέσματα τέτοιων μελετών για τις επιδόσεις με τη χρήση του εργαλείου Mininet, των SDN ελεγκτών POX, Ryu, ONOS και OpenDayLight το 2015 [32], για επιδόσεις ελεγκτών NOX, Beacon, Maestro και Floodlight με βάση τη διαχείριση της κυκλοφορίας (traffic handling) το 2016 [33], για τις επιδόσεις των ελεγκτών ONOS, Foodlight, ODL και Ryu με βάση τη διακίνηση (throughput) και τον χρόνο καθυστέρησης (latency) το 2017 [34], για τους ελεγκτές FloodLight και ODL το 2016 [35] και άλλες νεότερες που θα αναφερθούν στα επόμενα κεφάλαια.

- **Υποστήριξη της Νότιας Διεπαφής (Southbound Support)**

Η υποστήριξη της Νότιας Διεπαφής έχει ήδη ορισθεί ως ο τρόπος με τον οποίο ο ελεγκτής SDN, χειρίζεται τις δικτυακές συσκευές με σκοπό τη βελτιστοποίηση της κυκλοφοριακής ροής των δεδομένων που διακινούνται από το δίκτυο. Έχει αναφερθεί προηγουμένως ότι υπάρχουν πολλά πρωτόκολλα Νότιας Διεπαφής που μπορούν να χρησιμοποιηθούν, με το πρωτόκολλο OpenFlow να είναι το πλέον διαδεδομένο. Ορισμένες από τις βασικές λειτουργίες που πρέπει να υποστηρίζει οποιοσδήποτε ο ελεγκτής που χρησιμοποιεί OpenFlow, περιλαμβάνουν αντιστοίχιση πεδίων (field matching), εντοπισμός δικτύου με LLDP πρωτόκολλο (Link Layer Discovery Protocol) και άλλα. Σχετικά με την υποστήριξης της Νότιας Διεπαφής, αυτοί που την χρησιμοποιούν (προγραμματιστές,χειριστές δικτύου κλπ) οφείλουν να λάβουν υπόψη όχι μόνο χαρακτηριστικά των πρωτοκόλλων,αλλά και τις επεκτάσεις/νεότερες εκδόσεις αυτών, οι οποίες περιλαμβάνουν τις προσαρμογές στις τελευταίες εξελίξεις της τεχνολογίας. Χαρακτηριστικό παράδειγμα στην περίπτωση του OpenFlow, είναι ότι στην έκδοση 1.0 δεν υποστηρίζεται το IPv6 πρωτόκολλο, αλλά στη συνέχεια αυτό συμπεριλήφθηκε στην πρότυπη έκδοση v1.3. Αν και το πρωτόκολλο OpenFlow είναι το πιο ευρύτατα χρησιμοποιούμενο για την υποστήριξη εφαρμογών Νότιας Διεπαφής (SB APIs), υπάρχουν και άλλα πρωτόκολλα για τη διαχείριση των συσκευών του δικτύου, όπως το NETCONF (το οποίο έχει τυποποιηθεί από την IETF), το OF-Config (υποστηρίζεται από το ONF), το Opflex (υποστηρίζεται από τη Cisco) κ.ά. Ακόμα έχουν αναπτυχθεί πρωτόκολλα δρομολόγησης, ως νότιες διεπαφές για ορισμένους ελεγκτές, όπως το IS-IS, το OSPF και το BGP, με στόχο να εξυπηρετήσουν υβριδικά δίκτυα (hybrid networks), ή μη SDN δίκτυα, ή για εφαρμογή σε παραδοσιακά δίκτυα που λειτουργούν με τρόπο SDN.

- **Υποστήριξη της Βόρειας Διεπαφής (Northbound Support)**

Τα Northbound APIs χρησιμοποιούνται από το Επίπεδο Εφαρμογών για την επικοινωνία με τον Ελεγκτή και αντιστρόφως. Οι εφαρμογές αυτές είναι ίσως το πιο κρίσιμο τμήμα της αρχιτεκτονικής του ελεγκτή SDN, αφού η προστιθέμενη αξία της SDN δικτύωσης, συνδέεται με τις καινοτόμες εφαρμογές που αναπτύσσονται και εφαρμόζονται σ' αυτό το επίπεδο. Συνοπτικά τα NB APIs χρησιμοποιούνται για την πραγματοποίηση αφαιρέσεων (abstractions) δικτύου και την προγραματισιμότητα

(programmability) του. Λόγω της μεγάλης τους σημασίας, όπως τονίστηκε, θα πρέπει να είναι σε θέση να υποστηρίξουν μια πλειάδα εφαρμογών, όπως πλατφόρμες ενορχήστρωσης που επιτρέπουν τη σύνδεση με αυτοματοποιημένες στοίβες (stacks), όπως το OpenStack, ή CloudStack, που χρησιμοποιείται στη διαχείριση Νέφους. Το OpenStack είναι πλατφόρμα λογισμικού ανοικτού κώδικα που χρησιμοποιείται σε υπηρεσίες νέφους και αναπτύσσεται κυρίως ως infrastructure-as-a-service (Iaas).
Πρόσφατα ο ONF έστρεψε το ενδιαφέρον του στην τυποποίηση των Βόρειων Διεπαφών, δημιουργώντας μια Ομάδα Εργασίας για να γράψει κώδικα, να αναπτύξει πρωτότυπα και να διερευνήσει τη δυνατότητα δημιουργίας προτύπων. Σήμερα το πρωτόκολλο REST φαίνεται να είναι η ευρύτερα χρησιμοποιούμενη Βόρεια Διεπαφή από τους ελεγκτές SDN δικτύων.

- **Προγραμματισιμότητα (Programmability)**

Η προγραμματισιμότητα αποτελεί ένα από τα πιο πολύτιμα και χρήσιμα χαρακτηριστικά της SDN τεχνολογίας και των ελεγκτών της. Στο παρελθόν η διαμόρφωση δικτύου γινόταν μέσω της εφαρμογής σχετικών κανόνων σε κάθε μια συσκευή του δικτύου ξεχωριστά. Αναπόφευκτα, αυτή η στατική προσέγγιση σε πολλές περιπτώσεις ήταν πολύ χρονοβόρα και επιρρεπής σε σφάλματα οδηγώντας έτσι σε υποβάθμιση της συνολικής επίδοσης του δικτύου[36].Οι προγραμματιζόμενες διεπαφές αποτελούν τα βασικά τμήματα λογισμικού (components) κάθε ελεγκτή. Ένα από τα πιο συνήθη παραδείγματα προγραμματισιμότητας είναι ανακατεύθυνση της ροής (traffic redirection), που μπορεί να ζητηθεί για πολλούς σκοπούς όπως π.χ για την κατανομή της ροής με βάση το χρόνο, την τοποθεσία και την ασφάλεια. Σε πρώτη φάση ένα Northbound API καθιστά διαθέσιμες τις πληροφορίες του κέντρου ελέγχου, που έχουν συγκεντρωθεί στον ελεγκτή, στις δικτυακές εφαρμογές. Αυτές οι εφαρμογές με τη σειρά τους είναι πλέον σε θέση να τροποποιήσουν το δίκτυο, ώστε αυτό για παράδειγμα να προωθεί πακέτα χρησιμοποιώντας τη διαδρομή με το μικρότερο δυνατό κόστος, ή να αλλάξει τις ρυθμίσεις σχετικά με την ποιότητα των υπηρεσιών (Quality of Service - QoS) που σχετίζονται με το εύρος ζώνης και άλλους παράγοντες λειτουργίας. Ακόμη ένα παράδειγμα προγραμματισιμότητας στους SDN ελεγκτές είναι η δυνατότητα χρήσης φίλτρων με τα οποία καθορίζονται ποια πακέτα είναι αποδεκτά και ποιά όχι από τον εκάστοτε ελεγκτή. Τα δυναμικά αυτά φίλτρα μπορούν να βασίζονται σε αντιστοίχιση κεφαλίδας πακέτων, απλοποιώντας έτσι τη

λειτουργία τους, αλλά μπορούν επίσης να λειτουργήσουν πιο εκλεπτυσμένα όταν αποτελούνται από σύνθετους συνδυασμούς κεφαλίδων πολλαπλών πακέτων. Τέλος τα φίλτρα αυτά πρέπει να μπορούν να αναπτυχθούν δυναμικά με τον SDN ελεγκτή να αναλαμβάνει την ώθηση των αντίστοιχων καταχωρήσεων των πινάκων ροής στους μεταγωγείς.

- **Παρακολούθηση (Monitoring)**

Η παρακολούθηση του δικτύου αποτελεί άλλη μια εξίσου σημαντική ιδιότητα που διαθέτει ο ελεγκτής SDN. Μέσω πρωτοκόλλων (π.χ OpenFlow) και σχετικών εργαλείων ο ελεγκτής μπορεί να εντοπίσει τα προβλήματα που υπάρχουν στο δίκτυο και να διευκολύνει τις διαδικασίες αντιμετώπισης τους. Επιπροσθέτως κάποια από τα πλεονεκτήματα του ελεγκτή, περιλαμβάνουν λεπτομερή παρακολούθηση ροής, παρακολούθηση συγκεκριμένων κατηγοριών κυκλοφορίας κ.λ.π. Ακόμη ο ελεγκτής θα πρέπει να υποστηρίζει πρότυπα πρωτόκολλα και τεχνικές παρακολούθησης, έτσι ώστε οι πληροφορίες που αντλούνται να μπορούν να ενσωματωθούν σε άλλα συστήματα διαχείρισης και ενορχήστρωσης του δικτύου. Θα πρέπει, για παράδειγμα, να είναι δυνατή η παρακολούθηση της υγείας του ελεγκτή και των εικονικών δικτύων που υποστηρίζει χρησιμοποιώντας κάποιο Simple Network Management Protocol (SNMP). Το SDN Interactive Manager [37] και το OFMon [38] είναι δύο εφαρμογές δομοστοιχείων (modules) για την παρακολούθηση του δικτύου, τα οποία έχουν ενσωματωθεί σε ελεγκτές Floodlight και ONOS αντίστοιχα .

- **Εικονοποίηση Δικτύου (Network Virtualization)**

Εικονοποίηση δικτύου είναι η δυνατότητα δημιουργίας λογικών εικονικών δικτύων, τα οποία διαχωρίζονται από το υποκείμενο υλισμικό. Κοινά παραδείγματα εικονοποίησης δικτύου που υπάρχουν εδώ και αρκετά χρόνια είναι τα εικονικά LAN (Virtual LAN - VLAN) και η εικονική δρομολόγηση και προώθηση (Virual Rooting and Forwarding - VRF). Λόγω των ραγδαίων αλλαγών όσον αφορά τον όγκο του δικτύου, τις απαιτήσεις επιδόσεων και άλλα, οι παραπάνω μέθοδοι θεωρούνται πλέον περιορισμένοι τόσο ως προς το πεδίο εφαρμογής τους, όσο και ως προς την αξία τους. Οι ελεγκτές SDN διευκολύνουν την υλοποίηση της εικονοποίησης του δικτύου με καθολικό τρόπο (end-to-end) δίνοντας τη δυνατότητα στους οργανισμούς να

δημιουργούν δυναμικά εικονικά δίκτυα και να ανταποκρίνονται στις απαιτητικές ανάγκες που έχουν δημιουργηθεί.

Η ανάπτυξη των τεχνικών εικονοποίησης των δικτύων (Network Virtualization), έδωσαν στα δίκτυα SDN μια νέα διάσταση. Έτσι είναι δυνατός ο τεμαχισμός (slicing) του δικτύου και η πολλαπλή ενοικίαση υποδοχέων (Multi-tenant Hosting), σε υπάρχοντες φυσικούς πόρους του. Το FlowVisor [39] είναι η πιο δημοφιλής εφαρμογή που βασίζεται στα SDN, για την χρήση εικονικών δικτύων και αξιοποίηση των λειτουργιών OpenFlow, για την αφαίρεση υλισμικού από το υποκείμενο δίκτυο. Το VeRTIGO [40] αποτελεί επέκταση του FlowVisor, που παρέχει στους ελεγκτές επιλογές σε βάθος των απαιτούμενων αφαιρέσεων. Η επέκταση αυτή βελτιώνει την ευελιξία των διατάξεων SDN, όμως αυξάνει την πολυπλοκότητα του hypervisor. Οι Xingtao et al.[41] πρότειναν έναν ελεγκτή SDN βασισμένο στο docker [41] για τη βελτίωση της ταχύτητας της εφαρμογής και της επέκτασης της κινητικότητας. Στο [42] η ευελιξία του NOX, χρησιμοποιήθηκε ως container-based εικονοποιημένο δομοστοιχείο σε ένα ελεγκτή SDN για την αποτελεσματική απόκρυψη και διαχείριση επεικονίσεων μεταξύ εικονοποιημένων δικτύων και φυσικών μεταγωγέων.

Το Hyperflex [43] προτείνει ένα εικονοποιημένο μοντέλο Επιπέδου Ελέγχου, το οποίο στοχεύει σε επίτευξη κλιμακοδιαθεσιμότητας (scalability), ιδιωτικότητας και επεκτασιμότητας. Σ' αυτή την αρχιτεκτονική οι FlowVisor και Ryu ελεγκτές συνδυάζονται για την παροχή βασικών υπηρεσιών hypervisor, όπως και για τον έλεγχο του hypervisor του δικτύου.

Το OpenDayLight [44] έχει ενσωματωθεί με το OpenStack Habana [45] για την αξιολόγηση της αποτελεσματικότητας της αρχιτεκτονικής βασισμένης στο Νέφος του στα SDN δίκτυα για Κέντρα Δεδομένων, που βρίσκονται σε διαφορετικές περιοχές.Στη συγκεκριμένη αρχιτεκτονική, ο ελεγκτής επικοινωνεί με το Habana, χρησιμοποιώντας την REST API που είναι εγκατεστημένη σ΄αυτόν, για την εκτέλεση κρίσιμων εργασιών, όπως είνα η οικοδόμηση, η απομάκρυνση, ή η μετανάστευση εικονικών κλώνων (instances) που βρίσκονται στα περιβάλλοντα μεταξύ Κέντρων Δεδομένων (inter Data Centers) και εντός Κέντρων Δεδομένων (intra Data Centers).

- **Ευελιξία (Flexibility)**

Η ευελιξία έχει καταστεί ως ένας σημαντικός στόχος στο σχεδιασμό των δικτύων και των μηχανισμών του Επιπέδου Ελέγχου και του Επιπέδου Δεδομένων. Στην

πραγματικότητα ετερογενείς απαιτήσεις, που προέρχονται από διαφορετικά πεδία εφαρμογών (application domains) δημιουργούν υψηλές αξιώσεις από τα δίκτυα προκειμένου να είναι ευέλικτα (flexible). Αυτές οι απαιτήσεις μπορεί να περιλαμβάνουν την προσθήκη νέων ροών στο δίκτυο, ή και εικονικά δίκτυα κατά παραγγελία, χωρίς να επηρρεάζονται οι υφιστάμενες ροές, ή τα δίκτυα καθεαυτά, όπως και τη δυνατότητα προσωρινής επέκτασης της τοπολογίας του δικτύου προκειμένου να εξυπηρετηθούν κάποια συμβάντα (events).Η ευελιξία μπορεί να οριστεί για διαφορετικά πεδία και από διαφορετικές οπτικές γωνίες. Για τα δίκτυα και ειδικά τα SDN και NFV, η ευελιξία αναφέρεται στη δυνατότητά τους να προσαρμόζουν τους πόρους (resources), όπως για παράδειγμα τις ροές, ή την τοπολογία στις εκάστοτε αλλαγές των απαιτήσεων. Η προσαρμογή στις αλλαγές, μπορεί να αφορά σε προσαρμογή της διαμόρφωσης (configuration), ή της τοπολογίας του δικτύου ή και των λειτουργιών του και των τοποθετήσεων τους σ΄αυτό.

Τα τελευταία χρόνια έχουν αναδειχθεί τεχνολογίες και προσεγγίσεις σχετικά με την παροχή ευελιξίας στα δίκτυα. Η πλέον γνωστή και ευρέως διαδεδομένη είναι η προσέγγιση της δικτύωσης καθοριζόμενης από λογισμικό (SDN), η οποία παρέχει ευαλιξία στο δίκτυο με το διαχωρισμό του Επιπέδου Δεδομένων από ένα κεντρικοποιημένο (centralized) Ελέγχου Ελέγχου, με μία τυποποιημένη διεπαφή, που επιτρέπει προγραμματισιμότητα και ως εκ τούτου παρέχει ευελιξία στα συγκεκριμένα δίκτυα (SDN). Ένας ελεγκτής βασιζόμενος στο SDN (SDN- based network control) μπορεί να λειτουργήσει συμπληρωματικά στη λογική της εικονοποίησης δικτύου (NFV), βελτιώνοντας την ευελιξία ενός εικονικού δικτύου, η χρήση του οποίου σήμερα είναι ευρέως χρησιμοποιούμενη από πολλούς παρόχους υπηρεσιών.

Στο [46] παρουσιάζονται οι εξελίξεις (state of the art) σχετικά με τον ευέλικτο έλεγχο της διακίνησης (Flexible Traffic Control), τις ευέλικτες αρχιτεκτονικές δικτύων (Flexible Network Architectures), για τα ευέλικτα κινητά δίκτυα (Flexible Mobile Networks), για την ευέλικτη διαχείριση δικτύων (Flexible Management Network) και το ευέλικτο επίπεδο ελέγχου (Flexible Data Plane).

- **Τοπολογία (Topology)**

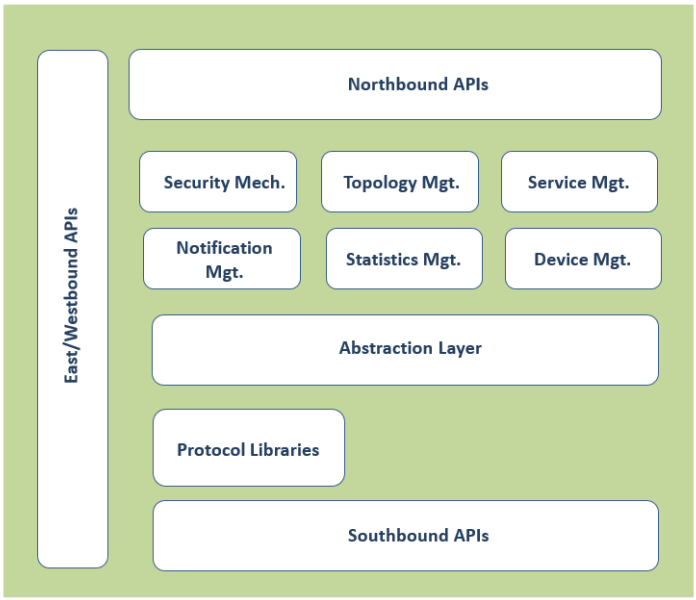
Οι ελεγκτές SDN αξιολογούνται επίσης να με βάση την προσαρμοστικότητά τους, όσον αφορά την τοπολογία του δικτύου. Η τοπολογία δικτύου είναι η διάταξη των

στοιχείων (ελεγκτές, μεταγωγείς κ.α) ενός δικτύου υπολογιστών. Στις περιπτώσεις δικτύων μεγάλης κλίμακας, πρέπει να εξετάζονται και να αναλύονται οι επιλογές σχετικά με την τοπολογία επιπλέον ελεγκτών. Επομένως η επικοινωνία μεταξύ ελεγκτών(Eastbound/Westbound επικοινωνία) αποτελεί επίσης μέρος της τοπολογίας. Επί του παρόντος, υπάρχουν κάποιες συζητήσεις στον κλάδο σχετικά με την τυποποίηση του τρόπου με τον οποίο οι ελεγκτές θα επικοινωνούν μεταξύ τους. Το BGP πρωτόκολλο αποτελεί μία συνήθη τεχνική για την ανταλλαγή πληροφοριών μεταξύ ελεγκτών. Τέλος, εξετάζονται θέματα σχετικά με το επίπεδο κεντρικοποίησης της αρχιτεκτονικής.

3.3 Σχεδιασμός Ελεγκτή (Components)

Η αρχιτεκτονική SDN που περιγράφτηκε στο Κεφάλιο 2.2, δεν αναφέρεται στον εσωτερικό σχεδιασμό ή την υλοποίηση του ελεγκτή. Ο ελεγκτής μπορεί να είναι ένα μόνο σύστημα λογισμικού, ή πολλαπλά συστήματα, οργανωμένα κατά τέτοιο τρόπο ώστε να εκτελούν λειτουργίες (functionalities), όπως η ισοστάθμιση φορτίου, η διαχείριση συσκευών κ.ά. Ακόμα μπορεί να "τρέχουν" σε τοπικούς πόρους, ή κατανεμημένους πόρους, όπως οι εικονικές μηχανές (VMs) σε Κέντρα Δεδομένων. Γενικά ο ελεγκτής θεωρείται ως ένα "μαύρο κουτί" που καθορίζεται από τις υπηρεσίες που παρέχει. Στο *Σχήμα 4* παρουσιάζεται ένα απλοποιημένο αρχιτεκτονικό πλαίσιο ενός SDN ελεγκτή, στο οποίο απεικονίζονται τα βασικά δομοστοιχεία (modules) και οι διεπαφές που το συγκροτούν.

Στις υπάρχουσες πλατφόρμες ελεγκτών, βρίσκονται καλά καθορισμένες στιβάδες, όπως αυτή των εφαρμογών (applications), της ενορχήστρωσης και των υπηρεσιών (orchestration & services), των βασικών λειτουργιών (core functions) του ελεγκτή και τα στοιχεία (elements) για τις νότιας κατεύθυνσης (southbound) επικοινωνίες [47]. Πρέπει να σημειωθεί ότι τα όρια των SDN ελεγκτών δεν είναι καθορισμένα με σαφήνεια. Ως αποτέλεσμα αυτού του γεγονότος, ορισμένοι θεωρούν ότι η διαχείριση υπηρεσιών αποτελεί αναπόσπαστο τμήμα ενός SDN ελεγκτή, ενώ άλλοι θεωρούν ότι πρόκειται για χωριστή εφαρμογή, η οποία τρέχει αναξάρτητα από αυτόν. Το ίδιο φαινόμενο παρατηρείται και με άλλες λειτουργίες. Μια ενδεικτική περιγραφή Τμημάτων Λογισμικού (components) /στιβάδας (layer) δίνεται στη συνέχεια.



Σχήμα 4 : Αρχιτεκτονική Δομή SDN Ελεγκτή

3.3.1 Στιβάδα Southbound

Στην αρχιτεκτονική των SDN ελεγκτών το southbound επίπεδο αντιπροσωπεύει ότι έχει αναφερθεί μέχρι στιγμής, ως νότια διεπαφή ή API και αποτελεί το κανάλι επικοινωνίας μεταξύ του ελεγκτή και των συσκευών δικτύου. Στις περισσότερες περιπτώσεις χρησιμοποιείται μια σύνδεση TLS (Transport Layer Security) μεταξύ των συσκευών μέσω της οποίας διενεργείται μια ασφαλής και επικυρωμένη επικοινωνία των συσκευών. Το επίπεδο Southbound αποτελείται από προγράμματα οδηγών συσκευών (device drivers) [47] δίνοντας στον ελεγκτή τη δυνατότητα να χρησιμοποιεί διαφορετικά API's και πολλαπλά πρωτόκολλα για τη διαχείριση μιας σειράς φυσικών και εικονικών συσκευών. Η επιλογή του πρωτοκόλλου southbound εξαρτάται από την περίπτωση χρήσης. Στη συνέχεια αναφέρονται ορισμένα πρωτόκολλα που υποστηρίζονται στο Southbound επίπεδο εκτός του OpenFlow που έχει ήδη αναφερθεί σε προηγούμενο κεφάλαιο.

- *NetConf Protocol*

Το NetConf πρωτόκολλο καθορίζει ένα μηχανισμό μέσω του οποίου γίνεται η διαμόρφωση και διαχείριση μίας συσκευής δικτύου. Χρησιμοποιεί επίσης ένα μηχανισμό που βασίζεται σε απομακρυσμένες κλήσεις διαδικασιών (RPC - Remote

Procedure Call) ώστε να διευκολυνθεί η επικοινωνία μεταξύ του πελάτη (client) και του εξυπηρετητή (server).

- *Path Computation Element (PCE) Protocol*

Αποτελεί ένα πρωτόκολλο πλήρους κλίμακας, που απλοποιεί τον υπολογισμό μονοπατιών (path computation), διαχωρίζοντας τον προσδιορισμό τοπολογίας δικτύου από την κατασκευή του μονοπατιού. Σε αντίθεση με άλλα πρωτοκολλα στα οποία μια SDN υλοποίηση απαιτεί την αντικατάσταση των ήδη υπαρχόντων στοιχείων δικτύου, στην περίπτωση του PCE πρωτοκόλλου απαιτείται μόνο η αναβάθμιση των head-end δρομολογητών. Το PCE πρωτοκόλλο είναι επίσης ικανό να ενσωματώνει παραμέτρους οπτικού δικτύου στον υπολογισμό διαδρομής [47] και να δημιουργεί διαδρομές σε τομείς δρομολόγησης (routing domains) .

- *Extensible Messaging and Presence (XMPP) Protocol*

Το πρωτόκολλο αυτό βασίζεται σε XML (Extensible Markup Language) και καθιστά δυνατή την ανταλλαγή δομημένων δεδομένων σε πραγματικό χρόνο (real-time) μεταξύ δύο οντοτήτων δικτύου. Επιπροσθέτως το XMPP θεωρείται ένα ώριμο πρωτόκολλο που επιτρέπει τη διαλειτουργικότητα με δίκτυα και συστήματα παλαιότερου τύπου (legacy networks).

- *Multiprotocol Label Switching – Transport Profile (MPLS-TP) Protocol*

Είναι μία έκδοση του MPLS πρωτοκόλλου που χρησιμοποιείται στα δίκτυα δεδομένων μεταγωγής πακέτων. Το πρωτόκολλο αυτό παρέχει μια αξιόπιστη τεχνολογία βασισμένη σε πακέτα και στα πρόσθετα χαρακτηριστικά του περιλαμβάνονται λειτουργίες συντήρησης, διαχείρισης της κυκλοφορίας δεδομένων παλαιού τύπου και άλλα.

- *Border Gateway Protocol (BGP)*

Αποτελεί ένα βασικό διαδικτυακό πρωτόκολλο δρομολόγησης (core-Internet Routing Protocol) το οποίο στην περίπτωση των SDN δικτύων μπορεί να χρησιμοποιηθεί στην ανακάλυψη τοπολογίας.

3.3.2 Στιβάδα Αφαίρεσης (Abstraction Layer)

Η στιβάδα αφαίρεσης είναι υπεύθυνη για την μετάφραση των αφαιρετικών μοντέλων που χρησιμοποιούνται στις λειτουργίες των ελεγκτών (όπως π.χ.Service Management, Resource Management) σε ένα data-specific μοντέλο δεδομένων[48].Το επίπεδο αφαίρεσης μπορεί να βασίζεται σε παραπάνω από ένα μοντέλα και αποτελεί ένα ενιαίο σημείο αναφοράς. Σημειώνεται εδώ ότι ο όρος στιβάδα αφαίρεσης μπορεί να αναφέρεται σε οποιαδήποτε στιβάδα χρησιμοποιείται για το διαχωρισμό δύο φυσικών ή λογικών οντοτήτων στην SDN αρχιτεκτονική. Το τυποποιητικό έγγραφο RFC7426 εισήγαγε πολλαπλές θεμελιώδεις SDN έννοιες με τη μορφή των "στιβάδων αφαίρεσης". Μία τέτοια έννοια είναι η Στιβάδα Αφαίρεσης Υπηρεσιών Δικτύου ή NSAL (Network Services Abstraction Layer) η οποία αναφέρεται στη στιβάδα που βρίσκεται μεταξύ των εφαρμογών υψηλού επιπέδου, των υπηρεσιών δικτύου και των SDN ελεγκτών. Το ίδιο έγγραφο (RFC7426) εισήγαγε και άλλα επίπεδα αφαίρεσης συμπεριλαμβανομένης της Στιβάδας Αφαίρεσης Ελέγχου ή CAL (Control Abstraction Layer), τη Στιβάδα Αφαίρεσης Διαχείρισης ή MAL (Management Abstraction Layer) και την Στιβάδα Αφαίρεσης Συσκευών και Πόρων ή DAL (Device and resource Abstraction Layer). Κάθε μία από τις προαναφερθείσες στιβάδες ή APIs παρέχουν στις υπερκείμενες στιβάδες με ένα ενιαίο τρόπο επικοινωνίας, τις απαιτήσεις της υποκείμενης στιβάδας ή των στιβάδων.

3.3.3 Στιβάδα Υπηρεσιών Δικτύου (Network Service Layer)

Η στιβάδα υπηρεσιών δικτύου αποτελείται από δομοστοιχεία/λειτουργικές μονάδες) που εκτελούν τις κύριες λειτουργίες ενός ελεγκτή SDN. Μερικά παραδείγματα τέτοιων modules παρουσιάζονται στη συνέχεια :

- **Μονάδα Διαχείρισης Υπηρεσιών (Service Management Module)**

Η λειτουργία διαχείρισης υπηρεσιών είναι υπεύθυνη για όλες τις πτυχές της δημιουργίας και διαχείρισης υπηρεσιών στο δίκτυο.Η διαχείριση υπηρεσιών ορίζει τα χαρακτηριστικά μιας υπηρεσίας όπως το πρωτόκολλο, τη δομή (π.χ. point-to-point, point-to-multipoint, multipoint-to-multipoint κ.α.), τα τελικά σημεία (end-points) και άλλα χαρακτηριστικά/ορίσματα (attributes), όπως το εύρος ζώνης,QoS, καθυστέρηση, προτεραιότητα, ασφάλεια, διαθεσιμότητα κ.α. Οι υπηρεσίες που παρέχονται από τη

Μονάδα Διαχείρισης Υπηρεσιών χρησιμοποιούνται από εξωτερικές εφαρμογές μέσω των βόρειων διεπαφών (Northbound Interfaces).

- **Μονάδα Διαχείρισης Τοπολογίας (Topology Management Module)**

Η μονάδα διαχείρισης τοπολογίας αφορά τη συντήρηση της ήδη υπάρχουσας τοπολογίας δικτύου αλλά και τον εντοπισμό κάθε νέας τοπολογικής αλλαγής. Ο εντοπισμός αυτός επιτυγχάνεται μέσω LLDP (Link Layer Discovery Protocol) μηνυμάτων τα οποία ανταλλάσονται μεταξύ των δικτυακών συσκευών.Για παράδειγμα αν το ληφθέν LLDP μήνυμα αντιστοιχεί σε ένα γνωστό μεταγωγέα τότε μια νέα σύνδεση (link) δημιουργείται στο δίκτυο.

- **Μονάδα Στατιστικών (Statistics Module)**

Ένας ελεγκτής παρέχει πολλές βασικές λειτουργίες, μία από αυτές να είναι η συλλογή στατιστικών στοιχείων, τα οποία μπορούν να διαμορφωθούν με βάση τις ανάγκες για να συμπεριληφθούν υπηρεσίες, πόροι ή και τα δύο. Αυτού του είδους οι πληροφορίες μπορούν να μεταβιβαστούν σε εξωτερικές εφαρμογές.

- **Μονάδα Διαχείρισης Προώθησης και Δρομολόγησης**

Η λειτουργία προώθησης και δρομολόγησης κανόνων είναι απαραίτητη για τον ορισμό του συνόλου των κανόνων προώθησης που πρέπει να κοινοποιούνται στις συσκευές δικτύου. Επιπλέον, πρέπει να παρέχονται οι διαδρομές μεταξύ της διεύθυνσης προέλευσης και προορισμού. Αυτή η λειτουργία απαιτεί εισροές δεδομένων (input) από άλλες υπηρεσίες, όπως η τοπολογία και η διαχείριση υπηρεσιών.

- **Μονάδα Διαχείρισης Συσκευών**

Αυτή η μονάδα είναι υπεύθυνη για τη διαχείριση των δικτυακών συσκευών και ειδοποιείται σε κάθε ενημέρωση που αφορά την κατάσταση τους (π.χ. προσθήκη, διαγραφή). Ενημερώνεται επίσης κατά την προσθήκη, ενημέρωση, αφαίρεση των IP διευθύνσεων των συσκευών αυτών.

- **Μονάδα Διαχείρισης Πόρων**

Η μονάδα αυτή επιτρέπει στα Northbound συστήματα να δημιουργούν, να τροποποιούν ή να διαγράφουν πόρους. Παρέχει επίσης λειτουργίες όπως βέλτιστη επιλογή διαδρομής (optimal path selection), ισοστάθμιση φορτίου σήραγγας (tunnel load balancing), μαζική βελτιστοποίηση (bulk optimization) και επαναδρομολόγηση της ροής (traffic re-route) κ.α.

- **Μονάδα OpenFlow**

Η μονάδα OpenFlow υπάρχει στους περισσότερους ελεγκτές ώστε να παρέχει λειτουργίες σχετιζόμενες με OpenFlow όπως μηνύματα, ενέργειες, εισροές ροής, αντιστοίχιση κανόνων ροής και στατιστικά στοιχεία. Όπως έχει ήδη επισημανθεί σε προηγούμενα κεφάλαια ένας ελεγκτής μπορεί να υποστηρίξει και άλλα πρωτόκολλα για να διαχειριστεί τους μεταγωγείς και τα υποκείμενα δίκτυα του, όπως απαιτείται από την εκάστοτε εφαρμογή.

3.3.4 Στιβάδα Northbound

Η αλληλεπίδραση/συνεργασία μεταξύ του ελεγκτή και των εφαρμογών επιτυγχάνεται μέσω μιας διεπαφής επικοινωνίας. Οι Βόρειες Διεπαφές (NB APIs), επιτρέπουν στον ελεγκτή να διευκολύνει εφαρμογές υψηλού επιπέδου, όπως η τοπολογία, οι ροές προώθησης, η εικονοποίηση του δικτύου, η ανίχνευση εισβολών στο υποκείμενο δίκτυο κλπ. Όπως έχει ήδη επισημανθεί σε προηγούμενα κεφάλαια, επί του παρόντος δεν έχει καθιερωθεί κάποιο τυπικό πρωτόκολλο για την διαχείριση της επικοινωνίας αυτής. Αντί αυτού τα APIs λογισμικού καθιστούν δυνατή τη προγραμματισιμότητα των ελεγκτών αποκαλύπτοντας υπηρεσίες δικτύου στο επίπεδο εφαρμογών.
Μέχρι στιγμής έχουν περιγραφεί διαφορετικές περιπτώσεις χρήσης ελεγκτή. Επιπλέον, έχει γίνει αναφορά στην ποικιλία δυνατοτήτων και χαρακτηριστικών που προσφέρουν οι ελεγκτές. Στη σημερινή αγορά υπάρχουν ελεγκτές ανοιχτού κώδικα (Open-Source) και ελεγκτές εμπορίου (Commercial). Όμως η επιλογή του κατάλληλου ελεγκτή εξαρτάται από διάφορους παράγοντες. Σήμερα υπάρχει πληθώρα SDN εφαρμογών και για αυτό το λόγο έχουν σχεδιαστεί αρκετοί ελεγκτές για να εξυπηρετούν τις ανάγκες των εφαρμογών αυτών.Υπάρχουν ελεγκτές σχεδιασμένοι για την υλοποίηση των WAN (Wide Area Networks), επιχειρηματικών

συγκεντρώσεων (enterprise campuses), κέντρων δεδομένων (data center) κ.ά. Δεν είναι όμως απαραίτητο ένας ελεγκτής να αποδίδει εξίσου ικανοποιητικά σε όλους τους τύπους εφαρμογών και ανάπτυξης. Ως παράδειγμα αναφέρονται οι ελεγκτές που σχεδιάζονται για ανάπτυξη των WAN, οι οποίοι μπορεί να μην αναποκρίνονται εξίσου καλά σε περιβάλλοντα Κέντρων Δεδομένων. Ακόμη πρέπει να ληφθούν υπόψη το πλήθος των εφαρμογών, το πεδίο εφαρμογής τους, όπως και ο χρόνος που αυτές οι εφαρμογές "τρέχουν".

Εκτός από τις ανωτέρω υλοποιήσεις (deployments), είναι επίσης σημαντικό να ικανοποιηθούν και οι επιχειρησιακές ανάγκες που υπάρχουν. Ειδικότερα ένας ελεγκτής θα πρέπει να είναι σε θέση να ενσωματώνει μέσω NB APIs και εφαρμογές που καλύπτουν συγκεκριμένες ανάγκες. Ακόμα η συμβατότητα ενός ελεγκτή με την συνολική αρχιτεκτονική του δικτύου είναι εξίσου σημαντική.

Αρχικά η SDN εφαρμογές για τα Κέντρα Δεδομένων (DC) απαιτούσαν ελεγκτές με ενσωματωμένες σ' αυτούς λύσεις. Στις περιπτώσεις αυτές οι ελεγκτές είχαν επικεντρωθεί στη διαχείριση των πόρων των DC, όπως οι υπολογιστές, η αποθήκευση, η απεικόνιση των εικονικών μηχανών και η κατάσταση του δικτύου. Στη συνέχεια οι ανάγκες στράφηκαν στις αφαιρέσεις (abstractions) του δικτύου και νέοι ελεγκτές αναδύθηκαν με εξειδίκευση στη διαχείριση του δικτύου. Οδηγός του νέου αυτού κύματος των ελεγκτών είναι η επέκταση των SDN εφαρμογών πέρα από τα DC, σε νέα πεδία δικτύων, όπου η διαχείριση των εικονικών πόρων (π.χ επεξεργασία και αποθήκευση) δεν είναι πλέον τόσο στενά συνδεδεμένη με την προτεινόμενη λύση. Η ενσωμάτωση (integration) με όρους εφαρμογών και δικτυακών συσκευών καθίσταται πολύπλοκη. Για παράδειγμα, ανάλογα με την περίπτωση χρήσης, ο ελεγκτής θα πρέπει να εσωματώνει πλατφόρμες Βόρειας κατεύθυνσης (NB platforms), για τη διαχείριση νέφους ή ενορχήστρωσης, όπως είναι η OpenStack ή CloudStack. Από την άλλη πλευρά οι ελεγκτές θα πρέπει να έχουν την ικανότητα ενσωμάτωσης διαφόρων τύπων μεταγωγέων, όπως είναι οι υβριδικοί, οι SDN μεταγωγείς και να είναι συμβατοί με διάφορα πρωτόκολλα που εφαρμόζονται.

Όσο η εξέλιξη των ελεκγτών ωριμάζει και κυκλοφορούν οι νέες εκδόσεις τους, η σταθερότητα των διεπαφών (APIs) θα πρέπει να λαμβάνεται σοβαρά υπόψη και να αξιολογείται κατά την επιλογή τους. Εν τέλει υπάρχουν πολλά ορίσματα (attributes) τα οποία παίζουν καταλυτικό ρόλο στην επιλογή του κατάλληλου ελεγκτή, όπως είναι η ωριμότητά του, η αξιοπιστία και η δυνατότητα ομαλής ενσωμάτωσής του στο

δίκτυο, κλπ. Στη συνέχεια παρατίθενται συνοπτικά ορισμένοι από τους ελεγκτές που κυκλοφορούν στην αγορά.

3.4 Εμπορικοί Ελεγκτές και Ελεγκτές Ανοιχτού Κώδικα

Η ύπαρξη εμπορικών και ανοιχτού κώδικα λύσεων σε ότι αφορά στους ελεγκτές δικτύων ήδη έχει αναφερθεί προηγουμένως. Λόγω της μεγάλης ανάπτυξης τα τελευταία χρόνια της SDN αρχιτεκτονικής στη δόμηση των σύγχρονων δικτύων, οι εξελίξεις στους ελεγκτές ανοιχτού κώδικα γνωρίζουν ιδιαίτερη άνθιση, με στόχο την επίλυση προβλημάτων που ανακύπτουν και την δημιουργία νέων καινοτόμων λύσεων. Από μια άλλη οπτική, η χρήση ελεγκτών ανοιχτού κώδικα σε σχέση με τους εμπορικούς ελεγκτές,δημιουργεί συνθήκες ανεξαρτησίας από στους ενδιαφερόμενους από τους προμηθευτές των τελευταίων. Το γεγονός όμως αυτό τείνει να ξεπεραστεί με δεδομένο ότι έχει καταβληθεί μεγάλη προσπάθεια για την αποφυγή εξαρτήσεων από τους προμηθευτές και με την τυποποίηση των αρχιτεκτονικών τμημάτων λογισμικού (components) όπως έγινε για παράδειγμα στην περίπτωση των Διεπαφών Νότιας Κατεύθυνσης (Southbound APIs). Στον Πίνακα ΙΙ που ακολουθεί παρουσιάζονται ενδεικτικά ελεγκτές ανοιχτού κώδικα, χωρισμένοι σε ενεργούς και ανενεργούς, όπως και οι εμπορικοί ελεγκτές με βάση το έτος 2017.

Σύμφωνα με το [26] μέχρι το 2016 ο πιο δημοφιλής ελεγκτής ανοιχτού κώδικα ήταν το OpenDayLight (ODL), με 100 και πλέον εφαρμογές, που υποστηρίζεται από το Linux Foundation, καθώς και από μια πλειάδα πολύ μεγάλων επιχειρήσεων όπως η Orange, η China Mobile, η AT &T, η T-Mobile, η Telefonica, κ.ά. Ένα άλλο σημαντικό εγχείρημα στον τομέα αυτό είναι το ONOS (Open Networking Operation System) το οποίο ξεκίνησε το 2014 και αυτό με την υποστήριξη του Linux Foundation το οποίο είναι προσανατολισμένο στην εξυπηρέτηση των αναγκών των δικτύων των παρόχων υπηρεσιών. Περισσοτερες λεπτομέρειες και στοιχεία για τους πιό σημαντικούς ελεγκτές ανοιχτού κώδικα (ODL, ONOS & Ryu) παρατίθενται στα Κεφάλαια 5,6 και 7 της παρούσας εργασίας.

ΠΙΝΑΚΑΣ ΙΙ

Κατηγοριοποίηση Ελεγκτών

Ελεγκτές Ανοιχτού Κώδικα (2017)		**Εμπορικοί Ελεγκτές (2017)**
Ενεργοί	**Ανενεργοί**	
OpenDayLight	Beacon	Cisco-ACI
ONOS	FlowER	VMware-vCloud/vSphere
Ryu	NOX	Nokia-Nouage
Floodlight	NodeFlow	Avaya
POX		Huawei
LOOM		Big Switch Network
Tremma		Jubiter-Contrail
Maestro		Ericcson
OpenMUL		Brocade
OpenContrail		
Faucet		

4. SDN Northbound Interfaces

4.1 Γενικά

Η Διεπαφή Βορείου Ορίου ή Βόρεια Διεπαφή (Northbound Interface - NBI), όπως αναφέρθηκε και στο προηγούμενο Κεφάλαιο, είναι ένας εκ των βασικών πυλώνων στην αρχιτεκτονική των Δικτύων που Καθορίζονται από Λογισμικό (SDN), καθώς μεταξύ των άλλων παρέχει και δυνατότητες αφαίρεσης στον προγραμματισμό της (programming abstraction)[2]. Λειτουργεί ως γέφυρα μεταξύ του Επιπέδου Ελέγχου και του Επιπέδου Διαχείρισης και όπως αναφέρθηκε παρέχει δυνατότητες υψηλού βαθμού "αφαίρεσης" στην ανάπτυξη των εφαρμογών (application development). Η ανάπτυξη εφαρμογών στο Επίπεδο Διαχείρισης δεν είναι τόσο εύκολη και ένας από τους κύριους λόγους είναι η έλλειψη τυποποίησης στον τομέα των NBIs.

Σε αντίθεση με το Πρωτόκολλο OpenFlow που χρησιμοποιείται στις SBI's (Διεπαφές Νότιας Κατεύθυνσης) δεν υπάρχει ένα μοναδικό API (Application Programming Interface) ή πρωτόκολλο που μπορεί να χρησιμοποιηθεί από τους προγραμματιστές και προμηθευτές για την ικανοποίηση των απαιτήσεών τους στη λειτουργία του NBI's. Τούτο οφείλεται στις παραλλαγές και τις απαιτήσεις που αυτοί θέτουν. Η Ομάδα Εργασίας που δημιούργησε ο ONF (Open Networking Foundation) [49] για την τυποποίηση των NBI's επιχειρεί να δώσει απαντήσεις και ολοκληρωμένες τυποποιημένες λύσεις υπό μορφή πρωτοκόλλων στα εμπλεκόμενα μέρη (π.χ προγραμματιστές, προμηθευτές υλισμικού και λογισμικού).

Λόγω της απουσίας τυποποίησης στα NBI's,όπως προαναφέρθηκε, χρησιμοποιούνται ορισμένοι Ελεγκτές (π.χ Onix, PANE κλπ) που παρέχουν υψηλές δυνατότητες για ανάπτυξη εφαρμογών API's στα SDN δίκτυα, αν και οι περισσότεροι προγραμματιστές και ελεγκτές χρησιμοποιούν το REST για ανάπτυξη εφαρμογών .

Η βιβλιογραφία σχετικά με τα NBI's, επικεντρώνεται κυρίως στις ακόλουθες ιδιότητες των διεπαφών αυτών. Στη φορητότητα (Portability), στη δυνατότητα προγραμματισμού (Programmability), στις βόρειες διεπαφές βασιζόμενες στον

[2]Αφαίρεση στον προγραμματισμό (Programming Abstraction) σημαίνει η εμφάνιση μόνο των σημαντικών πληροφοριών και η απόκρυψη άλλων.Με άλλα λόγια η αφαίρεση δεδομένων αναφέρεται μόνο στην παροχή σημαντικών πληροφοριών προς τα έξω, και την απόκρυψη background λεπτομερειών και στοιχείων της εφαρμογής.

ελεγκτή (Controller-based NBI'S), στις βόρειες διεπαφές σκοπού (Intent-based NBI'S), και στην εικονοποίηση (Virtualization).

4.2 Φορητότητα (Portability)

Μια από τις αιτίες που εμποδίζουν την ανάπτυξη εφαρμογών NBI's, είναι το κενό μεταξύ των κατασκευαστών υλισμικού και των προγραμματιστών εφαρμογών, γεγονός που μειώνει τη φορητότητα, δηλαδή την εγγύηση της σωστής επεξεργασίας πακέτων και της καλής απόδοσης τους σε ένα ευρύ φάσμα μεταγωγέων που χρησιμοποιούνται στο δίκτυο. Υπάρχει ένας μεγάλος αριθμός διαφορετικών μεταγωγέων υλισμικού και λογισμικού, που διατίθενται από δεκάδες προμηθευτές, οι οποίες διαφέρουν στο επίπεδο δεδομένων, στην αλληλεπίδραση μεταγωγέα - ελεγκτή και στη σταθερή/ευέλικτη προώθηση των πληροφοριών.

Η χρήση του λογισμικού Friendly Networking (SFNet) [50] είναι ένα παραδειγμα διεπαφής μεταξύ Επιπέδου Ελέγχου και Επιπέδου Εφαρμογής SDN, που έχει ως ρόλο την απόκρυψη των υποκείμενων πρωτόκολλων του δικτύου από την εφαρμογή. Πρόκειται για API υψηλού επιπέδου που αλληλεπιδρά άμεσα με το υποκείμενο δίκτυο. Μεταφράζει κατάλληλα τις απαιτήσεις εφαρμογής και τα προγράμματα των δίκτυων για την παροχή υπηρεσιών. Χρησιμοποιεί αρχεία JSON (JavaScript Object Notation)[3] για να στείλει αιτήματα παροχής πληροφοριών και έκθεση συμφόρησης (Congestion Report) από το δίκτυο, προκειμένου να βρεθεί μια καλύτερη διαδρομή στις διακινούμενες από το σύστημα πληροφορίες. Στις περιπτώσεις συμφόρησης, επιτρέπει στις εφαρμογές να υπαναχωρήσουν αν αποφασίσουν κάτι τέτοιο και μπορεί να είναι χρήσιμο σε περίπτωσεις καθυστέρησης ευαίσθητης μεταφοράς δεδομένων. Υποστηρίζει ακόμα επιφυλάξεις σχετικά με απαιτουμενο εύρος ζώνης και χορηγεί ή απορρίπτει το αίτημα ανάλογα με τη διαθεσιμότητα του απαιτούμενου εύρους ζώνης. Αυτή η απαίτηση μπορεί να προτεραιοποιηθεί για βίντεο κατόπιν ζήτησης, ή για μεταφορά Voice over IP (VoIP). Υποστηρίζει επίσης πολυεκπομπή (Multicasting) σε ένα σύνολο διευθύνσεων IP συμμετέχοντας σε αυτήν.

[3] Το **JSON (JavaScript Object Notation)** είναι ένα αρχείο που αποθηκεύει απλές δομές δεδομένων και αντικείμενα και αποτελεί μια τυπική μορφή ανταλλαγής δεδομένων. Χρησιμοποιείται κυρίως για τη μετάδοση δεδομένων μεταξύ μιας εφαρμογής ιστού και ενός διακομιστή. Τα αρχεία JSON είναι ελαφριά, βασισμένα σε κείμενο, αναγνώσιμα και μπορούν να υποστούν επεξεργασία επεξεργαστούν μέσω ενός επεξεργαστή κειμένου. https://fileinfo.com/extension/json

Το NOSIX [51] είναι μια διεπαφή, η οποία ασχολείται με το θέμα της βελτίωσης της απόδοσης ενός μεγάλου αριθμού μεταγωγέων, στο υποκείμενο σύστημα του δικτύου. Προτείνει τη χρήση Εικονικών Πινάκων Ροής (Virtual Flow Tables - VFT), οι οποίοι αποτελούν τα βασικα στοιχεία που χρησιμοποιούνται από τις εφαρμογές, για τον καθορισμό των κανόνων σχετικά με τη ροή των ενημερώσεων και των κοινοποιήσεων. Επιτρέπει στις εφαρμογές να προκαθορίζουν τις διαδρομές (διελεύσεις) των Εικονικών Πινάκων Ροής (Pipeline VFTs) και στη συνέχεια, την εγκατάσταση των κανόνων λειτουργίας τους. Ο προκαθορισμός της διαδρομής επιτρέπεται, επειδή είναι πολύ δύσκολο να γίνει δυναμική αναδιάταξη της φυσικής διαδρομής των μεταγωγέων. Επιπλέον, οι κανόνες για τους VFTs δεν χρειάζεται να βρίσκονται πάντα σε πίνακες φυσικής ροής ενός μεταγωγέα. Από την άλλη πλευρά, οι οδηγοί του μεταγωγέα (Switch Drivers)[4] απεικονίζουν τη δομοσειρά (Pipeline) των Εικονικών Πινάκων Ροής (VFT) πάνω σε μια φυσική δομοσειρά που είναι διαθέσιμη στον μεταγωγέα. Ο οδηγός του μεταγωγέα (switch driver) μπορεί να είναι τοποθετημένος είτε σε ένα υποκείμενο επίπεδο του ελεγκτή, είτε στον ίδιο τον μεταγωγέα. Θα ήταν ωστόσο χρησιμότερο οι οδηγοί των μεταγωγέων να βρίσκονται σε μια υποκείμενη στοίβα (Stack) του ελεγκτή, επειδή με τον τρόπο αυτό καθίσταται ευκολότερος ο προγραμματισμός του λογισμικού του στο επίπεδο του ελεγκτή, παρά στον μεταγωγέα. Στο NOSIX, οι εφαρμογές ελέγχου μπορούν να γραφούν (προγραμματιστούν) υπό μορφή δομοσειράς Εικονικών Πινάκων Ροής (VTF) και στη συνέχεια να μετασχηματιστούν σε διαμορφωμένους μεταγωγείς μέσω οδηγών (drivers) που μπορεί κάποιος να προμηθευτεί από την αγορά.

Μια άλλη λύση αποτελεί το tinyNBI [52], που είναι μια απλοποιημένη, ανεξάρτητη από τη γλώσσα προγραμματισμού διεπαφή (API), η οποία στηρίζεται στις πέντε υπάρχουσες εκδόσεις του Openflow, προκειμένου να επιλυθούν θέματα φορητότητας με διαφορετικό και πιο εύκολο τρόπο. Η συγκεκριμένη διεπαφή NBI σχεδιάστηκε σε γλώσσα C και παρέχει και διαχειρίζεται τις δυνατοτήτες όλων των εκδόσεων OpenFlow στους μεταγωγείς, χωρίς να απαιτείται πρόσθετη εργασία, παρέχοντας παράλληλα τη δυνατότητα ανάπτυξης και άλλων εφαρμογών. Ο κύριος σκοπός του

[4] Ονομάζουμε **οδηγό συσκευής** (Device Driver) το *λογισμικό* εκείνο που συνοδεύει συνήθως ένα περιφερειακό ενός *υπολογιστή,* π.χ. ένα *αποθηκευτικό μέσο,* όπως μια συσκευή CD-ROM, και το οποίο πρέπει να εγκατασταθεί στον υπολογιστή προκειμένου το περιφερειακό να λειτουργήσει. Κάθε συσκευή είναι κατασκευασμένη σύμφωνα με προδιαγραφές που ορίζονται συνήθως από διεθνείς οργανισμούς ή από συμφωνίες μεγάλων κατασκευαστών. Ο οδηγός συσκευής είναι ένα πρόγραμμα που επιτρέπει την επικοινωνία της συσκευής και του *λειτουργικού συστήματος*. https://el.wikipedia.org/wiki/Οδηγός_συσκευής

tinyNBI είναι να παρέχει μια βασική διεπαφή για την ανάπτυξη εφαρμογών. Χρησιμοποιεί ένα μοντέλο δεδομένων, που κάνει σαφή διάκριση μεταξύ των αφαιρέσεων στο επίπεδο ελέγχου και στο επίπεδο δεδομένων. Κάθε αφαίρεση περιλαμβάνει τρία τμήματα λογισμικού (components), τη διαμόρφωση (Configuration), τις δυνατότητες (Capabilities) και τις στατιστικές (Statistics). Η διαμόρφωση αφορά στην δυνατότητα τροποποίησης των δεδομένων από τη διεπαφή. Οι δυνατότητες, καθορίζουν τη συμπεριφορά της αφαίρεσης και είναι μη τροποποιήσιμη κατάσταση. Ενώ οι στατιστικές είναι δεδομένα μόνο για ανάγνωση και περιγράφουν τον τρόπο που έχουν λειτουργήσει οι αφαιρέσεις.

Ολες οι αφαίρεσεις δεν υπάρχουν σε όλες τις εκδόσεις OpenFlow (πχ. Meter Table, Group Table). Οι μη διαθέσιμες αφαιρέσεις είναι διαχειρίσιμες μέσω της απρόσκοπτης εξομοίωσης (Seamless Emulation), της εκφόρτωσης του μεταγωγέα(switch offloading), και των ένδειξεων σφάλματος (error indication).

Στον Πίνακα ΙΙΙ παρουσιάζοντα συνοπτικά οι διαφορετικές προτάσεις για φορητότητα σε περιβάλλον NBI. Αυτές οι λύσεις επικεντρώνονται είτε σε συσκευές επίπεδου δεδομένων είτε πρωτόκολλα, αλλά μία μόνο λύση και για τα δύο αυτά, αποτελεί μια σημαντική ερευνητική πρόκληση.

ΠΙΝΑΚΑΣ ΙΙΙ

Συνοπτική Παρουσίαση Των Λύσεων Φορητότητας στα NBI's

Βιβλιογραφία	Σκοπός	Λύση	Πλεονεκτήματα
SFNet	Αλληλεπίδραση μεταξύ εφαρμογών και υποκείμενων συσκευών	Χρήσεις πρόσθετων διεπαφών για διάφορες εφαρμογές	Επιτρέπει περιορισμούς εύρους ζώνης, πολυεκπομπή και υποστηρίζει την έρευνα συμφόρησης
tinyNBI	Στοιχειώδης διεπαφή βασισμένη στις εκδόσεις Πρωτοκόλλων OpenFlow	Χρήσεις μοντέλων δεδομένων για αφαίρεση προδιαγραφών	Υποστηρίζει πολλαπλές εκδόσεις OpenFlow και παρέχει επεκτασιμότητα
NOSIX	Ποικιλομορφία μεταγωγέων και βελτίωση της απόδοσης	Εικονικοί Πίνακες Ροής (VFT) και οδηγοί μεταγωγέων (Switch Drivers)	Παρέχει ευελιξία στους προγραμματιστές για τη δημιουργία ενός διαφορετικού τοπίου για τις συσκευές του Επιπέδου Δεδομένων (Data plane)

Συμπερασματικά, μια από τις σημαντικότερες λειτουργίες της Βόρειας Διεπαφής (NBI) είναι η παροχή πληροφοριών των συσκευών του υποκείμενου δικτύου στους προγραμματιστές. Υπάρχει ποικιλία συσκευών και χρησιμοποιούμενων πρωτοκόλλων στο υποκείμενο σύστημα του δικτύου. Η φορητότητα παρέχει λύσεις σε θέματα

συμβατότητας αναφορικά με την ποικιλία των στοιχείων του δικτύου και των χρησιμοποιούμενων πρωτοκόλλων.

4.3 Προγραμματισιμότητα (Programmability in NBI)

Όπως συνέβη στην ανάπτυξη εφαρμογών, όπου η γλώσσα Assembly[5] πραγματοποίησε στροφή στη χρήση γλωσσών υψηλού επιπέδου, όπως Python και Java, έτσι και οι χρησιμοποιούμενες οι γλώσσες δικτύου στράφηκαν επίσης από τη χρήση χαμηλού επιπέδου γλωσσών (π.χ. γλώσσα χαμηλού επιπέδου είναι αυτή που χρησιμοποιείται στο OpenFlow) σε υψηλού επιπέδου, όπως η Frenetic, η Procera κ.ά. Η χρήση των τελευταίων στο SDN, παρέχει ευελιξία στη διαχείριση δικτύου και το καθιστά λιγότερο επιρρεπές σε σφάλματα.

Τα στοιχεία (elements) του δικτύου συχνά εκτελούν ταυτόχρονα πολλαπλές εργασίες, όπως για παράδειγμα η δρομολόγηση, η παρακολούθηση, ο έλεγχος πρόσβασης κ.λ.π. Ωστόσο, η αποσύνδεση αυτών των εργασιών είναι σχεδόν αδύνατη, διότι οι εγκαταστημένοι κανόνες χειρισμού πακέτων μπορεί να προκαλέσουν συγκρούσεις μεταξύ των στοιχείων του δικτύου. Η διεπαφή OpenFlow καθορίζεται από χαμηλό επίπεδο της αφαίρεσης, το οποίο προσδιορίζει τις δυνατότητες του υλισμικού μεταγωγής. Για την εφαρμογή γλωσσών υψηλού επιπέδου, οι προγραμματιστές δεν μπορούν άμεσα να χρησιμοποιούν το πακέτο εντολών του πρωτοκόλλου OpenFlow. Ένα άλλο θέμα με το OpenFlow είναι ότι τα πακέτα επεξεργάζονται από τον ελεγκτή, εφόσον ο μεταγωγέας δεν είναι σε θέση να το κάνει λόγω έλλειψης πληροφοριών ροής και ως εκ τούτου οι προγραμματιστές θα πρέπει να προβούν σε προγραμματισμό δύο βαθμίδων, έναν για τα πακέτα που υποβάλλονται σε επεξεργασία στον ελεγκτή και έναν άλλο που χρειάζονται για την επεξεργασία στο επίπεδο του μεταγωγέα. Η βιβλιογραφία σχετικά με τις γλώσσες προγραμματισμού στην Βόρεια Διεπαφή (NBI) παρουσιάζεται στη συνέχεια, αφού προηγουμένως, εντοπίζονται οι διαφορετικές ιδιότητες αυτών των γλωσσών.

[5]Η γλώσσα Assembly είναι μια γλώσσα προγραμματισμού χαμηλού επιπέδου για έναν υπολογιστή ή άλλη προγραμματιζόμενη συσκευή, που είναι καθορισμένη σε μια συγκεκριμένη αρχιτεκτονική υπολογιστή, σε αντίθεση με τις περισσότερες γλώσσες προγραμματισμού υψηλού επιπέδου, οι οποίες είναι γενικά μπορούν να χρησιμοποιούνται σε πολλαπλά συστήματα. Η γλώσσα αυτή μετατρέπεται σε εκτελέσιμο κώδικα μηχανής από ένα βοηθητικό πρόγραμμα, που αναφέρεται ως συναρμολογητής όπως NASM, MASM κ.λπ. https://www.tutorialspoint.com/assembly_programming/

4.3.1 Ταξινόμηση Χαρακτηριστικών Γλωσσών Προγραμματισμού στα NBI's

Η εγκατάσταση ροής (Flow Installation) αναφέρεται στον τρόπο με τον οποίο οι κανόνες προώθησης μπορούν να εγκατασταθούν στους μεταγωγείς : δηλαδή ενεργητικά (Reactive) και προληπτικά (Proactive). Σχεδόν όλες οι γλώσσες παρέχουν δυνατότητες ενεργητικής εγκατάστασης, όμως ορισμένες προσφέρουν και προληπτικές δυνατότητες εγκατάστασης ροής. Στην ενεργητική προσέγγιση, όταν ένα νέο πακέτο δεδομένων φτάνει στο μεταγωγέα, και δεν υπάρχουν διαθέσιμες ορισμένες πληροφορίες ροής, τότε συγκεκριμένο το πακέτο προωθείται στον ελεγκτή. Η διαδικασία αυτή βασίζεται στην προγραμματισμένη λογική εγκαταστημένων ροών στον πίνακα ροής του μεταγωγέα. Αυτή η προσέγγιση είναι προφανές ότι δημιουργεί καθυστερήσεις, αφού κάθε νέο πακέτο θα σταλεί στον ελεγκτή για να πραγματοποιήσει την εγκατάσταση ροής. Με την προληπτική προσέγγιση, εξαλείφονται οι καθυστέρησεις, αφού κάθε νέο πακέτο δεν χρειάζεται να προωθηθεί στον ελεγκτή για εγκατάσταση κανόνων προώθησης, αφού έχουν εγκατασταθεί προκαθορισμένοι κανόνες και ενέργειες στους πίνακες ροής. Έτσι οι γλώσσες εκτελούν προληπτική εγκατάσταση και προ-υπολογισμό πινάκων προώθησης για ολόκληρο το δίκτυο και ο ελεγκτής παρεμβαίνει μόνο για να τροποποιήσει τους πίνακες ροής σε περίπτωσεις βλάβης της σύνδεσης (link) ή άλλων εξωτερικών συμβάντων.

Ο Ορισμός Πολιτικής (Policy Definition) είναι ένα σύνολο κανόνων για το δίκτυο και κάθε πολιτική είναι ένα σύνολο συνθηκών που συνοδεύονται από ένα σύνολο δράσεων (ενεργειών) που καθορίζουν ποιός κανόνας πολιτικής θα εφαρμοστεί κάθε φορά. Οι ορισμοί των πολιτικών ταξινομούνται σε στατικές και δυναμικές πολιτικές. Οι στατικές πολιτικές είναι ένα σύνολο προκαθορισμένων κανόνων και ενεργειών με τα φίλτρα τείχους προστασίας (Firewall Filters) να αποτελούν το πιο παραδοσιακό μέρος αυτών, ενώ οι δυναμικές πολιτικές μπορούν να ρυθμίζονται ανάλογα με τις συνθήκες του δικτύου. Αν και οι περισσότερες γλώσσες παρέχουν δυναμικές δυνατότητες ορισμού πολιτικών, υπάρχουν και ορισμένες που λειτουργούν μόνο με τον ορισμό στατικής πολιτικής.

Το Πρότυπο Προγραμματισμού (Programming Paradigm) αντικατοπτρίζει τον τρόπο με τον οποίο κατασκευάζονται η δομή και τα στοιχεία κάθε προγράμματος. Υπάρχουν δύο πρότυπα στον προγραμματισμό το SDN δικτύων: τα υποχρεωτικά

πρότυπα προγραμματισμού *(Imperative Programming Paradigm)* και τα δηλωτικά πρότυπα προγραμματισμού *(Declarative Programming Paradigm)*. Το υποχρεωτικό πρότυπο μπορεί να χαρακτηριστεί ως παραδοσιακή προγραμματική δομή, η οποία επιτρέπει στον προγραμματιστή να καθορίσει όλα τα βήματα για την επίλυση ενός συγκεκριμένου προβλήματος. Στο δηλωτικό πρότυπο ο προγραμματιστής πρέπει να καθορίσει τι πρόγραμμα πρέπει να υλοποιήσει και όχι πως να το υλοποιήσει. Το πιο χαρακτηριστικό παράδειγμα δηλωτικού προτύπου προγραμματισμού σχετίζεται με τη Δομημένη Γλώσσα Αναζήτησης (Structured Query Language - SQL), όπου καθορίζεται η αναζήτηση και η μηχανή αναζήτησης δεδομένων που την εκτελεί. Το δηλωτικό πρότυπο, έχει επιπλέον τρία υποσύνολα προτύπων : του λογικού προγραμματισμού (Logic Programming), του προγραμματισμού βάσει συμβάντων (Event Driven Programming) και του λειτουργικού προγραμματισμού (Functional Programming). Στο λογικό προγραμματισμό, ο μεταγλωττιστής εφαρμόζει ένα αλγόριθμο ο οποίος σαρώνει όλους τους δυνατούς συνδυασμούς σε ένα σύνολο καθορισμένων συμπερασμάτων, προκειμένου να αξιολογήσει και να απαντήσει σε μια αναζήτηση. Ο προγραμματισμός συμβάντων, επιτρέπει στο πρόγραμμα να αναταποκριθεί σε οποιοδήποτε συγκεκριμένο συμβάν. Μόλις αυτό παραληφθεί αυτομάτως ενεργοποιείται μια δράση/ενέργεια. Η τελευταία αυτή μπορεί να είναι, είτε υπολογιστικής μορφής (κάποιος υπολογισμός), είτε να ενεργοποιεί μια άλλη ενέργεια.Ο λειτουργικός προγραμματισμός λειτουργεί για την αξιολόγηση ορισμένων μαθηματικών λειτουργιών για την αποφυγή αλλαγών στην υπάρχουσα κατάσταση. Ο υποχρεωτικός προγραμματισμός (Imperative Programming) στο δηλωτικό πρότυπο διευκολύνει τα προγράμματα να αντιδράσουν σε εξωτερικά συμβάντα. Ας πάρουμε για παράδειγμα ένα υπολογιστικό φύλλο, το οποίο στις κυψέλες του έχει τιμές ή υπολογιστικούς τύπους. Όποτε λαμβάνουν χώρα αλλαγές στις κυψέλες, οι υπολογιστικοί τύποι προσαρμόζονται αυτομάτως. Ο συνδυασμός του λειτουργικού (Functional Programming) και του ενεργητικού προγραμματισμού (Reactive Programming), συγκροτεί τον Λειτουργικό Ενεργητικό Προγραμματισμό (Functional Reactive Programming - FRP)[6] το οποίο μοντελοποιεί την ενεργητική συμπεριφορά σε λειτουργικές γλώσσες.

[6]**Functional reactive programming** (**FRP**) is a programming paradigm for reactive programming (asynchronous dataflow programming) using the building blocks of functional programming (e.g. map, reduce, filter). FRP has been used for programming graphical user interfaces (GUIs), robotics, games, and music, aiming to simplify these problems by explicitly modeling time.

4.3.2 Γλώσσες Προγραμματισμού στα NBI's

Στη συνέχεια παρουσιάζονται οι γλώσσες προγραμματισμού που έχουν σχεδιαστεί για χρήση στις Βόρειες Διεπαφές (NBIs) ενός SDN δικτύου. Στον Πίνακα IV αυτές οι γλώσσες ταξινομούνται με βάση τα χαρακτηριστικά τους.

Η γλώσσα προγραμματισμού Frenetic,[53] η οποία είναι ενσωματωμένη στην Python, προτείνει δύο επίπεδα αφαίρεσης (abstraction) περιλαμβάνοντας ένα σύνολο χειριστών επιπέδου πηγής (source level operators) για την κατασκευή και τον χειρισμό ροών κίνησης δικτύου, και ένα δηλωτικό (declarative) σύστημα, το οποίο χειρίζεται όλες τις λεπτομέρειες της εγκατάστασης και της απεγκατάστασης κανόνων χαμηλού επιπέδου στους μεταγωγείς. Πρόκειται για μια δηλωτική λύση αρθρωτού (στοιχειακού) σχεδιασμού(Modular Design).Χρησιμοποιώντας τη γλώσσα Frenetic,οι προγραμματιστές δεν χρειάζεται να ανησυχούν για την εγκατάσταση κανόνα ροής που μπορεί να εμποδίσει τον ελεγκτή να αναλύει την άλλη κίνηση (traffic) στο δίκτυο. Η γλώσσα διαχείρισης βάσει ροής (Flow Management Language - FML) [54], είναι μια υψηλού επιπέδου δηλωτική γλώσσα, βασισμένη στο αρχείο καταγραφής δεδομένων (Data Log), για πολιτική διαμόρφωσης, σχετικά με την πιστότητα των καθηκόντων διαχείρισης, μέσα σε ένα ενιαίο πλαίσιο δικτύων για μεγάλες επιχειρήσεις. Κύριο θέμα με το FML είναι ότι εφαρμόζει πολιτική σε όλα τα πακέτα για οποιαδήποτε συγκεκριμένη ροή και δεν παρέχει μεγάλη ευελιξία.

ΠΙΝΑΚΑΣ IV

Συνοπτική Παρουσίαση των Χαρακτηριστικών των NBI Γλωσσών

Βιβλιογραφία	**Εγκατάσταση Ροής**		**Ορισμός Πολιτικής**	**Πρότυπο Προγραμ/σμού (*)**	**Περιγραφή**
	Ενεργητική	**Προληπτική**			
Frenetic	+	-	Δυναμική	FR	Σχεδιασμένη για την αποφυγή συνθηκών ανταγωνισμού με την χρήση καλά καθορισμένων υψηλού επιπέδου προγραμματισμένων αφαιρέσεων.
FML	+	-	Στατική	L	Σχεδιασμένη για ορισμό πολιτικών για δίκτυα επιχειρήσεων.
Flog	+	-	Δυναμική	L/ED	Εκτελεί προγράμματα σχετικά με την εμφάνιση συμβάντων στο δίκτυο.
Procera	+	-	Δυναμική	FR	Είναι μια εκφραστική γλώσσα προγραμματισμού, η οποία χρησιμοποιείται για χειρισμό πολιτικών και παρέχει ένα επεκτάσιμο και συνθετικό πλαίσιο λειτουργίας.
Pyretic	+	+	Δυναμική	I	Πρόκειται για αναβάθμιση της Frenetic, που χρησιμοποιείται για χειρισμό πολιτικών σε διαφανή δίκτυα.
Nettle	+	+	Δυναμική	FR	Επιτρέπει στους προγραμματιστές να ασχολούνται με ροές αντί συμβάντα και μπορεί να εκληφθεί ως λειτουργία σήματος.
NetKAT	+	+	Δυναμική	F	Παρέχει λογική για τη χαρτογράφηση του δικτύου και την απομόνωση της κυκλοφορίας, χρησιμοποιώντας

					άλγεβρα Kleene & Tests.
NetCore	+	+	Δυναμική	FR	Παρέχει τα μέσα για πολιτικές προώθησης πακέτων δεδομένων και δημιουργεί εντολές για εγκατάσταση ροών.
FlowLog	+	+	Δυναμική	F	Επιτρέπει στους προγραμματιστές να χρησιμοποιούν εξωτερικές βιβλιοθήκες με πλήρη χαρακτηριστικά.
FatTire	+	+	Δυναμική	F	Παρέχει ανοχή σφάλματος και περιγράφει διαδρομές δικτύου.
Kinetic	+	+	Δυναμική	ED	Ειδική γλώσσα τομέα (Domain), που επιτρέπει τον δυναμικό έλεγχο του δικτύου.
Merlin	+	-	Δυναμική	F	Αναθέτει επι μέρους πολιτικές σε διαφορετικούς κατόχους (tenants), επιτρέποντάς τους να τις τροποιούν αναλόγως των αναγκών τους.

(*) FR= Functional Reactive, L= Logic, F= Functional, ED= Event Driven, I= Imperative

Το Flog [55] είναι μια άλλη γλώσσα SDN βασισμένη στα γεγονότα και την αλυσίδα προώθησης (event driven and forward chaining), που συνδυάζει την FML και την Frenetic καθώς ακολουθεί την τεχνική λογικού προγραμματισμού όπως η FML και αναλυτικά προγράμματα τριών μερών όπως η Frenetic : μια μέθοδο για την αναζήτηση της κατάστασης του δικτύου, ένα τμήμα λογισμικού (component) για την επεξεργασία δεδομένων μετά την αναζητηση, και ένα μηχανισμό δημιουργίας κανόνων για την εγκατάσταση στοιχείων του δικτύου. Χρησιμοποιεί πρότυπα προγραμματισμού σκοπού και λογικού προγραμματισμού και προγράμματα της συγκεκριμένης γλώσσας, τα οποία εκτελούνται κατά την εμφάνιση ενός συμβάντος στα δίκτυο. Για το σχεδισμό των πολιτικών του δικτύου, μια γλώσσα πρέπει να διαθέτει εκφραστική επάρκεια για την καταγραφή αυτών των πολιτικών.

Η Procera, [56] είναι μια γλώσσα προγραμματισμού που συγκρινόμενη με την FML, παρέχει μεγαλύτερες εκφραστικές δυνατότητες και επιτρέπει τον χειρισμό πολιτικών με καλύτερο τρόπο. Επιτρέπει στους φορείς που λειτουργούν (operators) τα δίκτυα να καθορίζουν πολιτικές που αντιδρούν καλύτερα στις δυναμικές αλλαγές υπό διαφορετικές συνθήκες δικτύων. Ορισμένοι προγραμματιστές προτείνουν μια αναβάθμιση της Procera, γνωστή ως Pyretic[7],η οποία είναι μια πλατφόρμα βασισμένη στην Python, προκειμένου να είναι σε θέση να σχεδιάσουν πιο προηγμένες εφαρμογές στο δίκτυο. Με τον τρόπο αυτό επιχειρείται να αντιμετωπιστούν ελλείψεις και κενά του OpenFlow, τόσο ως προς τη φύση του, όσο και ως τον προγραμματισμό της

[7] Η Pyretic είναι μέλος της οικογένειας γλωσσών προγραμματισμού Frenetic για τα SDN δίκτυα. Πρόκειται για μια γλώσσα προγραμματισμού υψηλού επιπέδου, ανοικτού κώδικα, η οποία επιτρέπει στον προγραμματιστή τη δημιουργία ή τροποποίηση μια πολιτική του δικτύου με υψηλό επίπεδο αφαίρεσης, χωρίς τη χρήση μηχανισμών χαμηλού επιπέδου. https://www.researchgate.net/publication/315837452_Pyretic_SDN_Programming_Language

διεπαφής του μεταγωγέα στο δίκτυο. Η γλώσσα Pyretic υποστηρίζει τον αρθρωτό προγραμματισμό και διευκολίνει επίσης την δημιουργία δυναμικών πολιτικών.

Η NetCore [57] ως γλώσσα προγραμματισμού, παρέχει δυνατότητες πολιτικών για την προώθηση πακέτων, είναι εκφραστική[8], με δυνατότητες σύνθεσης και με καθορισμένες ερμηνείες (formal semantics). Αντί να χρησιμοποιεί τον SDN Ελεγκτή, το NetCore παρέχει μια συλλογή αλγορίθμων, τους οποίους συνδέει με το χρόνο εκτέλεσης του προγράμματος, δίνοντας εντολές κανόνων εγκατάστασης ροής. Η NetCore είναι αποκλειστικά συνδεδεμένη με την απλοποίηση των πινάκων ροής, αλλά στερείται εκφραστικότητας.

Η FlowLog[9] είναι μια γλώσσα προγραμματισμού χωρίς βαθμίδες (tier-less) που παρέχει δυνατότητες χρήσης εξωτερικών βιβλιοθηκών.

Η Nettle, [58] στηρίζεται στην φιλοσοφία του προγραμματισμού του δικτύου αντί της ρύθμισής του και υιοθετεί στοιχεία από άλλες γλώσσες προγραμματισμού όπως η (FRP), μεθοδολογία σχεδιασμού της Domain Specific Language (DSL) και ενσωματώνει ισχυρά στοιχεία της γλώσσας Haskell, που λειτουργεί ως γλώσσα υποδοχής[10].

Μια άλλη πρόταση για τον προγραμματισμό SDN δικτύων είναι η γλώσσα NetKAT [59], η οποία αποτελεί ένα πλαίσιο για τον προσδιορισμό, τον προγραμματισμό και τη λογική των δικτύων SDN, στηριζόμενη στην άλγεβρα Kleene για δοκιμές. Παρέχει θεωρητικές εξισώσεις για προσβασιμότητα, για την απομόνωση της κυκλοφορίας, και την ορθότητα των αλγορίθμων του μεταγλωττιστή. Είναι εμπνευσμένη από την NetCore, η οποία τροποιήθηκε και επεκτάθηκε σε κρίσιμους τομείς για να ανταποκρίνεται στο KAT[11], όπου η NetCore δεν είναι σε θέση να ανταποκριθεί.

Η FatTire [60], ως γλώσσα προγραμματισμού, παρέχει πολιτικές σχετικά με την προώθηση και τις ανοχές των σφαλμάτων. Επιτρέπει στους προγραμματιστές να

[8] Εκφραστική γλώσσα προγραμματισμού σημαίνει ότι έχει δυνατότητα ευκολίας γραφής κώδικα και γίνεται εύκολα κατανοητή τόσο από τον μεταγλωτιστή όσο και από τον αναγνώστη. https:// stack over flow. com/ questions/638881/what-does-expressive-mean-when-referring-to-programming-languages

[9] https://www.usenix.org/node/179782

[10] https://www.researchgate.net/publication/228712513_Nettle_Functional_Reactive_Programming_for_OpenFlow_Networks

[11] Το KAT (K Analysis Toolkit) είναι μια βιβλιοθήκη εργαλείων γραμμένη στην ίδια την K για την ανάλυση γλωσσών προγραμματισμού και προγραμμάτων. https://github.com/kframework/kat/blob/master/kat.md .
Η K είναι μια γλώσσα κειμένου που προορίζεται για σχεδιαστές /ή και προγραμματιστές για το σχεδιασμό και την υλοποίηση ενός συστήματος. http://www.theklanguage.com/tutorial.html

καθορίζουν νόμιμες διαδρομές σε όλο το δίκτυο μαζί με τις ανοχές αυτών των διαδρομών. Επειδή οι συνθήκες του δικτύου μεταβάλλονται συνεχώς, οι προγραμματιστές πρέπει να αλλάζουν τη διαμόρφωση χειροκίνητα.

Η Kinetic [61], η οποία βασίζεται στην Pyretic, προτείνει ένα ευκατανόητο (intuitive) μηχανισμό για την δυναμική αλλαγή των διαμόρφωσεων. Εκφράζει την πολιτική του δικτύου, ως ένα Finite State Machine - FSM (Μηχανή Πεπερασμένης Κατάστασης) να καταγράφει τη δυναμική και την ικανότητα για επαληθεύσεις. Επαληθεύει επίσης την ορθότητα των υψηλού επιπέδου προδιαγραφών.

Για το έλεγχο των δικτύων, οι διαχειριστές οφείλουν να είναι προσεκτικοί ως προς τις διαμορφώσεις του δικτύου, γιατί κάποιο ενδεχόμενο λάθος διαμόρφωσης μια συσκευής μπορεί να προκαλέσει ανεπιθύμητη συμπεριφορά σε όλο το δίκτυο. Με τη χρήση της γλώσσας Merlin [62], οι διαχειριστές του δικτύου μπορούν να καθορίσουν τις πολιτικές σε ένα υψηλό επίπεδο δηλωτικής γλώσσας. Ο μεταγλωττιστής Merlin χρησιμοποιεί πρόγραμμα κατάτμησης (Program Partitioning) για την μετατροπή των συνολικών πολιτικών, σε μικρότερες υπο-πολιτικές. Οι τελευταίες διανέμονται αυτομάτως στα επιμέρους στοιχεία του δικτύου, και ανατίθενται σε διαφορετικούς κατόχους, οι οποίοι μπορούν να τις τροποποιήσουν αναλόγως με τις δικές τους εξειδικευμένες απαιτήσεις.

Για την παροχή υπηρεσιών ποιότητας (QoS) στη δρομολόγηση και την κυκλοφορία χρησιμοποιώντας SDN και NBI's, μια εναλλακτική λύση είναι η εφαρμογή του Λογισμικού Καθοριζόμενου από Δευσμευτικό Προγραμματισμό Δρομολόγησης (Software Defined-Constrained Programming Routing – SCOR) [63], η οποία βασίζεται στο δεσμευτικό προγραμματισμό (Constrained Programming – CP). SCOR υλοποιείται στο MiniZinc το οποίο είναι ένα δηλωτικό-δεσμευτικό πρόγραμμα.

4.4 NBI's Βασισμένες στον Ελεγκτή και στο Σκοπό (Controller – based & Intent – based NBI's)

Λόγω της έλλειψη τυποποίησης, οι περισσότερες λύσεις που αφορούν στα NBI's καθορίζονται από τους κατασκευαστές. Ορισμένοι Ελεγκτές παρέχουν τη δική τους υψηλή διεπαφή, ενώ άλλοι χρησιμοποιούν Ad-hoc λύσεις για τον προγραμματισμό των διεπαφών εφαρμογής (APIs). Στην παρούσα εργασία εξετάζονται τέσσερις ελεγκτές αναφορικά με τις χρησιμοποιούμενες διεπαφές τους. Εκ των τεσσάρων

αυτών ελεγκτών, οι δύο (2) ONIX και PANE προτείνουν τις δικές τους υψηλού επιπέδου εφαρμογές στον προγραμματισμό των διεπαφών (APIs), ενώ οι πιο δημοφιλείς ελεγκτές OpenDaylight και ONOS χρησιμοποιούν πολλαπλές εφαρμογές προγραμματισμού διεπαφών ανάλογα με τις Northbound λειτουργίες τους.

4.4.1 Controller-based NBI's

Συμμετοχική δικτύωση (Participatory Networking - PANE) [64] *:* είναι ένα παράδειγμα OpenFlow ελεγκτή SDN, ο οποίος παρέχει τη δική του υψηλού επιπέδου διαπαφή (API). Η συγκεκριμένη εφαρμογή (API) η οποία βρίσκεται μεταξύ του επιπέδου ελέγχου και του επιπέδου εφαρμογών επιτρέπει την ανάγνωση της τρέχουσας κατάστασης του δικτύου και τη γραπτή διαμόρφωσή της. Η λειτουργία της είναι προσανατολισμένη στην αποσύνδεση του ελέγχου και της ορατότητας του δικτύου και στην επίλυση συγκρούσεων μεταξύ των εμπλεκομένων στοιχείων του δικτύου. Για την επίλυση των θεμάτων αυτών, χρησιμοποιεί τρεις τύπους μηνυμάτων, τα αιτήματα(requests), τα ερωτήματα/αναζητήσεις (queries) και τις συμβουλές/υποδείξεις (hints). Τα μηνύματα αιτημάτων χρησιμοποιούνται για τους πόρους του δικτύου (π.χ εύρος ζώνης, έλεγχος πρόσβασης κλπ).Τα ερωτήματα/αναζητήσεις χρησιμοποιούνται για για την ανάγνωση της κατάστασης του δικτύου (π.χ κυκλοφορία μεταξύ των κεντρικών υπολογιστών, και του διαθέσιμου εύρους ζώνης).Οι συμβουλές/υποδείξεις παρέχουν πληροφορίες δικτύου που μπορούν να συμβάλλουν στη βελτίωση της απόδοσής του. Επιπλέον παρέχεται μια διεπαφή (API), η οποία δίνει τη δυνατότητα στις τελικές εφαρμογές υποδοχής να αναζητήσουν δυναμικά πόρους του δικτύου (π.χ κράτηση εύρους ζώνης). Προς αποφυγή των περιορισμών και των υπερβάσεων των ορίων εύρους ζώνης που ορίζονται από τον διαχειριστή, χρησιμοποιείται από τη διεπαφή μια μηχανή επαλήθευσης (verification engine).

Το Onix [65] είναι ένα άλλο παράδειγμα ελεγκτή που παρέχει τη δική του Βόρεια Διεπαφή. Το Onix ορίζει μια γενική API, που επιτρέπει την κλιμακούμενη (scalable) ανάπτυξη εφαρμογών. Επιτρέπει επίσης τον έλεγχο των εφαρμογών για την ανάγνωση και την καταγραφή της κατάστασης των στοιχείων του δικτύου. Επιπλέον, χρησιμοποιεί μια προσέγγιση με βάση τα δεδομένα που παρέχει συνεκτικότητα μεταξύ των εφαρμογών ελέγχου και του υποκείμενου δικτύου συσκευών. Αποτελείται από ένα μοντέλο δεδομένων που αντιπροσωπεύει την υποδομή του δικτύου, όπου

κάθε στοιχείο δικτύου συνδέεται με ένα ή περισσότερα αντικείμενα δεδομένων. Ο προγραμματισμός του ελέγχου διαβάζει την τρέχουσα κατάσταση που σχετίζεται με ένα συγκεκριμένο αντικείμενο, παρεμβαίνει σε αυτό το αντικείμενο για να αλλάξει την κατάστασή του και καταχωρεί τις ενημερώσεις/ειδοποιήσεις (notifications) για τις αλλαγές της κατάστασης του συγκεκριμένου αντικειμένου. Αντίγραφο των καταχωρήσεων των ενημερώσεων και αλλαγών τοποθετείται στη βάση δεδομένων δικτύου (Network Information Base - NIB).

Το ONOS (Open Networking Operation System) [66] είναι ένας από τους πιο δημοφιλείς ελεγκτές SDN ανοιχτού κώδικα, ο οποίος δημιουργήθηκε από το Εργαστήριο OpenNetworks (ONLab) που ιδρύθηκε το 2012.Ο ONOS επικεντρώνεται κυρίως στην κλιμάκωση (scalability), στην υψηλή απόδοση (high performance), στην ανθεκτικότητα (resilience) και στην υποστήριξη επόμενης γενιάς συσκευών. Χρησιμοποιεί μια συλλογή από δέσμες της Open Services Gateway initiative (OSGi)[12] και αλληλεπιδρά με τις εφαρμογές χρησιμοποιώντας API Java και REST. Υποστηρίζει τόσο τη γραμμή εντολών όσο και τη γραφική διεπαφή του χρήστη, για την παροχή ευελιξίας στην ανάπτυξη εφαρμογών και τη διαχείριση δικτύου μέσω REST API. Παρέχει επίσης ένα ευρύ φάσμα προτύπων υποδειγμάτων για το ανάπτυξη νέων εφαρμογών. Λόγω της κατανεμημένης φύσης του ο συγκεκριμένος ελεγκτής, χρησιμοποιεί τη γενικού-σκοπού Απομακρυσμένη Διαδικασία Κλήσης (general-purpose Remote Procedure Call – gRPC), η οποία απλοποιεί τη δημιουργία κατανεμημένων εφαρμογών. Στο gRPC, οι μέθοδοι μιας εφαρμογής πελάτη μπορούν μπορούν να καλούνται απευθείας από την εφαρμογή του διακομιστή και να "τρεχουν" σε διαφορετική μηχανή, σαν να επρόκειτο για τοπικό αντικείμενο.

[12] Η προδιαγραφή OSGi (Open System Gateway initiative) περιγράφει ένα αρθρωτό σύστημα και μια πλατφόρμα υπηρεσιών για τη γλώσσα προγραμματισμού Java που υλοποιεί ένα ολοκληρωμένο και δυναμικό μοντέλο στοιχείων, κάτι που δεν υπάρχει σε αυτόνομα περιβάλλοντα Java / VM. Οι εφαρμογές ή τα στοιχεία, που έχουν τη μορφή δεσμών για ανάπτυξη, μπορούν να εγκατασταθούν, να ξεκινήσουν, να σταματήσουν, να ενημερωθούν και να απεγκατασταθούν από απόσταση, χωρίς να απαιτείται επανεκκίνηση.Η διαχείριση των πακέτων / τάξεων Java καθορίζεται με μεγάλη λεπτομέρεια. Η διαχείριση κύκλου ζωής εφαρμογών υλοποιείται μέσω API που επιτρέπουν την απομακρυσμένη λήψη πολιτικών διαχείρισης. Το μητρώο υπηρεσιών επιτρέπει στις δέσμες να εντοπίζουν την προσθήκη νέων υπηρεσιών ή την αφαίρεση υπηρεσιών και να προσαρμόζονται ανάλογα.Οι προδιαγραφές OSGi εξελίχθηκαν πέρα από τον αρχικό προσανατολισμό τους στις πύλες εξυπηρέτησης και τώρα χρησιμοποιούνται σε εφαρμογές που κυμαίνονται από τα κινητά τηλέφωνα μέχρι τον ανοιχτό κώδικα Eclipse IDE. Άλλοι τομείς εφαρμογής περιλαμβάνουν τα αυτοκίνητα, τον αυτοματισμό της βιομηχανίας, τον αυτοματισμό κτιρίων, τα PDA, το υπολογιστικό δίκτυο, την ψυχαγωγία, τη διαχείριση στόλου και τους διακομιστές εφαρμογών.

Ένας άλλος ελεγκτής SDN ανοιχτού κώδικα, είναι *το OpenDaylight - ODL*, ιδρυτικό μέλος του Linux Foundation Networking (LFN), το οποίο είναι χρησιμοποιείται ευρέως από τη βιομηχανία και την ερευνητική κοινότητα. Ο συγκεκριμένος ελεγκτής ξεκίνησε το 2013 και γράφτηκε στην γλώσσα προγραμματισμού Java, με έμφαση στην προγραμματισιμότητα (programmability) του δικτύου. Το ODL χρησιμοποιεί επίσης δέσμες OSGi, που λειτουργούν ως εξαρτήματα Apache Karaf[13]. Το DLUX σε ODL χρησιμοποιείται ως διαδικτυακή διεπαφή και αντιπροσωπεύει έναν αριθμό χαρακτηριστικών, συμπεριλαμβανομένων της γραφικής διεπαφής χρήστη για την αναπαράσταση τοπολογίας του δικτύου. Οι περισσότερες από τις διεπαφές μπορούν να οπτικοποιηθούν μέσω της διεπαφής Yang-User. Το Yang-UI είναι συλλογή API REST, που επιτρέπει στους προγραμματιστές να αναζητήσουν πληροφορίες δικτύου, καθώς και να τις διαμορφώσουν. Για παράδειγμα, το στοιχείο τοπολογίας δικτύου στο Yang-UI παρέχει ολοκληρωμένες πληροφορίες ολόκληρου του δικτύου και απογραφή των στοιχειών του, ενώ παρέχει μια λεπτομερή πληροφόρηση των στατιστικών στοιχείων.

4.4.2 Intent-based NBI's

Η υιοθέτηση του SDN εξαρτάται απόλυτα από την ικανότητά του να υποστηρίζει πολλαπλούς τύπους εφαρμογών μέσω των NBI διεπαφών. Οι περισσότερες από τις εφαρμοζόμενες λύσεις για τις συγκεκριμένες διεπαφές, είναι ad-hoc, ή εξαρτώνται ειδικά για τον προμηθευτή και ως εκ τούτου έχουν περιορισμένες δυνατότητες. Το μοντέλο των διεπαφών που βασίζεται στο σκοπό, επιχειρεί να δώσει απάντηση σ΄αυτά τα ζητήματα, με τη δήλωση εφαρμογής υψηλού επιπέδου πολιτικών, αντί του καθορισμού αναλυτικών προδιαγραφών για τους διαφορετικούς μηχανισμούς της δικτύωσης.Οι προαναφερθέντες ελεγκτές ONOS και OpenDaylight, υποστηρίζουν επίσης τις βόρειες διεπαφές σκοπού (Intent - Base NBIs).

Το Πλαίσιο που αφορά στο Σκοπό του ONOS [67], επιτρέπει στις εφαρμογές να προσδιορίζουν τις απαιτήσεις τους για έλεγχο δικτύου υπό τη μορφή πολιτικών αντί των μηχανισμών. Οι παρεχόμενες οδηγίες/κατευθύνσεις βάσει πολιτικής αναφέρονται ως Σκοποί/Προθέσεις.Αυτοί οι σκοποί υψηλού επιπέδου μεταφράζονται στους

[13] Το Apache Karaf είναι ένα σύγχρονος και πολυμορφικός περιέκτης. Μπορεί να χρησιμοποιηθεί αυτόνομα ως σύστημα αποθήκευσης , υποστηρίζοντας ένα ευρύ φάσμα εφαρμογών και τεχνολογιών. Υποστηρίζει επίσης την "εκτέλεση οπουδήποτε" (σε οποιοδήποτε μηχάνημα με Java, cloud, εικόνες docker, ...) χρησιμοποιώντας την ενσωματωμένη ρύθμισή του.

εγκαταστάσιμους κανόνες προώθησης, που είναι απαραίτητες λειτουργίες για τον έλεγχο του δικτύου. Επιπλέον, αυτοί οι σκοποί μπορούν να εντοπιστούν από χρησιμοποιώντας δύο παραμέτρους, τον Αριθμό Αναγνώρισης της Εφαρμογής (Application ID) και τον Αριθμό Αναγνώρισης του Σκοπού (Intent ID). Ο Αριθμός Αναγνώρισης της Εφαρμογής (Application ID) αντιπροσωπεύει μια εφαρμογή η οποία δημιουργεί ένα σκοπό, ενώ ο Αριθμός Αναγνώρισης του Σκοπού (Intent ID) δημιουργείται όποτε δημιουργείται ένας σκοπός. Μόλις υποβληθεί ένας σκοπός μετά από μια αίτηση, αποστέλλεται απευθείας στη φάση συλλογής. Η συγκεκριμένη φάση χρησιμοποιεί ένα συλλογέα σκοπού, για τη μετατροπή του αιτούμενου σκοπού σε εγκαταστήσιμο σκοπό. Αν μια εφαρμογή ζητήσει ένα μη διαθέσιμο αντικείμενο (π.χ συνδεσιμότητα μεταξύ μη συνδεδεμένων τομέων) αυτή η φάση θα αναζητήσει μια εναλλακτική λύση για την ανασύνταξη του αιτήματος. Αφού συντάξει το σκοπό, το προωθεί στη φάση εγκατάστασης, όπου ένας εγκαταστάτης σκοπού μετατρέπει το σκοπό σε κανόνες ροής. Ένας διαχειριστής σκοπού (intent manager) συντονίζει τον πάροχο σκοπού (intent provider) με τον εγκαταστάτη σκοπού (intent installer).

Το Network Intent Composition (NIC) [68] αφορά σε μια εσωτερική εφαρμογή του ODL, που βρίσκεται σε φάση επώασης και η οποία παρέχει τη δυνατότητα αλληλεπίδρασης μεταξύ των βασικών ενοτήτων του ODL, ή ικανοποίησης των επιθυμιών των χρηστών.Χρησιμοποιεί τρέχουσες λειτουργίες του δικτύου ODL και νότιες διεπαφές, για τον έλεγχο των εικονοποιημένων και των φυσικών στοιχείων του δικτύου.Ένα δομικό στοιχείο του ODL με την ονομασία renderer (δημιουργός) χρησιμοποιείται για να μετασχηματίσει τους σκοπούς σε κανόνες εφαρμογής ροής. Υπάρχουν πολλοί renderers που υποστηρίζουν τις διαφορετικές εκδόσεις του ODL και σε αυτούς περιλαμβάνονται, ο Network Modeling (NEMO) renderer, ο OpenFlow renderer, Virtual Tenant Network (VTN) renderer, και ο Group Based Policy (GBP) renderer. Υπάρχουν δύο βασικές λειτουργίες (π.χ hazelcast και MD-SAL) οι οποίες τροφοδοτούν τα βασικά μοντέλα για τη δυνατότητα NIC (Network Intent Composition). Πάνω από αυτές τις λειτουργίες, μπορεί να εγκατασταθεί ο renderer. Ο συγκεκριμένος renderer μετατρέπει ένα σκοπό χρησιμοποιώντας μια συγκεκριμένη εφαρμογή (π.χ. VTN, NEMO κ.λπ.) για πραγματοποιήσει τροποποιήσεις στο δίκτυο. Για παράδειγμα, ο renderer NEMO είναι μια λειτουργία που μετατρέπει ένα σκοπό, σε τροποποίηση του δικτύου με τη χρήση της εφαρμογής NEMO [69] στην ODL. Για τη δημιουργία νέων, όπως και για τη σύνθεση ή το διαχωρισμό, υφισταμένων

εφαρμογών δικτύου οι M. Pham and D.B. Hoang [70] πρότειναν μια συγκεκριμένη λύση Βόρειας Διεπαφής Σκοπού. Οι σχεδιαστικές αρχές της συγκεκριμένης πρότασης βασίζονται: στην αποκέντρωση των δεδομένων, στα στοιχεία υπηρεσιών διαδικτύου, στην απομόνωση των διεργασιών και στην ανθεκτικότητα (robustness). Επιπλέον προτάθηκε μια αρχιτεκτονική τριών βαθμίδων (tier), οι οποίες λειτουργούν ανεξάρτητα η μία από την άλλη, προκειμένου να υπάρχει ευελιξία μεταξύ τους (π.χ. να μπορεί να τροποιείται η μια βαθμίδα, χωρίς να επηρρεάζονται οι υπόλοιπες). Αυτές οι βαθμίδες είναι : η βαθμίδα της βάσης δεδομένων (database tier), η βαθμίδα της λογικής των εργασιών (business logic tier) και η βαθμίδα της παρουσίασης (presentation tier). Η κατάσταση των εφαρμογών είναι αποθηκευμένη στη βαθμίδα των δεδομένων, απ' όπου είναι δυνατόν να ανακληθούν για διαφορετικούς σκοπούς και χρήσεις. Η βαθμίδα της λογικής εργασιών (business logic tier) διαχειρίζεται τη δημιουργία και τη σύνθεση υπηρεσιών. Σ' αυτή τη βαθμίδα υπάρχει ένα μητρώο υπηρεσιών το οποίο χρησιμοποιείται για την εύρεση των υπαρχουσών υπηρεσιών και όπου μπορούν να δημιουργηθούν και νέες υπηρεσίες, ως μια νέα εξατομικευμένη υπηρεσία. Για την ενσωμάτωση νέων και υφιστάμενων υπηρεσιών χρησιμοποιείται ένα στοιχείο το service integration element. Με τη χρήση CLI, REST και διεπαφών προγραμματισμού, η βαθμίδα παρουσίασης λαμβάνει στοιχεία/δεδομένα του χρήστη. Για την ανάλυση των διεργασιών (processes) χρησιμοποιείται το Domain Driven Design (DDD), όπου οι προδιαγραφές χωρίζονται σε μικρότερα μέρη (προβλήματα), και στη συνέχεια αναπτύσσονται λύσεις για καθένα από αυτά. Ως παράδειγμα αναφέρονται οι σύνθετοι σκοποί (composed intents), οι οποίοι αναλύονται σε βασικούς σκοπούς, οι οποίοι περαιτέρω αναλύονται σε απαντήσεις (λύσεις) σκοπού.

4.5 Συμπεράσματα

Λόγω της ποικιλίας των χρησιμοποιούμενων βόρειων διεπαφρών που βασίζονται στους ελεγκτές (controller-based NBI), εφαρμογές που σχεδιάζονται για έναν ελεγκτή μπορεί να μην λειτουργούν σε έναν άλλο. Καλό θα ήταν λοιπόν να χρησιμοποιούνται βόρειες διεπαφές βασισμένες στο σκοπό, αλλά το ζήτημα που προκύπτει σ΄αυτές τις περιπτώσεις είναι το γεγονός ότι πολύ λίγοι ελεγκτές είναι σε θέση να τις υποστηρίξουν. Στον Πίνακα V που ακολουθεί παρουσιάζονται συνοπτικά οι διεπαφές βασιμένες στον ελεγκτή και αυτές βασιμένες στο σκοπό, όπου το ODL και ONOS υποστηρίζουν και τις δύο αυτές κατηγορίες διεπαφών, ενώ οι οι PANE και Onix

προσφέρουν τις δικές τους NBIs. Αντίθετα προς τις SBIs, στις NBIs είναι ζητούμενος ο βαθμός ωριμότητας και συνεκτικότητας, προκειμένου να υποστηρίξουν τα SDN και απαντήσουν στις ανάγκες τους.

ΠΙΝΑΚΑΣ V

Παρουσίαση Κατηγοριών Ελεγκτών στα NBI's

Βιβλιογραφία Ελεγκτή	NBI*	Βασισμένη στον Ελεγκτή/Σκοπό	GUI**	Βαθμός Ασφάλειας
ODL	OSGi & REST APIs	Ελεγκτή & Σκοπού	Ναι	Μεγάλος
Onix	Onix API	Μόνο Ελεγκτή	Όχι	Μέσος
PANE	PANE API	Μόνο Ελεγκτή	Όχι	Χαμηλός
ONOS	Java & REST APIs	Ελεγκτή & Σκοπού	Ναι	Μεγάλος
Pham et al.	Programming APIs, CLI & REST APIs	Μόνο Σκοπού	Ναι	Χαμηλός

*NBI = Northbound Interface, **GUI= Graphical User Interface

5. Open DayLight Project

5.1 Εισαγωγή

Όπως έχει ήδη αναφερθεί σε προηγούμενα κεφάλαια, η αρχιτεκτονική των Καθοριζόμενων από Λογισμικό Δικτύων (SDN), βασίζεται στην αποσύνδεση του ελέγχου του δικτύου από την προώθηση των δεδομένων και την άμεση δυνατότητα προγραμματισμού των λειτουργιών ελέγχου του δικτύου [14]. Η ευφυΐα του δικτύου είναι λογικά και κεντρικά προγραμματισμένη στους Ελεγκτές SDN, οι οποίοι και διατηρούν τη συνολική εικόνα του δικτύου. Το γεγονός αυτό έχει ως αποτέλεσμα το δίκτυο να εμφανίζεται στις "μηχανές" εφαρμογών και πολιτικής, ως ένας και μόνος "λογικός" μεταγωγέας. Στο Κεφάλαιο 4.4.1 αναφέρθηκαν τα ορισμένα στοιχεία σχετικά με το ODL και τα NBIs. Στη συνέχεια παρουσιάζεται διεξοδικότερα η πλατφόρμα αυτή, η φιλοσοφία της σχετικά με τα SDN, η αρχιτεκτονική της και οι λειτουργίες που εκτελεί σε ένα SDN δίκτυο.

5.2 Πλαίσιο

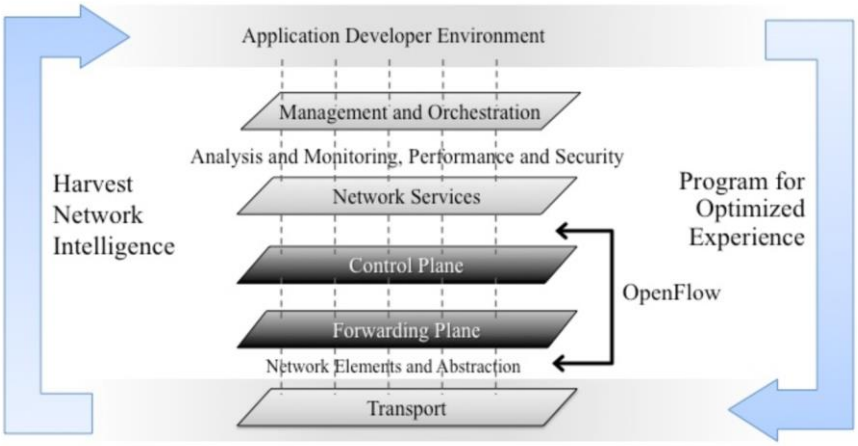
5.2.1 Ορισμένα Στοιχεία για τα SDN Δίκτυα

Η SDN αρχιτεκτονική αναπτύχθηκε από τον ONF, ο οποίος ηγείται της ανάπτυξης και της τυποποίησης των OpenFlow πρωτοκόλλων [71] [72]. Το OpenFlow επιτρέπει στις λειτουργίες ελέγχου του δικτύου SDN, να προγραμματίζουν εξ αποστάσεως τις δραστηριότητες προώθησης των πακέτων δεδομένων. Με την πάροδο των ετών αναπτύχθηκε ένας ικανός αριθμός Ελεγκτών, όπως ο Beacon, ο FloodLight, ο NOX, ο POX, ο Ryu και τελευταία το ONOS, όπου θα αναλυθεί εκτενέστερα στο επόμενο κεφάλαιο. Κάθε ελεγκτής εφαρμόζει το Openflow, ως μοναδικό southbound (SB) πρωτόκολλο, προς τις λειτουργίες προώθησης δεδομένων και παρέχει πρόσβαση στις λειτουργίες του επιπέδου ελέγχου μέσω Northbound (NB) REST APIs. Αυτές οι πλατφόρμες ελεγκτών υποστηρίζουν κυρίως εφαρμογές στο Επίπεδο Ελέγχου, με πιο σημαντική αυτή της εικονοποίησης του δικτύου.

Παρ' ότι η αποσύνδεση των λειτουργιών του Επιπέδου Ελέγχου από αυτές του Επιπέδου Προώθησης αποτελεί μια πολύ σημαντική και έγκυρη περίπτωση ως προς τη χρήση της από το SDN, εν τούτοις το τελευταίο μπορεί να οριστεί και ευρύτερα πέρα από τη χρήση OpenFlow Πρωτοκόλλων Αφαίρεσης. Το OpenFlow υποστηρίζει

μόνο την ανάκτηση και τον προγραμματισμό δεδομένων που σχετίζονται με τις λειτουργίες προώθησης του δικτύου. Όμως το δίκτυο από μόνο του είναι μια πολύ μεγάλη πηγή δεδομένων, τα οποία μπορούν να χρησιμοποιηθούν για SDN εφαρμογές. Έτσι δεδομένα μπορούν να ανακαλούνται και τα δίκτυα να προγραμματίζονται μέσα από μια ευρεία ποικιλία πρωτοκόλλων και APIs σε διαφορετικά επίπεδα αφαίρεσης πληροφοριών όπως το BGP, NETCONF, NetFlow/IP-Fix, SNMP ή και εξειδικευμένων CLIs. Βασικά το SDN καθορίζει ένα γενικό βρόγχο στη βάση/ανάδραση/έλεγχος/πολιτική μεταξύ των εφαρμογών και του δικτύου, με αποτέλεσμα οι εφαρμογές να λαμβάνουν δεδομένα/πληροφορίες από το δίκτυο, τα οποία στη συνέχεια υφίστανται επεξεργασία, και λαμβάνονται αποφάσεις πολιτικής/ελέγχου που στη συνέχεια οδηγούνται πίσω στο δίκτυο για τα περαιτέρω, όπως φαίνεται και στο *Σχήμα 5* που ακολουθεί.

Οι χρήσεις των SDN εξελίχθηκαν αρχικά από τις ειδικές εφαρμογές ενορχήστρωσης των Κέντρων Δεδομένων, σε μια πλειάδα διαφορετικών εφαρμογών, όπου λαμβάνοντας πληροφορίες από το δίκτυο, τις χρησιμοποιούν μέσω των διαθέσιμων APIs για τον προγραμματισμό του. Τέτοια παραδείγματα εκτός από την εικονοποίηση [73]&[74], που προαναφέρθηκε είναι εφαρμογές όπως η βελτιστοποίηση των πολυστιβαδικών διαδρομών, οι δικτυακές αναλύσεις, οι πολιτικές διαμόρφωσης [75] [76], οι αλυσίδες υπηρεσιών και η ασφάλεια [77].



Σχήμα 5 : Ανάδραση Μεταξύ Εφαρμογών στα SDN Δίκτυα

5.2.2 Model-Driven Software Engineering – MDSE

Το OpenDaylight (ODL) αξιοποιεί την τεχνολογία λογισμικού βασιζόμενη στο μοντέλο (Model Driven Software Engineering - MDSE), όπως αυτή ορίζεται από την OMG (Object Management Group). Το MDSE καθορίζει ένα πλαίσιο που στηρίζεται στις σταθερές σχέσεις μεταξύ των διαφόρων μοντέλων, τυποποιεί απεικονίσεις (mappings) και διατάξεις που επιτρέπουν την δημιουργία/παραγωγή μοντέλων και κατ' επέκταση την παραγωγή code/API από τα μοντέλα. Αυτή η γενική προσέγγιση μπορεί να καλύψει κάθε γλώσσα μοντελοποίησης.

Αν και το OGM επικεντρώνει τις MDA (Model-Driven Architecture) λύσεις του σε UML, η γλώσσα YANG αναδεικνύεται ως η γλώσσα της μοντελοποίησης των πεδίων (domains) στις δικτυώσεις.Τα μοντέλα που δημιουργούνται είναι φορητά (διατρέχουν την πλατφόρμα) και μπορούν να αξιοποιηθούν για να περιγράψουν πακέτα εργασιών, λειτουργίες της πλατφόρμας, ή ειδικές δυνατότητες των components (καθέτως), που γίνονται αντιληπτές στις διεπαφές υπηρεσιών (για τις περιπτώσεις SDN) και οι οποίες παρέχονται από μια οντότητα ελέγχου του δικτύου (οριζόντια και κάθετα εντός του πλαισίου εφαρμογής μιας υπηρεσίας). Η MDSE επιτρέπει τη μεσολάβηση σε τρέχοντα χρόνο και την εσωτερική απεικόνιση (inter-mapping) των μοντέλων, για παράδειγμα, όταν πρόκειται για μοντέλα δύο διαφορετικών τύπων πεδίων, όπως J2EE[14] και UML[15].

5.2.3 Model-Driven Network Management/Programmability

Η βασισμένη στο μοντέλο προσέγγιση χρησιμοποιείται συνεχώς και περισσότερο στους τομείς/πεδία (domain) των δικτύων, για να περιγράψει τη λειτουργικότητα των συσκευών τους, των υπηρεσιών [78], των πολιτικών [79],[80], και των διεπαφών προγραμματιζόμενων εφαρμογών (APIs) [81],[82]. Τα πρωτόκολλα επιλογής σ' αυτές τις περιπτώσεις είναι το NETCONF και RESTCONF και η προτιμώμενη γλώσσα προγραμματισμού είναι η YANG.

Το NETCONF [83] είναι ένα πρωτόκολλο διαχείρισης που δημιουργήθηκε από την IEFT, το οποίο καθορίζει τη διαμόρφωση και το πλαίσιο των επιχειρησιακών αποθηκών δεδομένων και μια σειρά λειτουργιών (CRUD) Δημιουργίας (Create),

[14] Η J2EE (Java 2 Enterprise Edition) είναι μια πλατφόρμα ανάπτυξης και χρήσης εφαρμογών, οι προδιαγραφές της οποίας καθορίζονται από την Sun Microsystems Inc.

[15] Η ULM (Unified Modeling Language) είναι μια τυποποιημένη γλώσσα για τη δημιουργία μοντέλων λύσεων λογισμικού, δόμησης εφαρμογών, συμπεριφοράς συστημάτων και επιχειρησιακών διεργασιών.

Ανάκτησης (Retrieve), Εκσυγχρονισμού (Update) και Διαγραφής (Delete), οι οποίες μπορούν να χρησιμοποιηθούν στις συγκεκριμένες αποθήκες δεδομένων. Επιπρόσθετα προς τις λειτουργίες CRUD των αποθηκών δεδομένων, το NETCONF υποστηρίζει την απλή εξ Αποστάσεως Διαδικασία Κλήσης (Remote Procedure Call - PRC) και τις λειτουργίες Ενημέρωσης (Notifications). Οι λειτουργίες NETCONF εκτελούνται πάνω σε μια απλή στιβάδα κλήσης (RPC). Το NETCONF χρησιμοποιεί μια XML γλώσσα (Extensible Markup Language)[16], που βασίζεται στην κωδικοποίηση των δεδομένων για τη διαμόρφωση και την επιχειρησιακή λειτουργία τους, όπως και για το πρωτόκολλο των μηνυμάτων της.

Το RESTCONF [84] είναι ένα άλλο πρωτόκολλο τύπου REST, το οποίο παρέχει μια διεπαφή προγραμματισμού επί το HTTP για πρόσβαση σε δεδομένα που καθορίζονται στην YANG, χρησιμοποιώντας αποθήκες δεδομένων που έχουν οριστεί σ' αυτό (NETCONF). Διαμορφωμένα δεδομένα και δεδομένα κατάστασης εμφανίζονται ως πόροι του συστήματος και μπορούν να ανακληθούν με τη μέθοδο HTTP GET. Πόροι που αφορούν δεδομένα διαμόρφωσης μπορούν να τροποποιηθούν με τη μέθοδο HTTP DELETE, PATCH, POST και PUT. Τα δεδομένα κωδικοποιούνται είτε σε XML, είτε σε JSON γλώσσες.

Η YANG [85] αρχικά αναπτύχθηκε για την τυποποίηση της μοντελοποίησης των διαμορφώσεων και της κατάστασης των δεδομένων στις συσκευές του δικτύου, μπορεί όμως να χρησιμοποιηθεί και για την περιγραφή άλλων στοιχείων του δικτύου, όπως οι πολιτικές [78], τα πρωτόκολλα ή οι συνδρομητές. Η YANG είναι δομημένη υπό μορφή "δένδρου" αντί να είναι "προσανατολισμένου αντικειμένου" (object-oriented), τα δεδομένα είναι και αυτά δομημένα σε μορφή "δένδρου" και μπορούν να περιλαμβάνουν πολύπλοκους τύπους, όπως λίστες και ενώσεις (unions). Πλέον των ορισμών δεδομένων, η YANG υποστηρίζει δομές για την οργάνωση Διαδικασιών Κλήσης εξ Αποστάσεως (RPCs) και Ενημερώσεων (Notifications) που την καθιστούν κατάλληλη για Διεπαφές Περιγραφής Γλώσσας (Interface Description Language - IDL) σε ένα σύστημα βασισμένο στο μοντέλο.

5.3 Απαιτήσεις

Ένας ελεγκτής SDN οφείλει να είναι ταυτόχρονα μια πλατφόρμα ανάπτυξης εφαρμογών και να παρέχει το κατάλληλο περιβάλλον για τη δημιουργία αυτών των

[16] Η XML είναι μια γλώσσα σήμανσης που ορίζει τους κανόνες για την κωδικοποίηση εγγραφών σε μορφή που να είναι αναγνώσιμη από ανθρώπους και μηχανές.

συγκεκριμένων εφαρμογών. Μια SDN πλατφόρμα επίσης οφείλει να είναι δομημένη κατά τέτοιο τρόπο ώστε να ικανοποιεί τις ακόλουθες απαιτήσεις :

- Flexibility (Ευελιξία): Πρέπει να είναι σε θέση να φιλοξενεί μια πλειάδα διαφορετικών εφαρμογών και ταυτόχρονα οι εφαρμογές του ελεγκτή να χρησιμοποιούν ένα κοινό πλαίσιο και μοντέλο προγραμματισμού, για την παροχή σταθερών APIs στους πελάτες του. Αυτό είναι σημαντικό για την αντιμετώπιση προβλημάτων, την ενοποίηση του συστήματος, για τον συνδυασμό εφαρμογών στο ανώτερο επίπεδο ενορχηστρωμένων ροών εργασίας.
- Scale the development process (Κλιμάκωση ανάπτυξης διεργασιών): Δεν θα πρέπει να υπάρχει υποσύστημα στην υποδομή του ελεγκτή, όπου μια προσθήκη (plugin) σ' αυτό θα πρέπει να προσθέσει κώδικα. Η αρχιτεκτονική θα πρέπει να επιτρέπει στις προσθήκες (plugins) να αναπτύσσονται ανεξάρτητα μεταξύ τους, αλλά και με την υποδομή του ελεγκτή και θα πρέπει να υποστηρίζουν μικρούς χρόνους ενσωμάτωσης στο σύστημα. Αυτή την περίοδο υπάρχουν 18 ενεργά έργα ODL,ενώ 15 ακόμα βρίσκονται στη φάση των προτάσεων.Τα συγκεκριμένα έργα λειτουργούν αυτόνομα και αναπτύσσονται από αυτόνομες ομάδες, με ελάχιστη ή καθόλου συνεργασία μεταξύ τους.
- Run-time Extensibility (Επεκτασιμότητα σε τρέχοντα χρόνο): Ο ελεγκτής θα πρέπει να είναι σε θέση να "φορτώνει" νέα πρωτόκολλα και προσθήκες (plugins)[17] υπηρεσιών/εφαρμογών σε τρέχοντα χρόνο. Η υποδομή του ελεγκτή θα πρέπει να προσαρμόζεται σε μοντέλα δεδομένων τα οποία είτε προσλαμβάνονται μέσω δυναμικής φόρτωσης από νέες προσθήκες, είτε ανακαλύπτονται από τις συσκευές του δικτύου. Η επεκτασιμότητα σε τρέχοντα χρόνο (RtEx) επιτρέπει στον ελεγκτή να προσαρμόζεται στις αλλαγές του δικτύου (νέες συσκευές /ή και νέα χαρακτηριστικά) και να αποφεύγει μακροσκελείς κύκλους απελευθέρωσης που αποτελούν τυπικό φαινόμενο στα παλαιού τύπου (legacy) EMS/NMS συστήματα, όπου κάθε νέο χαρακτηριστικό σε μια συσκευή του δικτύου επιβάλλει χειροκίνητη αλλαγή του μοντέλου της συσκευής στο NMS/EMS.

[17] Plugin : "Προσθήκη" ονομάζεται ένα πρόσθετο λογισμικό που είναι εγκαταστημένο σε ένα πρόγραμμα με σκοπό την ενίσχυση των δυνατοτήτων του. Για παράδειγμα αν χρειάζεται να παρακολουθήσει κάποιος ένα βίντεο σε έναν ιστότοπο, μπορεί να χρειάζεται μια "προσθήκη" για να το παίξει επειδή το πρόγραμμα περιήγησης που είναι διαθέσιμο δεν έχει τα κατάλληλα εργαλεία. Σε ένα απλό παράδειγμα αυτό γίνεται όταν διαθέτει κάποιος ένα δίσκο blu-ray, για να τον ακούσει χρειάζεται ένα blu-ray player (plugin).

- Performance & Scale (Απόδοση & Κλιμάκωση): Ένας ελεγκτής θα πρέπει να είναι σε θέση να αποδίδει εξίσου καλά τόσο σε μια ευρεία ποικιλία διαφορετικών φορτίων/εφαρμογών, όσο και σε μια μεγάλη σειρά από περιβάλλοντα.Αυτά όμως δεν θα πρέπει να γίνονται σε βάρος της αρθρωμένης μορφής του (modularity). Η αρχιτεκτονική του ελεγκτή θα πρέπει επίσης να επιτρέπει την οριζόντια κλιμάκωσή (επέκτασή) του για περιβάλλοντα συστάδων(clusters)/νέφους (cloud).

Για την υποστήριξη των SDN εφαρμογών, ο ελεγκτής οφείλει να παρέχει (ή να συνεργάζεται με) ένα περιβάλλον ανάπτυξης εφαρμογών, που θα πρέπει να πληροί τις ακόλουθες προϋποθέσεις:

i) Να χρησιμοποιεί μια εξειδικευμένη γλώσσα μοντελοποίησης πεδίου (domain-specific modeling language) για την περιγραφή της εξωτερικής και εσωτερικής συμπεριφοράς του συστήματος.

ii) Να δημιουργεί κώδικα από τα μοντέλα, ο οποίος πρέπει να χρησιμοποιείται εφαρμόζοντας πρότυπες συμβάσεις APIs και να παράγει ένα στερεότυπο κώδικα που να αποδίδει κατά επαναλήψιμο τρόπο και με μηδενικό σφάλμα τους ελέγχους όλων των παραμέτρων του συστήματος.

iii) Τα εργαλεία που προκύπτουν από την εξειδικευμένη γλώσσα μοντελοποίησης πεδίου (domain) και την παραγωγή κώδικα, θα πρέπει να δίνουν τη δυνατότητα ταχείας εξέλιξης των APIs και των πρωτοκόλλων (ευελιξία συστήματος).

iv) Η δημιουργία κώδικα θα πρέπει να παράγει λειτουργικά ισοδύναμες APIs για διαφορετικές γλωσσικές συνδέσεις (language bindings).

v) Τα εργαλεία μοντελοποίησης για τον ελεγκτή θα πρέπει να συμφωνούν με τα αντίστοιχα των συσκευών του δικτύου. Στη συνέχεια μπορεί να χρησιμοποιηθεί μια κοινή αλυσίδα εργαλείων και για τα δύο μέρη (ελεγκτή και συσκευές) και τα μοντέλα των συσκευών μπορούν να επαναχρησιμοποιηθούν στον ελεγκτή, μέσω της δημιουργίας μιας διαδρομής μηδενικής επαφής (zero-touch path)μεταξύ της συσκευής και μιας εφαρμογής/προσθήκης του ελεγκτή χρησιμοποιώντας τα μοντέλα του (ελεγκτή).

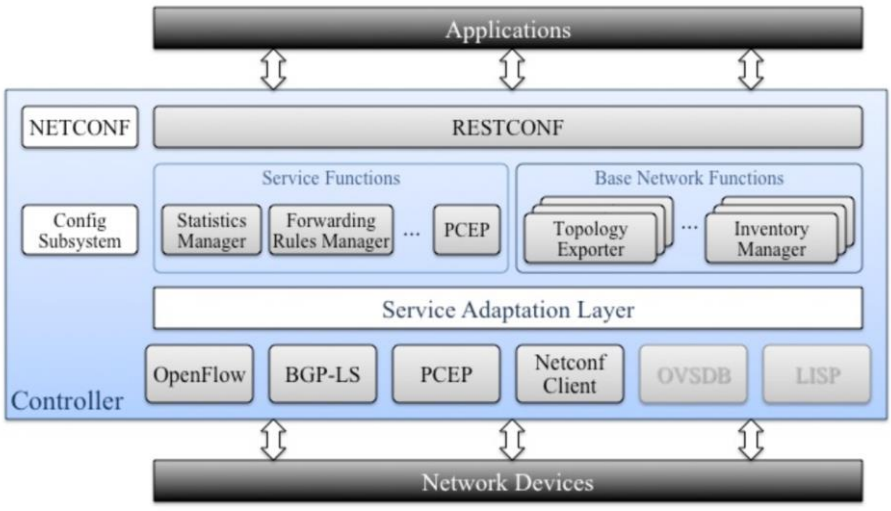
vi) Οι ειδικές γλώσσες πεδίου/τεχνολογίες/εργαλεία που χρησιμοποιούνται στον ελεγκτή πρέπει να μπορούν να χρησιμοποιηθούν για την μοντελοποίηση γενικών δικτυακών κατασκευών, όπως οι υπηρεσίες, οι αλυσίδες υπηρεσιών, η διαχείριση συνδρομητών και οι πολιτικές.

vii) Η αλυσίδα εργαλείων θα πρέπει να υποστηρίζει τη δημιουργία κώδικα για προσαρμογές μοντέλο-προς-μοντέλο, τόσο για τις υπηρεσίες όσο και για τις συσκευές.

5.4 Αρχιτεκτονική Ελεγκτή

5.4.1 Επισκόπηση

Το OpenDaylight (ODL) ήταν έμπνευση του Beacon, το οποίο εισήγαγε τη χρήση της Διεπαφής Ανοιχτής Υπηρεσίας Πύλης (Open Service Gateway Interface - OSGi), η οποία αποτελεί το κλειδί για την αρθρωτή μορφή και τη φόρτωση σε τρέχοντα χρόνο των τμημάτων λογισμικού (components) των προσθηκών (plugins) εντός του ελεγκτή. Το πρώτο νέο στοιχείο που εισήγαγε το ODL και το οποίο αποτελέσε μια σοβαρή καινοτομία ήταν η Στιβάδα Προσαρμοσμένης Υπηρεσίας(Service Adaptation Layer) γνωστή και ως SAL, η οποία διαχωρίζει τις νότιες προσθήκες πρωτοκόλλων (southbound protocol plugins) και τις βόρειες προσθήκες υπηρεσιών/εφαρμογών (northbound service/application plugins). Η συγκεκριμένη αρχιτεκτονική, όπως φαίνεται και στο *Σχήμα 6*, συγκροτείται από τρεις στιβάδες: τα SB πρωτόκολλα προσθηκών, την ίδια τη SAL και τις NB λειτουργίες εφαρμογών/υπηρεσιών.



Σχήμα 6 : Αρχιτεκτονική ODL με την προσθήκη SAL

Οι νότιες (SB) προσθήκες (πρωτοκόλλων) βρίσκονται σε διεπαφή με τις συσκευές του δικτύου. Η SAL προσαρμόζει τις λειτουργίες των SB προσθηκών στο ανώτερο

επίπεδο λειτουργιών Εφαρμογών/Υπηρεσιών, το οποίο οδηγεί μέσω των NB APIs του ελεγκτή στις εφαρμογές. Παραδείγματα λειτουργιών Εφαρμογών/Υπηρεσιών είναι ο εξαγωγέας τοπολογίας (Topology Exporter), ο διαχειριστής απογραφής (Inventory Manager), όπως και ο OpenFlow διαχειριστής στατιστικών δεδομένων. Η συγκεκριμένη στιβαδοποιημένη αρχιτεκτονική παρέχει τη δυνατότητα στον ελεγκτή να υποστηρίξει πολλαπλά SB πρωτόκολλα (μέσω των SB προσθηκών πρωτοκόλλων), καθώς και την παροχή μια σειράς ενιαίων υπηρεσιών και APIs προς τις εφαρμογές μέσω μιας κοινής σειράς NB APIs.

Στην αρχική SAL, η οποία ονομαζόταν API-Driven-SAL (AD-SAL), οι προγραμματιστές έπρεπε να καθορίσουν τις SAL APIs που χρησιμοποιούνταν από τις προσθήκες και να κωδικοποιήσουν την λειτουργία προσαρμογής μεταξύ των SB και των NB προσθηκών. Γρήγορα όμως έγινε προφανές ότι η χειροκίνητη κωδικοποίηση των SAL APIs και οι προσαρμογές για την υποστήριξη της λειτουργικότητας των νέων προσθηκών δεν θα οδηγούσε στην επιθυμητή κλιμάκωση του μεγέθους που απαιτούσε το ODL.

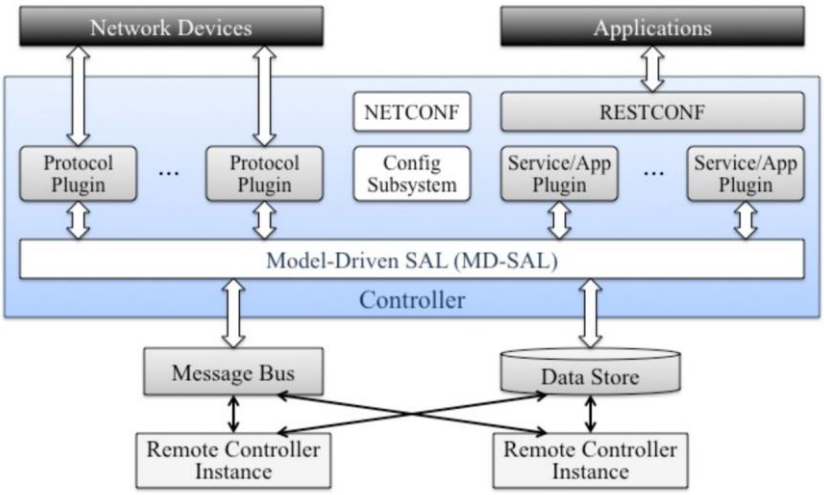
5.4.2 Εξέλιξη προς την MD-SAL

Για την ικανοποίηση των απαιτήσεων που πρέπει να πληρούνται για έναν ελεγκτή (αναφέρθηκαν στο Κεφ. 5.2.4), προτάθηκε μια καινούργια αρχιτεκτονική βασιζόμενη στο μοντέλο (model-driven) για τη Στιβάδα Προσαρμογής Υπηρεσιών (SAL), η οποία αναφέρεται ως Model - Driven Service Adaptaion Layer ή MD-SAL. Η συγκεκριμένη αρχιτεκτονική δομήθηκε γύρω από τις έννοιες των πρωτόκολλων και των γλωσσών που αναφέρθηκαν στο Κεφ. 5.2. Οι προσθήκες του ελεγκτή μπορούν να είναι είτε πάροχοι δεδομένων/υπηρεσιών, είτε καταναλωτές δεδομένων/υπηρεσιών. Ο πάροχος διαθέτει δεδομένα/υπηρεσίες μέσω δικών του APIs.Ο καταναλωτής αναλώνει υπηρεσίες/δεδομένα που είναι διαθέσιμα από ένα ή περισσότερους παρόχους. Για παράδειγμα το πρωτόκολλο προσθηκών OpenFlow παρέχει υπηρεσίες για πρόσθεση, τροποποίηση, διαγραφή ροών από μεταγωγείς, που είναι συνδεδεμένες σε αυτές τις προσθήκες. Ένας από τους καταναλωτές των υπηρεσιών των προσθηκών του OpenFlow, είναι ο Διαχειριστής Κανόνων Προώθησης (Forwarding Rules Manager), ο οποίος παρέχει υψηλού επιπέδου υπηρεσίες προγραμματισμού ροών στους πελάτες του ελεγκτή.

Το περιβάλλον ανάπτυξης του ODL περιλαμβάνει και εργαλεία για τη δημιουργία αυτού του κώδικα (codecs και Java APIs). Τα εργαλεία διατηρούν τύπους ιεραχίας YANG, διατηρούν ιεραρχίες δεδομένων δένδρου και ιεραρχίες διευθύνσεων δεδομένων. Η SAL δεν περιέχει προσθήκες συγκεκριμένου κωδικά ή APIs και επομένως είναι σε θέση να

προσαρμοστεί σε οποιεσδήποτε προσθήκες ή υπηρεσίες/εφαρμογές "φορτωθούν", στον ελεγκτή.

Από πλευράς υποδομής δεν υπάρχει καμία διαφορά μεταξύ μιας προσθήκης πρωτοκόλλου και μιας προσθήκης εφαρμογής/υπηρεσίας. Όλοι οι κύκλοι ζωής των προσθηκών είναι ίδιοι, κάθε προσθήκη είναι μία δέσμη (bundle) OSGi που περιέχει καθορισμένα μοντέλα APIs. Δεδομένου ότι όλες οι προσθήκες είναι ίδιες στην υποδομή του ελεγκτή, η αρχιτεκτονική του μοντέλου μπορεί να δομηθεί υπό τη μορφή που παρουσιάζεται στο Σχήμα 7 που ακολουθεί.



Σχήμα 7 : Αρχιτεκτονική ODL-Συνδέσεις Πρωτοκόλλων και Υπηρεσιών/Εφαρμογών

Παρουσιάζει επίσης πως σε μια συστάδα (cluster) πολλαπλών κλώνων (instaces) ελεγκτών, ένας κλώνος της MD-SAL εντός JVM περιέκτη (container),θα συνδεθεί με το μήνυμα διαύλου (message bus) της συστάδας και την αποθήκη δεδομένων της.

5.4.3 OpenDaylight Μοντέλα Yang

Η έκδοση Hydrogen[18] του ODL, ως κώδικας παραγωγής περιέχει 110 μοντέλα YANG, εκ των οποίων 3 ορίζονται σε IETF RFCs (Remote Function Controls), 8 σε προσχέδια

[18] Η πρώτη έκδοση της πλατφόρμας ODL βγήκε τον Φεβρουάριο του 2014, με την κωδική ονομασία **Hydrogen** και ακολούθησε σύντομα και η δεύτερη βελτιωμένη έκδοση **Helium**, τον Οκτώβριο του 2014. Η τελευταία, σηματοδότησε μια σημαντική αλλαγή κατεύθυνσης στη φιλοσοφία του ODL, που επηρέασε τις μεταγενέστερες εξελίξεις στην αρχιτεκτονική των ελεγκτών. Η αλλαγή αφορούσε στη SAL του ελεγκτή, η οποία στην νέα έκδοση απομόνωνε τα SB πρωτόκολλα, όπως το OpenFlow, από τις NB εφαρμογές. Έτσι ενώ στην έκδοση Hydrogen η SAL ήταν βασισμένη στα APIs (APIs-driven SAL, AD-SAL), στην έκδοση Helium βασίστηκε στην μοντελοποιημένη SAL (Model-driven SAL, MD-SAL), η οποία παρείχε μεγαλύτερες δυνατότητες ευελιξίας και προσαρμοστικότητας.

IETF και τα υπόλοιπα αφορούν ειδικά στον ελεγκτή και τις εφαρμογές του. Περίπου 10 μοντέλα περιγράφουν IETF πρωτόκολλα PDUs (Protocol Data Units), 27 μοντέλα περιγράφουν διάφορες πτυχές του OpenFlow πρωτοκόλλου, 35 μοντέλα ορίζουν τις εσωτερικές συνδέσεις του ελεγκτή και της διαμόρφωσής του. Περίπου 15 πρόσθετα μοντέλα χρησιμοποιούνται ως υποδείγματα και για την απόδειξη της ορθότητας της χρήσης τους.

5.4.4 Model-Driven Protocol & Application Plugins

Με την κυκλοφορία της έκδοσης Hydrogen, εφαρμόστηκαν αρκετά πρωτόκολλα προσθηκών βασισμένα στην MD-SAL. Η OpenFlow προσθήκη εφαρμόζει την έκδοση OF 1.0 & 1.3 [86]. Η προσθήκη BGP-LS/PCEP εφαρμόζει μαζί ένα BGP ακροατή (listener) που υποστηρίζει IPv4 & IPv6/AFI/SAFIs [87], το σύνδεσμο κατάστασης (link-state) AFI/SAFI [88] και ένα PCEP (Path Computation Element Protocol) [89] ηχείου το οποίο υποστηρίζει το φίλτρο προστασίας του. Ο σύνδεσμος NETCONF, όχι αυστηρά ως μια προσθήκη, αλλά ως μέρος του MD-SAL, εφαρμόζει από την πλευρά του αποδέκτη (πελάτη) το NETCONF πρωτόκολλο. Επιπλέον των πρωτοκόλλων προσθηκών, ένας αριθμός MD-SAL, βασίζεται στις λειτουργίες Εφαρμογών/Υπηρεσιών για την υλοποίηση της έκδοσης Hydrogen του ODL. Αυτές κατατάσσοντια σε δύο κατηγορίες, στις βασικές δικτυακές λειτουργίες (base network functions) και τις λειτουργίες υπηρεσιών (service functions). Στην πρώτη κατηγορία συναντάμε δύο βασικές υπηρεσίες, την Τοπολογία (Topology Explorer) και την Απογραφή (Inventory Manager). Στη δεύτερη κατηγορία των υπηρεσιών, αυτές υλοποιούνται μέσω του OpenFlow (Forwarding manager & Statistic Manager plugins) και το PCEP (Transaction programming & RSVP-TE tunnels).

5.5.5 MD-SAL –Αναλυτική Περιγραφή & Σχεδιασμός

1. Ορισμοί

Οι ορισμοί που χρησιμοποιούνται στο σχεδιασμό μιας MD-SAL είναι οι ακόλουθοι: *Η Διαδικασία Κλήσης εξ Αποστάσεως (Remote Procedure Call-RPC),* πρόκειται για μια κλίση ενός-προς-ένα, η οποία προκαλείται από ένα Καταναλωτή και η οποία είναι

αλλαγών στις εφαρμογές του ελεγκτή στο υποκείμενο δίκτυο. Ακολούθησαν οι εκδόσεις Lithium (2015), η οποία επεξέτεινε τις δυνατότητες προγραμματισμότητας του δικτύου, Beryllium (2/2016) και Boron ((/2016).

δυνατόν να υποστεί επεξεργασία από ένα Πάροχο, είτε τοπικά είτε εξ αποστάσεως. *Η Ενημέρωση (Notification)* είναι ένα συμβάν που ο Καταναλωτής ενδιαφέρεται να λάβει και η οποία προκαλείται σε ένα Πάροχο. *Αποθήκη Δεδομένων(Data Store)* είναι ένα νοητό δένδρο δεδομένων το οποίο περιγράφεται από σχήματα YANG. Η *Διαδρομή (Path)* είναι ένας μοναδικός "εντοπιστής" ενός φύλλου ή υπο-δένδρου στο νοητό δένδρο δεδομένων. Τέλος *η Προσάρτηση (Mount),* η οποία είναι ένας λογικός ένθετος (logic-nested) κλώνος (instance) στην MD-SAL, που μπορεί να χρησιμοποιεί μια διαφορετική σειρά μοντέλων YANG και η οποία υποστηρίζει τις δικές της εξ Αποστάσεων Κλήσεις (RPCs) και Ενημερώσεις και επιτρέπει την επανάχρηση μοντέλων συσκευών και ένα περιβάλλον στο ευρύτερο πλαίσιο του δικτύου, χωρίς να απαιτείται ο επανακαθορισμός των μοντέλων συσκευών στον ελεγκτή.

2. Λειτουργικότητα MD-SAL & Αναπαραστάσεις Δεδομένων (MD-SAL Functionality & Data Representations)

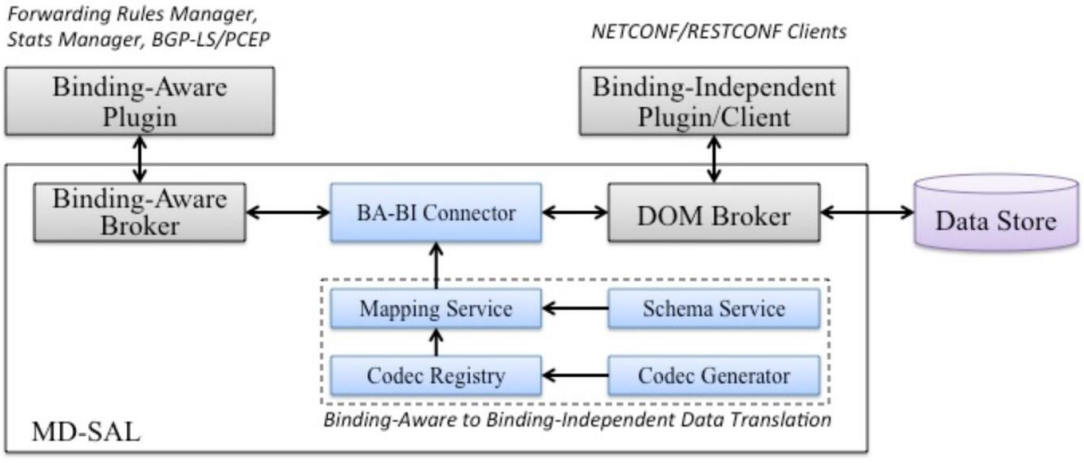
Η MD-SAL παρέχει μια πλειάδα λειτουργιών που απαιτούνται για την προσαρμογή της συνεργασίας μεταξύ Καταναλωτών και Παρόχων. Αρχικά δρομολογεί RPC κλήσεις μεταξύ Καταναλωτών και Παρόχων (RPC Call Router). Κατά δεύτερον παρέχει μηχανισμούς βασισμένους στις συνδρομές για την παράδοση των Ενημερώσεων από τους Εκδότες στους Συνδρομητές (Notification Broker). Τρίτον δρομολογεί αναγνώσεις δεδομένων από τους καταναλωτές σε μια συγκεκριμένη αποθήκη δεδομένων (Data Stores) και συντονίζει τις αλλαγές/τροποποιήσεις μεταξύ Παρόχων (Data Broker). Τέλος, δημιουργεί και διαχειρίζεται Προσαρτήσεις (Mount manager).

Η εφαρμογή των προαναφερθεισών λειτουργιών SAL, απαιτεί τη χρήση δύο αναπαραστάσεων δεδομένων και δύο σειρές από SAL APIs προσθήκες. Ο Ανεξάρτητος Διασύνδεσης μορφότυπος δεδομένων/εφαρμογών(Binding-Independent data format/APIs) είναι ένα Μοντέλο Αντικειμένου Δεδομένων (Data Object Model-DOM) αναπαραστάσεων δένδρων YANG. Αυτός ο μορφότυπος είναι κατάλληλος για κοινά δομικά στοιχεία του δικτύου, όπως οι αποθήκες δεδομένων, ο Συνδετήρας NETCONF (NETCONF Connector), το RESTCONF, που μπορεί να αναπτύξει τη συμπεριφορά του από το ίδιο το μοντέλο YANG. Η Σύνδεση Ενημέρωσης του μορφότυπου δεδομένων/εφαρμογών (Binding-Aware data format/APIs) είναι μια

εξειδικευμένη γλώσσα σύνδεσης YANG προς την JAVA, η οποία καθορίζει πως τα JAVA DOTs (Data Transfer Objects) και οι εφαρμογές (APIs) παράγονται από το μοντέλο YANG. Ο ορισμός αυτών των APIs εφαρμογών αυτών των DOTs, η διεπαφές κλήσης/εφαρμογής RPCs, διαπαφές που περιέχουν Ενημερώσεις επανάκλησης, δημιουργούνται κατά το χρόνο μεταγλώττισης. Codecs[19] για την μετάφραση μεταξύ των JAVA DOTs και των DOM απεικονίσεων, παράγονται κατά παραγγελία σε τρέχοντα χρόνο.

3. Σχεδιασμός MD-SAL

Ο λειτουργικός χειρισμός των δεδομένων γίνεται με τη παρέμβαση δύο διακριτών Διαμεσολαβητών (Brokers): ενός ανεξάρτητου διασυνδεδεμένου DOM Διαμεσολαβητή (binding – independent DOM Broker), ο οποίος ερμηνεύει τα μοντέλα YANG σε τρέχοντα χρόνο και είναι ένα κομβικό στοιχείο της MD-SAL σε τρέχοντα χρόνο, και ενός Διασυνδεδεμένου Διαμεσολαβητή Ετοιμότητας (Binding-Aware Broker), ο οποίος αποκαλύπτει τις προσθήκες για JAVA εφαρμογές (APIs), χρησιμοποιώντας τις αναπαραστάσεις δεδομένων του Διαμεσολαβητή Ετοιμότητας (JAVA DOTs). Οι προαναφερθέντες Διαμεσολαβητές με τα υποστηρικτικά στοιχεία παρουσιάζονται στο *Σχήμα 8*.



Σχήμα 8 : Σχεδιασμός MD-SAL

Ο Διαμεσολαβητής DOM χρησιμοποιεί εφαρμογές (APIs) δεδομένων YANG για να περιγράψει δεδομένα και εξειδικευμένες Αναγνωρίσεις Κλώνων (Instance

[19] Ο όρος codec προέρχεται από τη σύντμηση των λέξεων coder (κωδικοποιητής) και decoder (αποκωδικοποιητής).

Identifiers)[20] YANG, για να καθορίσει τις κατάλληλες διαδρομές προς τα δεδομένα του συστήματος. Οι δομές των δεδομένων στον Διαμεσολαβητή Ετοιμότητας, που είναι ορατές στις εφαρμογές, παράγονται από τα μοντέλα YANG σε εργαλεία YANG. Ο Διαμεσολαβητής DOM στηρίζεται στην παρουσία σχημάτων YANG, τα οποία ερμηνεύονται σε τρέχοντα χρόνο για ειδικούς λειτουργικούς σκοπούς, όπως είναι η RPC δρομολόγηση, η οργάνωση αποθηκών δεδομένων και επαλήθευση διαδρομών.

Ο Διαμεσολαβητής Ετοιμότητας βασίζεται σε JAVA εφαρμογές (APIs), οι οποίες δημιουργούνται από μοντέλα YANG, καθώς και στις κοινές ιδιότητες των JAVA DOTs, που επιβάλλονται από κώδικα δημιουργίας. Ως εκ τούτου η βελτιστοποίηση της μεταφοράς των δεδομένων (zero-copy) καθίσταται εφικτή, όταν τόσο ο Καταναλωτής Δεδομένων, όσο και ο Πάροχος Δεδομένων είναι Συνδεδεμένης Ετοιμότητας.

Ο Διαμεσολαβητής Ετοιμότητας συνδέεται στον Διαμεσολαβητή DOM μέσω ενός Συνδέσμου BA-BI (binding independence-binding aware), ώστε η σύνδεση ετοιμότητας (B-A) του Καταναλωτή/εφαρμογών του Παρόχου/προσθηκών να έχουν τη δυνατότητα επικοινωνίας με τους ανάλογους B-I ομολόγους τους. Ο BA-BI Σύνδεσμος από κοινού με την Υπηρεσία Απεικόνισης (Mapping Service), την Υπηρεσία Σχημάτων (Schema Service), με το Μητρώο Codec, και τον Δημιουργό Codec, υλοποιούν μια δυναμική ύστερη σύνδεση: τα codecs τα οποία μεταφράζουν YANG αναπαραστάσεις δεδομένων μεταξύ ενός BI-DOM μορφότυπου και DOTs, πού είναι εξειδικευμένες σε JAVA συνδέσεις, παράγονται αυτόματα όταν αυτό ζητηθεί (on demand).

Οι φυσικές Αποθήκες Δεδομένων μπορούν να προστεθούν στο σύστημα και η MD-SAL παρέχει μια διεπαφή SPI (Serial Peripheral Interface), μέσω της οποίας διαφορετικές εφαρμογές Αποθηκών Δεδομένων μπορούν να προστεθούν στο δίκτυο.

Η έννοια της Προσάρτησης (Mount) και η υποστήριξη για APIs που παράγονται από τα μοντέλα, επιτρέπουν στις εφαρμογές που "συνομιλούν" με τις συσκευές NETCONF και να μεταγλωττίζονται απευθείας στα μοντέλα συσκευών. Τα μοντέλα συσκευών "φορτώνονται" στο ελεγκτή από μια συσκευή NETCONF, όταν ο ελεγκτής συνδέεται με τη συγκεκριμένη συσκευή και οι εφαρμογές μπορούν πλέον να εργάζονται απευθείας με αυτές.

[20] Instance Identifier είναι μια σειριακή συσκευή αναγνώρισης, η οποία ξεχωρίζει μια συσκευή από άλλες του ίδιου τύπου, σε ένα υπολογιστή.

6. Open Network Operating System (ONOS)

6.1 Γενικά για το ONOS

Αναφορές και στοιχεία σχετικά με το ONOS έχουν παρουσιαστεί στο Κεφάλαιο 4.4.1, όταν περιγράφτηκαν οι Βόρειες Διεπαφές (NBIs), σε σχέση με τους ελεγκτές που μπορούσαν να υποστηρίξουν αποτελεσματικά τις λειτουργίες του επιπεδου εφαρμογών ενός SDN δικτύου. Στο παρόν Κεφάλαιο παρουσιάζεται αναλυτικότερα η αρχιτεκτονική του ONOS, οι δυνατότητες που παρέχει σε ένα SDN και τα επιμέρους στοιχεία που το συγκροτούν, καθώς και οι λειτουργίες τους στο δίκτυο.

Το Ανοιχτό Δικτυακό Λειτουργικό Σύστημα (ONOS) [90],[66] είναι το πρώτο δικτυακό λειτουργικό σύστημα ανοικτού κώδικα για Δίκτυα Καθοριζόμενα από Λογισμικό (SDN), το οποίο στοχεύει στην εξυπηρέτηση Παρόχων Υπηρεσιών και δικτύων κρίσιμης σημασίας (mission critical networks)[21]. Ο σκοπός της δημιουργίας του ONOS είναι η δυνατότητα παροχής υψηλής διαθεσιμότητας (high availability), οριζόντιας κλιμάκωσης (scale-out) και απόδοσης (performance) στις συνεχώς αυξανόμενες απαιτήσεις και ανάγκες των δικτύων. Επιπρόσθετα στο ONOS δημιουργήθηκαν χρήσιμες NB (Northbound) αφαιρέσεις και διεπαφές (APIs), που παρέχουν τη δυνατότητα εύκολης ανάπτυξης εφαρμογών,καθώς και SB (Southbound) αφαιρέσεις και διεπαφές, ώστε να είναι σε θέση να ελέγχει την ετοιμότητα του OpenFlow όπως και τις παλαιού τύπου συσκευές (legacy devices)[22]. Ως εκ τούτου το ONOS :

- Μεταφέρει χαρακτηριστικά κλιμακούμενης ικανότητας αναβάθμισης (carrier grade) στο δίκτυο όπως η κλιμάκωση, η διαθεσιμότητα, η απόδοση κλπ,
- Παρέχει τη δυνατότητα ευκινησίας τύπου Ιστού (Web).

[21] Ένας παράγοντας κρίσιμης σημασίας (mission critical factor) για ένα σύστημα είναι οποιοσδήποτε παράγοντας (στοιχείο, εξοπλισμός, προσωπικό, διαδικασία, λογισμικό κ.λπ.) που είναι απαραίτητος για την επιχειρησιακή λειτουργία σε έναν οργανισμό. Η αποτυχία ή η διατάραξη των κρίσιμων παραγόντων της αποστολής θα έχει σοβαρές επιπτώσεις στις επιχειρηματικές δραστηριότητες ή σε έναν οργανισμό και μπορεί ακόμη και να προκαλέσει κοινωνική αναταραχή και καταστροφές.
https://en.wikipedia.org/wiki/Mission_critical#Safety.

[22] Παλαιού τύπου (απαρχαιωμένες) συσκευές ή εξοπλισμός.

- Βοηθά τους παρόχους υπηρεσιών να μεταφέρουν τα υπάρχοντα δίκτυά τους σε "λευκά κουτιά"[23].
- Μειώνει το κόστος της επένδυσης (CapEx) και λειτουργίας (OpEx) του Παρόχου Υπηρεσιών.

Το ONOS αναπτύχθηκε με την σύμπραξη μεγάλων Παρόχων Υπηρεσιών (AT & T, NTT Communications), απαιτητικών κατασκευαστών δικτυακού εξοπλισμού (Ciena, Ericsson, Fujitsu, Huawei, Intel, NEC), φορέων εκμετάλλευσης δικτύων Έρευνας και Εκπαίδευσης (Internet 2, CNIT, CREATE-NET), άλλων συνεργατών (SRI, Infobox), με τον ONF να επαληθεύει την αρχιτεκτονική του μέσω πραγματικών περιπτώσεων χρήσης του.

Η εκθετική αύξηση των κινητών συσκευών τα τελευταία χρόνια, οι OTT (over-the-top) υπηρεσίες[24] και η χρήση υπηρεσιών νέφους, έχουν οδηγήσει τα δίκτυα των Παρόχων Υπηρεσιών σε σημείο που να απαιτείται άμεσα η επανεφεύρεσή τους. Οι Πάροχοι Υπηρεσιών επιδιώκουν να καταστήσουν τα δίκτυά τους ευέλικτα και αποτελεσματικά, προκειμένου να αντιμετωπίσουν τις προκλήσεις της εκθετικής ζήτησης εύρους ζώνης, καθώς και να είναι σε θέση να δημιουργήσουν νέες οικονομικές ροές προσφέροντας καινοτόμες υπηρεσίες και νέα επιχειρησιακά πρότυπα.

Όπως ήδη αναφέρθηκε σε προηγούμενα Κεφάλαια, αυτό που επιτρέπει την ανάπτυξη καινοτομιών στα SDN δίκτυα είναι ο διαχωρισμός κατά ολοκληρωμένο κάθετο τρόπο του Επιπέδου Ελέγχου από το Επίπεδο Δεδομένων/Προώθησης, μεταξύ των συσκευών του δικτύου. Ένα ανοιχτό πρωτόκολλο όπως το OpenFlow, επιτρέπει στο Επίπεδο Ελέγχου να προγραμματίζει το Επίπεδο Δεδομένων με περισσότερο ανοιχτό και αποτελεσματικό τρόπο. Επιπρόσθετα αυτός ο διαχωρισμός επιτρέπει στο υλισμικό και το λογισμικό να αναπτύσσονται ανεξάρτητα, διευκολύνοντας έτσι την αντικατάσταση δαπανηρού υλισμικού και μνημών με άλλα κοινά υλικά διαθέσιμα στην αγορά, καθώς και με λογισμικό ανοιχτού κώδικα.

[23] Η δικτύωση λευκού κουτιού "White box networking" αναφέρεται στη δόμηση ενός δικτύου με τη χρήση μεταγωγέων, «λευκών κουτιών», οι οποίοι είναι κατασκευασμένοι από υλισμικό ευρείας κατανάλωσης (commodity hardware) και "τρέχουν" σε ένα δικτυακό λειτουργικό σύστημα (NOS) κατά προτίμηση ανοιχτού τύπου.

[24] OTT (over-the-top) είναι μια υπηρεσία πολυμέσων, ενός οποιουδήποτε παρόχου διαδικτυακού περιεχομένου, η οποία προσφέρει υπηρεσίες περιεχομένου ροής πολυμέσων (streaming media), ως αυτόνομο προϊόν. Ο όρος συνήθως αναφέρεται σε πλατφόρμες βίντεο κατά παραγγελία, αλλά επίσης αναφέρεται και για περιεχόμενο ροής ήχου (audio streaming), υπηρεσίες ανταλλαγής μηνυμάτων, λύσεις φωνητικής κλήσης μέσω διαδικτύου κλπ.

Η ύπαρξη ενός λειτουργικού προγράμματος που είναι σε θέση να παρακολουθεί τους πόρους του δικτύου και να διαχειρίζεται τις αφαιρέσεις και τις APIs, την παρακολούθηση και τον προγραμματισμό των συσκευών του, απλοποιεί σε μεγάλο βαθμό τη δημιουργία καινοτόμων και χρήσιμων εφαρμογών δικτύου και υπηρεσιών, σε ένα ευρύ φάσμα υλισμικού.

Το ONOS (Open Network Operating System), δημιουργήθηκε και αναπτύχθηκε, ως ένα τέτοιας μορφής λειτουργικό σύστημα, που υπηρετεί τους ακόλουθους σκοπούς:

i. Απελευθερώνει τους προγραμματιστές εφαρμογών του δικτύου από τις επιπλοκές που προέρχονται από τη χρήση λογισμικού συγκεκριμένων κατασκευαστών.

ii. Επιτρέπει στους χειριστές του δικτύου να ανεξαρτητοποιηθούν από τις λειτουργικές πολυπλοκότητες των διεπαφών και των πρωτοκόλλων, που προέρχονται από συγκεκριμένους κατασκευαστές.

iii. Επιτρέπει την εφαρμογή καινοτομιών τόσο στο υλισμικό, όσο και το λογισμικό ανεξαρτήτως κλίμακας.

6.2 Network Operating System (OS)

Υπάρχουν και κυκλοφορούν στην αγορά πολλοί ανοιχτού κώδικα Ελεγκτές SDN. Είναι λοιπόν λογικό να τίθεται το ερώτημα σχετικά με το ρόλο και τη σημασία του ONOS και του OS στους Ελεγκτές SDN.

Η συνεργασία ειδικών στα δίκτυα από τα πανεπιστήμια του Stanford και του Berkley και της εταιρείας Nicira, είχε ως αποτέλεσμα την ανάπτυξη ενός αριθμού ελεγκτών ανοιχτού κώδικα, όπως ο NOX, ο Beacon, ο SNAC και ο POX, τα τελευταία χρόνια. Οι συγκεκριμένοι ελεγκτές σχεδιάστηκαν με τρόπο ώστε να αξιοποιούν τις δυνατότητες που παρέχουν τα SDN δίκτυα. Είναι σημαντικό να γίνει αντιληπτό, ότι η ανάπτυξη των προαναφερθέντων ελεγκτών δεν έγινε για εμπορικούς σκοπούς, δηλαδή τη δημιουργία συγκεκριμένων εμπορικών προϊόντων. Και τούτο γιατί οι εν λόγω ελεγκτές δεν διαθέτουν τα ειδικά χαρακτηριστικά και ιδιότητες, όπως η επεκτασιμότητα, η υψηλή διαθεσιμότητα και η απόδοση. Επιπλέον έχουν περιορισμένες δυνατότητες προγραμματισμού και αφαιρέσεων προσανατολισμένες στις αφαιρέσεις βασισμένες στις συσκευές του δικτύου (device-oriented abstractions).

Οι προαναφερθέντες ελεγκτές ουσιαστικά μεταφέρουν/μεταδίδουν μηνύματα OpenFlow απευθείας στις εφαρμογές, οι οποίες στη συνέχεια δημιουργούν μηνύματα OpenFlow προς τις συσκευές του δικτύου. Με τη λειτουργία τους αυτή, οι ελεγκτές ενεργούν περισσότερο ως οδηγοί συσκευών (device drivers) και δεν σχεδιάστηκαν για να παρέχουν τα κρίσιμης σημασίας στοιχεία επεκτασιμότητας, διαθεσιμότητας και απόδοσης για την πλήρη αξιοποίηση των δυνατοτήτων μιας ολοκληρωμένης SDN πλατφόρμας. Η δημιουργία του OS-ONOS ήρθε να καλύψει αυτή την ανάγκη.

Σημειώνεται ότι ένα λειτουργικό σύστημα (OS) είναι υπεύθυνο για τις ακόλουθες λειτουργίες του δικτύου:

α. Διαχειρίζεται πεπερασμένους πόρους (finite resources) για λογαριασμό των καταναλωτών πόρων και διασφαλίζει στους τελευταίους πρόσβαση σ' αυτούς κατά τα τρόπο που να ικανοποιούνται οι ανάγκες τους και να έχουν όλοι ισότιμη μεταχείριση.

β. Απομονώνει τους χρήστες του NOS, προστατεύοντας παράλληλα τον ένα από τον άλλο, προκειμένου καθένας να έχει πρόσβαση σε ένα πλήρες σετ πόρων. Επιτρέπει πολύπλοκες λειτουργίες μεταξύ πολλαπλών εφαρμογών και πολλαπλών συσκευών. Μπορεί επίσης να εικονοποιεί μερικούς ή όλους τους πόρους, προκειμένου να επιτρέπει στους καταναλωτές (πόρων) να έχουν την δική τους εικονοποιημένη εκδοχή του OS.

γ. Παρέχει χρήσιμες αφαιρέσεις (abstractions) επιτρέποντας στους χρήστες να χρησιμοποιούν (καταναλώνουν) υπηρεσίες και πόρους που διαχειρίζεται το σύστημα, χωρίς να απαιτείται η κατανόηση της όλης πολυπλοκότητάς του. Διαθέτει μηχανισμούς για την προσθήκη διαφόρων συσκευών με εύκολο τρόπο στο δίκτυο και τον έλεγχό τους από το λειτουργικό σύστημα, χωρίς να απαιτείται τροποποίηση της εφαρμογής.

δ. Διασφαλίζει την προστασία των χρηστών του λειτουργικού συστήματος από εξωτερικές πηγές.

ε. Προμηθεύει με υπηρεσίες του χρήστες του λειτουργικού συστήματος, ώστε να μην απαιτείται η εκ νέου ανάπτυξή τους.

Τα ανωτέρω αποτελούν τα απαραίτητα στοιχεία που απαιτούνται για να "τρέξουν" οι εφαρμογές σε ένα δίκτυο. Ο ελεγκτής είναι τυπικά περιορισμένος σε ότι αφορά στο πεδίο εφαρμογής του – απλά ελέγχει μια συσκευή του δικτύου. Δεν είναι απαραίτητο

να παρέχει χρήσιμες αφαιρέσεις, δεν προστατεύει απαραιτήτως τους διαφορετικούς χρήστες του ελεγκτή μεταξύ τους και δεν παρέχει πρόσθετες υπηρεσίες. Το ONOS δημιουργήθηκε για να παρέχει λειτουργικότητα στο χειρισμό ενός συστήματος (operating system) και όχι μόνο λειτουργικότητα στον ελεγκτή. Έτσι η αρχιτεκτονική του ONOS, σχεδιάστηκε εξ αρχής έχοντας στο επίκεντρο τα χαρακτηριστικά του **παρόχου υπηρεσιών.**

6.3 Αρχιτεκτονική του ONOS

Όπως ήδη αναφέρθηκε το ONOS σχεδιάστηκε έχοντας ως προτεραιότητα τους Παρόχους Υπηρεσιών. Η υψηλή διαθεσιμότητα, οι δυνατότητες επέκτασης και η απόδοση, όπως και οι ισχυρές δυνατότητες αφαιρέσεων στις NB & SB διεπαφές, αποτέλεσαν τις βασικές παραμέτρους σχεδιασμού του. Τα ακόλουθα αποτελούν εξ ορισμού τα χαρακτηριστικά του ONOS και διακρίνονται στο *Σχήμα 9*:

- *Distributed Core(Κατανεμημένος Πυρήνας)*, ο οποίος παρέχει επεκτασιμότητα, υψηλή διαθεσιμότητα και απόδοση, δίνοντας ταυτόχρονα και χαρακτηριστικά κλιμακούμενης ικανότητας αναβάθμισης (carrier grade) στο Επίπεδο Ελέγχου του SDN δικτύου. Η ικανότητα του ONOS να "τρέχει" ως συστάδα (cluster) είναι ένα χαρακτηριστικό που καταδεικνύει την δυνατότητά ευελιξίας τύπου Web που δίνει στο Επίπεδο Ελέγχου του SDN στα δίκτυα των Παρόχων Υπηρεσιών.

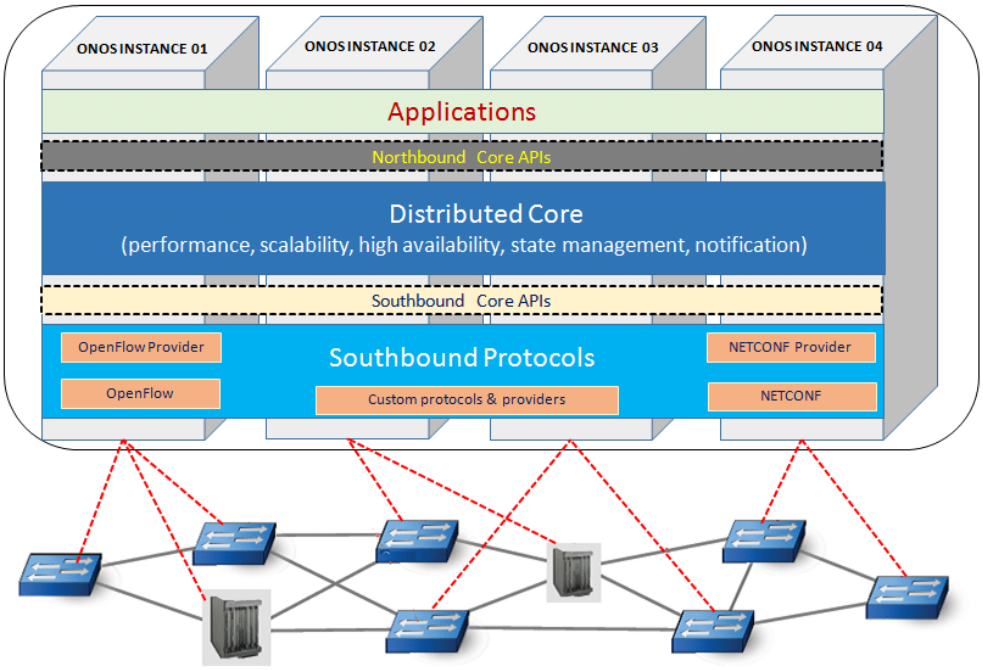
- *Northbound (NB) abstractions/APIs (Προς Βορρά Αφαιρέσεις & Προγραμματισμένες Εφαρμογές Διεπαφών)* που περιλαμβάνουν γράφημα του δικτύου και εφαρμογών σκοπού (application intents), που διευκολύνουν την ανάπτυξη υπηρεσιών ελέγχου, διαχείρισης και διαμόρφωσης. Αυτή η αφαίρεση αποτελεί ένα ακόμα καλό παράδειγμα για το πως το ONOS δημιουργεί ευελιξία τύπου παγκόσμιου ιστού στο Επίπεδο Ελέγχου του SDN και στους Παρόχους Υπηρεσιών.

- *Southbound (SB) abstractions/APIs (Προς Νότο Αφαιρέσεις & Προγραμματισμένες Εφαρμογές Διεπαφών)* που παρέχουν τη δυνατότητα προσθηκών (pluggable) πρωτοκόλλων για τον έλεγχο τόσο του OpenFlow, όσο και του παλαιού τύπου συσκευών (legacy devices). Η νότια αφαίρεση απομονώνει τον πυρήνα του ONOS, από τις λεπτομέρειες των διαφόρων συσκευών και πρωτοκόλλων που χρησιμοποιούνται στο δίκτυο. Η SB διεπαφή είναι το στοιχείο "κλειδί" που

διευκολύνει τη μετάβαση από τις παλαιού τύπου συσκευές, στα βασιζόμενα στο OpenFlow "λευκά κουτιά".

-*Software Modularity(Η Αρθρωτή Μορφή Λογισμικού)* η οποία καθιστά εύκολη την ανάπτυξη, την εξάλειψη σφαλμάτων (debugging), την συντήρηση και την αναβάθμιση του ONOS ως συστήματος λογισμικού, από τους προγραμματιστές και τους Παρόχους Υπηρεσιών.

Καθένα από τα προαναφερθέντα χαρακτηριστικά περιγράφεται συνοπτικά στη συνέχεια.



Σχήμα 9 : Αρχιτεκτονική του ONOS

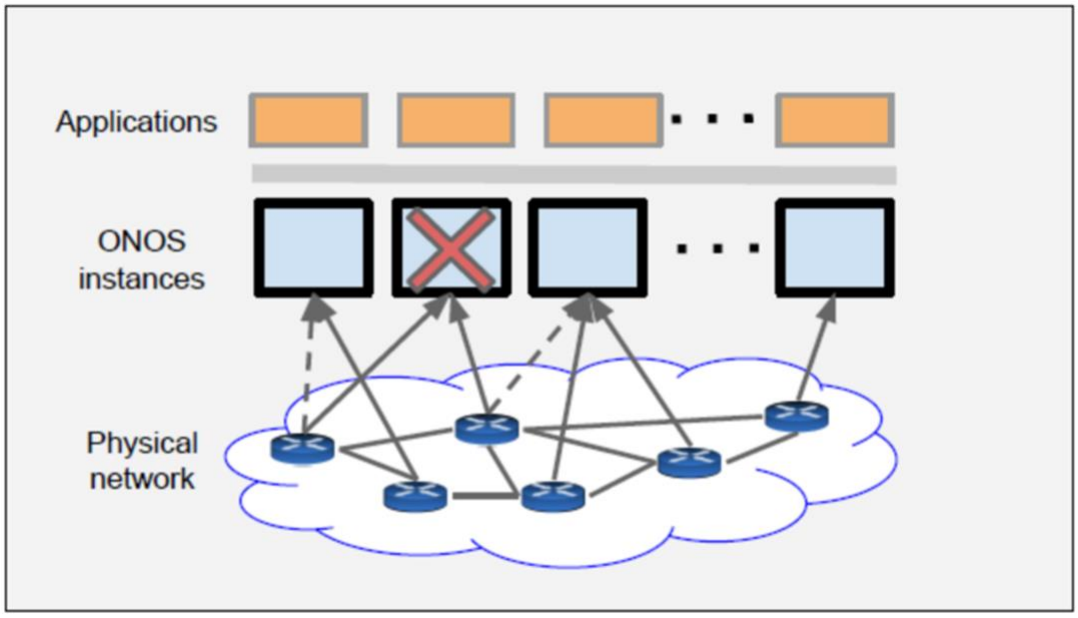
6.3.1 Κατανεμημένος Πυρήνας

Το ONOS έχει αναπτυχθεί ως μια υπηρεσία σε μια συστάδα από διακομιστές (cluster servers) και το ίδιο λογισμικό ONOS "τρέχει" σε κάθε ένα από αυτούς τους διακομιστές. Η ανάπτυξη της συμμετρίας του συγκεκριμένου συστήματος αποτελεί σημαντική παράμετρο σχεδιασμού του, γιατί επιτρέπει την ανάταξη του συστήματος στις περιπτώσεις που κάποιος από τους διακομιστές του συστήματος αντιμετωπίσει πρόβλημα/βλάβη. Ο χειριστής του δικτύου μπορεί να προσθέτει σταδιακά και χωρίς διακοπή ή διατάραξη του συστήματος διακομιστές, όποτε χρειάζεται πρόσθετη δυναμικότητα στο σύστημα. Όλοι οι κλώνοι του ONOS (ONOS instances),

συνεργάζονται προκειμένου να δημιουργήσουν ότι εμφανίζεται στο υπόλοιπο δίκτυο και τις εφαρμογές, ως μια ενιαία πλατφόρμα. Οι εφαρμογές και οι συσκευές του δικτύου δεν χρειάζεται να γνωρίζουν αν εργάζονται με έναν ή περισσότερους κλώνους του ONOS. Αυτό το συγκεκριμένο χαρακτηριστικό καθιστά επεκτάσιμη τη χωρητικότητα της πλατφόρμα ONOS, χωρίς προβλήματα. Ο κατανεμημένος πυρήνας είναι αυτός που εκτελεί το βαρύ έργο και υλοποιεί τις δυνατότητες της πλατφόρμας. Ο κατανεμημένος πυρήνας παρέχει μηνύματα, διαχείριση της κατάστασης και είναι ο οδηγός επιλογής υπηρεσιών (leader election services) προς τους κλώνους, αλλά και μεταξύ αυτών. Ως αποτέλεσμα αυτού του γεγονότος πολλαπλοί κλώνοι, συμπεριφέρονται ως μια και μοναδική λογική οντότητα. Χρησιμοποιώντας υψηλής ταχύτητας μηνύματα σε ένα δημοσιευμένο ή συνδρομητικό μοντέλο,κλώνοι μπορούν ταχύτατα να ενημερώνουν τους άλλους κλώνους για τις εκτελούμενες αναβαθμίσεις. Ακόμα στο ONOS υπάρχουν πρωτόκολλα ανάκτησης, τα οποία λειτουργούν για τις περιπτώσεις απώλειας των αναβαθμίσεων και για τις περιπτώσεις αποτυχίας των κλώνων. Χρησιμοποιώντας διάφορους μηχανισμούς είναι δυνατή η διαχείριση της λειτουργικής κατάστασης μεταξύ των κλώνων, καθένας δε από αυτούς είναι κατάλληλος για ένα τύπο λειτουργικής κατάστασης. Τρία παραδείγματα περιλαμβάνουν, τις εφαρμογές σκοπού (application intents), την τοπολογική βάση δεδομένων και τους πίνακες ροής όπου καθένα από αυτά τα στοιχεία έχει ένα μοναδικό μέγεθος, αποτύπωμα ανάγνωσης/γραφής και απαιτήσεις αντοχής. Ο ηγέτης επιλογής υπηρεσίας (leader election service) εξασφαλίζει ότι κάθε μεταγωγέας έχει ένα και μοναδικό κύριο κλώνο (master instance). Από κοινού αυτοί οι τρείς μηχανισμοί, των μηνυμάτων, της διαχείρισης κατάστασης και ο οδηγός επιλογής υπηρεσιών, καθιστούν δυνατή την υψηλή διεκπεραίωση, τη χαμηλή αδράνεια και την υψηλή διαθεσιμότητα της συστάδας. Αυτό σημαίνει για τις συσκευές ότι πάντοτε θα υπάρχει ένας κύριος κλώνος και στην περίπτωση που αυτός καταπέσει, αυτές μπορούν να συνδεθούν με ένα άλλο κλώνο, χωρίς να απαιτείται η επαναδημιουργία και ο επανασυγχρονισμός των πινάκων ροής. Ώστοσο οι εφαρμογές αυτές μπορούν να στηρίζονται στο γεγονός ότι θα έχουν συνεχή εικόνα του δικτύου, μέσω του γραφήματος αφαίρεσης του δικτύου. Επιπλέον η αποτυχία (κατάπτωση) ενός κλώνου ή μια αποτυχία στο Επίπεδο Δεδομένων/Προώθησης γίνεται αντιληπτή στην εφαρμογή. Το γεγονός αυτό απλοποιεί σε μεγάλο βαθμό την ανάπτυξη εφαρμογών και τη διαχείριση σφαλμάτων.

Από επιχειρησιακή άποψη, ο κατανεμημένος πυρήνας δημιουργεί ένα υψηλής διαθεσιμότητας περιβάλλον, ώστε οι εφαρμογές του δικτύου να λειτουργούν χωρίς χρονικούς περιορισμούς και καθυστερήσεις. Έτσι ο Πάροχος Υπηρεσιών μπορεί εύκολα να προσθέτει δυναμικότητα στο Επίπεδο Ελέγχου όσο το δίκτυο επεκτείνεται, χωρίς να προκαλεί διαταραχές σ' αυτό. Μέσω του ίδιου μηχανισμού ο διαχειριστής του δικτύου έχει τη δυνατότητα αναβάθμισης του λογισμικού, χωρίς χρονικές υστερήσεις, αποσύροντας ένα κλώνο (instance) από το σύστημα, αναβαθμίζοντάς τον και επαναφέροντάς τον σ' αυτό.

Συμπερασματικά ο κατανεμημένος πυρήνας αποτελεί βασικό αρχιτεκτονικό χαρακτηριστικό του ONOS, δεδομένου ότι παρέχει σημαντικές δυνατότητες βελτίωσης των ιδιοτήτων και των επιδόσεων του Επιπέδου Ελέγχου ενός SDN δικτύου.



Σχήμα 10 : Μορφή Κατανεμημένου Πυρήνα

6.3.2 Northbound Abstractions

Η πλατφόρμα του ONOS παρέχει δύο πολύ ισχυρές αφαιρέσεις: Το Πλαίσιο Σκοπού (Intent Framework) και τη Συνολική Εικόνα του Δικτύου (Global Network View).

Το Πλαίσιο Σκοπού επιτρέπει σε μια εφαρμογή να αιτηθεί για τη λήψη μιας υπηρεσίας από το δίκτυο, χωρίς να είναι απαραίτητη η γνώση λεπτομερειών για την απόδοσή της. Επιτρέπει στους χειριστές του δικτύου, όπως και στους υπεύθυνους

ανάπτυξης εφαρμογών, να προγραμματίζουν σε υψηλό επίπεδο το δίκτυο, καθορίζοντας απλώς το Σκοπό τους που είναι μια δήλωση πολιτικής ή απαιτήσεις συνδεσιμότητας. Αναφέρονται μερικά παραδείγματα Σκοπών:

- Δημιουργία σύνδεσης μεταξύ υποδοχέα (Host) A και υποδοχέα B.
- Δημιουργία μιας Οπτικής Διαδρομής (Optical Path) από τον μεταγωγέα X στον μεταγωγέα Y με Z εύρος ζώνης.
- Απαγόρευση επικοινωνίας μεταξύ των μεταγωγέων A και B.

Το Πλαίσιο Σκοπού λαμβάνει τέτοιου είδους αιτήματα και καθορίζει ποιά από αυτά μπορούν να ενταχθούν στο σύστημα και ποια όχι, επιλύει τυχόν συγκρούσεις μεταξύ των εφαρμογών, υλοποιεί πολιτικές που καθορίζονται από ένα διαχειριστή, προγραμματίζει το δίκτυο για την παροχή της αιτηθείσας λειτουργικότητας και προωθεί (παραδίδει) την αιτηθείσα υπηρεσία στην εφαρμογή.

Ένας Σκοπός μετατρέπεται σε πολλαπλά αντικείμενα (multiple objectives), όπως για παράδειγμα ένας Σκοπός που έχει ως στόχο τη σύνδεση δύο υποδοχέων, μεταφράζεται σε δύο αντικείμενα (objects), το καθένα από τα οποία παρέχει μια κατεύθυνση ροής. Τα αντικείμενα συντάσσονται υπό μορφή οδηγιών, οι οποίες αποστέλλονται στις συσκευές. Η συγκεκριμένη διεργασία γίνεται υπό τον έλεγχο των πολιτικών, που έχουν καθοριστεί από τον Διαχειριστή του δικτύου, με τρόπο ώστε να επιλύονται συγκρούσεις μεταξύ διαφορετικών σκοπών.

Η Συνολική Εικόνα του Δικτύου (Global Network View),περιλαμβάνει την εφαρμογή η οποία δίνει την εποπτεία (εικόνα) για όλο το δίκτυο, τους υποδοχείς, τους μεταγωγείς, τις συνδέσεις και οποιαδήποτε άλλη κατάσταση που σχετίζεται με το δίκτυο, όπως είναι η χρήση του. Η εικόνα του δικτύου πραγματοποιείται προγραμματίζοντας μια εφαρμογή μέσω κατάλληλων Διεπαφών Προγραμματισμού Εφαρμογών (APIs). Μια τέτοιας μορφής διεπαφή προγραμματισμού, επιτρέπει σε μια εφαρμογή να “βλέπει” την εικόνα του δικτύου υπό μορφή γραφήματος (network graph). Αναφέρονται στη συνέχεια ορισμένα παραδείγματα, σχετικά με το τι μπορεί να κάνει το γράφημα δικτύου:

- Δημιουργεί μια απλή εφαρμογή για να υπολογίζει τις μικρότερες διαδρομές μεταφοράς δεδομένων κλπ, αφού η εφαρμογή έχει ήδη τη γραφική εικόνα του δικτύου.

- Μεγιστοποιεί τη χρήση του δικτύου, μέσω της παρακολούθησης της εικόνας του και προγραμματισμού αλλαγών στις διαδρομές για τη ρύθμιση του διακινούμενου φορτίου (traffic engineering).

- Απομάκρυνση της κίνησης από ένα μέρος του δικτύου το οποίο αναβαθμίζεται ή έχει τεθεί σε καραντίνα για κάποιο λόγο.

Τεχνικά οι προς βορρά αφαιρέσεις (NB abstractions) και διεπαφές (APIs) απομονώνουν τις εφαρμογές από λεπτομέρειες του δικτύου, τις οποίες δεν χρειάζονται. Οι αφαιρέσεις είναι επίσης σε θέση να απομονώσουν εφαρμογές από συμβάντα (events) στο δίκτυο, όπως π.χ κατάπτωση μια σύνδεσης (link down), όταν αυτό είναι επιθυμητό από την εφαρμογή.

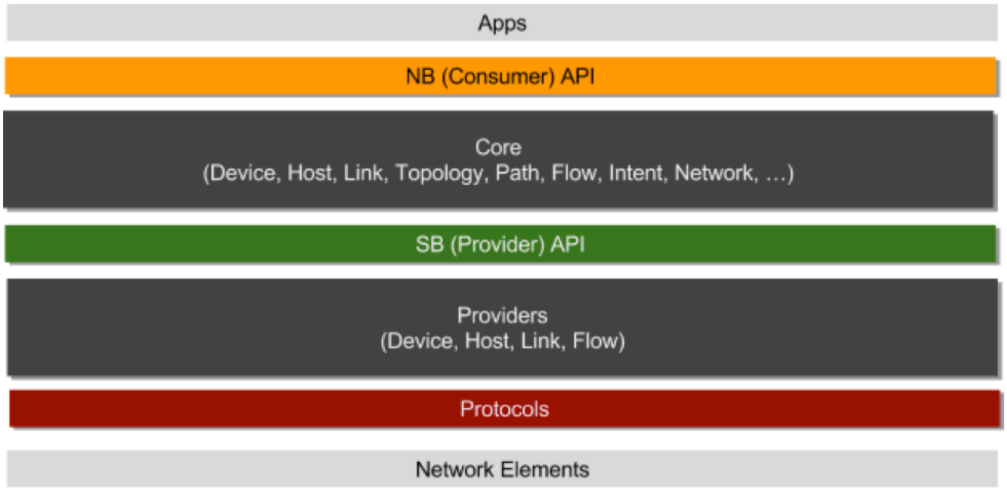
Αντιστρόφως, απομονώνει το λειτουργικό σύστημα (OS) από τις εφαρμογές, επιτρέποντας σε αυτό να εκτελέσει το έργο του, αναφορικά με τη διαχείριση αιτημάτων από πολλαπλές και ανταγωνιστικές μεταξύ τους εφαρμογές.

Από πλευράς επιχειρησιακής λειτουργίας του δικτύου οι προς βορρά αφαιρέσεις του ONOS, αυξάνουν την ταχύτητα ανάπτυξης εφαρμογών και επιτρέπει στο δίκτυο αλλαγές χωρίς καθυστερήσεις στις εφαρμογές.

6.3.3 Southbound Abstractions

Οι προς νότο αφαιρέσεις (SB abstractions) έχουν δομηθεί χρησιμοποιώντας στοιχεία του δικτύου, όπως είναι οι μεταγωγείς, οι υποδοχείς και οι συνδέσεις. Οι προς νότο αφαιρέσεις στο ONOS αντιστοιχίζουν κάθε ένα στοιχείο του δικτύου ως ένα "αντικείμενο" (object), στη γενική του μορφή. Μέσω της αφαίρεσης, ο κατανεμημένος πυρήνας συντηρεί την κατάσταση ενός δικτυακού στοιχείου χωρίς να απαιτείται η γνώση των εξειδικεύσεων του στον υποκείμενο διακομιστή. Στην ουσία επιτρέπει στον πυρήνα να συμπεριφέρεται ως προς νότο πρωτόκολλο ανεξάρτητα από τον κατασκευαστή της συσκευής. Η δυνατότητα αφαίρεσης ενός στοιχείου του δικτύου είναι αυτό που επιτρέπει την προσθήκη νέων συσκευών και πρωτοκόλλων στο δίκτυο. Το ONOS και οι προς νότο αφαιρέσεις επιτρέπουν προσθήκες (plug-ins) διαφόρων πρωτοκόλλων και συσκευών, όπου μια προσθήκη απεικονίζει ή μετατρέπει τη γενική περιγραφή ενός κοινού στοιχείου του δικτύου ή την λειτουργία μιας συσκευής στις ειδικότερες λεπτομέρειες και αντιστρόφως. Ως εκ τούτου οι προς νότο αφαιρέσεις παρέχουν τη δυνατότητα στο ONOS να ελέγχει ή να διαχειρίζεται

πολλαπλές διαφορετικές συσκευές, ακόμα και αν αυτές χρησιμοποιούν διαφορετικά πρωτόκολλα (π.χ OpenFlow, NetConf κλπ).



Σχήμα 11 : Επίπεδα Λειτουργικότητας του ONOS

Αρχιτεκτονικά η νότια διεπαφή (SB API) αποτελείται από τις στιβάδες που παρουσιάζονται στο ανωτέρω *Σχήμα 11*. Στο κάτω μέρος βρίσκονται οι συσκευές και τα στοιχεία του δικτύου. Το ONOS επικοινωνεί (interacts) με τις συσκευές μέσω πρωτοκόλλων. Οι λεπτομέρειες του πρωτοκόλλου αφαιρούνται, από ένα στοιχείο προσθήκη (element plug-in) ή από ένα προσαρμογέα (adaptor). Αυτό το γεγονός έχει ως αποτέλεσμα, ο πυρήνας της νότιας διεπαφής να υποστηρίζει τα δικά του στοιχεία-αντικείμενα του δικτύου (συσκευές, υποδοχείς, συνδέσεις), χωρίς να απαιτείται να γνωρίζει τις ειδικές λεπτομέρειες των πρωτοκόλλων και στοιχείων του δικτύου. Μέσω ενός προσαρμογέα API, ο κατανεμημένος πυρήνας ενημερώνεται συνεχώς για την κατάσταση των στοιχείων-αντικειμένων του δικτύου. Ο προσαρμογέας API φροντίζει και απομονώνει τον κατανεμημένο πυρήνα από τη γνώση λεπτομερειών σχετικά με τα πρωτόκολλα και τα στοιχεία του δικτύου.

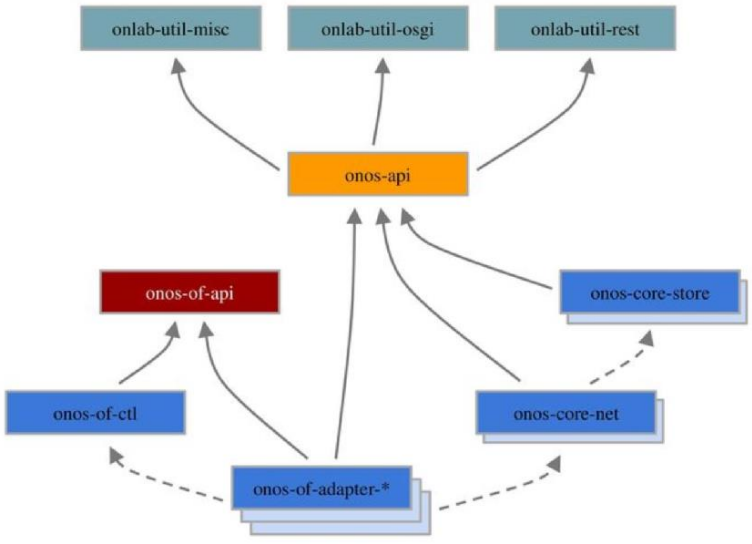
Στα κύρια πλεονεκτήματα των νότιων αφαιρέσεων περιλαμβάνονται:

- Η δυνατότητα διαχείρισης διαφορετικών συσκευών που χρησιμοποιούν διαφορετικά πρωτόκολλα χωρίς αυτό να έχει καμία επίδραση στον κατανεμημένο πυρήνα του συστήματος.
- Η δυνατότητα προσθήκης νέων συσκευών και πρωτοκόλλων στο σύστημα.

- Η εύκολη μετάβαση από τις παλιές συσκευές και πρωτόκολλα σε "λευκά κουτιά" υποστηριζόμενα από OpenFlow.

6.3.4 Αρθρωμένο Λογισμικό (Software Modularity)

Αρθρωμένο (δομοστοιχειωμένο) λογισμό σημαίνει ότι το λογισμικό έχει δομηθεί υπό μορφή στοιχείων τα οποία συνδέονται μεταξύ τους. Όπως φαίνεται και στο *Σχήμα 12* που ακολουθεί, οι κύριες δομές του ONOS είναι οι βαθμίδες του (tiers) οι οποίες συγκεντρώνονται (κεντροποιούνται) γύρω από τον κατανεμημένο πυρήνα του. Όπως προκύπτει από το σχήμα σε μάκρο-επίπεδο οι Βόρειες και Νότιες Διεπαφές (NB APIs & SB APIs), αποτελούν τη βάση για την απομόνωση των Εφαρμογών, του Πυρήνα και των Προσαρμογέων μεταξύ τους. Νέες εφαρμογές ή νέοι προσαρμογείς πρωτοκόλλων μπορούν να προστεθούν εφ' όσον υπάρχει ανάγκη, χωρίς να απαιτείται καθένας από αυτούς να γνωρίζει τον άλλο. Κατ' αναλογία, κάτω από τις μακρο-επιπέδου, απεικονίσεις υπάρχουν μικρότερες υπό-δομές εντός του πυρήνα, οι οποίες υπάρχουν για να υποστηρίζουν την μείωση του μεγέθους οποιουδήποτε υποσυστήματος και να υποστηρίξουν την αρθρωτή επεκτασιμότητα. Αυτά πάλι στηρίζονται σε μεγάλο βαθμό στις διεπαφές οι οποίες λειτουργούν ως "συμβάσεις" για τις επικοινωνίες μεταξύ των διαφόρων μερών του πυρήνα, επιτρέποντας σε μέρος να αναπτύσσεται ανεξάρτητα από τα άλλα.



Σχήμα 12 : Οι Κύριες Δομές του ONOS

Αυτό δίνει τη δυνατότητα εισαγωγής νέων αλγορίθμων, ή πιο αποτελεσματικών δομών δεδομένων, χωρίς να επηρεάζονται μεγάλα τμήματα του δικτύου ή των εφαρμογών του.

Είναι προφανές ότι έχει δοθεί μεγάλη προσοχή στην διασφάλιση του διαχωρισμού ενδιαφερόντων και αρμοδιοτήτων των διεπαφών με στόχο η επικοινωνία μεταξύ των υποσυστημάτων να είναι κατά το δυνατόν απλή και φυσική, αφού αυτό το γεγονός αποτελεί σημαντικό παράγοντα για την εξέλιξη του λογισμικού. Για παράδειγμα, έχει ληφθεί μέριμνα αύξησης του επιπέδου αφαιρέσεων στη SB-API, προκειμένου να αποφευχθούν γενικά μεροληπτικές συμπεριφορές προς συγκεκριμένα πρωτόκολλα και να εφαρμόζονται κανόνες (conventions) δημιουργίας από τον Πυρήνα και όχι από τους Προσαρμογείς μοντέλων αντικειμένων του δικτύου.

- Η πηγαία δομή δένδρου του ONOS έχει οργανωθεί κατά τέτοιο τρόπο, ώστε όχι μόνο να ακολουθεί αλλά και να εφαρμόζει αυτές τις αρχές. Τα δομοστοιχεία/λειτουργικές μονάδες (modules) έχουν διατηρηθεί σε σχετικά μικρό μέγεθος και οι εξαρτήσεις μεταξύ τους σχηματίζουν ένα μη-κυκλικό γράφημα, όπου οι άμεσες εξαρτήσεις μεταξύ των δομοστοιχείων υλοποιούνται μέσω δομοστοιχείων APIs, όπως φαίνεται και στο ανωτέρω διάγραμμα.

Τα πλεονεκτήματα του αρθρωμένου λογισμικού συνοψίζονται στα ακόλουθα:

- Στην αρχιτεκτονική ακεραιότητα και συνοχή.
- Στην απλοποιημένη δοκιμή δομής (test structure) που επιτρέπει πιο εμπεριστατωμένες δοκιμές .
- Στην ευκολότερη συντήρηση με λιγότερες παράπλευρες επιπτώσεις των αλλαγών.
- Επεκτασιμότητα και εξειδίκευση των στοιχείων του δικτύου.
- Αποφυγή κυκλικών εξαρτήσεων.

6.4 Πρόσθετα Χαρακτηριστικά του ONOS

6.4.1 Πολυστιβαδικός SDN έλεγχος του πυρήνα οπτικού-πακέτου

Όπως ήδη έχει αναφερθεί το ONOS είναι προσανατολισμένο στην εξυπηρέτηση των αναγκών και απαιτήσεων Πάροχων Υπηρεσιών οι οποίοι λειτουργούν πολύ-στιβαδικά δίκτυα. Για παράδειγμα ένας πάροχος υπηρεσιών λειτουργεί ένα IP packet network (διαδικτυακό πρωτόκολλο για πακέτο δικτύου) σε ένα δίκτυο μεταφορών[25] ή σε ένα οπτικό δίκτυο[26]. Μια στιβάδα σήραγγας (tunneling layer) που είναι δυνατόν να βρίσκεται επί του IP υποστρώματος για την παροχή υπηρεσιών, όπως μια εικονική IP στιβάδα δικτύων. Σήμερα η κάθε μια από αυτές τις στιβάδες λειτουργεί ανεξάρτητα ως προς τη διαχείρισή της, γεγονός που έχει ως αποτέλεσμα τη χαμηλή χρήση του δικτύου, υψηλές λειτουργικές δαπάνες, και κύκλους επαναδιαμόρφωσης που απαιτούν μεγάλους χρόνους υλοποίησης (μήνες αντί για κάποια λεπτά). Έτσι στη σημερινή πραγματικότητα οι σχεδιαστές δικτύων είναι υποχρεωμένοι να προνοούν για πρόσθετη χωρητικότητα, προκειμένου να διαχειριστούν αστοχίες (failures) και εκρηκτικές καταστάσεις διακίνησης φορτίων με αποτέλεσμα να είναι απαραίτητη 4-5 φορές μεγαλύτερη χωρητικότητα από αυτή που χρειάζονται. Ο SDN έλεγχος των δικτύων πολλαπλών στιβάδων (SDN control of multi layer networks), δίνει την απάντηση στο συγκεκριμένο πρόβλημα. Η λύση περιλαμβάνει τρία δομικά στοιχεία:

(α) Την προσθήκη προγραμματιμότητας τύπου OpenFlow σε ένα οπτικό στοιχείο μεταφοράς, όπως ROADMs[27]

(β) Το ONOS δημιουργεί ένα δικτυακό γράφημα κάθε στιβάδας του δικτύου, διατηρώντας την απεικόνιση ή την επικοινωνία μεταξύ τους.

(γ) Την ανάπτυξη μιας εφαρμογής PCE (Path Computation Element) στο ONOS η οποία συναρμολογεί και αποσυναρμολογεί διαδρομές, λαμβάνοντας υπόψη της πολλαπλούς συσχετισμούς γραφημάτων του δικτύου.

---

[25] Ως δίκτυο μεταφορών (transport network) αναφέρεται η υλοποίηση ενός χωρικού δικτύου, το οποίο περιγράφει μία δομή που επιτρέπει είτε την κίνηση ροών οχημάτων, είτε κάποιου προϊόντος (commodity). Παραδείγματα είναι τα οδικά δίκτυα, τα σιδηροδρομικά δίκτυα, τα δίκτυα ενέργειας, οι δρομολογήσεις αεροπλάνων, δίκτυα ύδρευσης κλπ.

[26] Οπτικό Δίκτυο (optical network) είναι αυτό το οποίο χρησιμοποιεί ένα πομπό που μετατρέπει τα ηλεκτρικά σήματα που λαμβάνει από ένα δίκτυο σε ένα τύπο επικοινωνίας δεδομένων, το οποίο στη συνέχεια μέσω (καλωδίων) οπτικών ινών μεταφέρεται σε μια συσκευή υποδοχής.

[27] ROADM (Reconfigurated Optical add-drop multiplexer) πρόκειται για μια συσκευή η οποία μπορεί να προσθέσει, να εμποδίσει, να επιτρέψει την είσοδο, ή να επανακατευθύνει διαμορφωμένες υπέρυθρες (IR) και ορατές ακτίνες φωτός διαφόρων μηκών κύματος σε ένα δίκτυο οπτικών ινών. Τα ROADMs χρησιμοποιούνται σε συστήματα που εφαρμόζουν διαιρεμένα πολυπλεξικά μήκη κύματος (multiplexing division wavelength).

Η PCE εφαρμογή στο ONOS έχει την ικανότητα να διαμορφώνει, να παρακολουθεί και να ενορχηστρώνει τις στιβάδες, ως να επρόκειτο για μια μοναδική οντότητα. Για παράδειγμα, αλλαγές συνδέσεων IP μπορούν να δώσουν το έναυσμα για αυτόματες προβλέψεις οπτικών διαδρομών. Ο χειριστής έχει πλέον τη δυνατότητα να μειώσει το κόστος λειτουργίας του δικτύου μέσω του κεντρικού ελέγχου, να αυξήσει τη χρήση του δικτύου σε όλες τις στιβάδες και να επαναδιαμορφώσει το δίκτυο κάποια λεπτά.

Μια από τις πρώτες εφαρμογές αυτής της δυνατότητας ήταν η εφαρμογή χρονο-προγραμματισμού εύρους ζώνης του δικτύου (bandwidth calendaring application), η οποία επιτρέπει το εύρος ζώνης να δεσμεύεται στο μέλλον. Το ONOS προβλέπει την παροχή εύρους ζώνης από τις στιβάδες πακέτου και οπτικές στιβάδες και κατόπιν παρακολουθεί τους πόρους να πραγματοποιούν επαναδρομολόγηση και ρυθμίσεις σε πραγματικό χρόνο στο δίκτυο ή και άλλες αλλαγές.

Ο SDN έλεγχος πολύ-στιβαδικών δικτύων, είναι ένα χαρακτηριστικό παράδειγμα για το πώς εφαρμόζεται ο συγκεκριμένος έλεγχος μέσω του ONOS, γεγονός που μπορεί να οδηγήσει σε σημαντικό περιορισμό του κόστους επένδυσης (CapEx) και λειτουργίας (OpEx) του δικτύου, καθώς και δυνατότητες νέων υπηρεσιών.

6.4.2 SDN – IP Ομοτιμία & Επέκταση του SDN Επιπέδου Ελέγχου

Σήμερα τα Αυτόνομα Συστήματα (Autonomous Systems – OS), συνδέονται στο διαδίκτυο αλλά και μεταξύ τους χρησιμοποιώντας ένα Πρωτόκολλο Διασυνοριακής Πύλης Σύνδεσης (BGP - Border Gateway Protocol)[28] για να μοιράζονται πληροφορίες δρομολόγησης. Η SDN – IP peering application[29] (εφαρμογή ομοτιμίας πρωτοκόλλου διαδικτύου) είναι σχεδιασμένη για να βοηθήσει τους παρόχους υπηρεσιών να αυξήσουν τις δράσεις τους. Με τη συγκεκριμένη εφαρμογή ένα SDN δίκτυο εμφανίζεται ως άλλο ένα αυτόνομο σύστημα στο υπόλοιπο διαδίκτυο. Με την πάροδο του χρόνου ένας πάροχος υπηρεσιών μπορεί συνεχώς να επεκτείνει το δίκτυό του και να απολαμβάνει όλα τα πλεονεκτήματα αυτής του της ενέργειας, χωρίς να επηρεάζεται η ομοτιμία του με το Internet. Ακόμα περισσότερο ένας Πάροχος μπορεί

[28] Πρόκειται για ένα πρωτόκολλο δρομολόγησης, το οποίο χρησιμοποιείται για να δρομολογεί κίνηση (traffic) στο διαδίκτυο.

[29] Peering (ομοτιμία):Είναι μια εθελοντική διασύνδεση μεταξύ ανεξάρτητα διαχειρίσιμων internet δικτύων με σκοπό την ανταλλαγή κίνησης μεταξύ των χρηστών καθενός δικτύου. Πρόκειται για συμφωνία δύο ή περισσότερων δικτύων για ομοτιμία ασφάλειας με φυσική διασύνδεση των δικτύων, ή ανταλλαγή πληροφοριών δρομολόγησης , μέσω BGP (Border Gateway Protocol) πρωτοκόλλου δρομολόγησης και σε μερικές ειδικές περιπτώσεις με επίσημα συμβολαιοποιημένα έγγραφα. https://en.wikipedia.org/wiki/Peering

χρησιμοποιώντας την συγκεκριμένη εφαρμογή (SDN-IP peering application) για τη διασυνδεσή του με πολλαπλά δίκτυα για τη δημιουργία ενός μεγαλύτερου Αυτόνομου Συστήματος, το οποίο διατηρεί την ομοτιμία του με το Ίντερνετ, κατά τον ίδιο τρόπο όπως και οποιοδήποτε άλλο Αυτόνομο Σύστημα. Επιπρόσθετα η ίδια εφαρμογή μπορεί να χρησιμοποιηθεί για την επέκταση του SDN Επιπέδου Ελέγχου που βασίζεται στο ONOS.

6.4.3 Λειτουργίες Δικτύου ως μια Υπηρεσία (NFaaS) σε κεντρικά γραφεία

Οι τηλεπικοινωνιακοί πάροχοι διαθέτουν σήμερα στα δίκτυά τους ένα πολύτιμο περιουσιακό στοιχείο, που είναι τα Κεντρικά Γραφεία. Είναι εγκαταστημένα γεωγραφικά πλησίον σε μεγάλους αριθμούς χρηστών και ως εκ τούτου είναι μεγάλης στρατηγική σημασίας, αφού τους επιτρέπουν να προσφέρουν υπηρεσίες μεγάλης κλίμακας σε πολυάριθμους πελάτες με απόδοση και προβλεψιμότητα. Για παράδειγμα η AT & T διαθέτει 4,500 Κεντρικά Γραφεία στις ΗΠΑ, και κάθε ένα από αυτά είναι ένα μεγάλο Μητροπολιτικό Δίκτυο (Metropolitan area network-MAN)[30], που είναι σε θέση να εξυπηρετούν μέχρι 100K ενσύρματους (wired), έως 100K κινητούς (cellular) συνδρομητές και 100 περίπου επιχειρήσεις.

Ένα τυπικό Κεντρικό Γραφείο διαθέτει υψηλές δυνατότητες πρόσβασης σ' αυτό, μεγάλη χωρητικότητα μεταγωγέων και πολλαπλού σκοπού "ενδιάμεσα κουτιά" που εξυπηρετούν διάφορες λειτουργίες του δικτύου. Τα κεντρικά γραφεία είναι στρατηγικής σημασίας για τους χειριστές των δικτύων τηλεπικοινωνιών και έχουν εξελιχθεί σε πολύ μεγάλο βαθμό τις τελευταίες δεκαετίες, με αποτέλεσμα να είναι εξαιρετικά πολύπλοκα σε όρους μεταγωγής/δρομολόγησης, ενδιάμεσων "κουτιών" και τελικών καταναλωτών, όπως και σημαντικής οικονομικής σημασίας για τους παρόχους από πλευράς κόστους επένδυσης και λειτουργίας.

Είναι λοιπόν προφανής η επιδίωξη του ανασχεδιασμού των κέντρων δεδομένων από τους παρόχους υπηρεσιών, προκειμένου να αποκτήσουν περισσότερη ευκινησία και να επιτύχουν μεγαλύτερες οικονομίες κλίμακας. Σχεδιάζοντας αρχιτεκτονικά και

[30] Ένα μητροπολιτικό δίκτυο (metropolitan area network-MAN) είναι ένα δίκτυο υπολογιστών που συνήθως καλύπτει μια πόλη ή μια πανεπιστημιούπολη. Ένα MAN συνήθως συνδέει μεταξύ τους τοπικά δίκτυα υπολογιστών (LANs) χρησιμοποιώντας ένα δίκτυο κορμού (backbone technology) υψηλού εύρους ζώνης, όπως οι οπτικές ίνες και παρέχει διασυνδέσεις προς δίκτυα ευρείας περιοχής ή το Διαδίκτυο. http://el.wikipedia.org/wiki/Μητροπολιτικά_δίκτυα.

κατασκευάζοντας τη δομή ενός δικτύου με τη χρήση SDN, σημαίνει τη μετατροπή των εφαρμοζόμενων λειτουργιών χρησιμοποιώντας ενδιάμεσων "κουτιά" σκοπού, την εκτέλεση προγραμμάτων λογισμικού σε διακομιστές x86 (γνωστό και ως Network Function Virtualization-NFV) και την υλοποίηση ενός συστήματος ενορχήστρωσης το οποίο οργανώνει τις λειτουργίες και τις ροές του δικτύου εξυπηρετώντας δυναμικά ένα συνδρομητή ή μια επιχείρηση, πραγματοποιώντας παράλληλα διάφορες πολιτικές τόσο για τον πελάτη όσο και για τον πάροχο.

Ο μετασχηματισμός των κεντρικών γραφείων με τον παραπάνω τρόπο επιτρέπει στους παρόχους υπηρεσιών την εύκολη δημιουργία, εφαρμογή και προσφορά υπηρεσιών προς τους πελάτες σε μεγάλη κλίμακα και ταυτόχρονα την επίτευξη σημαντικών οικονομιών κλίμακας στο επενδεδυμένο κεφάλαιο και το κόστος λειτουργίας του δικτύου.

Στην περίπτωση αυτή το ONOS μπορεί να προσφέρει λύσεις μικρής κλίμακας στον ανασχεδιασμό των κεντρικών γραφείων. Το ONOS, οι εφαρμογές του και το OpenFlow δίνει τη δυνατότητα στους μεταγωγείς να μετατρέψουν την δομή ενός κεντρικού γραφείου,σε ένα SDN, γεγονός που του παρέχει την απαραίτητη ευελιξία για Λειτουργία Δικτύου ως μια Υπηρεσία (Network Function as a Service). Έτσι καταδεικνύεται ότι η εφαρμογή το ONOS στην περίπτωση αυτή, είναι το κατάλληλο Λειτουργικό Σύστημα (OS) για τον ανασχεδιασμό ενός Κεντρικού Γραφείου με όρους κλίμακας και απόδοσης και πως το SDN και η Λειτουργία Δικτύου ως μια Υπηρεσία ταιριάζει στη συνολικότερη αρχιτεκτονική του Κεντρικού Γραφείου.

6.4.4 Τμηματική Δρομολόγηση – Εξέλιξη & Βελτίωση MPLS

Τα δίκτυα IP/MPLS[31] (Internet Protocol/Multiprotocol Label Switching) έχουν πολύπλοκη μορφή και είναι δύσκολα στη διαχείρισή τους. Η περιγραφική κατανομή (label distribution), η τεχνολογία κίνησης (traffic engineering) και τα εικονικά ιδιωτικά δίκτυα (Virtual Private Networks-VPNs) είναι πολύ σύνθετες λειτουργίες

[31] MPLS (Multiprotocol Label Switching): Πρόκειται για μια τεχνική δρομολόγησης στα τηλεπικοινωνιακά δίκτυα, η οποία κατευθύνει δεδομένα από ένα κόμβο στον επόμενο, βασιζόμενη σε μια σύντομη διαδρομή αντί σε μακρές δικτυακές διευθύνσεις. Με τον τρόπο αυτό αποφεύγονται πολύπλοκες αναζητήσεις πληροφοριών στους πίνακες δρομολόγησης και επιταχύνονται οι ροές κίνησης. Οι επισημάνσεις (labels) αναγνωρίζουν εικονικές συνδέσεις (διαδρομές) μεταξύ απομακρυσμένων κόμβων. Τα MPLS είναι σε θέση να ενθυλακώνουν πακέτα διαφόρων δικτυακών πρωτοκόλλων, και ως εκ τούτου η αναφορά σε πολύ-πρωτόκολλα είναι η ονομασία που τους δίνεται. Τα MPLS υποστηρίζουν ένα ευρύ πεδίο τεχνολογιών πρόσβασης στις οποίες περιλαμβάνονται T1/E1, ATM, FRAMERELAY και DLS.

και υπηρεσίες που εξαρτώνται από μια συλλογή κατανεμημένων πρωτοκόλλων, τα οποία βρίσκονται εγκαταστημένα στο Επίπεδο Ελέγχου του δικτύου. Επιπλέον η δυνατότητα περιορισμού των σφαλμάτων (debugging) στα δίκτυα αυτά, είναι εξαιρετικά δυσχερής, δεδομένων των θεμάτων συγχρονισμού και διαχείρισης της κατάστασης μεταξύ των πολλαπλών πρωτοκόλλων στο Επίπεδο Ελέγχου, καθώς και μιας τοπικής αντικατάστασης σημαντικής επισήμανσης (locally-significant-label swapped) στο Επίπεδο Δεδομένων.

Η IETF εισήγαγε την ιδέα της Τμηματικής Δρομολόγησης (Segment Routing-SR) για την MPLS (γνωστή από την IEFT και ως SPRING). Εισήγαγε ακόμα την έννοια των καθολικών σημαντικών επισημάνσεων (global-significant labels) οι οποίες δεν χρειάζεται να ανταλλαγούν (swapped) σε κάθε αναπήδηση (hop). Ακόμα εισήγαγε την βασισμένη στην πηγαία δρομολόγηση επισημάνση (source-routing based labels), η οποία περιορίζει την εξάρτηση από σύνθετα πρωτόκολλα, όπως το LDP (Label Distributed Protocols) RSVP (Resource Reservation Protocol) για κατανομή επισημάνσεων και ρυθμίσεις επισήμανσης μεταγωγέα διαδρομής (Label Switching Path –LSP). Η Τμηματική Δρομολόγηση ως εκ τούτου, απλοποιεί τόσο το Επίπεδο Ελέχγου, όσο και το Επίπεδο Δεδομένων των MPLS δικτύων. Και ενώ η Τμηματική Δρομολόγηση συνεχίζει να εξαρτάται από ένα Ενσωματωμένο Επεξεργαστή Γραφικών (Integrated Graphic Processor-IGP) για την κατανομή της δρομολόγησης και των επισημάνσεων, παρέχεται η δυνατότητα χρήσης ενός εξωτερικού ελεγκτή για τον προγραμματισμό από άκρο σε άκρο (end-to-end) σηράγγων που προέρχεται από τον πηγαίο δρομολογητή.

Όπως προκύπτει από τις εφαρμογές που ήδη περιγράφτηκαν, η Τμηματική Δρομολόγηση (SR), η οποία προωθήθηκε με τη συνεργασία ο ONF και ειδικών, μέσω του SPRING-OPEN project, καταδεικνύεται πως μπορεί αυτή (SR) να υλοποιηθεί χρησιμοποιώντας το ONOS και μια SR εφαρμογή στο Επίπεδο Ελέγχου ενός SDN δικτύου, με δρομολογητές και υλισμικό που απαντάται εύκολα στην αγορά. Η συγκεκριμένη λύση δεν χρησιμοποιεί IGP ενσωματωμένο στο δρομολογητή, αντ' αυτού χρησιμοποιεί μια εφαρμογή δρομολόγησης του ONOS. Η συγκεκριμένη εφαρμογή προγραμματίζει τους Ακραίους Δρομολογητές (Edge Router)[32] και τους

[32] Edge Router (Ακραίος Δρομολογητής) είναι ένας ειδικός δρομολογητής ο οποίος είναι εγκαταστημένος στην άκρη ή στο όριο ενός δικτύου. Αυτός εξασφαλίζει τη συνδεσιμότητα του δικτύου του, με άλλα εξωτερικά δίκτυα, δίκτυα ευρείας περιοχής και το διαδίκτυο. Ο συγκεκριμένος όρος συχνά αναφέρεται ως δρομολογητής πρόσβασης (access router), ή δρομολογητής πυρήνα (core router).

δρομολογητές πυρήνα, για την προώθηση αυτών των τμηματοποιημένων κανόνων δρομολόγησης, για προεπιλεγμένη δρομολόγηση και δρομολόγηση βασισμένη σε πολιτική. Με το ONOS να παρέχει τον έλεγχος και την διαχείριση της εφαρμογής των επισημάνσεων, οι διαχειριστές των δικτύων μπορούν να καθορίσουν τις απαιτήσεις της πολιτικής προς τον ελεγκτή και οι εφαρμογές μαζί με το ONOS να εφαρμόσουν την πολιτική στο IP/MPLS δίκτυο.

Εν τέλει η Τμηματική Δρομολόγηση δείχνει με ποιό τρόπο οι Πάροχοι Υπηρεσιών μπορούν να δομήσουν το MPLS Επίπεδο Δεδομένων χωρίς την πολυπλοκότητα του Επιπέδου Ελέγχου, πετυχαίνοντας ταυτόχρονα απλούστευση του SR MPLS του Επιπέδου δεδομένων και ευελιξία στο Επίπεδο Ελέχγου.

7. Ryu Controller

7.1 Γενικά

Η λέξη Ryu στην Ιαπωνική γλώσσα αναφέρεται σε ένα δράκο και μια σχολή σκέψης και σημαίνει "ροή". Με ένα δράκο στο λογότυπό του, ο Ελεγκτής Ryu, συνήθως αναφέρεται ως ένα βασιζόμενο σε τμήματα λογισμικού (components) ανοιχτού κώδικα, που καθορίζεται από ένα δικτυακό πλαίσιο (SDN). Όπως ήδη έχει αναφερθεί ο SDN Ελεγκτής αποτελεί τον εγκέφαλο του συγκεκριμένου τύπου δικτύωσης, ο οποίος επικοινωνεί πληθώρα πληροφοριών προς τους μεταγωγείς και τους δρομολογητές του υποκειμένου τμήματος του δικτύου, μέσω SB APIs και προς τις εφαρμογές και την επιχειρησιακή λογική στο υπερκείμενο τμήμα του δικτύου, μέσω NB APIs. Ο Ελεγκτής Ryu υποστηρίζεται από την NTT (Nipon Telegraph & Telephon Co) και εφαρμόζεται σε Κέντρα Δεδομένων της συγκεκριμένης εταιρείας.

Ο Ελεγκτής Ryu [91], [92] παρέχει στοιχεία λογισμικού, με σαφώς καθορισμένες διεπαφές προγραμματισμού εφαρμογών, που καθιστούν εύκολη τη δημιουργία νέων δικτυακών εφαρμογών, ελέγχου και διαχείρισης από τους προγραμματιστές του δικτύου. Η συγκεκριμένη προσέγγιση μέσω στοιχείων λογισμικού βοηθά στην εξατομικευμένη χρήση του Ryu για την αντιμετώπιση συγκεκριμένων αναγκών, δίνοντας τη δυνατότητα στους προγραμματιστές εύκολα και γρήγορα να τροποποιούν τα υπάρχοντα στοιχεία λογισμικού (components) ή και να εφαρμόζουν δικά τους, προκειμένου να διασφαλίζουν ότι το υποκείμενο δίκτυο είναι σε θέση να ανταποκριθεί στις αλλαγές της ζήτησης, από τις εφαρμογές.

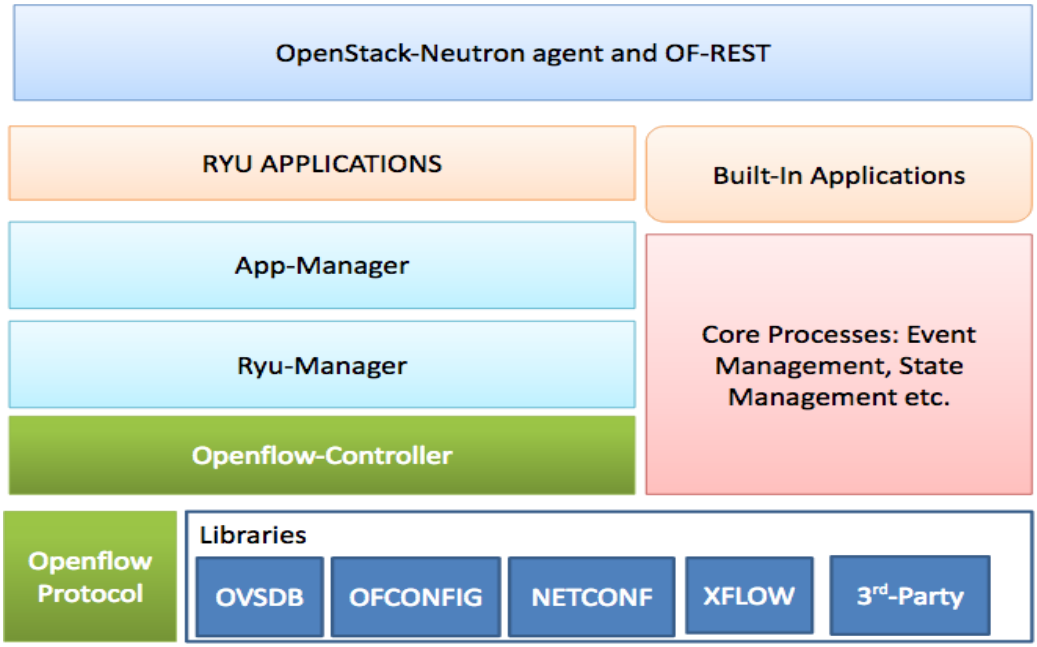
Ο πηγαίος κώδικας ενός ελεγκτή Ryu, φιλοξενείται, διατηρείται και συντηρείται από την ανοιχτή κοινότητα Ryu. Η κοινότητα συνεργάζεται σε ανοιχτή βάση με τα ενδιαφερόμενα μέρη, επικεντρώνοντας το έργο της στην ανάπτυξη ενός λειτουργικού συστήματος νέφους που ελέγχει τον υπολογιστή και τις δικτυακές πηγές ενός οργανισμού, καθώς και την υποστήριξη χρήσεων και εφαρμογών του Ryu, ως SDN ελεγκτή.

Ο ελεγκτής Ryu είναι γραμμένος εξ ολοκλήρου στην υψηλού επιπέδου και γενικής χρήσης γλώσσα προγραμματισμού Python και όλοι οι κώδικες είναι διαθέσιμοι και ανοιχτοί στον οποιονδήποτε από την κατόπιν αδείας έκδοση Apache 2.0. Υποστηρίζει

πολλαπλά πρωτόκολλα για διαχείριση συσκευών όπως NETCONF (Network Configuration Protocol) και OF-config (OpenFlow Management and Configuration Protocol), ενώ για το τελευταίο σημειώνεται ότι είναι από τα πρώτα και πλέον ευρέως χρησιμοποιούμενα πρότυπα επικοινωνίας στα δίκτυα. Υποστηρίζει ακόμα και επεκτάσεις (extensions) Nicira.

7.2 Η Αρχιτεκτονική ενός Ελεγκτή Ryu

Όπως κάθε Ελεγκτής SDN δικτύου, έτσι και ο Ryu μπορεί να δημιουργήσει και να αποστείλει μηνύματα OpenFlow, να ακούσει ασύγχρονα συμβάντα (asynchronous events) όπως είναι η μετακίνηση κάποιας ροής (flow_removed), να αναλύσει και να διαχειριστεί εισερχόμενα πακέτα. Στο *Σχήμα 13* που ακολουθεί παρουσιάζεται η αρχιτεκτονική του πλαισίου ενός τυπικού ελεγκτή Ryu.



Σχήμα 13 : Αρχιτεκτονική Ελεγκτή Ryu

Στοιχεία της ανωτέρω αρχιτεκτονικής είναι :

- ***Οι βιβλιοθήκες Ryu (Ryu Libraries)***: Το πλαίσιο περιέχει μια εντυπωσιακή συλλογή βιβλιοθηκών, που υποστηρίζουν από τα SB πρωτόκολλα μέχρι τη λειτουργία και την επεξεργασία δικτυακών πακέτων. Αναφορικά με τα SB Πρωτόκολλα, το Ryu υποστηρίζει OFconfig, OVSDB (Open vSwitch Database Management Protocol), XFlow (Netflow & Sflow) και άλλα πρωτόκολλα τρίτου

μέρους. Το Netflow υποστηρίζεται από τη Cisco και άλλους και αναφέρεται ειδικά στο IP. Το Netflow και το Sflow υποστηρίζουν τη δειγματοληψία και τη συγκέντρωση πακέτων, τα οποία κυρίως χρησιμοποιούνται στην μέτρηση της κίνησης (traffic measurement). Οι βιβλιοθήκες τρίτου μέρους περιλαμβάνουν την Open vSwitch Python binding, την βιβλιοθήκη διαμόρφωσης Oslo, τη βιβλιοθήκη Python, για τον NETCONF πελάτη. Οι βιβλιοθήκες Ryu βοηθούν στην ανάλυση και την ανάπτυξη διαφόρων πακέτων πρωτοκόλλων, όπως VLAN, MPLS, GRE κλπ.

- ***Πρωτόκολλο OpenFlow & Ελεγκτής OpenFlow***: Ο Ryu υποστηρίζει μέχρι και την πιο πρόσφατη έκδοση (1.4) του OpenFlow και περιλαμβάνει βιβλιοθήκη κωδικοποίησης και αποκωδικοποίησης του συγκεκριμένου πρωτοκόλλου. Ένα από τα κύρια χαρακτηριστικά της συγκεκριμένης αρχιτεκτονικής Ryu είναι ο Ελεγκτής OpenFlow. Είναι υπεύθυνος για τη διαχείριση μεταγωγέων OpenFlow, οι οποίοι χρησιμοποιούνται για τη διαμόρφωση των ροών, τη διαχείριση των συμβάντων κλπ. Ο Ελεγκτής OpenFlow είναι μια από τις εσωτερικές πηγές συμβάντων στην αρχιτεκτονική Ryu. Στον Πίνακα VI που ακολουθεί συνοψίζονται τα μηνύματα του Ryu OpenFlow πρωτοκόλλου.

ΠΙΝΑΚΑΣ VI
Μηνύματα Ryu OpenFlow Πρωτοκόλλου

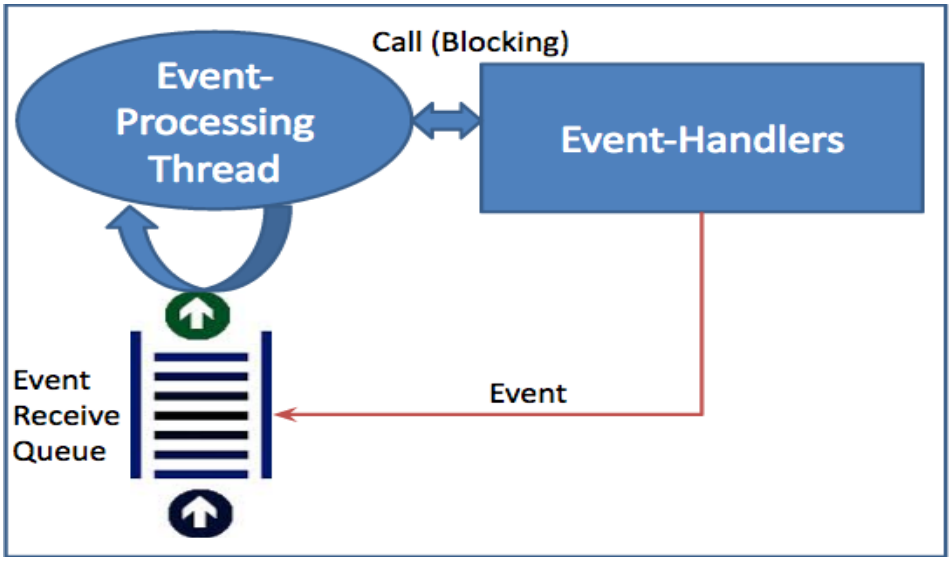
Controller to switch Messages	Asynchronous Messages	Symmetric Messages	Structures
Handshake,switch-config, flow-table-config,modify/read state,queue-config,packet-out,barrier,role-respect	Packet-in,flow removed,port-satus and Error.	Hello,Echo-Request & Reply,Error,experime-nter	Flow Match
Send_msg API and packet builder API's	Set_ev_cls API and packet parser API's	Both Send and Event API's	

- ***Διαχειριστές και Κύριες Διεργασίες (Managers & Core-Processes)***: Ο Διαχειριστής Ryu είναι το κύριο εκτελέσιμο πρόγραμμα. Όταν αυτό εκτελείται, ακούει μια καθορισμένη διεύθυνση IP (π.χ: 0.0.0.0) και μια συγκεκριμένη θύρα, την 6633, όπως είναι προκαθορισμένη. Κατόπιν αυτού κάθε συμβατός με OpenFlow μεταγωγέας (υλισμικό, ή Open vSwitch, ή OVS), μπορεί να συνδεθεί με τον Διαχειριστή Ryu. Η εφαρμογή manager αποτελεί θεμελιακό τμήμα λογισμικού (component) για όλες τις εφαρμογές Ryu και όλες κληρονομούν την κλάση RyuApp, από τη συγκεκριμένη εφαρμογή. Η αρχιτεκτονική της κύριας διεργασίας

(core-process) του θεμελιώδους τμήματος λογισμικού (component) περιλαμβάνει συμβάντα/γεγονότα διαχείρισης, ανταλλαγή μηνυμάτων, in-memory state διαχείριση, κλπ. Είναι ενδιαφέρον το γεγονός ότι η υπηρεσία ανταλλαγής μηνυμάτων του Ryu, υποστηρίζει θεμελιώδη τμήματα λογισμικού που έχουν αναπτυχθεί σε άλλες γλώσσες.

- ***Βόρεια Διεπαφή Ryu (Ryu Northbound)***: Στη στιβάδα Διεπαφής Προγραμματισμού Εφαρμογών (API layer), ο Ryu περιλαμβάνει μια επέκταση λογισμικού (plugin) που ονομάζεται OpenStack Neutron και η οποία υποστηρίζει τόσο την διαμόρφωση/παραμετροποίηση για τη δημιουργία εικονικού τοπικού δικτύου (VLAN), όσο και GRE τοπολογίας. Επί πλέον ο ελεγκτής Ryu, υποστηρίζει τη δημιουργία διεπαφής REST (Representational State Transfer).Επιπρόσθετα, χρησιμοποιώντας WSGI (Web Server Gateway Interface) που είναι ένα πλαίσιο για τη σύνδεση εφαρμογών web και διακομιστές Web σε Python, μπορεί εύκολα να εισάγει νεώτερες REST Διεπαφές Προγραμματισμού Εφαρμογών (APIs) σε μια εφαρμογή.

- ***Εφαρμογές Ryu (Ryu Applications)***: Ο ελεγκτής Ryu είναι ένα κατανεμημένο λογισμικό πολλαπλών εφαρμογών, όπως simple_switch, router, isolation, firewall, GRE tunnel, topology, VLAN, κ.ά. Οι εφαρμογές Ryu είναι μονής σύνδεσης οντότητες/ή οντότητες μονού νήματος (single – threaded entities) που είναι σε θέση να υλοποιήσουν πολλαπλές λειτουργίες και κατά τη λειτουργία τους ανταλλάσουν ασύγχρονα μηνύματα μεταξύ τους. Η αρχιτεκτονική λειτουργία μιας εφαρμογής Ryu παρουσιάζεται στο *Σχήμα 14* που ακολουθεί. Κάθε εφαρμογή διατηρεί συγκεκριμένη σειρά (queue) για τα συμβάντα/γεγονότα, τα οποία ακολουθούν τη λογική FIFO (First In/ First Out), δηλ. το πρώτο που συμβαίνει εξυπηρετείται πρώτο. Κάθε συμβάν/γεγονός, που λαμβάνει χώρα με την προαναφερθείσα λογική, υφίσταται κατάλληλη επεξεργασία και στη συνέχεια ο αρμόδιος διαχειριστής γεγονότων (event handler), δεσμεύει την υπόλοιπη διαδικασία έως ότου αυτή ολοκληρωθεί. Οι εφαρμογές Ryu μπορούν να εκτελεστούν και να διαμορφωθούν μεταφέροντας στον διαχειριστή Ryu τον φάκελο διαμόρφωσης (python script). Μια από τις σημαντικότερες γενικές επιλογές (generic options) στην τελευταία αυτή εντολή είναι οι apps_lists, η οποία περιλαμβάνει ονόματα των δομοστοιχείων εφαρμογών που εκτελούνται.



Σχήμα 14 : Αρχιτεκτονική Λειτουργία Εφαρμογής Ryu

7.3 Δημιουργία Εφαρμογών στον Ryu

Μια εφαρμογή Ryu αποτελεί ένα δομοστοιχείο (module) Python.Πρόκειται για μονής σύνδεσης/μονού νήματος οντότητες που μπορούν να δώσουν ποικίλες δυνατότητες λειτουργίας στον Ryu, που από μόνος του δεν έχει, περιορίζοντας έτσι τη λειτουργία του υφιστάμενου δικτύου στην προώθηση οποιουδήποτε πακέτου στον ελεγκτή, με τη δημιουργία ενός συμβάντος/γεγονότος PacketIn. Ως συμβάν ή γεγονός ορίζεται ένα μήνυμα από την εφαρμογή στον ελεγκτή.

7.3.1 Μοντέλο Προγραμματισμού Ryu

Όπως ήδη αναφέρθηκε οι εφαρμογές Ryu ανταλλάσουν μεταξύ τους ασύγχρονα γεγονότα. Πηγή όμως γεγονότων μπορεί να υπάρξει και εσωτερικά στον ελεγκτή, προερχόμενη από ένα πρωτόκολλο OpenFlow.

Κάθε εφαρμογή για την εξυπηρέτηση των γεγονότων που λαμβάνει, διατηρεί μια σειρά τύπου FIFO, προκειμένου να τηρήσει προτεραιότητα εξυπηρέτησης σε αυτά που προηγούνται χρονικά των υπολοίπων. Επιπρόσθετα, κάθε εφαρμογή δημιουργεί ένα νήμα (thread) επεξεργασίας γεγονότων. Ο ρόλος του νήματος είναι η αποδέσμευση της σειράς σταδιακά από τα γεγονότα με την κλήση του αρμόδιου διαχειριστή (event handler). Κάθε δομοστοιχείο-εφαρμογή αποτελεί υπο-κλάση της κλάσης ryu.base.app_manager.Ryu.App. Αν οριστούν δύο ή περισσότερες τέτοιες υποκλάσεις τότε επιλέγεται αυτή που προηγείται αλφαβητικά.

Η παρακολούθηση συγκεκριμένων συμβάντων/γεγονότων από μια εφαρμογή και η ενεργοποίηση κάποιας μεθόδου σε συγκεκριμένες περιπτώσεις γίνεται με τη διαμεσολάβηση ενός διακοσμητή Python (decorator), ενώ η δημιουργία ενός γεγονότος γίνεται με τη χρήση κατάλληλης μεθόδου κλάσης.

7.3.2 Εφαρμογή Κόμβου Μεταγωγής (Switching Hub)

Ο Κόμβος Μεταγωγής είναι μια βασική εφαρμογή του Ryu, η οποία χρησιμοποιείται από πολλές άλλες. Ο τρόπος δημιουργίας αυτών των εφαρμογών περιλαμβάνει τις ακόλουθες λειτουργίες :

1. Ο Κόμβος Μεταγωγής μαθαίνει μια διεύθυνση MAC ενός υποδοχέα (host), που είναι συνδεδεμένος με μια θύρα του μεταγωγέα και τη διατηρεί σε σχετικό πίνακα.

2. Όταν λαμβάνει ένα πακέτο (δεδομένων) με προορισμό κάποιο υποδοχέα, τη διεύθυνση του οποίου έχει αποθηκευμένη στον προαναφερθέντα πίνακα, προωθεί το πακέτο στον συγκεκριμένο υποδοχέα.

3. Όταν λαμβάνει ένα πακέτο (δεδομένων) με άγνωστη διεύθυνση προορισμού το αντιγράφει και το προωθεί προς όλες τις θύρες (τεχνική πλημμύρας-flooding).

Ένας ελεγκτής Ryu, μπορεί να δώσει οδηγίες στο υποκείμενο δίκτυο, σε ένα μεταγωγέα Openflow να πραγματοποιήσει ενέργειες σχετικά με την κίνηση ενός πακέτου. Έτσι μπορεί να γράψει ξανά τη διεύθυνση του εισερχομένου πακέτου ή να το προωθήσει, μπορεί να μεταφέρει το εισερχόμενο πακέτο στον ελεγκτή (Packet In, ή μπορεί να μεταφέρει το προωθημένο από τον ελεγκτή πακέτο (Packet Out). Όλες αυτές οι ενέργειες μπορούν να συνδυαστούν σε μια εφαρμογή.

Η συνάρτηση packet-in χρησιμοποιείται για να γίνει γνωστή η διεύθυνση MAC, όπως επίσης χρησιμοποιείται από τον ελεγκτή για να δεχθεί πακέτα από τον μεταγωγέα. Ο μεταγωγέας αναλύει τα εισερχόμενα πακέτα για να μάθει τη διεύθυνση MAC του υποδοχέα, όπως και άλλες πληροφορίες για τη συνδεδεμένη θύρα και στη συνέχεια μεταφέρει τα ληφθέντα πακέτα. Για την μεταφορά των πακέτων ο μεταγωγέας διερευνά αν η διεύθυνση προορισμού MAC ανήκει σε μια από αυτές που γνωρίζει και αναλόγως ακολουθεί μια από τις ακόλουθες διαδικασίες:

(α) Αν η διεύθυνση MAC είναι γνωστή χρησιμοποιεί τη συνάρτηση packet-out για την προώθηση του πακέτου σε μια διασυνδεδεμένη θύρα.

(β) Αν η διεύθυνση MAC είναι άγνωστη μέσω της συνάρτησης packet-out προωθεί το πακέτο προς όλες τις θύρες μέσω πλημμύρας (flooding).

7.3.3 Εφαρμογή Spanning Tree

Το Πρωτόκολλο Απλωμένου Δένδρου (Spanning Tree Protocol-STP) διασφαλίζει την εξέλειψη βρόγχων σε ένα τοπικό δίκτυο μεταγωγής, όταν πολλοί μεταγωγείς συνδέονται μέσω πολλαπλών διαδρομών, με στόχο να επιτρέπει την καλύτερη διαδρομή μεταξύ δύο μεταγωγέων. Οι επιπλέον διαδρομές δεσμεύονται και διατηρούνται ως εφεδρικές για την περίπτωση που μια ενεργή διαδρομή παύει να λειτουργεί ή αντιμετωπίζει πρόβλημα. Σε αυτή την περίπτωση αντικαθίσταται από μιά εφεδρική. Το συγκεκριμένο πρωτόκολλο έχει αρκετούς τύπους, όπως το STP, το οποίο είναι και το πιο απλό, το MSTP (Multiple STP) και το Rapid STP, RVST+ (Rapid per VLAN Spanning Tree) που είναι κατάλληλα για VLAN περιβάλλοντα. Προκειμένου το STP να πραγματοποιήσει την αποστολή του, αντιλαμβάνεται το δίκτυο ως ένα λογικό δένδρο και κατά συνέπεια αποκόβει λογικά μια ζεύξη και όχι φυσικά. Σύμφωνα με την τεκμηρίωση του συγκεκριμένου πρωτοκόλλου, οι δικτυακές συσκευές στις οποίες εφαρμόζεται αναφέρονται ως γέφυρες (bridges) και όχι ως μεταγωγείς. Σημειώνεται ότι οι γέφυρες είναι δικτυακές συσκευές προώθησης, που συνδέουν δύο περιοχές του δικτύου που χρησιμοποιούν τα ίδια πρωτόκολλα. Η λειτουργία τους συνοψίζεται στην απόφαση που λαμβάνουν αναφορικά με την κατεύθυνση στην οποία θα προωθηθεί το μήνυμα.

Η λειτουργία του STP βασίζεται στην κυκλοφορία μηνυμάτων BPDU (Bridge Protocol Data Unit), τα οποία ανταλλάσσονται από τις γέφυρες, προκειμένου να γίνει σύγκριση των πληροφοριών που τις γέφυρες και τις θύρες τους και να αποφασιστεί κατά πόσο η εκάστοτε ζεύξη μπορεί να συμμετέχει ενεργά στο δίκτυο.

8. Σύγκριση Ελεγκτών ONOS,OpenDaylight,Ryu

8.1 Κριτήρια

Το τοπίο στα Δίκτυα Καθοριζόμενα από Λογισμικό (SDN) εξελίσσεται με ταχείς ρυθμούς τα τελευταία 2-3 χρόνια. Εξαιτίας της αναπτυξιακής φύσης και του μεγάλου οικονομικού και τεχνολογικού ενδιαφέροντος που παρουσιάζει ο τομέας των Ελεγκτών SDN, υπάρχει πληθώρα λογισμικών που είναι διαθέσιμα προς χρήση. Όπως έχει αναφερθεί επανειλημμένα στα προηγούμενα κεφάλαια, ο πυρήνας της φιλοσοφίας που διέπει τα SDN δίκτυα είναι ο διαχωρισμός της ευφυΐας και του ελέγχου (π.χ δρομολόγηση), από τα στοιχεία προώθησης (π.χ μεταγωγείς) και η επικέντρωση στον έλεγχο του δικτύου, στη διαχείριση και στη λειτουργία του μέσα από ένα λογικά κεντρικό κώδικα λογισμικού (logically centralized component), δηλαδή έναν Ελεγκτή SDN.

Σήμερα υπάρχουν διαθέσιμοι στην αγορά και χρησιμοποιούνται αρκετοί σύγχρονοι, αλλά και παλαιότεροι SDN Ελεγκτές, που εξυπηρετούν διαφορετικές ανάγκες στα SDN δίκτυα. Έχουν ως εκ τούτου διεξαχθεί πολλές μελέτες σχετικά μα τα ποιοτικά τους χαρακτηριστικά από την ακαδημαϊκή κοινότητα και την αγορά, με σκοπό να αξιολογήσουν τις δυνατότητες και τα πλεονεκτήματα-μειονεκτήματα που παρουσιάζουν ανάλογα με την περίπτωση χρήσης τους, αλλά και τις απαιτήσεις σε ένα SDN δίκτυο. Οι συγκριτικές αυτές μελέτες στην μεγάλη πλειοψηφία τους είναι κυρίως ποιοτικού χαρακτήρα, συγκρίνοντας τα χαρακτηριστικά στοιχεία και λειτουργίες που παρέχει κάθε ελεγκτής.
Στην [93] που δημοσιεύτηκε στις αρχές του 2019 γίνεται ποιοτική αξιολόγηση 34 γνωστών στην αγορά ελεγκτών (βλ.Πίνακας VII), με βάση 12 κριτήρια όπως η αρχιτεκτονική, η αποτελεσματικότητά (η επίδοση, η αξιοπιστία, η κλιμάκωση και η ασφάλεια), η υποστήριξη βόρειων και νότιων διεπαφών εφαρμογών, τις γλώσσες προγραμματισμού που χρησιμοποιούν οι ελεγκτές, η τεκμηρίωση κλπ.
Ως γενικό σχόλιο στον Πίνακα VII με βάση επί μέρους τα χαρακτηριστικά του κάθε ελεγκτή, σημειώνεται ότι οι ελεγκτές ODL, ONOS και Ryu, από πλευράς δυνατοτήτων καλύπτουν επαρκώς και τα δώδεκα (12) κριτήρια που αξιολογούνται, γεγονός που τους καθιστά ικανούς να υποστηρίξουν αποτελεσματικότερα SDN λύσεις, σε σχέση με τους υπόλοιπους ελεγκτές. Τους τρεις παραπάνω ελεγκτές

ακολουθούν οι Onix, Floodlight, PANE, OpenContrail, Faucet και έπονται οι υπόλοιποι, οι οποίοι παρουσιάζουν περιορισμούς και λιγότερες δυνατότητες.
Διαπιστώνεται επίσης ότι οι δημοφιλέστεροι ελεγκτές, όπως ο ONOS, και OpenDayLight, διαθέτουν ως κύριο χαρακτηριστικό την οριζόντια κατανεμημένη αρχιτεκτονική έναντι της κεντροποιημένης στους υπόλοιπους ελεγκτές, γεγονός παρέχει περισσότερες δυνατότητες στη διαχείριση των SDN δικτύων, όπως είναι η βελτιωμένη κλιμάκωση (scalability), η αξιοπιστία (reliability), ο περιορισμός των στενώσεων (bottlenecks) στην απόδοση και η μικρότερη καθυστέρηση (latency). Αυτός είναι ίσως και ο κυριότερος λόγος που οι δύο προαναφερθέντες ελεγκτές είναι εξαιρετικά δημοφιλείς στη χρήση πολύ μεγάλου μεγέθους δικτύων (Κέντρα Δεδομένων, Τηλεπικοινωνίες , Μεταφορές & WANs). Ο ελεγκτής Ryu αποτελεί μια διαφορετική περίπτωση στην αρχιτεκτονική του, όπως θα αναλυθεί στη συνέχεια, που όμως φαίνεται να επιτυγχάνει ανάλογα αποτελέσματα με το ONOS και OpenDaylight. Άλλοι ελεγκτές κατανεμημένης αρχτεκτονικής φαίνεται να έχουν λιγότερα λειτουργικά χαρακτηριστικά γεγονός που τους καθιστά χρήσιμους σε μικρότερης έκτασης δίκτυα.
Ακόμα οι πολύ καλές επιδόσεις των τριών ελεγκτών ONOS, ODL και Ryu σε άλλα κρίσιμα κριτήρια όπως οι γλώσσες προγραμματισμού, η υποστήριξη των διεπαφών SBAPI's & NBPAI's, η δομοστοιχείωση (modularity), η σταθερότητα κλπ, σε σχέση με τους υπόλοιπους ελεγκτές, τους φέρνουν στην πρώτη θέση από πλευράς επιλογής από τους διαχειριστές των μεγάλων δικτύων και τις μεγάλες επιχειρήσεις.

Στη συνέχεια παρουσιάζονται τα στοιχεία της πιο πρόσφατης μελέτης [95] του Ιουλίου 2019, που εξετάζει και συγκρίνει τους πέντε πλέον δημοφιλείς Ελεγκτές Ανοιχτού Κώδικα, ONOS, OpenDayLight, Ryu, OpenKilda και Faucet, που χρησιμοποιούνται σήμερα. Για τις ανάγκες της παρούσας εργασίας και λόγω της μεγάλης δημοφιλίας τους στην αγορά, αντλήθηκαν στοιχεία μόνο για τους τρεις πρώτους ελεγκτές, χρησιμοποιώντας ως κριτήρια (τα ίδια χρησιμοποιούνται και στις προαναφερθείσες μελέτες) τα ακόλουθα:

- Αρχιτεκτονική (Architecture)
- Δομοστοιχείωση & Επεκτασιμότητα (Modularity & Extensibility)
- Κλιμακοδιαθεσιμότητα (Scalability)
    - Κλιμακοδιαθεσιμότητα Συστάδας (Cluster Scalability)

- Αρχιτεκτονική Κλιμακοδιαθεσιμότητα (Architectural Scalability)

- Διεπαφές (Interfaces)
  - Υποστήριξη Εφαρμογών Βόρειας Διεπαφής (Northbound APIs)
  - Υποστήριξη Εφαρμογών Νότιας Διεπαφής (Southbound APIs)
- Τηλεμετρικά Δεδομένα (Telemetry)
- Ανθεκτικότητα και Ανοχή σε Σφάλμα (Resilience & Fault Tolerance)
- Γλώσσα Προγραμματισμού (Programming Language)
- Κοινότητα (Community)

Τα ανωτέρω κριτήρια παρουσιάζονται αναλυτικά στα επόμενα υποκεφάλαια για τον καθένα ελεγκτή και στο τέλος δίνεται σε πίνακα η σταθμισμένη βαθμολογία τους.

ΠΙΝΑΚΑΣ VII

Σύγκριση Χαρακτηριστικών των SDN Ελεγκτών

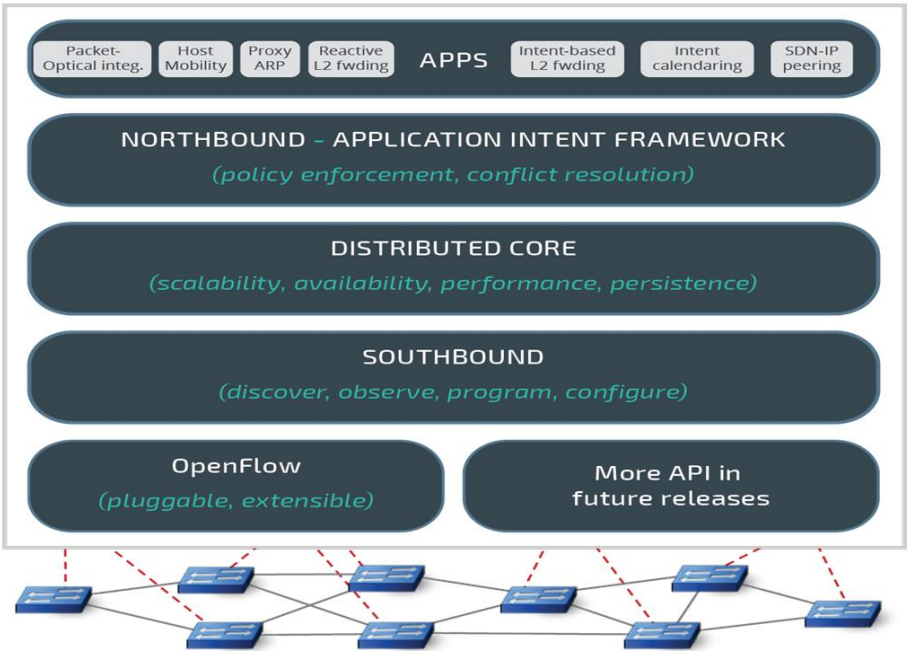
Name	Programming Language	Architecture	Northbound API	Southbound API	EastWestbound API	Supported Platform	Interface	License	Multithreading	Modularity	Consistency	Documentation
Beacon [42]	Java	Centralized	ad-hoc	OpenFlow 1.0	-	Linux, MacOS, Windows	CLI, Web UI	GPL 2.0	Yes	Fair	No	Fair
Beehive [43]	Go	Distributed Hierarchical	REST	OpenFlow 1.0, 1.2	-	Linux	CLI	Apache 2.0	Yes	Good	Yes	Limited
DCFabric [44]	C, Javascript	Centralized	REST	OpenFlow 1.3	-	Linux	CLI, Web UI	LGPL 3.0	Yes	Good	Yes	Fair
Disco [45]	Java	Distributed Flat	REST	OpenFlow 1.0	AMQP	-	-	Proprietary	-	Good	No	Limited
Faucet [46]	Python	Centralized	-	OpenFlow 1.3	-	Linux	CLI, Web UI	Apache 2.0	Yes	-	Yes	Good
Floodlight [14]	Java	Centralized	REST, Java RPC, Quantum	OpenFlow 1.0, 1.3	-	Linux, MacOS, Windows	CLI, Web UI	Apache 2.0	Yes	Fair	Yes	Good
FlowVisor [47]	C	Centralized	JSON RPC	OpenFlow 1.0, 1.3	-	Linux	CLI	Proprietary	-	-	No	Fair
HyperFlow [48]	C++	Distributed Flat	-	OpenFlow 1.0	Publish and subscribe messages	-	-	Proprietary	Yes	Fair	No	Limited
Kandoo [49]	C, C++, Python	Distributed Hierarchical	Java RPC	OpenFlow 1.0-1.2	Messaging Channel	Linux	CLI	Proprietary	Yes	High	No	Limited
Loom [50]	Erlang	Distributed Flat	JSON	OpenFlow 1.3-1.4	-	Linux	CLI	Apache 2.0	Yes	Good	No	Good
Maestro [51]	Java	Centralized	ad-hoc	OpenFlow 1.0	-	Linux, MacOS, Windows	Web UI	LGPL 2.1	Yes	Fair	No	Limited
McNettle [52]	Haskell	Centralized	-	OpenFlow 1.0	-	Linux	CLI	Proprietary	Yes	Good	No	Limited
Meridian [53]	Java	Centralized	REST	OpenFlow 1.0, 1.3	-	Cloud-based	Web UI	-	Yes	Good	No	Limited
Microflow [54]	C	Centralized	Socket	OpenFlow 1.0-1.5	-	Linux	CLI, Web UI	Apache 2.0	Yes	-	No	Limited
NodeFlow [55]	JavaScript	Centralized	JSON	OpenFlow 1.0	-	Node.js	CLI	Cisco	-	-	No	Limited
NOX [12]	C++	Centralized	ad-hoc	OpenFlow 1.0	-	Linux	CLI, Web UI	GPL 3.0	Yes (Nox-MT)	Low	No	Limited
Onix [56]	C++	Distributed Flat	Onix API	OpenFlow 1.0, OVSDB	Zookeeper	-	-	Proprietary	Yes	Good	No	Limited
ONOS [16]	Java	Distributed Flat	REST, Neutron	OpenFlow 1.0, 1.3	Raft	Linux, MacOS, Windows	CLI, Web UI	Apache 2.0	Yes	High	Yes	Good
OpenContrail [57]	C, C++, Python	Centralized	REST	BGP, XMPP	-	Linux	CLI, Web UI	Apache 2.0	Yes	High	Yes	Good
OpenDaylight [15]	Java	Distributed Flat	REST, RESTCONF, XMPP, NETCONF	OpenFlow 1.0, 1.3	Akka, Raft	Linux, MacOS, Windows	CLI, Web UI	EPL 1.0	Yes	High	Yes	Good
OpenIRIS [58]	Java	Distributed Flat	REST	OpenFlow 1.0-1.3	Custom Protocol	Linux	CLI, Web UI	Apache 2.0	Yes	Fair	No	Limited
OpenMul [59]	C	Centralized	REST	OpenFlow 1.0, 1.3, OVSDB, Netconf	-	Linux	CLI	GPL 2.0	Yes	High	No	Good
PANE [60]	Haskell	Distributed Flat	PANE API	OpenFlow 1.0	Zookeeper	Linux, MacOS	CLI	BSD 3.0	-	Fair	No	Fair
POF Controller [61]	Java	Centralized	-	OpenFlow 1.0, POF-FIS	-	Linux	CLI, GUI	Apache 2.0	-	-	No	Limited
POX [13]	Python	Centralized	ad-hoc	OpenFlow 1.0	-	Linux, MacOS, Windows	CLI, GUI	Apache 2.0	No	Low	No	Limited
Ravel [62]	Python	Centralized	ad-hoc	OpenFlow 1.0	-	Linux	CLI	Apache2.0	-	-	Yes	Fair
Rosemary [63]	C	Centralized	ad-hoc	OpenFlow 1.0, 1.3, XMPP	-	Linux	CLI	Proprietary	Yes	Good	No	Limited
RunOS [64]	C++	Distributed Flat	REST	OpenFlow 1.3	Maple	Linux	CLI, Web UI	Apache2.0	Yes	High	Yes	Fair
Ryu [17]	Python	Centralized	REST	OpenFlow 1.0-1.5	-	Linux, MacOS	CLI	Apache 2.0	Yes	Fair	Yes	Good
SMaRtLight [65]	Java	Distributed Flat	REST	OpenFlow 1.3	BFT-SMaRt	Linux	CLI	Proprietary	-	-	No	Limited
TinySDN [66]	C	Centralized	-	OpenFlow 1.0	-	Linux	CLI	BSD 3.0	No	-	No	Limited
Trema [67]	C, Ruby	Centralized	ad-hoc	OpenFlow 1.0	-	Linux	CLI	GPL 2.0	-	Good	No	Fair
Yanc [68]	C, C++	Distributed Flat	REST	OpenFlow 1.0-1.3	yanc File System	Linux	CLI	Proprietary	-	-	No	Limited
ZeroSDN [69]	C++	Distributed Flat	REST	OpenFlow 1.0, 1.3	ZeroMQ	Linux	CLI, Web UI	Apache 2.0	-	High	Yes	Fair

8.2 Συγκριτικά Στοιχεία Ελεγκτή ONOS

Ο ONOS, έχει σχεδιαστεί ως ένας μόνιμα κατανεμημένος και κλιμακοδιαθέσιμος Ελεγκτής (distributed & scalable), η χρήση του οποίου είναι προσανατολισμένη στα δίκτυα Παρόχων Υπηρεσιών. Πρόκειται για την μοναδική πλατφόρμα SDN Ελεγκτή που υποστηρίζει την μετάβαση από παλαιού τύπου δίκτυα (legacy/brown field networks) σε νέου τύπου "πράσινα" δίκτυα (green field networks), παρέχοντας νέες συναρπαστικές δυνατότητες και ανατρεπτικές (disruptive) εφαρμογές, καθώς και βελτιώσεις στο κόστος λειτουργίας στους διαχειριστές των δικτύων.

Αρχιτεκτονική (Architecture)

Η αρχιτεκτονική του ONOS είναι σχεδιασμένη σε τρείς βαθμίδες (tier) όπως φαίνεται στο κάτωθι *Σχήμα 15*:



Σχήμα 15 : Βαθμίδες Αρχιτεκτονικής ONOS

- Η βαθμίδα 1 (tier 1) συνίσταται από δομοστοιχεία (modules) πού σχετίζονται με τα πρωτόκολλα τα οποία επικοινωνούν με τις συσκευές του δικτύου (Διεπαφή Southbound στο Σχήμα).

- Η βαθμίδα 2 (tier 2) αποτελείται από τον πυρήνα του ONOS και παρέχει την κατάσταση του δικτύου χωρίς να βασίζεται σε οποιοδήποτε πρωτόκολλο.
- Η βαθμίδα 3 (tier 3), συνίσταται από τις εφαρμογές (applications), ONOSapps, οι οποίες χρησιμοποιούν πληροφορίες για την κατάσταση του δικτύου, τις οποίες λαμβάνουν από τη βαθμίδα 2.

Δομοστοιχείωση & Επεκτασιμότητα (Modularity & Extensibility)

Το ONOS έχει ενσωματωμένους μηχανισμούς για τη σύνδεση/αποσύνδεση τμημάτων λογισμικού (components), ενώ ο Ελεγκτής βρίσκεται εν λειτουργία. Το γεγονός αυτό παρέχει μια ιδιαίτερα ευέλικτη προσέγγιση στο ζήτημα της προσθήκης νέων λειτουργιών στον Ελεγκτή.

Κλιμακοδιαθεσιμότητα (Scalability)

Το ONOS έχει σχεδιστεί ώστε να παρέχει δυνατότητες οριζόντια κλιμάκωσης για καλύτερη απόδοση και γεω-πλεονασμό (geo-redundancy) μεταξύ μικρών περιοχών του δικτύου.

- *Κλιμακοδιαθεσιμότητα Συστάδων (Cluster Scalability)*

Η παραμετροποίηση των συστάδων είναι απλοποιημένη, με τους νέους Ελεγκτές να είναι σε θέση να συμμετέχουν ή να αποχωρούν με δυναμικό τρόπο από το δίκτυο, δίνοντας με το τρόπο αυτό ιδιαίτερη ευελιξία στη λειτουργία.
Η κατανεμημένη βάση αποθηκευσης δεδομένων Atomix, η οποία προτεραιοποιεί την πυκνότητα των δεδομένων, έχει τη δυνατότητα να μειώνει τις διακοπές που οφείλονται στην προτεραιοποίηση της συστάδας, ώστε όλοι οι υποδοχείς να έχουν εγγυημένα σωστά δεδομένα.
Με δεδομένο ότι μια συστάδα συνεχώς διευρύνεται, η επικοινωνία και ο συντονισμός αυξάνουν με ταχείς ρυθμούς, περιορίζοντας έτσι τα οφέλη από την προσθήκη νέων μελών στη συστάδα.

- *Αρχιτεκτονική Κλιμακοδιαθεσιμότητα (Architectural Scalability)*

Το ONOS περιλαμβάνει εγγενώς δυνατότητες δρομολόγησης BGP (Border Gateway Protocol), προκειμένου να συντονίζει την κίνηση ροών (traffic flows) μεταξύ νησίδων

SDN. Υπάρχουν αρκετοί τεκμηριωμένοι κλώνοι (instances) του ONOS (π.χ ICONA, SDN-IP) που χρησιμοποιούνται με επιτυχία στην αρχιτεκτονική του γεω-πλεονασμού (geo-redundancy), για τον έλεγχο μεγάλης κλίμακας SD-WANs.

Διεπαφές (Interfaces)

Το ONOS υποστηρίζει ένα μεγάλο εύρος Νότιων Διεπαφών (Southbound Interfaces), περιλαμβανομένων και των πλέον ευρέως χρησιμοποιούμενων όπως το OpenFlow, το P4, το NTCONF, το TL1, το SNMP, το RESTCONF και PCEP. Σε ότι αφορά στις Βόρειες Διεπαφές (Northbound Interfaces), το ONOS προσφέρει το μεγαλύτερο πακέτο τέτοιων διεπαφών με gRPC και RESTful APIs. Σχετκά με το GUI (Graphical User Interface), το ONOS παρέχει μια single-page web-εφαμοργή, η οποία παρέχει οπτική διεπαφή του Ελεγκτή ή της συστάδας Ελεγκτών (cluster of controllers).Το ONOS εφαρμόζει ένα Πλαίσιο Βασισμένο στο Σκοπό (Intent base framework) το οποίο είναι εσνωματωμένο σ' αυτό.

Τηλεμετρικά Δεδομένα (Telemetry)

Η τροφοδοσία τηλεμετρικών δεδομένων γίνεται μέσω μιας προσθήκης δομοστοιχείου η οποία συνοδεύεται από λογισμικό, με προσθήκες Influx Db και Grafana να περιλαμβάνονται στην τελευταία έκδοσή του.

Ανθεκτικότητα και Ανοχή σε Σφάλμα (Resilience & Fault Tolerance)

Το ONOS διαθέτει ένα πολύ απλό μηχανισμό διαχείρισης για τις συστάδες με εγγενείς εντολές για προσθήκη και αφαίρεση μελών (Ελεγκτών). Περιλαμβάνει επίσης και Ανοχή σε Σφάλμα στο σύστημα για περιττό αριθμό Ελεγτών. Στην περίπτωση αποτυχίας (failure) του Κύριου Κόμβου (Master Node), ένας νέος επιλέγεται να πάρει τη θέση του σε ότι αφορά στον έλεγχο του δικτύου.

Γλώσσα Προγραμματισμού (Programming Language)

Η Γλώσσα προγραμματισμού του ONOS είνα η Java.

Κοινότητα (Community)

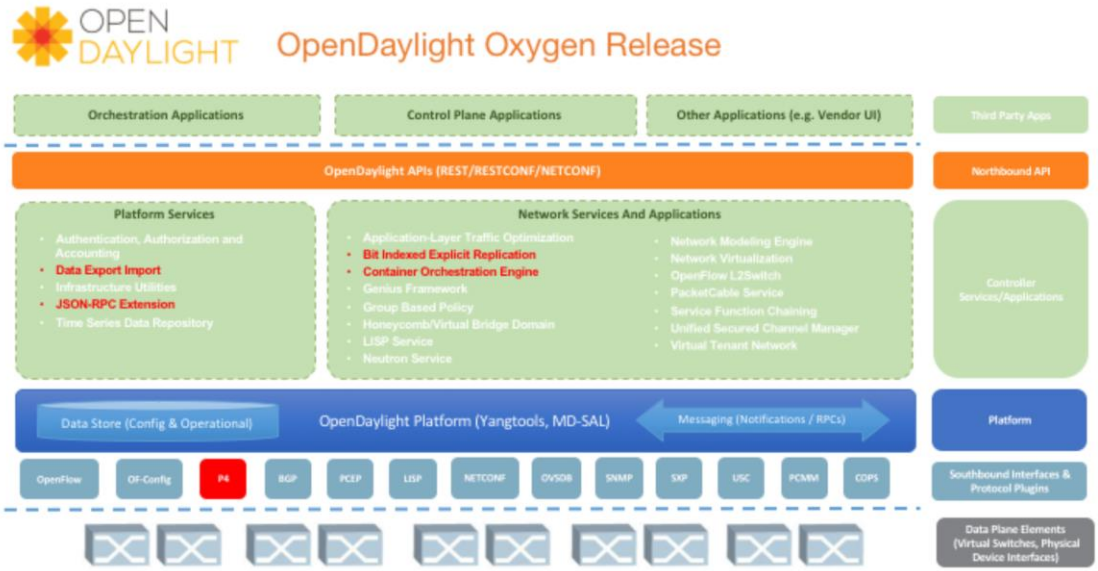
Το ONOS υποστηρίζεται από το Linux Foundation Networking και από μια ιδιαίτερα μεγάλη κοινότητα προγραμματιστών και χρηστών.

8.3 Συγκριτικά Στοιχεία Ελεγκτή OpenDaylight

Το OpenDaylight (ODL) είναι μια δομοστοιχειωτή/αρθρωτή (modular) ανοιχτή πλατφόρμα για την εξατομίκευση και την αυτοματοποίηση δικτύων οποιουδήποτε μεγέθους και κλίμακας.Το OpenDaylight Project προέκυψε από την κίνηση SDN, με σαφή εστίαση στην προγραμματισιμότητα του δικτύου. Σχεδιάστηκε εξ αρχής, έχοντας ως θεμέλιο την παροχή επιχειρηματκών λύσεων που απευθύνονται σε διάφορες περιπτώσεις χρήσης και σε υφιστάμενα περιβάλλοντα δικτύου. Είναι επικεντρωμένο στα SD-LAN και στην ενσωμάτωση χώρων που σχετιζονται με το Νέφος (Cloud Integration).

Αρχιτεκτονική (Architecture)

Το ODL αποτελείται από τρείς στιβάδες όπως φαίνεται και στο *Σχήμα 16* που ακολουθεί:



Σχήμα 16 : Η Έκδοση Oxygen του OpenDaylight

- Υπάρχουν πρόσθετα Νότιας κατεύθυνσης (Southbound plugins) για την επικοινωνία του Ελεγκτή με τις συσκευές του (υποκείμενου) δικτύου.
- Οι κύριες (βασικές) υπηρεσίες του Ελεγκτή χρησιμοποιούν την SAL (Service Abstraction Layer) η οποία βασίζεται στην OSGi (Open Service Gateway initiative) προκειμένου να υποστηρίζει τμήματα λογισμικού (components) να ενσωματωνονται ή να αποκόπτονται από τον Ελεγκτή, όσο αυτός βρίσκεται σε λειτουργία.
- Οι Βόρειες Διεπαφές (π.χ REST/NETCONF) επιτρέπουν στους χειριστές του δικτύου να εφαρμόζουν υψηλού επιπέδου πολιτικές για τις συσκευές ή και για την ενοποίηση με άλλες πλατφόρμες.

Δομοστοιχείωση & Επεκτασιμότητα (Modularity & Extensibility)

Υπάρχουν ενσωματωμένοι μηχανισμοί στο ODL που απλοποιούν τη σύνδεση μεταξύ των κωδικοποιημένων δομοστοιχείων. Ο ελεγκτής αξιοποιώντας τον περιέκτη OSGi για τη φόρτωση δεσμίδων σε τρέχοντα χρόνο, επιτρέπει με τον τρόπο αυτό μια πιό ευέλικτη προσέγγιση στη προσθήκη και νέων λειτουργιών.

Κλιμακοδιαθεσιμότητα (Scalability)

Το ODL χρησιμοποιεί την προσέγγιση που βασίζεται στο μοντέλο (model-based approach) η οποία παρέχει την συνολική ενσωματωμένη στη μνήμη του ελεγκτή εικόνα του δικτύου, η οποία απαιτείται για να πραγματοποιηθούν οι λογικοί υπολογισμοί. Η τελευταία έκδοση του ODL προωθεί περαιτέρω τη δομοστοιχείωση και την ευρωστία της πλατφόρμας, μέσω νέων δυνατοτήτων υποστήριξης πολυκομβικών (multi-site) εφαρμογών για γεωγραφική εμβέλεια, απόδοση και ανοχή σε σφάλματα.

- *Κλιμακοδιαθεσιμότητα Συστάδας (Cluster Scalability)*

Το ODL έχει ενσωματωμένη εσωτερική λειτουργία για τη συντήρηση της Συστάδας των Ελεγκτών, AKKA η οποία είναι μια κατανεμημένη αποθήκη δεδομένων για το διαμοιρασμό της τρέχουσας κατάστασης του δικτύου, επιτρέπωντας σε όλους τους ελεγκτές να ανακατευθύνουν τη λειτουργία τους σε περίπτωση διαίρεσης της συστάδας.

Με δεδομένο ότι μια συστάδα συνεχώς διευρύνεται η επικοινωνία και ο συντονισμός αυξάνουν με ταχείς ρυθμούς, περιορίζοντας έτσι τα οφέλη από την προσθήκη νέων μελών στη συστάδα.

- *Αρχιτεκτονική Κλιμακοδιαθεσιμότητα (Architectural Scalability)*

Το ODL έχει εγγενώς ενσωματωμένες BGP (Border Gateway Protocol) δυνατότητες δρομολόγησης για το συντονισμό των ροών κίνησης μεταξύ των SDN νησίδων.
Η εισαγωγή του ODL στο OpenStack παρέχει πολυκομβική δικτύωση,αυξάνοντας ταυτόχρονα την απόδοση του δικτύου.

Διεπαφές (Interfaces)

Νότια Διεπαφή: Η πλατφόρμα υποστηρίζει μια εκτεταμένη λίστα από Νότιες Διεπαφές περιλαμβανομένων και των OpenFlow, P4, NETCONF, SNMP, BGP, RESTCONF και PCP.
Βόρεια Διεπαφή: Το ODL προσφέρει το μεγαλύτερο πακέτο Βόρειων Διεπαφών με με gRPC και RESTful APIs. Οι Βόρειες Διεπαφές που υποστηρίζονται από το ODL, περιλαμβάνουν OSGi εφαρμογές στην ίδια χωρική διεύθυνση (address space) όπως και του ελεγκτής και της RESTful διεπαφής. DLUX χρησιμοποιείται για να την οπτικοποίηση της Βόρεια Διεπαφής, προκειμένου να διευκολυνθούν οι εργασίες ενσωμάτωσης και ανάπτυξης.

Τηλεμετρικά Δεδομένα (Telemetry)

Το ODL παρουσιάζει περιορισμένες λειτουργικές δυνατότητες σχετικά με την τηλεμετρία της Βόρειας Διεπαφής, αν και έχουν δρομολογηθεί βελτιώσεις στο θέμα αυτό, οι οποίες βρίσκονται σε εξέλιξη.

Ανθεκτικότητα και Ανοχή σε Σφάλμα (Resilience & Fault Tolerance)

Ο μηχανισμός ανοχής σφάλματος ODL είναι παρόμοιος με αυτόν του ONOS, με έναν περιττό αριθμό ελεγκτών SDN να απαιτείται για την παροχή ανοχής σφάλματος στο σύστημα. Σε περίπτωση αποτυχίας κύριου κόμβου, θα επιλεγεί ένας νέος για να

αναλάβει τον έλεγχο του δικτύου. Ο μηχανισμός επιλογής ενός κόμβου είναι ελαφρώς διαφοροποιημένος σε αυτούς τους δύο ελεγκτές - με το ONOS να επικεντρώνεται στην τελική συνέχεια και συνέπεια και το ODL να επικεντρώνεται στην υψηλή διαθεσιμότητα.

Γλώσσα Προγραμματισμού (Programming Language)

Η γλώσσα προγραμματισμού του ODL είναι η Java.

Κοινότητα (Community)

Το ODL είναι η δεύτερη πλατφόρμα για SDN Ελεγκτές που υποστηρίζεται από το Linux Foundation Networking. Έχει την υποστήριξη από τη μεγαλύτερη κοινότητα εμπλεκομένων μερών, σε σχέση με όλους τους άλλους SDN ελεγκτές ανοιχτού κώδικα που είναι διαθέσιμοι στην αγορά, με μερικά από τα μεγαλύτερα επιχειρηματικά ονόματα να προωθούν την ανάπτυξή του. Σημειώνεται ότι λόγω της μεγάλης διάδοσης της χρήσης του στην αγορά, το ODL έχει ενσωματωθεί σε άλλες ανοιχτού κώδικα SDN/NFV (Network Function Virtualization) λύσεις συντονισμού (orchestration) και διαχείρισης, όπως το OpenStack, Kubernetes, OPNFV και ONAP, οι οποίες είναι εξαιρετικά δημοφιλείς ως προς τη χρήση τους σε τηλεπικοινωνιακά δίκτυα.

8.4 Συγκριτικά Στοιχεία Ελεγκτή Ryu

Το Ryu αποτελεί μια εντελώς διαφορετική πρόταση για Ελεγκτές SDN, συγκρινόμενομενο με το ONOS και το OpenDayLight. Αν και χρησιμοποιεί ένα πακέτο προγραμμάτων τα οποία "τρέχουν" ως πλατφόρμα Ryu, θα ήταν προτιμότερο να θεωρείται ως μια εργαλειοθήκη (toolbox) η οποία μπορεί να αναπτυχθεί σε μια SDN λειτουργία.

Το Ryu, σε σχέση με το ONOS και το OpenDayLight, είναι ένα SDN πλαίσιο το οποίο βασίζεται σε τμήματα λογισμικού (components), παρέχοντας τμήματα λογισμικού με καλά καθορισμένες Διεπαφές Προγραμματισμού Εφαρμογών (APIs), που δίνουν τη δυνατότητα στους προγραμματιστές να δημιουργήσουν νέες διαχειριστικές και ελεγκτικές δικτυακές εφαρμογές.

Αρχιτεκτονική (Architecture)

Το Ryu SDN, συντίθεται από τα ακόλουθα Τμήματα Λογισμικού (components):

- Οι Νότιες Διεπαφές (Southbound Interfaces) επιτρέπουν την επκοινωνία με τους SDN μεταγωγείς και τους ελεγκτές.
- Οι κύριες λειτουργίες του Ελεγκτή Ryu υποστηρίζουν περιορισμένες εφαρμογές (π.χ ανεύρεση τοπολογίας, εκμάθηση μεταγωγέα) και βιβλιοθήκες.
- Εξωτερικές εφαρμογές μπορούν να αναπτύξουν δικτυακές πολιτικές στο επίπεδο δεδομένων μέσω καλά καθορισμένων Βόρειων Διεπαφών Προγραμματισμού Εφαρμογών (Northbound APIs), όπως το REST.

Δομοστοιχείωση & Επεκτασιμότητα (Modularity & Extensibility)

Το Ryu είναι δομημένο διαφορετικά σε σχέση με τις άλλες δύο λύσεις, παρέχοντας απλή υποστήριξη της υποδομής, όπου οι χρήστες της πλατφόρμας πρέπει να γράψουν το δικό τους κώδικα για την επιθυμητή χρήση του ελεγκτή.Αν και αυτό το τελευταίο απαιτεί τεχνογνωσία για την ανάπτυξή του, δίνεται η δυνατότητα πλήρους ευελιξίας για την δημιουργία εξειδικευμένων SDN λύσεων.

Κλιμακοδιαθεσιμότητα (Scalability)

Το Ryu, δεν διαθέτει εγγενώς δυνατότητες για λειτουργία συστάδας (clustering), αλλά απαιτεί εξωτερικά εργαλεία για το διαμοιρασμό πληροφοριών σχετικά με την κατάσταση του δικτύου και αντικατάστασης μελών της συστάδας.

- *Κλιμακοδιαθεσιμότητα Συστάδας (Cluster Scalability)*

Εξωτερικά εργαλεία, όπως το Zookeeper, κατανέμουν την επιθυμητή κατάσταση (desire state) στα μέλη της συστάδας. Πρόσθετοι κλώνοι (instances) του Ελεγκτή μπορούν να χρησιμοποιηθούν υπό την προϋπόθεση δεν αλλάζει η παραμετροποίησή τους.

- *Αρχιτεκτονική Κλιμακοδιαθεσιμότητα (Architectural Scalability)*

Αν και ο Ryu διαθέτει υψηλή διαθεσιμότητα μέσω του Zookeeper, δεν είναι σε θέση να υποστηρίξει ακόμα μια συστάδα ελεγκτών.

Διεπαφές (Interfaces)

Νότιες Διεπαφές (Southbound): Υποστηρίζει πολλαπλά πρωτόκολλα νότιας διεπαφής για τη διαχείριση συσκευών του δικτύου, όπως το OpenFlow, OF-Config και μερικώς το P4.
Βόρειες Διεπαφές (Northbound): Προσφέρει μόνο RESTful APIs, που είναι όμως περιορισμένες σε σχέση με το ONOS και το OpenDayLight.

Τηλεμετρικά Δεδομένα (Telemetry)

Ο Ryu δεν διαθέτει λειτουργία παροχής τηλεμετρικών δεδομένων και για το σκοπό αυτό απαιτείται η χρήση εξωτερικών εργαλείων.

Ανθεκτικότητα και Ανοχή σε Σφάλμα (Resilience & Fault Tolerance)

Δεν διαθέτει ενσωματωμένο μηχανισμό συστάδας και βασίζεται στη χρήση εξωτερικών εργαλείων για την υποστήριξη της διαθεσιμότητας. Η υψηλή διαθεσιμότητα επιτυγχάνεται μέσω της χρήσης πολλαπλών όμοιων ως προς τη διαμόρφωση (παραμετροποίηση) κλώνων ή και ενός μόνο κλώνου που ελέγχεται από ένα εξωτερικό πλαίσιο που ανιχνεύει και επαναλειτουργεί κόμβους που έχουν παρουσιάζει προβλήματα.
Η ανοχή σε σφάλμα παρέχεται από το Zookeeper, για την παρακολούθηση των ελεγκτών, για τις περιπτώσεις κατάπτωσής τους και για το διαμοιρασμό της κατάστασης μεταξύ των μελών της συστάδας.

Γλώσσα Προγραμματισμού (Programming Language)

Ο Ryu είναι γραμμένος σε γλώσσα προγραμματισμού Python.

Κοινότητα (Community)

Ο Ryu είναι ένας καλά υποστηριζόμενος και στοχευμένος ελεγκτής, εξαιρετικά δημοφιλής στην ακαδημαϊκή κοινότητα, ενώ διαθέτει και μια ενεργή κοινότητα που

ασχολείται με την ανάπτυξή του. Χρησιμοποιείται επίσης ως ελεγκτής δικτύου στο OpenStack.

8.5 Σύγκριση και Αξιολόγηση ONOS, OpenDaylight & Ryu

Η συγκριτική αξιολόγηση των διαφόρων SDN ελεγκτών που έχουν χρησιμοποιούνται σήμερα, πρέπει να λαμβάνει σοβαρά υπόψη της τα κίνητρα τα οποία οδήγησαν στην ανάπτυξή τους. Ο σχεδιασμός ενός ελεγκτή γίνεται με βάση τις διαφορετικές περιπτώσεις χρήσης του από τις επιχειρήσεις και την ακαδημαϊκή κοινότητα, λαμβάνοντας υπόψη του την κουλτούρα κάθε οργανισμού, αλλά και τις ανάγκες του ίδιου του έργου το οποίο έρχεται να υπηρετήσει. Υπό το πρίσμα αυτό για τους ελεγκτές SDN ONOS, ODL και Ryu που εξετάστηκαν ανωτέρω σημειώνονται τα ακόλουθα:

Αρχιτεκτονική (Archtecture)

Όπως συμβαίνει στις περισσότερες πλατφόρμες, υπάρχουν συμβιβασμοί οι οποίοι πρέπει να γίνουν προκειμένου να υπάρξει δυνατότητα σύγκρισης των δύο βασικών κατηγοριών ελεγκτών SDN, των κεντροποιημένων (centralized), ως προς την αρχιτεκτονική τους με ισχυρά συνδεδεμένο επίπεδο ελέγχου (π.χ ONOS, ODL, Ryu) και των εναλλακτικών αποκεντρωμένων (decentralized) με χαλαρά συνδεδεμένο επίπεδο ελέγχου (π.χ OpenKilda, Faucet).

Οι κεντροποιημένες αρχιτεκτονικές στην ανάπτυξη ελεγκτών SDN, όπως αυτές του ONOS και OpenDayLight έχουν την τάση να είναι ευκολότερες στη συντήρησή τους, να παρουσιάζουν μικρότερους χρόνους υστέρησης (lower latency) μεταξύ των πολύ στενά συνδεδεμένων νότιων διεπεφών (SBAPI), του PCE (Path Computation Element) και των βόρειων διεπαφών (NBAPIs). Ωστόσο, καθώς αυξάνεται η κλιμάκωση (scale), οι κεντρικοποιημένοι ελεγκτές μπορεί να παρουσιάζουν φαινόμενα συμφόρησης (bottleneck) στη λειτουργία τους. Σε περιπτώσεις δικτύωσης καθοριζόμενη από λογισμικό σε δίκτυα ευρείας περιοχή (SD-WAN) αυτό το γεγονός μπορεί να αυξήσει την καθυστέρηση του επιπεδου ελέγχου, η οποία όμως μπορεί να μετριαστεί από μια κατανεμημένη αρχιτεκτονική.
Ο Ryu αποτελεί μια διαφορετική περίπτωση όπως ήδη αναφέρθηκε προηγουμένως.

Οι καταναμημένες (distributed) αρχιτεκτονικές σε SDN ελεγκτές (π.χ OpenKilda, Faucet) αν και είναι γενικά πιο πολύπλοκες στην εγκακτάσταση, ανάπτυξη και συντήρησή τους, επιτρέπουν αποτελεσματικότερα την κλιμάκωση της πλατφόρμας, αποφεύγοντας φαινόμενα συμφόρησης στην απόδοσή τους.

Δομοστοιχείωση & Επεκτασιμότητα (Modularity & Extensibility)

Η δομοστοιχείωση (modularity) κάθε ελεγκτή καθορίζεται από την εστίαση του σχεδιασμού και τη γλώσσα προγραμματισμού. Το ONOS και το ODL διαθέτουν ενσωματωμένους μηχανισμούς για τη σύνδεση δομοστοιχείων κώδικα, σε βάρος της κεντρικοποιημένης επεξεργασίας κάθε ελεγκτή. Οι συγκεκριμένοι βασισμένοι στην Java ελεγκτές, αξιοποιούν τους OSGi containers, για τη φόρτωση δεσμίδων σε τρέχοντα χρόνο, επιτρέποντας με τον τρόπο αυτό μια πολύ ευέλικτη προσέγγιση στην προσθήκη λειτουργιών.
Οι ελεγκτές με βάση την Python, όπως εν προκειμένω ο Ryu, παρέχουν καλά καθορισμένες Διεπαφές Προγραμματισμού Εφαρμογών (APIs) στους προγραμματιστές για την αλλαγή του τρόπου διαχείρισης και διαμόρφωσης των τμημάτων λογισμικού (components) του ελεγκτή.

Κλιμακοδιαθεσιμότητα (Scalability)

Το ONOS και το ODL διαθέτουν εσωτερικές λειτουργίες για τη συντήρηση μιας συστάδας (cluster) ελεγκτών. Κάθε μια από αυτές τις λειτουργίες υποστηρίζεται από μια αποθήκη δεδομένων(datastore) που διαμοιράζει την τρέχουσα κατάσταση του SDN δικτύου, επιτρέποντας την εναλλακτική σύνδεση (failover) ελεγκτών σε περιπτώσεις διαχωρισμού της συστάδας. Στις νεές εκδόσεις των συγκεκριμένων ελεγκτών (ONOS, ODL) η κατάσταση αυτή φαίνεται να εξελίσσεται περαιτέρω.
Ο Ryu δεν διαθέτει εγγενώς δυνατότητα δημιουργίας συστάδας και χρειάζεται εξωτερικά εργαλεία, όπως το Zookeeper, για την κατανομή της επιθυμητής κατάστασης (desire state) στα μέλη της συστάδας. Πρόσθετοι κλώνοι (instances) του Ελεγκτή μπορούν να χρησιμοποιηθούν, υπό την προϋπόθεση δεν αλλάζει η διαμόρφωσή (παραμετροποίηση) τους. Μια PCE λειτουργία θα μπορούσε να προωθηθεί προς τα κάτω στον κλώνο υπό μορφή δομοστοιχείου στον ελεγκτή Ryu, υποστηριζόμενη από μια επιλεγμένη συστάδα.

Καθώς η κλιμάκωση του SDN δικτύου αυξάνει, καθίσταται εξαιρετικά δύσκολη η διαχείριση από μια τοπική συστάδα του φορτίου που προέρχεται από κάθε ένα μεταγωγέα του δκτύου.Αφήνοντας κατά μέρος τη γεωγραφική κατανομή των ελεγκτών, η διάσπαση του δικτύου σε μικρότερες λογικές νησίδες, μειώνει την ανάγκη μαζικής κλιμάκωσης μιας μοναδικής προς νότο συστάδας. Με αυτή τη σχεδίαση, ο συντονισμός μεταξύ των νησίδων καθίσταται κρίσιμος και ενώ είναι αναγκαία ακόμα μια κεντρικοποιημένη εικόνα του δικτύου, η απουσία PCE και τηλεμετρικών δεδομένων, δεν επηρρεάζει την σταθερότητα του επιπέδου δεδομένων από τη στιγμή που έχουν διαμορφωθεί οι ροές.

Και οι τρεις ελεγκτές (ONOS, ODL και RYU) μπορούν να κλιμακώνονται με τον τρόπο αυτό, εφόσον συμπεριλαμβάνουν εγγενώς δυντότητες δρομολόγησης BGP (Border Gateway Protocol), για το συντονισμό των ροών κυκλοφορίας μεταξύ των SDN νησίδων.

Διεπαφές (Interfaces)

Λαμβάνοντας υπόψη τις μελλοντικές απαιτήσεις συμβατότητας, για τον έλεγχο της νότιας κατεύθυνσης, το ONOS, το ODL και το Ryu περιλαμβάνουν και πρωτόκολλα πέραν του OpenFlow. Netconf, P4 και OF-Config που μπορούν να δώσουν πρόσθετες δυνατότητες επιλογής μεταγωγέων υλισμικού για περαιτέρω εξέλιξη, όποτε αυτό καταστεί αναγκαίο.

Οι Βόρειες Διεπαφές (NB APIs) αποτελούν ένα από τους βασικούς παράγοντες διαφοροποίησης μεταξύ των τριών συγκεκριμένων πλατφορμών. Το ONOS και το ODL, προσφέρουν μεγαλύτερη σειρά διεπαφών προς βορρά, με τα gRPC και RESTful API - που είναι διαθέσιμα μεταξύ των άλλων - καθιστώντας τις συγκεκριμένες διεπαφές ευκολότερα ενσωματώσιμες στους ελεγκτές. Το Ryu προσφέρει περιρισμένες δυνατότητες RESTful API, συγκρινόμενο με τις δύο προηγούμενες πλατφόρμες.

Τηλεμετρικά Δεδομένα (Telemetry)

Ένα από τα πρωταρχικής σημασίας θέματα που αφορούν στη διατήρηση και συντήρηση ενός δικτύου SDN είναι η εξαγωγή και χρήση τηλεμετρικών δεδομένων, για την παρακολούθηση και την τεκμηρίωση της κατάστασης του δικτύου, καθώς και της αποκατάστασης των όποιων θεμάτων/προβλημάτων προκύπτουν κατά τη διάρκεια της λειτουργίας του.

Στο συγκεκριμένο πεδίο το ODL στερείται μιας τέτοιας εφαρμογής,με την τηλεμετρία να αποτελεί ένα πειραματικό δομοστοιχείο στην επερχόμενη έκδοση του. Το ONOS, διαθέτει δομοστοιχεία που παρέχουν τηλεμετρικά δεδομένα χρησιμοποιώντας εργαλεία όπως το Grafana και το InfluxDB. Το Ryu δεν διαθετει τηλεμετρική λειτουργία συλλογής δεδομένων και γιαυτό απαιτείται η χρήση εξωτερικών εργαλείων.

Ανθεκτικότητα και Ανοχή σε Σφάλμα (Resilience & Fault Tolerance)

Οι πλατφόρμες ONOS και ODL έχουν ενσωματωμένες εφαρμογές για συστάδα. Η πλατφόρμα Ryu χρειάζεται εξωτερικά εργαλεία για την συγκεριμένη εφαρμογή. Αυτό το γεγονός απλοποιεί την αρχιτεκτονική του ελεγκτή απαλλάσσοντάς τον από τα βάρη της διατήρησης καταναμημένων βάσεων για την παροχή πληροφοριών κατάστασης του δικτύου. Η υψηλή διαθεσιμότητα στο Ryu επιτυγχάνεται μέσω της χρήσης πολλαπλών όμοιων ως προς τη διαμόρφωση (παραμετροποίηση) κλώνων ή και ενός μόνο κλώνου που ελέγχεται από ένα εξωτερικό πλαίσιο που ανιχνεύει και επαναλειτουργεί κόμβους που έχουν παρουσιάζει προβλήματα.
Το ONOs και το ODL παρέχουν ανοχή σφάλματος στο σύστημα για περιττο αριθμό ελεγκτών SDN. Σε περίπτωση συμβάντος αποτυχίας (κατάπτωσης) του κύριου κόμβου του δικτύου, επιλέγεται ένας νέος κόμβος που αναλαμβάνει τον έλεγχο του δικτύου. Ο μηχανισμός επιλογής του νέου κόμβου είναι ελαφρώς διαφοροποιημένος στις συγκεκριμένες δύο πλατφόρμες. Στην πλατφόρμα Ryu η ανοχή σε σφάλμα παρέχεται από το Zookeeper, για την παρακολούθηση των ελεγκτών, για τις περιπτώσεις κατάπτωσής τους και για το διαμοιρασμό της κατάστασης μεταξύ των μελών της συστάδας.

Γλώσσες Προγραμματισμού

Το ONOS και το ODL χρησιμοποιούν Java ως γλώσσα προγραμματισμού, η οποία είναι μια πολύ γνωστή γλώσσα, η οποία υποστηρίζεται πολύ καλή τεκμηριωμένη και μεγαλό αριθμό βιβλιοθηκών. Οι διαδικασίες Java απαιτούν κατάλληλη διαχείριση και διαμόρφωση, προκειμένου να παραμείνουν αποτελεσματικές. Το Ryu χρησιμοποιεί ως γλώσσα προγραμματισμού τη Python, που διαθέτει καλή τεκμηρίωση και απευθύνεται κυρίως σε προγραμματιστές. Η Python δεν είναι μια γρήγορη γλώσσα

και έχει εγγενείς περιορισμούς, συγκρινόμενη με άλλες γλώσσες όπως η Java, η C++ κλπ.

Κοινότητα (Community)

Οι πλατφόρμες του ONOS και του ODL υποστηρίζονται απόμεγάλες κοινότητες προγραμματιστών και χρηστών υπό την αιγίδα του Linux Foundation. Πολλοί μεγάλοι διεθνείς παίκτες σε επίπεδο επιχειρήσεων έχουν εμπλεκεί ενεργά στην ανάπτυξη και την διακυβέρνησή τους, γεγονός που προσφέρει ασφάλεια και μακροβιότητα αναφορικά με τη χρήση τους. Ένα ίσως αρνητικό στοιχείο στις περιπτώσεις αυτές είναι οι πολλοί εμπλεκόμενοι, οι απόψεις των οποίων μπορούν να προκαλέσουν προβλήματα σταθερότητας στην ανάπτυξη και την ταχύτητα εξέλιξης τους.
Αναφορικά με το Ryu σημειώνεται ότι είναι μια καλά υποστηριζόμενη και στοχευμένη πλαφτόρμα και δεδομένης της αναδυόμενης ανάπτυξης του συγκεκριμένου τομέα, φαίνεται να έχει ελπιδοφόρο μέλλον, με μια απλούστερη και βελτιωμένη προσέγγιση σε ότι αφορά την αλλαγές στην υποβολή και τον έλεγχο (submission & test) .

Βαθμολόγηση της αξιολόγησης

Σταθμίζοντας τα παραπάνω κριτήρια, ο μελετητής Ferzaneh Pakzad [95] κατέληξε στον ακόλουθο Πίνακα VIII όπου γίνεται αξιολόγηση των τριών πλατφορμών. Στην μελέτη αξιολόγησης που προαναφέρθηκε είχαν συμπεριληφθεί και δύο άλλοι γνωστοί κατανεμημένοι ελεγκτές (OpenKilda & Faucet) οι οποίοι βαθμολογήθηκαν με ίδια κριτήρια και την σταθμισμένη βαθμολογία, όπως και το ONOS, το ODL και το Ryu. Η αξιολόγηση δεν διαφοροποιείται αφού οι συγκεκριμένοι ελεγκτές βρέθηκαν αρκετά πίσω στη βαθμολόγισή τους, από τους ανωτέρω.

ΠΙΝΑΚΑΣ VIII

Πίνακας Αξιολόγησης ONOS, OpenDaylight, Ryu

Κριτήρια	**Στάθμιση**	**ONOS**	**ODL**	**Ryu**
Υποστήριξη OpenFlow	*20.0*	*20.0*	*19.0*	*20.0*
Υποστήριξη Βόρειας Διεπαφής (Northbound API support)	*20.0*	*20.0*	*20.0*	*16.0*
Υποστηρίξη Νότιας Διεπαφής (Southbound API support)	*10.0*	*10.0*	*10.0*	*8.0*
Γλώσσα Προγραμματισμού (Programming Language)	*5.0*	*4.0*	*4.0*	*4.5*
Βασικά Τμήματα Λογισμικού-Χαρακτηριστικά /Υπηρεσίες (Core Component features/services)	*5.0*	*4.5*	*4.5*	*2.0*
Ενσωματωμένες Δυνατότητες Συστάδας (Native Clustering Capabilities)	*10.0*	*9.0*	*7.0*	*2.0*
ΤυπικήΑρχιτεκτονική (Typical Architecture)	*3.0*	*2.7*	*2.4*	*2.4*
Οριζόντια Κλιμακοδιαθεσιμότητα (Horizontal Scalability)	*5.0*	*3.5*	*3.0*	*1.0*
Κάθετη Κλιμακοδιαθεσιμότητα (Vertical Scalability)	*5.0*	*3.5*	*3.0*	*4.5*
Επεκτασιμότητα (Extensibility)	*2.0*	*1.6*	*1.8*	*1.8*
Μέγεθος Κοινότητας/Συνεργασίες (Community Size & Partnerships)	*5.0*	*4.5*	*4.5*	*4.5*
Ανθεκτικότητα & Ανοχή σε Σφάλμα (Resilience & Fault Tolerance)	*5.0*	*4.0*	*3.0*	*4.0*
Υποστήριξη Λειτουργιών (Operations Support)	*5.0*	*4.5*	*2.5*	*2.5*
Σταθμισμένο Αποτέλεσμα	*100.0*	*92.0*	*84.5*	*73.2*

Συμπεράσματα

Στην παρούσα εργασία εξετάστηκαν, η Δικτύωση Καθοριζόμενη από Λογισμικό (SDN), ο ρόλος και η σημασία της Βόρειας Διεπεφής (Northbound Interface) στη λειτουργία της συγκεκριμένης δικτυακής αρχιτεκτονικής, όπως και οι δυνατότητες των τριών δημοφιλέστερων ελεγκτών ανοιχτού κώδικα, του OpenDayLight, του ONOS και του Ryu. Λαμβάνοντας υπόψη τα ανωτέρω συνάγονται τα ακόλουθα συμπεράσματα:

- Η έννοια της Δικτύωσης Καθοριζόμενης από Λογισμικό (SDN) είναι γνωστή για πολλά χρόνια. Εν τούτοις, άρχισε να επικρατεί ως αντίληψη και εφαρμογή μόλις τα τελευταία χρόνια εξαιτίας της εκρηκτικής αύξησης της διακίνησης των δεδομένων και των αναγκών προγραμματισιμότητας στα δίκτυα, λόγω της ανάπτυξης των τομέων των κινητών συσκευών, της εικονοποίησης των δικτύων, κλπ. Παρά τις εξελίξεις που έχουν σημειωθεί, υπάρχουν ακόμα τομείς της αρχιτεκτονικής των SDN δικτύων που δεν έχουν διερευνηθεί επαρκώς, όπως για παράδειγμα οι Βόρειες Διεπαφές, η ασφάλεια και άλλα.
- **Στα πλεονεκτήματα** της χρήσης των Δικτύων καθοριζόμενων από Λογισμικό, συγκρινόμενα με τα παραδοσιακά δίκτυα, περιλαμβάνονται: (α) η κεντροποιημένη (centralized) διαχείριση και αυτοματοποιημένη λειτουργία των δικτυακών συσκευών, (β) οι βελτιωμένες υπηρεσίες στους τελικούς χρήστες, (γ) οι δυνατότητες ευελιξίας και κλιμάκωσης, καθώς και η αποδοτικότητα. **Στα μειονεκτήματα** των SDN καταγράφονται η ανάγκη αλλαγών στο σύνολο της δικτυακής υποδομής, προκειμένου να εφαρμοστούν τα SDN πρωτόκολλα και να λειτουργήσουν οι SDN ελεγκτές. Ως εκ τούτου απαιτείται η πλήρης διαμόρφωση του δικτύου γεγονός που αυξάνει το κόστος. Λόγω της πολυπλοκότητας που παρουσιάζουν τα SDN δίκτυα στην εγκατάσταση και λειτουργία τους, αλλά και των συνεχών εξελίξεων στα εργαλεία διαχείρισης που ενσωματώνονται σ' αυτά, απαιτείται συνεχής εκπαίδευση του προσωπικού που ασχολείται με αυτά. Η ασφάλεια είναι μια άλλη επίσης μεγάλη πρόκληση για τα SDN δίκτυα, όπως και η single point failure.
- Οι ελεγκτές αποτελούν τον πυρήνα της SDN αρχιτεκτονικής, καθώς οι ίδιοι συγκροτούν το κεντρικό σημείο του δικτύου, το οποίο είναι υπεύθυνο για την ενεργοποιήση και ενορχήστρωση της επικοινωνίας μεταξύ των εφαρμογών που αντιπροσωπεύουν τις συσκευές δικτύου και την επιχειρηματική λογική. Η επικοινωνία αυτή είναι εφικτή μέσω των Northbound και Southbound API's, στις οποίες οφείλει όλες τις απλοποιήσεις και αυτοματοποιήσεις που υφίσταται στο δίκτυο. Οι SDN ελεγκτές λόγω της κεντρικοποιημένης λειτουργίας τους, δίνουν τη

δυνατότητα στους χειριστές του δικτύου να έχουν εικόνα της συνολικής κατάστασης του δικτύου (visibility), γεγονός που τους παρέχει ευκολία αλλαγών και βελτιώσεων, όποτε αυτό απαιτείται, ενώ ταυτόχρονα μπορεί να αυξάνουν και την ταχύτητα του. Παράλληλα, οι ελεγκτές παρέχουν δυνατότητες καλύτερης ασφάλειας και εντοπισμού προσβολών, τις οποίες μπορούν να αναχαιτίσουν με τον καθορισμό ασφαλών διαδρομών στο δίκτυο χωρίς να μεσολαβεί τείχος προστασίας (firewall). Από την άλλη πλευρά, για να μην είναι ευάλωτοι στα θέματα ασφάλειας και λειτουργίας τους (vulnerability), οι ελεγκτές θα πρέπει να παρακολουθούνται προσεκτικά από τους διαχειριστές του δικτύου και να ελέγχεται σχολαστικά η πρόσβαση σ΄αυτούς. Ένα άλλο τρωτό σημείο των SDN ελεγκτών είναι προστασία τους από καταναμημένες επιθέσεις άρνησης εξυπηρέτησης (Distributed Denial of Service Attacks-DDoS), που μπορούν να θέσουν εκτός λειτουργίας τις υπηρεσίες του δικτύου. Ακόμα ο περιορισμός της χρήσης φυσικών δρομολογητών και μεταγωγέων και η αντικατάστασή τους από εικονικούς, περιορίζει την ασφάλειά τους αφού πλέον δεν υπάρχει το τείχος προστασίας (Firewall), που τους συνοδεύει, με αποτέλεσμα το δίκτυο να είναι περισσότερο ευάλωτο σ' αυτόν τον τομέα.

- Η Βόρεια Διεπαφή (NBI) είναι ένας εκ των πυλώνων της SDN αρχιτεκτονικής, αφού λειτουργεί ως γέφυρα μεταξύ του Επιπέδου Ελέγχου και του Επιπέδου Διαχείρισης του δικτύου. Μέσω αυτής αναπτύσσονται στο Επίπεδο Διαχείρισης καινοτόμες εφαρμογές, δημιουργώντας νέες υπηρεσίες και προοπτικές για τους διαχειριστές των δικτύων. Η ανάπτυξη εφαρμογών στο Επίπεδο Διαχείρισης, δεν είναι τόσο εύκολη, εξαιτίας της έλλειψης τυποποίησης στον τομέα των NBIs και των σχετικών πρωτοκόλλων. Οι σημερινοί SDN ελεγκτές, προσφέρουν μια ποικιλία Βόρειων Διεπαφών (NBAPIs), οι οποίες μπορούν αν χωριστούν σε τρεις κατηγορίες : (α) στις RETSful APIs, (β) στις ειδικές ad-hoc λύσεις APIs και (γ) στις γλώσσες προγραμματισμού που χρησιμοποιούνται για τις NBAPIs. Οι RESTful APIs, οι οποίες είναι ευρύτατα διαδεδομένες, παρουσιάζουν πολλά πλεονεκτήματα, όπως η αποκεντρωμένη διαχείριση των πόρων, η σύνθεση των υπηρεσιών, η τοπική μετανάστευση (local migration),οι ετερογενείς πελάτες και η κλιμάκωση (scalability) των διακομιστών κλπ. Οι γλώσσες προγραμματισμού (Frenetic, Procera, Pyretic, Nettle, NetCore κλπ) από την άλλη προσφέρουν λύσεις με υψηλού επιπέδου αφαιρέσεις απλοποιώντας τον προγραμματισμό των συσκευών προώθησης, επιταχύνουν την ανάπτυξη SDN εφαρμογών, προσφέρουν καλύτερη αντιμετώπιση θεμάτων δρομολόγησης και παρακολούθησης του δικτύου, οι αφαιρέσεις επιτρέπουν τη δημιουργία εικονικών τοπολογιών του δικτύου και τέλος παρέχουν δυνατότητες

φορητότητας (portability).Υπάρχουν σήμερα στην αγορά και SDN ελεγκτές που χρησιμοποιούν τις δικές τους εξατομικευμένες APIs, όπως είναι το Onix, MobilFlow και PANE, που χρησιμοποιούν NVP NBAPI, SDMN API και PANE API. Η χρήση Βόρειων Διεπαφών βασισμένων στο σκοπό, φαίνεται να δίνει ικανοποιητικά αποτελέσματα, όμως λίγοι ελεγκτές είναι σε θέση να τις υποστηρίξουν. Υπάρχουν σήμερα ελεγκτές (π.χ ONOS και OpendayLight) που μπορούν να υποστηρίξουν βόρειες διεπαφές που βασίζονται τόσο στον ελεγκτή (controller-based NB APIs) όσο και στον σκοπό (intent-based NB APIs).

- Οι τρεις πιο γνωστοί και ευρέως χρησιμοποιούμενοι ελεγκτές (ODL, ONOS, Ryu) ανοιχτού κώδικα, παρουσιάστηκαν ως προς τις βασικές αρχές που τους διέπουν και αναλύθηκαν η λειτουργία και τα χαρακτηριστικά τους, όπως η αρχιτεκτονική τους, η δομοστοιχείωση (modularity) και η επεκτασιμότητα, η κλιμακοδιαθεσιμότητα (scalability), οι βόρειες και νότιες διεπαφές, οι γλώσσες προγραμματισμού κλπ. Από τη σχετική βιβλιογραφία προέκυψε ότι και οι τρεις αυτοί ελεγκτές υποστηρίζουν πολύ καλά τις βόρειες διεπαφές γεγονός που τους καθιστά ικανούς να συνεργάζονται με όλες τις εφαρμογές,αλλά και μεγάλο αριθμό δικτυακών συσκευών. Αρχιτεκτονικά αλλά και από πλευράς υποστήριξης εφαρμογών και οι τρείς αυτοί ελεγκτές διαφέρουν μεταξύ τους. Στο γεγονός αυτό οφείλεται ότι καθένας τους είναι προσανατολισμένος στην εξυπηρέτηση διαφορετικού τύπου δικτύων. Έτσι το ONOS είναι καταλληλότερο στην εξυπηρέτηση πολύ μεγάλων δικτύων παρόχων τηλεπικοινωνικών υπηρεσιών και μεταφορών, ενώ το ODL είναι προσανατολισμένο στα μεγάλα κέντρα δεδομένων. Υπάρχουν αρκετές βιβλιογραφικές αναφορές ποιοτικής συγκριτικής αξιολόγησης των ONOS, ODL και Ryu. Από αυτές προκύπτει ότι το ODL και το ONOS παρέχουν καλύτερες και καλά καθορισμένες λύσεις (π.χ NBAPIs, clustering (intent-based NB APIs) και επιλογές στους διαχειριστές δικτύων SDN, με το ONOS να υπερέχει ελαφρώς έναντι του ODL. Ο Ryu υπολείπεται των δύο άλλων, γεγονός που οφείλεται στην ανάγκη χρήσης πολλών εξωτερικών εργαλείων για διάφορες εφαρμογές που δεν είναι ενσωματωμένες στον συγκεκριμένο ελεγκτή.

Παράρτημα Α’

Η ιδέα της δημιουργίας ευφυών προγραμματιζόμενων δικτύων, και οι προσπάθειες ξεκινούν από τα μέσα της δεκαετίας του ’90. Παρατίθενται στη συνέχεια ορισμένα στοιχεία σχετικά με τις πρώιμες προσπάθειες που καταβλήθηκαν προς την κατεύθυνση αυτή.

The Open Signaling (OPENSIG)

Το 1995 ξεκίνησε μια σειρά εργαστηριών (workshops) την ονομασία Open Signaling, που ήταν αφιερωμένα στην πιο ανοικτή, επεκτάσιμη και προγραμματιζόμενη λειτουργία των ATM, του Διαδικτύου και των δικτύων κινητής τηλεφωνίας. Η κύρια ιδέα ήταν ότι ήταν απαραίτητος ο διαχωρισμός μεταξύ του υλισμικού επικοινωνίας και του λογισμικού ελέγχου γεγονός που θα είχε πολλά πλεονεκτήματα στα ενδιαφερόμενα μέρη. Ο πυρήνας ττων συμπερασμάτων ήταν να υπάρχει πρόσβαση στο υλισμικό του δικτύου μέσω ανοικτών, προγραμματιζόμενων διεπαφών δικτύου.

General Switch Management Protocol (GSMP)

Συγκροτήθηκε ομάδα εργασίας του Internet Engineering Task Force (IETF), η οποία ήταν υπεύθυνη για τον καθορισμό του πρωτοκόλλου GSMP, που περιγράφει τον τρόπο ελέγχου ενός μεταγωγέα. Το GSMP επιτρέπει σε έναν ελεγκτή να δημιουργεί και να απελευθερώνει συνδέσεις μέσω ενός διακόπτη, να προσθέτει και να διαγράφει συνδέσεις πολυεκπομπής, να διαχειρίζεται τις θύρες μεταγωγέων, να ενημερώνει τις παραμέτρους διαμόρφωσης, να ζητά και να διαγράφει την κράτηση του πόρου διακόπτη και άλλα. Η ομάδα εργασίας ολοκλήρωσε τις εργασίες της και η πρότασή της για τα σχετικά πρότυπα (GSMPv3) δημοσιεύθηκε τον Ιούνιο του 2002.

Active Networking

Η Υπηρεσία Προηγμένων Ερευνητικών Έργων Άμυνας (Defense Advanced Research Project Agency - DARPA) πρότεινε μια προγραμματιζόμενη δικτυακή υποδομή για προσαρμοσμένες υπηρεσίες, στα μέσα της δεκαετίας του '90. Η πρόταση είχε επικεντρωθεί σε δύο βασικές προσεγγίσεις: προγραμματιζόμενους από το χρήστη μεταγωγείς και κάψουλες. Οι κάψουλες αφορούσαν σε μέρη του προγράμματος που θα μεταφέρονται στα μηνύματα χρήστη και θα μπορούσαν να ερμηνευτούν και να εκτελεστούν από δρομολογητές. Παρά τη σημαντική δραστηριότητα, η ενεργός δικτύωση ποτέ δεν συγκέντρωσε κρίσιμη μάζα ή εκτεταμένη χρήση από τη βιομηχανία, κυρίως λόγω πρακτικών προβλημάτων ασφάλειας και απόδοσης.

Tempest

Το Tempest ήταν ένα πλαίσιο για ασφαλή προγραμματιζόμενα δίκτυα, το οποίο εισήχθη το 1998. Το πλαίσιο Tempest παρέχει ένα προγραμματιζόμενο περιβάλλον δικτύου επιτρέποντας την εισαγωγή και την τροποποίηση των υπηρεσιών δικτύου σε δύο επίπεδα. Αυτά τα χαρακτηριστικά του πλαισίου Tempest επέτρεψαν στους παρόχους υπηρεσιών να γίνουν ουσιαστικά φορείς εκμετάλλευσης δικτύων για ορισμένα σαφώς καθορισμένα τμήματα του φυσικού δικτύου. Αυτό τους παρείχε τη δυνατότητα να επωφεληθούν από τις γνώσεις που διέθεταν για το πώς θα χρησιμοποιηθούν οι πόροι του δικτύου, προγραμματίζοντας τη δική τους ειδικά σχεδιασμένη αρχιτεκτονική ελέγχου.

Path Computation Element (PCE)

Το Path Computation Element (Πρότυπο IETF), εισήχθη το 2004 και είναι ένα πρωτόκολλο ελέγχου που λειτουργεί σε δίκτυα MPLS και εν μέρει αφαιρεί την ευθύνη από τους δρομολογητές στον καθορισμό των διαδρομών του δικτύου. Η αρχιτεκτονική PCE, που ορίζεται στο RFC 4655 (2006), απλοποιεί τον υπολογισμό διαδρομής διαχωρίζοντας τον προσδιορισμό της τοπολογίας του δικτύου από τη δημιουργία της διαδρομής.

The 4D project

Το 4D Projectδημοσιεύτηκε το 2004 και αφορούσε στον διαχωρισμό μεταξύ των (λογικών) πρωτοκόλλων απόφασης δρομολόγησης. Το επίπεδο "απόφασης" έχει μια συνολική εικόνα του δικτύου και σε συνεργασία με τις υπηρεσίες των επιπέδων «διάδοσης» και «ανακάλυψης», έχουν τον έλεγχο στο επιπέδο «δεδομένων» για την προώθηση της κυκλοφορίας. Η "ανακάλυψη" αναφέρεται στις πληροφορίες σχετικά με τους πόρους που διατίθενται στον ελεγκτή δικτύου και η "διάδοση" αναφέρεται στον τρόπο ανίχνευσης της τοπολογίας του δικτύου. Αργότερα έργα όπως το NOX στηρίχτηκαν σε αυτές τις ιδέες, οι οποίες πρότειναν ένα λειτουργικό σύστημα για δίκτυα με δυνατότητα OpenFlow. Το τελευταίο πρότυπο δημοσιεύθηκε τον Ιούνιο του 2011.

Βιβλιογραφία - Αναφορές

[1] S. Kemp, "Digital in 2018: World's Internet Users Pass The 4 Billion Mark," 2018 Accessed on: 10-Nov-2018. Available: https://wearesocial.com/blog/2018/01/global-digital-report-2018.

[2]W. Yangyang and B. Jun, "Survey of Mechanisms for Inter-Domain SDN," accessed on: 9-Nov-2018. https://www.zte.com.cn/global/about/magazine/zte-communications/2017/3/en 225/465746

[3] D. L. Tennenhouse, J. M. Smith, W. D. Sincoskie, D. J. Wetherall, and G. J. Min-den, "A survey of active network research," IEEE Communications Magazine, vol. 35, no. 1, pp. 80–86, Jan 1997.

[4] A. T. Campbell, H. G. De Meer, M. E. Kounavis, K. Miki, J. B. Vicente & D. Vil-lela, "A survey of programmable networks," SIGCOMM Comput. Commun. Rev, vol. 29, no. 2, pp. 7–23, Apr. 1999.

[5] https://opensource.com/resources/what-is-software-defined-networking

[6]https://www.ionos.com/digitalguide/server/know-how/software-defined-network-ing/

[7] https://www.opennetworking.org/sdn-definition/

[8] RFC 7426 "Software-Defined-Networking (SDN): Layers & Architecture Terminology", IEFT (2015).

[9] https://community.fs.com/blog/unveil-the-myths-about-sdn-switches.html

[10] https://en.wikipedia.org/wiki/OpenFlow

[11]https://www.sdxcentral.com/articles/contributed/sdn-openflow-tcam-need-to-know/2012/07/

[12] https://www.sdxcentral.com/networking/sdn/definitions/sdn-controllers/

[13]"Distributed SDN Control: Survey, Taxonomy and Challenges" F. Bennur, S. Souihi and A. Melluk, IEEE Communications Surveys & Tutorials PP(99):1-1, December 2017.

[14] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, and J. Turner, "Openflow: Enabling innovation in campus networks," SIGCOMM Comput. Commun. Rev., vol. 38,no. 2, pp. 69–74, Mar. 2008.

[15] E. Haleplidis, J. H. Salim, J. M. Halpern, S. Hares, K. Pentikousis, K. Ogawa, W. Wang, S. Denazis, and O. Koufopavlou, "Network programmability with forces," IEEE Communications Surveys Tutorials, vol. 17, no. 3, pp. 1423–1440, 2015.

[16] B. Pfaff and B. Davie, “The Open vSwitch Database Management Protocol,” Informational, Internet Engineering Task Force, RFC 7047, December 2013. http://www.ietf.org/rfc/rfc7047.txt

[17] H. Song, “Protocol-oblivious forwarding: Unleash the power of sdn through a future-proof forwarding plane,” in Proceedings of the Second ACM SIGCOMM Workshop on Hot Topics in Software Defined Networking, ser. HotSDN ’13. New York, NY, USA: ACM, 2013, pp.127–132.

[18] M.Smith, M.Dvorkin, V.Laribi, V.Pandey, P.Gerg, and N.Weidenbacher, “ OpFlex Control Protocol,” accessed on: 31-July-2018. https://tools.ietf.org/ html/draft-smith-opflex-03

[19] G. Bianchi, M. Bonola, A. Capone, and C. Cascone, “Openstate: Programming platform-independent stateful openflow applications inside the switch,” SIGCOMM Comput. Commun. Rev., vol. 44, no. 2, pp. 44–51, Apr. 2014.

[20] Bruno Astuto A. Nunes, Marc Mendonca, Xuan-Nam Nguyen, Katia Obraczka, and Thierry Turletti, “A Survey of Software-Defined Networking: Past, Present, and Future of Programmable Networks”, Communications Surveys and Tutorials, IEEE Communications Society, Institute of Electrical and Electronics Engineers, 2014, 16 (3), pp.1617 – 1634.

[21] Pritesh Ranjan, Pankaj Pande, Ramesh Oswal, Zainab Qurani, Rajneeshkaur Bedi, “A Survey of Past, Present and Future of Software Defined Networking”, Volume 2, Issue 4, International Journal of Advance Research in Computer Science and Management Studies, 2014.

[22] B. Heller, S. Seetharaman, P. Mahadevan, Y. Yiakoumis, P. Sharma, S. Banerjee, and N. McKeown. Elastictree: Saving energy in data center networks. In Proceedings of the 7th USENIX conference on Networked systems design and implementation, pages 17–17. USENIX Association, 2010.

[23] H. Shirayanagi, H. Yamada, and K. Kono. Honeyguide: A vm migration-aware network topology for saving energy consumption in data center networks. In Computers and Communications (ISCC), 2012 IEEE Symposium on, pages 000460–000467. IEEE, 2012.

[24] C.-Y. Hong, S. Mandal, M. Al-Fares, M. Zhu *et al.*, “B4 and After: Managing Hierarchy, Partitioning, and Asymmetry for Availability and Scale in Google’s Software-defined WAN,” in Proceedings of the 2018Conference of the A ACM Special Interest Group on Data Communication, 2018, pp. 74–87.

[25] Hyojoon Kim, Nick Feamster, “Improving Network Management with Software Defined Networking”, IEEE Communications Magazine, 2013.

[26] SDX Central, “The Future of Network Virtualization and SDN Controllers”, https://www.sdxcentral.com, 2016

[27] Hyojoon Kim, Nick Feamster, “Improving Network Management with Software Defined Networking”, IEEE Communications Magazine, 2013

[28] Joshua Reich, Christopher Monsanto, Nate Foster, Jennifer Rexford, and David Walker "Modular SDN Programming with Pyretic", https://www.cs.princeton.edu, 2013.

[29] POX Controller Manual Current Documentation." https://noxrepo.github.io/pox-doc/html/

[30] David Erickson, "The Beacon OpenFlow controller", HotSDN'13, August 16, 2013, Hong Kong, China

[31] Big Switch Networks, "Project Floodlight."http://www.projectfloodlight.org/floodlight/

[32] Alexandru L. Stancu, Simona Halunga, Alexandru Vulpe, George Suciu, Octavian Fratu, Eduard C. Popovici "A comparison between several SDN Controllers", TELSIKS 2015 conference

[33] Abhishek Rastogi, Abdul Bais, "Comparative Analysis of Software Defined Networking (SDN) Controllers – In Terms of Traffic Handling Capabilities", Multi-Topic Conference (INMIC), 2016 19th International

[34] Pedro Bispo, Daniel Corujo, Rui L. Aguiar "A Qualitative and Quantitative assessment of SDN controllers", 20171nlemalional Young Engineers Forum (YEF~ECE), 2017

[35] Shiva Rowshanrad, Vajihe Abdi and Mamijeh Keshtgari, "Performance Evaluation of SDN Controllers: FloodLight and OpenDaylight", IIUM Engineering Journal, Vol. 17, No. 2, 2016.

[36] Ola Salman, Imad Elhajj, Ayman Kayssi, Ali Chehab, "SDN Controllers: A Comparative Study", 2016

[37] P. H. Isolani, J. A. Wickboldt, C. B. Both, J. Rochol, and L. Z.Granville, "Sdn interactive manager: An openflow-based sdn manager," in IFIP/IEEE International Symposium on Integrated Network Management, 2015, pp. 1157–1158.

[38] W. Kim, J. Li, J. W. K. Hong, and Y. J. Suh, "Ofmon: Openflow monitoring system in onos controllers," in 2016 IEEE NetSoft Conference and Workshops (NetSoft), 2016, pp. 397–402.

[39] R. Sherwood, G. Gibb, K.-k. Yap *et al.*, "FlowVisor: A Network Virtualization Layer," *Network*, p. 15, 2009.

[40] R. D. Corin, M. Gerola, R. Riggio, F. D. Pellegrini, and E. Salvadori, "Vertigo: Network virtualization and beyond," in European Workshop on Software Defined Networking, 2012, pp. 24–29.

[41] L. Xingtao, G. Yantao, W. Wei, Z. Sanyou, and L. Jiliang, "Network virtuali-zation by using software-defined networking controller based docker," in IEEE Information Technology, Networking, Electronic and Automation Control Conference, 2016, pp. 1112–1115.

[42] D. Drutskoy, E. Keller, and J. Rexford, "Scalable network virtualization in software-defined networks," IEEE Internet Computing, vol. 17, no. 2, pp. 20–27, March 2013.

[43] A. Blenk, A. Basta, and W. Kellerer, "Hyperflex: An sdn virtualization architecture with flexible hypervisor function allocation," in *IFIP/IEEE* International Symposium on Integrated Network Management, 2015, pp. 397–405.

[44] A. Mayoral, R. Vilalta, R. Muoz *et al.*, "Sdn orchestration architectures and their integration with cloud computing applications," Optical Switching and Networking, vol. 26, pp. 2 – 13, 2017.

[45] "OpenStack Havana Release." https://www.openstack.org/software/havana/

[46] W. Kellerer, A. Basta, A. Blenk "Using a flexibility measure for network design space analysis of SDN and NFV" IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), 2016

[47] Diego Kreutz, Fernando M. V. Ramos, Paulo Verissimo, Christian Esteve Rothenberg, Siamak Azodolmolky and Steve Uhlig, Member, "Software-Defined Networking: A Comprehensive Survey", https://arxiv.org/abs/1406.0440.

[48]Verizon,"SDN-NFV Reference Architecture",http://innovation.verizon.com,2016.

[49]"NBIWG," accessed on: 31-July-2018. https://www.opennetworking.org/tag/nbi-working-group/

[50] K.-K. Yap, T.-Y. Huang, B. Dodson, M. S. Lam, and N. McKeown, "Towards software-friendly networks," in Proceedings of the First ACM Asia-pacific Workshop on Workshop on Systems, ser. APSys '10. New York, NY, USA: ACM, 2010, pp. 49–54.

[51] M. Yu, A. Wundsam, and M. Raju, "Nosix: A lightweight portability layer for the sdn os," SIGCOMM Comput. Commun. Rev., vol. 44, no. 2, pp. 28–35, Apr. 2014.

[52] C. J. Casey, A. Sutton, and A. Sprintson, "tinynbi: Distilling an API from essential openflow abstractions," CoRR, vol. abs/1403.6644, 2014

[53] N. Foster, R. Harrison, M. J. Freedman, C. Monsanto, J. Rexford, A. Story, and D. Walker, "Frenetic: A network programming language," SIGPLAN Not., vol. 46, no. 9, pp. 279–291, Sep. 2011.

[54] T. L. Hinrichs, N. S. Gude, M. Casado, J. C. Mitchell, and S. Shenker, "Practical declarative network management," in Proceedings of the 1st ACM Workshop on Research on Enterprise Networking, ser. WREN '09. New York, NY, USA: ACM, 2009, pp. 1–10.

[55] N. P. Katta, J. Rexford, and D. Walker, "Logic programming for software-defined networks," accessed on: 31-July-2018. http://frenetic-lang.org/publi-cations/logic-programmingxldi12.pdf.

[56] A. Voellmy, H. Kim, and N. Feamster, "Procera: A language for highlevel reactive network control," in Proceedings of the First Workshop on Hot Topics in Software Defined Networks, ser. HotSDN '12. New York, NY, USA: ACM, 2012, pp. 43–48.

[57] C. Monsanto, N. Foster, R. Harrison, and D. Walker, “A compiler and run-time system for network programming languages,” SIGPLAN Not., vol. 47, no. 1, pp. 217–230, Jan. 2012.

[58]] A. Voellmy and P. Hudak, “Nettle: Taking the sting out of programming network routers,” in Proceedings of the 13th International Conference on Practical Aspects of Declarative Languages, ser. PADL’11. Berlin, Heidelberg: Springer-Verlag, 2011, pp. 235–249.

[59] C. J. Anderson, N. Foster, A. Guha, J.-B. Jeannin, D. Kozen, C. Schlesinger, and D. Walker, “Netkat: Semantic foundations for networks,” SIGPLAN Not., vol. 49, no. 1, pp. 113–126, Jan. 2014.

[60] M. Reitblatt, M. Canini, A. Guha, and N. Foster, “Fattire: Declarative fault tolerance for software-defined networks,” in Proceedings of the Second ACM SIGCOMM Workshop on Hot Topics in Software Defined Networking, ser. HotSDN ’13. New York, NY, USA: ACM, 2013, pp. 109–114.

[61] H. Kim, J. Reich, A. Gupta, M. Shahbaz, N. Feamster, and R. Clark, “Kinetic: Verifiable dynamic network control,” in Proceedings of the 12th USENIX Conference on Networked Systems Design and Implementation, ser. NSDI’15. Berkeley, CA, USA: USENIX Association, 2015, pp. 59–72.

[62] R. Soule, S. Basu, R. Kleinberg, E. G. Sirer, and N. Foster, “Managing ´ the network with merlin,” in Proceedings of the Twelfth ACM Workshop on Hot Topics in Networks, ser. HotNets-XII. New York, NY, USA: ACM, 2013, pp. 24:1–24:7.

[63] S. Layeghy, F. Pakzad, and M. Portmann, “Scor: Constraint programming-based northbound interface for sdn,” in 2016 26th International Telecommunication Networks and Applications Conference (ITNAC), Dec 2016, pp. 83–88.

[64] A. D. Ferguson, A. Guha, C. Liang, R. Fonseca, and S. Krishnamurthi, “Participatory networking: An api for application control of sdns,” SIGCOMM Comput. Commun. Rev., vol. 43, no. 4, pp. 327–338, Aug. 2013.

[65] T. Koponen, M. Casado, N. Gude, J. Stribling, L. Poutievski, M. Zhu, R. Ramanathan, Y. Iwata, H. Inoue, T. Hama, and S. Shenker, “Onix: A distributed control platform for large-scale production networks,” in Proceedings of the 9th USENIX Conference on Operating Systems Design and Implementation, ser. OSDI’10. Berkeley, CA, USA: USENIX Association, 2010, pp. 351–364.

[66] P. Berde, M. Gerola, J. Hart, Y. Higuchi, M. Kobayashi, T. Koide,B. Lantz, B. O’Connor, P. Radoslavov, W. Snow, and G. Parulkar,“Onos: Towards an open, distributed sdn os,” in Proceedings of theThird Workshop on Hot Topics in Software Defined Networking, ser.HotSDN ’14. New York, NY, USA: ACM, 2014, pp. 1–6.

[67]“Intent Framework,” accessed on: 08-August-2018. https://wiki.onosproject.org/display/ONOS/Intent+Framework

[68] Y. Rodrigues, “OpenDaylight ODL: Network Intent Composition (NIC) - A real Intent-based solution, challenges and next stepsIntent Framework,” https://www.

serro.com/ opendaylight - network - intent - composition-a- real-intent-based-solution-challenges-and-next-steps/, accessed on: 08-August-2018.

[69] "Network Intent Composition (NIC) Developer Guide," accessed on: 08-August-2018. https://docs.opendaylight.org/en/stable-oxygen/developer-guide/network-intent-composition- (nic)-developer-guide.html

[70] M. Pham and D. B. Hoang, "Sdn applications - the intent-based northbound interface realisation for extended applications," in 2016 IEEE NetSoft Conference and Workshops (NetSoft), June 2016, pp. 372–377.

[71]"Software Defined Networking: The New Norm of Networks", ONF White Paper, April 13, 2012 .

[72]A. Clemm, "Navigating device management and control interfaces in the age of SDN" February 28, 2014.http://blogs.scisco.com/getyourbuildon/navigating-device-management-and-contoll-interfaces-in-the-age-of-sdn

[73] "VMWare NSX – the platform for network virtualization" http://www.vmware.com/files/pdf/products/nsx/VMware-NSX-Datasheet.pdf

[74] "Virtual Tenant Network (VTN) OpenDaylight Project", http://wiki.opendaylight.org/view/OpenDayLight_Virtual_Tenant_Network_VTN:Main.

[75] "OpenDOVE OpenDayLight Project", http://wiki.opendaylight.org/view/Open_DOVE:Main.

[76] "Group Based Policy OpenDayLight Project", http://wiki.opendaylight.org/view/Group_Policy:Main.

[77] "Defense4All Opendaylight Project", http://wiki.opendaylight.org/view/Defense4All:Main

[78] S. Wallin and C. Wikstrom, "Automating network and service configuration using NETCONF and YANG", Usenix LISA '11 December 4-9 2011Boston, MA

[79] " The Cisco Application Policy Infrastructure Controller," http://cisco.com/en/us/solutions/collateral-data-center-virtualization/unified-fabric/white-paper-c11-730021.pdf.

[80] "Application centric infrastructure object-oriented data model: gain advanced network control and programmability," http://cisco.com/en/us/solutions/collateral /data-center-vitrualization/unified-fabric/white-paper-c11-730021.pdf

[81] S. Raza, D. Lenrow, P. Menezes, E. Dom, T. Tsou and F. Schneider "Open Networking Foundation North bound interface working group (NBI-WG), chapter v.1.1" October 10, 2013.

[82] "Interface to the Routing System (i2rs) IETF working group," http://datatracker.ietf.org/wg.i2rs

[83] R. Enns, M. Bjorkloud, J. Schoenwaelder, A. Bienman, "Network Configuration Protocol (NETCONF)," http://tools.ietf.org/html/rfc6241

[84] A. Bierman, M. Bjorklund, K. Watsen, and R. Fernando, "RESTCONF Protocol, draft-bierman-netconf-restconf-04" February 13, 2014, http://datatracker.ietf.org/doc/draft-bierman-netconf-restconf/

[85] M. Bjorklund et.al. "YANG – A data modeling language for network configuration protocol (NETCONF), RFC 6020, October 2010, http://datatracker.ietf .org/doc/rfc6020

[86] "Openflow Protocol Specifications," http://opennetworking.org/images/stories/downloads/sdn-resources/onf-specifications/openflow/opeflow-spec-v1.3.1.pdf

[87] T. Bates, R. Chandra,D. Katz,Y. Rekhter,"Multiprotocol Extensions for BGR-4" http://tools.ietf.org/html/rfc4760

[88] H. Gredler, J. Medved, S. Previdi, A. Farrel, S. Ray "North-Bound Distribution of Link-State and TE Information using BGP" http://tools.ietf.org/html/draft-ietf-idr-ls-distribution

[89] E. Crabbe, J. Medved, I. Minei, R Varga, "PCEP Extensions for Stateful PCE" http://tools.ietf.org./html/draft-ietf-pce-stateful-pce

[90] White Paper "Introducing ONOS – a SDN network operating system for Service Porviders" ON.LAB, 2014, https://onosproject.org/wp-content/uploads/2014/11 /Whitepaper-ONOS-final.pdf

[91] SDN Series Part Four: Ryu, a Rich-Featured Open Source SDN Controller Supported by NTT Labs, https://thenewstack.io/sdn-series-part-iv-ryu-a-rich-featured-open-source-sdn-controller-supported-by-ntt-labs/

[92] R. Kubo, T. Fujita, Y. Agawa "Ryu SDN Framework-Open-source SDN Platform Software", NTT Technical Review, Augast 2014, vol.12, No 8.

[93] L. Zhu, M. Karim, K. Sharif, F. Li, X. Du, M.Gurazani "SDN Controllers: Benchmarking & Performance Evaluation", 2/2019, http://arvix.org/pdf/1902.04491.pdf

[94] O. Salam, I. Elhajj, A. Kayssi, a. Chehab "SDN Controller: A comparative study", 4/2016, http://www.researchgate.net/publication/304457462-sdn-controller-A-copmarative-study

[95] F. Pakzad "Comparison of Software Defined Network (SDN) Controllers: Part II, III, V and VII", http://aptira.com/comparison-of-software-defined-network-controllers