Εθνικο Μετσοβιο Πολυτεχνειο
Σχολη Ηλεκτρολογων Μηχανικων και Μηχανικων Υπολογιστων
Τομεας Τεχνολογιας Πληροφορικης και Υπολογιστων

Ασφάλεια στα SDN Δίκτυα με την χρήση του **Blockchain**

Διπλωματικη Εργασια

του

ΟΡΕΣΤΗ ΠΑΝΑΡΑ

Επιβλέπων: Συμεών Παπαβασιλείου
Καθηγητής Ε.Μ.Π.

Εργαστηριο Διαχειρισης Δικτυων και Βελτιστου Σχεδιασμου
Αθήνα, Μάιος 2020

Εθνικό Μετσόβιο Πολυτεχνείο
Σχολή Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών
Τομέας Τεχνολογίας Πληροφορικής και Υπολογιστών
Εργαστήριο Διαχείρισης Δικτύων και Βέλτιστου Σχεδιασμού

Ασφάλεια στα SDN Δίκτυα με την χρήση του Blockchain

Διπλωματικη Εργασια

του

ΟΡΕΣΤΗ ΠΑΝΑΡΑ

Επιβλέπων: Συμεών Παπαβασιλείου
Καθηγητής Ε.Μ.Π.

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την 21 Μαίου 2020.

(Υπογραφή)	*(Υπογραφή)*	*(Υπογραφή)*
.....	.....	.....
Συμεών Παπαβασιλείου	Θεοδώρα Βαρβαρίγου	Ιωάννα Ρουσσάκη
Καθηγητής Ε.Μ.Π.	Καθηγήτρια Ε.Μ.Π.	Επίκουρη Καθηγήτρια Ε.Μ.Π.

Αθήνα, Μάιος 2020

(Υπογραφή)

.....

Ορεστης Παναρας

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

Εθνικό Μετσόβιο Πολυτεχνείο
Σχολή Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών
Τομέας Τεχνολογίας Πληροφορικής και Υπολογιστών
Εργαστήριο Διαχείρισης Δικτύων και Βέλτιστου Σχεδιασμού

Περίληψη

Στο Διαδίκτυο του Μέλλοντος, η πληροφορία, ο πυρήνας των υπολογιστικών συστημάτων, διαμειράζεται κατά βάση μέσω δικτύων. Για την διαχείρηση και τον διαμοιρασμό της, χρειάζεται η χρήση αξιόπιστων μέσων μετάδοσης και μηχανισμών που εξασφαλίζουν την ακεραιότητά της. Απαραίτητος είναι ένας μηχανισμός διασφάλισης έτσι ώστε να καθιστά σίγουρο ότι αυτή, δεν έχει αλλιωθεί και παραποιηθεί σε κάτι πιθανόν κακόβουλο. Στην ερώτηση πως γίνεται η πληροφορία να είναι κακόβουλη, ένα παράδειγμα είναι τα υπολογιστικά συστήματα στις σύγχρονες υποδομές, όπως η πληθώρα αρχείων διαμόρφωσης ή κανόνων τοιχών προστασίας σε κρίσιμες εφαρμογές. Ακόμα και σήμερα, είναι συνηθισμένη η διαχείριση και ο σχεδιασμός ολόκρηρων υποδομών σε ένα κεντρικό υπολογιστικό σύστημα ή μία εφαρμογή, όπως το Windows Admin Center της Microsoft. Είναι σημαντικό να αναρωτηθούμε πόσο αξίζει μία αποκεντρωμένη προσσέγγιση για τον διαμοιρασμό πληροφορίας, με τρόπο που να εξασφαλίζεται η ακεραιότητά της.
Στην παρούσα εργασία θα ασχοληθούμε με δύο νέες τεχνολογίες. Τα Καθορισμένα από το Λογισμικό Δίκτυα (SDN) και τις Κατανεμημένες Εγγραφές (BlockChain). Στα SDN, βασική ιδέα είναι ο συγκεντρωτισμός της πληροφορίας (Intelligence centralization), αλλά και βασικό μειονέκτημα όσον αφορά την ασφάλεια αυτών. Αντίθετα στο Blockchain, ο αποσυγκεντρωτισμός της πληροφορίας (decentralization via ledgers), παρέχει αξιοπιστία χωρίς κάποια κεντρική αρχή (central authority). Τα SDN, διαχωρίζοντας το επίπεδο ελέγχου με το επίπεδο δεδομένων, καταφέρνουν να υλοποιήσουν δίκτυα τα οποία είναι εξ ορισμού δυναμικά και προγραμματιζόμενα, αυξάνοντας την ευελιξία και την ικανότητα διαχείρισης αυτών. Το BlockChain είναι υλοποιείται μέσω μίας λίστας καταχωρήσεων οι οποίες μετά την δημιουργία τους είναι πολύ δύσκολο να παραποιηθούν αφού δημιουργηθούν. Αυτό γίνεται υλοποιήσιμο, με τον κατακερματισμό. Οι παραπάνω δύο τεχνολογίες ξεκίνησαν κατά βάση την ίδια χρονιά, το 2008,τα (SDN), με το API για το πρωτόκολλο OpenFLow και το (blockchain) με το γνωστό πλέον white paper για το Bitcoin με ομώνυμο τίτλο: “Bitcoin: A Peer-to-Peer Electronic Cash System”.

Λέξεις Κλειδιά

Ασφάλεια, εμπιστοσύνη, σύστημα καταγραφής, αλυσίδες κοινοποιήσεων, SDN, ledgers, κεντροκοποιημένα και κατανεμημένα συστήματα, δίκτυο ομότιμων χρηστών.

Abstract

The core of the Internet of the Future, information, is greatly distributed over networks. For the management, distribution of information as data sets, it is necessary to use reliable transmission media and mechanisms that ensure integrity. Moreover when such a distribution of data takes place, safeguard mechanisms are needed to make sure that no alteration of critical information has taken place which can potentially be harmful to any third parties. Asking how any part of information can be malicious in any way the answer is obvious. Data are not only images or document files, but configuration system files, firewall rules, passwords. The integrity and security of these is a major field of research, especially when in many infrastructures, someone can find sensitive information for the design and management of whole datacenters in a single central computing system or cleartext files. Therefore it is worth asking, how much value does a decentralized approach has in the sharing of any information, especially if the integrity and reliability of the latest can be ensured.

In this thesis we will study and make a new approach to two very new technologies. The first one is Software Defined Networks (SDNs), and the other is the technology of BlockChain.

SDN in a sentence is a technology that seperates the level of control and data plane in a network and therefore manages to implement networks that their configuration can basically be dynamic and programmable, resulting in increased performance, flexibility, and management.

Blockchain, or a chain of blocks, is a method of storing a list of entries that after creation, it is very difficult to unlikely to change, and this is basically done by fragmentation.

In this theses, we combine two core concepts of the above technologies which are opposite. For SDNs, the there is the idea of intelligence of centralization, which in fact is the main drawback regarding their security. In contrast, for BlockChain, decentralization via distributed ledgers, the core concept, provides credibility of any data which is written in the ledgers, without any central authority.

Keywords

Security, trusting identity. logging system, digital signatures, blockchains, ledgers, software defined networks. centralized and decentralized systems, Peer to peer network.

Ευχαριστίες

Η παρούσα διπλωματική εργασία εκπονήθηκε στον τομέα Επικοινωνιών, Ηλεκτρονικής και Συστημάτων Πληροφορικής, στο εργασηριο Διαχείρισης Δικτύων και Βέλτιστου Σχεδιασμού (Netmode) και επισφραγίζει τις σπουδές μου στο Εθνικό Μετσόβιο Πολυτεχνείο. Θα ήθελα να ευχαριστήσω ιδιαιτέρως τον καθηγητή μου κ. Παπαβασιλείου Συμεών για την ευκαιρία που μου έδωσε να αχοληθώ με ένα τόσο ενδιαφέρον και επίκαιρο αντικείμενο καθώς και για την εμπιστοσύνη που μου έδειξε κατά την διάρκεια αυτή. Επίσης, θα ήθελα να ευχαριστήσω τον Δεχουνιώτη Δημήτριο για την εξαιρετική συνεργασία και την πολύτιμη βοήθεια που μου πρόσέφερε καθώς και για την υπομονή και κατανόηση που είχε κατά την προσπάθεια μου ολοκλήρωσης της παρούσας εργασίας. Ακόμα, ένα μεγάλο ευχαριστώ αξίζει ο Παπαδάκης Κωνσταντίνος για τον χρόνο του, την άμεση ανταπόκριση και την τεχνική υποστήριξη που μου προσέφερε όποτε την χρειάστηκα. Τέλος, θα ήθελα να πω ένα ευχαριστώ στην οικογένειά μου για την άπλετη και πολύπλευρη στήριξη και κατανόηση καθόλη την διάρκεια των σπουδών μου, καθώς και τους δικούς μου ανθρώπους και φίλους, που βρίσκονται τώρα σε διάφορες χώρες του πλανήτη κυνηγώντας τα όνειρά τους. Ευχαριστώ Κωνσταντίνε Α., Παναγιώτη Φ., Μαρία Μ. Αριστοτέλη Π., Βασίλη Κ., Κορίνα Τ., Κική Π., Βασίλη Λ., Γιώργο Σ., Αλέξανδρο Α. για την υπομονή, την υποστήριξη, την εμπιστοσύνη που μου δείξατε και την έμπνευση που αποτελέσατε για εμένα, όλο αυτό το διάστημα μέχρι σήμερα. Τέλος θέλω να ευχαριστήσω τους υπόλοιπους ανθρώπους του εργαστηρίου Netmode (Δημήτρης Σ., Γιάννης Δ., Μάριος Α.), οι οποίοι συνέβαλαν σε ένα πολύ φιλικό και παραγωγικό για εμένα κλίμα καθόλη την διάρκεια που συναναστράφηκα μαζί τους.

Περιεχόμενα

Κατάλογος Σχημάτων

Κεφάλαιο 1

Εισαγωγή

1.1 Θεωρητικό Υπόβαθρο

1.1.1 Κεντρικοποιημένα συστήματα

Κεντρικοποιημένο σύστημα είναι ένα σύστημα στο οποίο όλες οι αποφάσεις για έναν συγκεκριμένο στόχο, τον στόχο του συστήματος, καθορίζονται από έναν κεντρικό μηχανισμό και μεταβιβάζονται στα υπόλοιπα επιμέρους στοιχεία του. Όταν πρωτοεμφανίστηκαν τα υπολογιστικά συστήματα, δεν είχαν ξεχωριστά τερματικά, αλλά ενσωματωμένες συσκευές εισόδου εξόδου, και η δρομολόγηση των διεργασιών στον χώρο του πυρήνα γινόταν με μία υλοποίηση τύπου pipeline. Όπως για παράδειγμα για να εισέλθει στον χώρο του πυρήνα η διεργασία B, πρέπει να έχει τελειώσει πρώτα η διεργασία A. Όμως για λόγους κόστους και λόγω έλλειψης υπολογιστικών πόρων, σιγά σιγά τα πρώτα υπολογιστικά συστήματα άρχισαν να υποστηρίζουν πολλούς χρήστες ταυτόχρονα οι οποίοι με διάφορες τεχνικές και αλγορίθμους δρομολόγησης διεργασιών (παραδείγματος χάριν Round Robin), μπορούσαν πρακτικά να εκτελέσουν διεργασίες σχεδόν ταυτόχρονα.

Έτσι γεννήθηκε η ιδέα των κεντρικοποιημένων υπολογιστικών συστημάτων η οποία κυριάρχησε για αρκετό καιρό στην σχεδίαση και ανάπτυξη εφαρμογών για τους μετέπειτα υπολογιστές. Ουσιαστικά, λόγω της ανάπτυξης, της υλοποίησης αλλά και της κατασκευής των πρώτων υπολογιστικών συστημάτων, που ήταν αρκετά πιο γρήγορη σε ρυθμό, από αυτή των αρχικών δικτύων για την μεταξύ τους διασύνδεση, υπήρχε μία άμεση ανάγκη το κάθε ένα υπολογιστικό μηχάνημα να μπορεί να χρησιμοποιηθεί είτε ταυτόχρονα είτε μεμονωμένα, από πολλούς διαφορετικούς χρήστες. Όλες οι διεργασίες ή εφαρμογές έτρεχαν στο ίδιο σύστημα και έτσι ο εκάστοτε χρήστης μπορούσε να το χρησιμοποιήσει κατά το δοκούν. Επίσης η γεωγραφική κατανομή των πρώτων συστημάτων αλλά και η έλλειψη δικτυακών πρωτοκόλλων για την διασύνδεσή τους, οδήγησαν σε αυτό που αποκαλείται κεντρικοποιημένη υπηρεσία ή κεντρικοποιημένο σύστημα.

Με την άνοδο του διαδικτύου την δεκαετία του 1980, και την εξάπλωσή του πέρα από τα εταιρικά ή ερευνητικά μικρής κλίμακας δίκτυα, η ιδέα για την απομάκρυνση από ένα κεντρικοποιημένο υπολογιστικό μοντέλο όπως αυτό του server / client ήταν μία μελλοντική προοπτική.

Μέχρι και σήμερα, τα περισσότερα διαδικτυακά συστήματα υπηρεσιών ακολουθούν αυτό το μοντέλο για την εξυπηρέτηση πελατών τους αλλά και την διασύνδεση με τα επιμέρους στοιχεία τους.

Ορισμός: Κεντρικοποιημένο υπολογιστικό σύστημα είναι μία αρχιτεκτονική η οποία έχει ως βασική ιδέα ότι όλη, ή σχεδόν όλη, η δρομολόγηση υπολογιστικών διεργασιών πραγματοποιείται σε έναν κεντρικό υπολογιστή, ο οποίος με την σειρά του είναι υπεύθυνος να παρέχει την υπολογιστική του ισχύ και τους πόρους του, στα υπόλοιπα στοιχεία-υπολογιστές που είναι συνδεδεμένοι με αυτόν και θέλουν να εξυπηρετηθούν. Ο κεντρικός υπολογιστής ονομάζεται εξυπηρετητής και οι υπόλοιποι πελάτες του.

Η παραπάνω αρχιτεκτονική είναι εύκολα υλοποιήσιμη για μία υπηρεσία, όταν υπάρχει ένα αρκετά έμπιστο δίκτυο που εξασφαλίζει την συνδεσιμότητα του server και του client. Όμως έχει πολλά μειονεκτήματα όταν μιλάμε για υπηρεσίες που έχουν για παράδειγμα εκατομμύρια χρηστών, για τους οποίους δεν υπάρχει γνώση της χρονικής στιγμής ή της διάρκειας για την οποία θέλουν να συνδεθούν, αλλά και αρκετές επιμέρους παραμέτρους. Ακόμα πιο μεγάλο πρόβλημα με την παραπάνω αρχιτεκτονική υπάρχει όταν αυτή πρέπει να υλοποιηθεί για υπηρεσίες οι οποίες πρέπει να είναι διαθέσιμες (high availability), όπως οι διαδικτυακές υπηρεσίες τραπεζικών συστημάτων, ή συστήματα μετάδοσης δεδομένων σε πραγματικό χρόνο όπως το πρόγραμμα EFFIS [1] στην Ελλάδα.

Στο παραπάνω πρόβλημα, υπάρχουν δύο λογικές λύσεις. Η πρώτη είναι η αύξηση των πόρων του κεντρικού υπολογιστικού συστήματος, κομμάτι της τεχνολογίας που ασχολείται με τους γνωστούς super-computers (όπως το σύστημα ARIS [2]), στο οποίο βέβαια τεχνολογικά υπάρχει και κάποιο όριο. Η δεύτερη είναι η σύνδεση πολλών υπολογιστικών πόρων μεταξύ τους, υλοποιώντας ένα δίκτυο μηχανημάτων, τα οποία φυσικά έχουν έναν κοινό σκοπό. Ο σκοπός αυτός είναι ο σκοπός του συστήματος και μπορεί να ποικίλλει αναλόγως την περίσταση και τους στόχους που επιθυμεί να επιτύχει η εκάστοτε υπηρεσία. Προφανώς, όλο αυτό, εγείρει άλλα προβλήματα και παράγοντες, όπως ο διαμοιρασμός των πόρων ή της πληροφορίας σε ένα δίκτυο ή την δρομολόγηση μέσα σε αυτό και κατά πόσο η τελευταία είναι αξιόπιστη και συνεπής. Βέβαια, αυτό δεν είναι το αντικείμενο της παρούσας εργασίας και τέτοιου είδους προβλήματα είναι αντικείμενο μελέτης για την επιστήμη των κατανεμημένων και αποκεντρωμένων συστημάτων τα οποία αναφέρουμε στην συνέχεια.

1.1.2 Κατανεμημένα συστήματα

Περίπου την ίδια εποχή με την εμφάνιση της ιδέας του κατακερματισμού στην κρυπτογραφία, στην οποία θα αναφερθούμε σε επόμενη ενότητα, άρχισε να εμφανίζεται παράλληλα και με την άνοδο της χρήσης των Ethernet δικτύων και μία νέα προσέγγιση στον διαμοιρασμό πόρων στα υπολογιστικά συστήματα. Οι διεργασίες που μέχρι τότε δρομολογούνταν όπως αναφέραμε προηγουμένως με το γνωστό pipelining σε ένα κεντρικοποιημένο σύστημα, με την άνοδο των πλέον παραδοσιακών δικτύων, άρχισαν να διαμοιράζονται και σε διαφορετικά συστήματα υπολογιστικών πόρων (clusters). Αρχικά τέτοιοι πόροι ανήκαν στο ίδιο δίκτυο, για παράδειγμα το τοπικό δίκτυο (LAN). Αργότερα και με την εξάπλωση των δικτυακών πρωτοκόλλων και την πιο ευρεία χρήση τους, ο διαμοιρασμός πληροφορίας γινόταν μεταξύ φαινομενικά ανεξάρτητων πόρων, ίσως με διαφορετικές αρχιτεκτονικές, οι οποίοι ανήκαν όχι μόνο σε διαφορετικά υποδίκτυα αλλά ήταν και φυσικά απομακρυσμένοι.

Ορισμός: Κατανεμημένο είναι ένα σύστημα του οποίου τα επιμέρους συστατικά στοιχεία, είναι διαμοιρασμένα σε διαφορετικούς, αλλά δικτυωμένους μεταξύ τους, υπολογιστές. Οι υπολογιστές αυτοί μεταφέρουν και συντονίζουν τις ενέργειές τους, μεταφέροντας μηνύματα μεταξύ τους (message passing) με σκοπό την επίτευξη ενός κοινού στόχου.

Τρία βασικά χαρακτηριστικά στοιχεία ενός κατανεμημένου υπολογιστικού συστήματος είναι τα εξής κατώθι. Αρχικά η συνάφεια των επιμέρους στοιχείων του, δηλαδή των μηχανημάτων που το απαρτίζουν και των μεθόδων και των πρωτοκόλλων που χρησιμοποιούν για να επικοινωνήσουν. Δεύτερον η έλλειψη ενός παγκόσμιου ρολογιού. Τελευταίο και πιο σημαντικό χαρακτηριστικό, το οποίο κάνει και την ειδοποιό διαφορά σε σχέση με ένα κεντρικοποιημένο σύστημα, είναι η ανεξάρτητη αποτυχία των συνιστωσών του. Αυτό πρακτικά σημαίνει πως σε περίπτωση που ένα ή πολλά επιμέρους τμήματα του δικτύου αποτύχουν για οποιονδήποτε λόγο, το σύστημα παραμένει λειτουργικό εξυπηρετώντας την αρχική λειτουργία του. Το τελευταίο αυτό χαρακτηριστικό καθιστά τα κατανεμημένα συστήματα άτρωτα σε αποτυχίες ενός μόνο σημείου (single point of failure), παράγοντας που αποτελεί μεγάλο πλεονέκτημα σε σχέση με τις προαναφερθέντες κεντρικοποιημένες υπηρεσίες.

Ορισμένα παραδείγματα και τύποι κατανεμημένων συστημάτων είναι για παράδειγμα τα συστήματα βασισμένα σε SOA (Service Oriented Architecture), όπου διαφορετικά κομμάτια ενός λογισμικού συνεργάζονται διαμοιρασμένα πάνω σε ένα δίκτυο για να προσφέρουν μία υπηρεσία. Άλλο σημαντικό και ευρέως γνωστό παράδειγμα είναι τα διομότιμα δίκτυα (peer-to-peer networks - P2P). Ίσως για τους περισσότερους, τα παραπάνω είναι γνωστά από τα μεταδεδομένα των αρχείων που διαμοιράζονται σε τέτοιου είδους δίκτυα, τα γνωστά αρχεία torrent. Αξίζει να αναφέρουμε ότι τα πρώτα διομότιμα δίκτυα που δημιουργήθηκαν και είχαν ως αρχικό σκοπό την ανταλλαγή μουσικών αρχείων μεταξύ κάθε ομότιμου μέλους, ήταν όχι περισσότερο από είκοσι χρόνια πρίν. Μέχρι και σήμερα η ίδια λογική χρησιμοποιείται για τον διαμοιρασμό πληροφορίας και αρχείων στο διαδίκτυο.

Οι διαφορές μεταξύ των αποκεντρωμένων (decentalized) συστημάτων με τα κατανεμημένα (distributed) συστήματα, όπως είναι τα P2P δίκτυα, είναι λεπτές. Στην ουσία ένα αποκεντρω-

μένο σύστημα είναι απλά ένα υποσύνολο ενός κατανεμημένου [3]. Οι βασικές διαφορές μεταξύ τους είναι ουσιαστικά δύο. Η πρώτη αφορά τον τρόπο με τον οποίο λαμβάνεται ομόφωνα μία απόφαση μεταξύ των κόμβων του δικτύου (θα εξηγηθεί στην πορεία τι ακριβώς σημαίνει απόφαση και γιατί μία απόφαση είναι απαραίτητη). Η δεύτερη είναι ο τρόπος με τον οποίο διαμοιράζεται η πληροφορία στο δίκτυο. Σε ένα αποκεντρωμένο σύστημα, δεν υπάρχει ένα ακριβές σημείο-κόμβος από τον οποίο λαμβάνεται μία απόφαση κάθε φορά. Κάθε κόμβος λαμβάνει μία απόφαση και το τελικό αποτέλεσμα είναι το σύνολο των αποφάσεων, οι οποίες συγκεντρώνονται και παράγουν κάτι τελικώς ορισμένο, όπως για παράδειγμα η κοινή γνώμη. Από την άλλη σε ένα κατανεμημένο σύστημα η επεξεργασία των δεδομένων κατανέμεται στις οντότητες του δικτύου αλλά οι τελικές αποφάσεις και τα αποτελέσματα μπορεί να είναι κεντρικοποιημένα και να εκτιμώνται εν τέλει, με βάση όλη την εικόνα του δικτύου και με βάση τα αποτελέσματα των υπολογισμών όλων των κόμβων. Κύριο χαρακτηριστικό των αποκεντρωμένων συστημάτων, είναι το γεγονός ότι τυπικά, κανένας κόμβος από μόνος του δεν μπορεί να έχει πλήρη γνώση όλης της πληροφορίας του συστήματος. Σε αντίθεση με τα κατανεμημένα, τα οποία μπορούν παρόλη την κατανομή της πληροφορίας και των υπολογιστικών πόρων σε διαφορετικές οντότητες, αυτές να έχουν πλήρη γνώση για όλο το σύστημα.

1.1.3 Εμπιστοσύνη στα υπολογιστικά δίκτυα και το διαδίκτυο

Η αλληλεπίδραση ενός ατόμου με ένα άλλο, ή με έναν οργανισμό και το αντίθετο, είναι μία φράση που μπορεί να περιγράψει αρκετές καθημερινές οικονομικές ή μη συναλλαγές, που λαμβάνουν χώρα σε μία κοινωνία ή ένα δίκτυο ατόμων. Για παράδειγμα μία παραγγελία ενός προϊόντος, στην γνωστή πλατφόρμα του amazon, συμπληρώνοντας μόνο τα στοιχεία πελάτη, στα πεδία μίας ηλεκτρονικής φόρμας, έχοντας πολλές φορές ως μόνο κριτήριο μία περιγραφή του προϊόντος, εικόνες αυτού και ίσως κριτικές από αντίστοιχους καταναλωτές που το αγόρασαν προηγουμένως. Η αντίστοιχα παραγγελίες για τις οποίες δεν υπάρχει οποιαδήποτε γνώση του δικτύου και του μηχανισμού που πιστοποιεί ότι η αγορά του προϊόντος όντως πραγματοποιήθηκε. Χωρίς γνώση του τρόπου με τον οποίον η αγοραπωλησία υλοποιήθηκε και τα χρήματα όντως από έναν λογαριασμό σε έναν άλλον. Τα παραπάνω είναι ο τρόπος με τον οποίο λειτουργούν και αγοράζουν αγαθά μεγάλο ποσοστό ανθρώπων σήμερα.

Ταυτόχρονα, όταν η τεχνολογία έχει τόσο στενές σχέσεις με την ανθρώπινη καθημερινότητα, όταν μεγάλο ποσοστό από οικογένειες και επιχειρήσεις εμπιστεύονται ηλεκτρονικά συστήματα, είναι επιτακτική ανάγκη η ασφάλεια, η πιστοποίηση αλλά πάνω από όλα η εμπιστοσύνη στα δικτυακά αυτά συστήματα. Ειδικά όταν το μεγαλύτερο ποσοστό των χρηστών κάθε δικτυακής υπηρεσίας έχει απόλυτη άγνοια για τον τρόπο που λειτουργεί, για παράδειγμα, η ταυτοποίηση τους στο σύστημα, η πίστωση χρημάτων σε έναν τραπεζικό λογαριασμό ή η εγγραφή τους σε μία διαδικτυακή πλατφόρμα, αυτή η ανάγκη γίνεται ακόμα πιο κρίσιμη και άμεση, ιδίως για όποιον προσφέρει μία τέτοια υπηρεσία.

Τίθεται λοιπόν ένα ερώτημα εμπιστοσύνης στις σχέσεις μεταξύ οντοτήτων ενός δικτύου. Είναι επιτακτικός, για όλα τα παραπάνω και για πλήθος ακόμα αντίστοιχων παραδειγμάτων, ένας μηχανισμός ο οποίος μπορεί να εξασφαλίσει ότι μέσα σε ορισμένα πλαίσια, μία οντότητα A,

μπορεί να εμπιστεύεται μία δεύτερη εξουσιοδοτημένη οντότητα B, η οποία λειτουργέι ως αρχή πιστοποίησης, δημιουργώντας επί της ουσίας ένα δίκτυο μελών εμπιστοσύνης (trusted party), το οποίο βασίζεται σε αυτήν [4]. Ο σκοπός είναι η εξασφάλιση πιστοποίησης και η ασφάλεια στο δίκτυο αυτό, όπως ακριβώς ένας πελάτης στο online κατάστημα μιας τράπεζας μπορεί να εμπιστευθεί τις διαδικτυακές του συναλλαγές στην τράπεζα. Αυτό γίνεται εφικτό με τα λεγόμενα ψηφιακά πιστοποιητικά (digital certificate) και τις αρχές πιστοποίησης (certificate authorities), τα οποία λειτουργούν όπως ακριβώς η λειτουργεί η αστυνομική ταυτότητα ή το διαβατήριο στην ζωή εκτός διαδικτύου. Θα αναφερθούμε εκτενώς σε αυτά σε επόμενη ενότητα.

1.1.4 Κρυπτογραφημένες & Κατανεμημένες Αλυσίδες Κοινοποιήσεων

Οι ηλεκτρονικές συναλλαγές που λαμβάνουν μέρος σε ένα δίκτυο χρηστών όπως το διαδίκτυο, είναι πλέον ένα από τα οφέλη και επιτεύγματα της τεχνολογίας όπως τη γνωρίζουμε σήμερα. Το ηλεκτρονικό υποκατάστημα μίας τράπεζας, από το οποίο μπορεί κάποιος να πληρώσει τον λογαριασμό του ρεύματος ή της τηλεφωνικής του σύνδεσης. Οι διάφοροι διαδικτυακοί μηχανισμοί μέσω των οποίων είναι δυνατές οικονομικές συναλλαγές κάθε είδους και για οποιοδήποτε προϊόν μπορεί κανείς να φανταστεί. Κάθε αγαθό το οποίο γενικότερα έχει μία χρηματική αξία, μπορεί να αποτελέσει προιόν διαδικτυακής αγοραπωλησίας μέσω ενός μεσάζοντα, όπως ακριβώς μία τράπεζα στον πραγματικό κόσμο. Για να επιτευχθεί όμως το παραπάνω, είναι απαραίτητη προϋπόθεση ο μεσάζοντας, η τράπεζα ή οποιοσδήποτε κατέχει τον ρόλο αυτό, να είναι αξιόπιστος και να μπορεί να πιστοποιήσει ανά πάσα στιγμή, σε κάθε συναλλαγή, ορισμένα πράγματα όπως το είδος του αγαθού που αγοράστηκε, την τιμή του, και ότι πλέον το χρηματικό έναντι έχει όντως μεταφερθεί από το τραπεζικό λογαριασμό του αγοραστή, σε αυτόν του εκάστοτε πωλητή.

Για την υλοποίηση λοιπόν ενός απλού μοντέλου οικονομικών, και μη, συναλλαγών που λαμβάνουν μέρος ηλεκτρονικά, μέσω ενός δικτύου όπως το διαδίκτυο, είναι απαραίτητο αυτό που ονομάζεται, αξιόπιστη τρίτη οντότητα (trusted third party). Ο ρόλος της είναι να λειτουργεί και να διαμεσολαβεί για την πιστοποίηση μεταξύ δύο ή περισσότερων μελών τα οποία επιθυμούν να κάνουν μία συναλλαγή. Με βάση την παραπάνω αναλογία, άρχισαν να δημιουργούνται τα πρώτα ηλεκτρονικά νομίσματα και εικονικό χρήμα.

Όπως όμως και με τις πραγματικές τράπεζες, έτσι και οι προαναφερθείσες αξιόπιστες οντότητες που κατέχουν ρόλο διαμεσολάβησης στις συναλλαγές, κρίνεται απαραίτητο να έχουν μηχανισμούς που τις καθιστούν ασφαλείς ως προς κακόβουλες επιθέσεις από τρίτους. Όπως ακριβώς οι κάμερες και οι φύλακες που εναλλάσσονται με βάρδιες σε μία τραπεζική υπηρεσία, έτσι και σε αυτές τις ηλεκτρονικές αρχές πιστοποίησης (Digital Certificate Authorities), χρειάζεται έξτρα λογισμικό, προφανώς με την βοήθεια της κρυπτογραφίας, που να τις καθιστά όσο το δυνατόν περισσότερο ασφαλείς σε επιθέσεις. Παρόλα αυτά, όσα εργαλεία κρυπτογράφησης και να χρησιμοποιηθούν και όση ασφάλεια και να προσαρτηθεί σε ένα τέτοιο σύστημα, αυτό για την σωστή λειτουργία του και παραμετροποίηση του, θα συνδέεται πάντα με μία κεντρική μονάδα ή οντότητα διαχείρισης. Οποιοσδήποτε και να διαχειρίζεται ένα τέτοιο σύστημα, το ίδιο

το σύστημα θεωρείται ότι πάσχει από ένα κεντρικό σημείο αποτυχίας (single point of failure). Το παραπάνω, είναι μία αδυναμία που δεν μπορεί να αποφευχθεί παρά μόνο εάν αλλάξει δομή το κεντρικοποιημένο αυτό σύστημα.

Την αλλαγή στο παραπάνω πρόβλημα έφερε το 2008 μία δημοσίευση, σχετικά με το πρώτο αποκαλούμενο κατανεμημένο ηλεκτρονικό νόμισμα ή πιο σωστά κρυπτονόμισμα, το Bitcoin [5]. Η δημοσίευση αυτή, πρότεινε με ήδη υπάρχουσες τεχνολογίες και μαθηματικά εργαλεία κρυπτογράφησης, ένα σύστημα που κατέρριψε, την παραπάνω αδυναμία της κεντρικοποιημένης αρχής πιστοποίησης. Βέβαια όλα τα προηγούμενα είχαν ήδη προταθεί και υλοποιηθεί με έτοιμο κώδικα από τον David Chaum, 25 χρόνια πριν, με μία μόνο παράληψη η οποία όμως ήταν σημαντική για την ανάπτυξη αυτής της τεχνολογίας. Η παράληψη του Chaum [5] [6] ήταν η ιδέα ενός μηχανισμού συναίνεσης (consensus). Έτσι με την προσθήκη ενός μηχανισμού συναίνεσης για το P2P δίκτυο, γεννήθηκε το πρώτο κρυπτονόμισμα, δηλαδή νόμισμα που χρησιμοποιεί τεχνικές εφαρμοσμένης κρυπτογράφησης για την υλοποίησή του. Τα κρυπτονομίσματα εισήγαγαν μία νέα έννοια στον τεχνολογικό ορίζοντα και η ιδέα αυτή δεν ήταν άλλη από το ενΒλοςκςηαιν. Σε άμεση μετάφραση, το blockchain είναι μία αλυσίδα ή ακολουθία από blocks, τα οποία ουσιαστικά αποτελούν μία ομαδοποίηση συναλλαγών (οικονομικών ή όχι) που αλληλοσυνδέονται μεταξύ τους και κατανέμονται μεταξύ των χρηστών ενός δικτύου.

Ορισμός: Blockchain ορίζεται ως μία άφθαρτη (immutable) πρακτικά ψηφιακό αρχείο (digital ledger), αρχικά οικονομικών συναλλαγών που πλέον μπορεί να προγραμματιστεί έτσι ώστε να καταγράφει όχι μόνο χρηματοοικονομικές συναλλαγές, αλλά σχεδόν οτιδήποτε έχει κάποια αξία και μπορεί να αναπαρασταθεί σε μορφή ηλεκτρονικής πληροφορίας.

Σε αναλογία με το προηγούμενο παράδειγμα και το ρόλο της τράπεζας ως αρχή πιστοποίησης στις συναλλαγές, το blockchain αποδεσμεύει την κεντρικοποιημένη αρχή ως μεσάζοντα. Αντί αυτής, επιτυγχάνει την διαδικασία της πιστοποίησης των συναλλαγών μέσω ενός αρχείου (ledger) το οποίο διαμοιράζεται, εξ αρχής και σε κάθε συναλλαγή που συμβαίνει στο δίκτυο, μεταξύ όλως των χρηστών του δικτύου. Έτσι η απόφαση αν μία συναλλαγή είναι έγκυρη επιτυγχάνεται μέσω της ομοφωνίας του πλήθους των χρηστών με βάση του τι έχουν καταγράψει προηγουμένως στο ledger τους. Στην τελική του μορφή λειτουργεί ως ένα αρχείο συναλλαγών που ανανεώνεται συνεχώς και διατηρείται πρακτικά αμετάβλητο με την κρυπτογράφησή του, η οποία προηγείται του διαμοιρασμού του στους χρήστες του δικτύου. Η εξάρτηση από μία εξωτερική αρχή ή οντότητα επικύρωσης για την γνησιότητα και την ακεραιότητα των δεδομένων, δεν είναι με αυτό το μοντέλο πλέον απαραίτητη με τον ίδιο τρόπο όπως στα κεντρικοποιημένα συστήματα εμπιστοσύνης. Επίσης το γεγονός ότι στις κατανεμημένες αλυσίδες από blocks, μπορεί να αποθηκευτεί και να παραμείνει αμετάβλητη, οποιαδήποτε πληροφορία επιθυμούμε και όχι μόνο οικονομικές συναλλαγές είναι σημαντικό πλεονέκτημα και καθιστά το blockchain μία τεχνολογία για την ανάπτυξη πληθώρας πρακτικών εφαρμογών.

1.1.5 Δίκτυα Καθοριζόμενα από το Λογισμικό

Τα δίκτυα έχουν καταστεί απολύτως απαραίτητο στοιχείο στις σύγχρονες επιχειρηματικές δραστηριότητες. Όλες οι εταιρίες είτε μικρές είτε μεγάλες έχουν κάποιου είδους δικτυακή υποδομή για τις ανάγκες του οποιουδήποτε έργου επιτελούν και τους στόχους που επιτυγχάνουν. Η δικτυακή αυτή υποδομή μπορεί είτε να είναι εξ αρχής και ανεπτυγμένη εξ ολοκλήρου στις φυσικές εγκαταστάσεις της εταιρίας, είτε βασίζεται σε εγκαταστάσεις υπολογιστικού νέφους (cloud networking), είτε ακόμα μπορεί να είναι κάποιο υβριδικό σύστημα μετάξύ των δύο. Τα δίκτυα παρέχουν τις ζωτικές επικοινωνιακές συνδέσεις τις οποίες οι επιχειρήσεις, χρειάζονται για να εκτελέσουν τις εφαρμογές τους, να παραδώσουν σε τρίτους τις υπηρεσίες τους και να είναι ανταγωνιστικές, επιτελώντας έναν σκοπό βιωσιμότητας. Τα δίκτυα καθοριζόμενα από το λογισμικό (Software Defined Networks - SDN) ή αλλιώς η καθορισμένη δικτύωση λογισμικού, αντιπροσωπεύει έναν εντελώς νέο τρόπο με τον οποίο προσεγγίζονται οι ρυθμίσεις ενός δικτύου, αλλά και ο έλεγχος και η λειτουργία αυτών.

Υπάρχουν πολλές διαφορετικές προσεγγίσεις για τα SDN δίκτυα, καθώς είναι μία νέα, ανοιχτή τεχνολογία που συνεχώς εξελίσσεται, ωριμάζει και νέες λύσεις και ιδέες εισάγονται και παρουσιάζονται. Η καθορισμένη δικτύωση λογισμικού σημαίνει ότι τα δίκτυα ελέγχονται από εφαρμογές λογισμικού και οντότητες που ονομάζονται ελεγκτές (controllers), και όχι από τα παραδοσιακά τερματικά και εντολές διαχείρισης δικτύων, οι οποίες απαιτούσαν μεγάλη επιβάρυνση στο διαχειριστικό κομμάτι με αποτέλεσμα να είναι δύσκολη η διεύθυνση και ο έλεγχος των ρυθμίσεων ενός δικτύου σε μεγάλη κλίμακα. Συνεπώς, από την ανάγκη για έναν αρκετά πιο ευέλικτο τρόπο της εύκολης διαχείρισης των επιχειρησιακών δικτύων, γεννήθηκε ο έλεγχος μέσω λογισμικού και τα SDN δίκτυα. Το παραπάνω είναι εφικτό με τα SDN δίκτυα καθώς με την χρήση τους γίνεται διαχωρισμός του επιπέδου ελέγχου (control plane) της πληροφορίας που επεξεργάζεται ένα δίκτυο, με το επίπεδο δεδομένων (data plane) του δικτύου. Το επίπεδο ελέγχου στην ουσία είναι αυτό που καθορίζει πως θα δρομολογηθεί, θα προωθηθεί και θα επεξεργαστεί η πληροφορία, ενώ το δεύτερο, είναι τα ίδια τα δεδομένα που δρομολογούνται. Έτσι προκύπτουν σημαντικά πλεονεκτήματα, όπως η ευελιξία, μία αρκετά πιο δυναμική ροή της πληροφορίας και η βελτιστοποίηση των πόρων.

Βέβαια όλα τα παραπάνω, συνοδεύονται με ένα σημαντικό και βασικό μειονέκτημα που προκαθορίζει τα SDN δίκτυα εξ ορισμού. Δεδομένου ότι για να υλοποιηθεί ένα τέτοιου είδους δίκτυο, πρέπει να υπάρχουν οι κεντρικές οντότητες των ελεγκτών, οι οποίοι όπως είπαμε διατηρούν μία συνολική εικόνα του δικτύου με σκοπό την καλύτερη κατανομή των πόρων κατά την δρομολόγηση. Αυτό αυτομάτως μεταφράζεται και σε αποτυχίες ενός μόνο σημείου για το σύστημα, τις οποίες αναφέραμε παραπάνω και χαρακτηρίζουν εξ ορισμού ένα κεντρικοποιημένο σύστημα. Τέλος, είναι σημαντικό να αναφέρουμε πως, μέχρι σήμερα δεν υπάρχει παρά ελάχιστη έρευνα για το πως μπορούν οι δικτυώσεις καθορισμένου λογισμικού να αποτρέψουν τέτοιου είδους αποτυχίες.

1.2 Αντικείμενο διπλωματικής

1.2.1 Το πρόβλημα

Όπως αναφέραμε προηγουμένως όσον αφορά την δικτύωση καθορισμένου λογισμικού (SDN), για να μπορεί να λειτουργήσει και να προσφέρει ευελιξία στην διαχείριση μίας δικτυακής υποδομής, βασικό και αναπόσπαστο κομμάτι αποτελούν οι κανόνες δρομολόγησης. Από την άλλη μεριά βέβαια, ο τρόπος με τον οποίο δημιουργείται μία τέτοιου είδους δικτύωση, προασπίζει και εν γένει βασίζεται στην ίδια λογική με αυτή των κεντρικοποιημένων συστημάτων. Οι βασικές οντότητες του SDN δικτύου, οι ελεγκτές (controllers), λειτουργούν ως μία κεντρικοποιημένη αρχή στο δίκτυο. Όπως προαναφέραμε οι ελεγκτές ορίζουν κανόνες δρομολόγησης με βάση τις ανάγκες του δικτύου. Τίθενται λοιπόν ερωτήματα σχετικά με το κατά πόσο είναι ασφαλής μία τέτοιου είδους δικτύωση. Για παράδειγμα σε ένα σενάριο όπου το λογισμικό του ελεγκτή μπορεί να έχει μία δυσλειτουργία ή ανωμαλία και δεν μπορεί να επιτελέσει τον απαραίτητο ρόλο του στο δίκτυο. Επίσης αν τα κανάλια επικοινωνίας του ελεγκτή με τους δρομολογητές έχουν για οποιοδήποτε λόγο καταστραφεί και για να αποκατασταθεί επικοινωνία πρέπει οι υπηρεσίες να επανεκκινηθούν, πράγμα που μπορεί να σημαίνει την απώλεια των κανόνων δρομολόγησης. Ακόμα και στην ακραία περίπτωση όπου μία κακόβουλη οντότητα όπως ένας διαχειριστής, έχει πάρει τον έλεγχο του SDN ελεγκτή με σκοπό την εκμετάλλευση του δικτύου ή μέρους του δικτύου το οποίο αυτός ελέγχει. Σε όλες αυτές τις περιπτώσεις, θα ήταν εύλογο και ουσιώδες να υπάρχει ένας μηχανισμός ο οποίος στην περίπτωση μίας ασυνέχειας στην διαχείριση του δικτύου, είτε λόγω διακοπής, είτε για οποιονδήποτε λόγο, θα μπορεί να επαναφέρει μία προηγούμενη κατάσταση στον ελεγκτή και κατ'επέκταση στους δρομολογητές. Ταυτόχρονα, σημαντικό είναι να υπάρχει η δυνατότητα αμετάβλητων στον χρόνο εγγραφών, έτσι ώστε να υπάρχει ένα ιστορικό από αυτές και κατ' επέκταση μία όσο το δυνατόν πιο σίγουρη συνέχεια στην διαχείριση του δικτύου.

Το πρόβλημα που θα εξετάσουμε στην παρούσα εργασία σχετίζεται με την καταγραφή και την αποθήκευση των κανόνων δρομολόγησης ενός SDN Δικτύου. Όπως προαναφέρθηκε, οι κανόνες δρομολόγησης καθορίζουν το μονοπάτι που ακολουθεί ένα πακέτο στο δίκτυο, ορίζονται από τον ελεγκτή και προωθούνται (propagated) στους εικονικούς δρομολογητές (virtual switches) οι οποίοι τους αποθηκεύουν σε μία απλή βάση ο καθένας. Αν για οποιονδήποτε λόγο η διαφήμιση ή η αποθήκευση αποτύχει ή ακόμα και αν είναι με κακόβουλο σκοπό από μέλος του δικτύου που δεν έχει πιστοποίηση, αυτό υπονομεύει την ακεραιότητα του δικτύου. Αυτό ακριβώς είναι και αυτό που εξετάζουμε στην παρούσα διπλωματική.

1.2.2 Συνεισφορά

Το παραπάνω πρόβλημα έρχεται να λύσει η τεχνολογία των κατανεμημένων εγγραφών. Με την τεχνολογία του BlockChain, είναι δυνατή η καταγραφή πληροφορίας σε ηλεκτρονικά αρχεία, τους λεγόμενους ledgers. Ως αποτέλεσμα, η οποιαδήποτε είδους πληροφορία που καταγράφεται είναι πρακτικά αμετάβλητη, και έτσι δεν μπορεί να αλλοιωθεί από τρίτους. Η

μεταβλητότητα εκφράζει το κατά πόσο μία οντότητα μπορεί να αλλάξει και να τροποποιηθεί σε ένα περιβάλλον. Ειδικά σε ένα περιβάλλον που αλλάζει διαρκώς, όπως οι τοπολογίες δικτύων και οι εγγραφές στους πίνακες των δρομολογητών, η έννοια της μεταβλητότητας, είναι μείζονος σημασίας.

Την αμεταβλητότητα των κατανεμημένων εγγραφών εκμεταλλευόμαστε και στην παρούσα εργασία για να παρουσιάσουμε ένα σενάριο στο οποίο οι εγγραφές που υπάρχουν στις βάσεις των δρομολογητών ενός SDN δικτύου καταγράφονται σε μία κατακερματισμένη αλυσίδα εγγραφών.

Σκοπός του σεναρίου είναι ο συνδυασμός των τεχνολογιών των SDN και του BlockChain, οι οποίες προσεγγίζουν την έννοια της πληροφορίας με έναν φαινομενικά αντίθετο τρόπο. Τα μεν καθοριζόμενα από το λογισμικό δίκτυα υπάρχουν και έχουν σχεδιαστεί με βάση ένα κεντρικοποιημένο μοντέλο πληροφορίας. Αποτελούνται από δρομολογητές, και έναν ή περισσότερους κεντρικούς διαχειριστές, οι οποίοι είναι υπεύθυνοι για τον έλεγχο και την ρύθμιση της πληροφορίας, όπου στην περίπτωσή μας είναι οι κανόνες δρομολόγησης στους συνδεδεμένους με αυτούς δρομολογητές. Από την άλλη, η τεχνολογία των κατανεμημένων εγγραφών αλυσίδας προσφέρει την αποκέντρωση της πληροφορίας σε ένα διομότιμο δίκτυο από κόμβους και την καταγραφή αυτής σε κατακερματισμένες και πιστοποιημένες εγγραφές που ονομάζονται ledgers. Η πληροφορία για ένα διομότιμο δίκτυο κατανεμημένων εγγραφών, περιλαμβάνει οτιδήποτε μπορεί να καταγραφεί στο ledger για κάθε χρήστη δικτύου αυτού.

Στην παρούσα εργασία επομένως, και για τις δύο παραπάνω περιπτώσεις, θεωρούμε ως πληροφορία τους κανόνες δρομολόγησης, οι οποίοι καταγράφονται στο μπλοκ αλυσίδας των κατανεμημένων εγγραφών, δηλαδή τους ledgers του διομότιμου δικτύου που αποτελεί το BlockChain.

Συνδυάσαμε λοιπόν ένα blockchain δίκτυο το οποίο κατασκευάσαμε βασιζόμενοι στο Hyperledger Fabric και ένα SDN δίκτυο το οποίο βασίζεται στον προσομοιωτή mininet και στο πρωτόκολλο OpenFlow. Το αποτέλεσμα του συνδυασμού αυτού είναι το γεγονός πως οι κανόνες δρομολόγησης του τελευταίου, μπορούν αρχικά να καταγράφονται στους ledgers ενός διομότιμου δικτύου. Αυτό έχει ως αποτέλεσμα, οι κανόνες αυτοί να έιναι άμεσα διαθέσιμοι ανά πάσα στιγμή καθώς επίσης χαρακτηριστικό τους είναι η αμεταβλητότητα τους, ως προς κακόβουλες τρίτες οντότητες, όπως θα δούμε στα επόμενα κεφάλαια.

1.3 Οργάνωση Κειμένου

Η εργασία που υλοποιήθηκε στα κεφάλαια που ακολουθούν οργανώθηκε με βάση τις παρακάτω θεματικές περιοχές. Αρχικά τίθενται τα θεμέλια για την καλύτερη ανάλυση και κατανόηση των εννοιών της παρούσας διπλωματικής οριοθετώντας το θεωρητικό υπόβαθρο. Συγκεκριμένα, γίνεται μία εισαγωγή για τα κεντρικοποιημένα αλλά και τα κατανεμημένα συστήματα. Περιγράφεται η σημασία εμπιστοσύνης στα δίκτυα καθώς και αναφέρονται οι έννοιες του blockchain και του SDN. Στην επόμενη υποενότητα περιγράφεται συνοπτικά το πρόβλημα και η συνεισφορά της παρσούσας εργασίας.

Στο δεύτερο κεφάλαιο συγκεντρώνονται συγγενικές εργασίες που βρίσκονται θεματικά στην ίδια περιοχή ενδιαφέροντος. Καταγράφονται οι ήδη υπάρχουσες εργασίες για την συγκεκριμένη θεματική περιοχή οι οποίες αφορούν τόσο το blockchain, τα SDN δίκτυα αλλά και συνδυασμό αυτών των δύο.

Στην συνέχεια περιγράφεται και αναλύεται η αρχιτεκτονική και τα δομικά στοιχεία ενός SDN δικτύου, όπως αυτό χρησιμοποιείται στο επόμενο κεφάλαιο για την υλοποίηση του σεναρίου μας. Επίσης αναλύονται η βασική δομή και τα στοιχεία ενός blockchain δικτύου που βασίζεται στο Hyperledger Fabric, το οποίο επίσης χρησιμοποιείται στην πορεία αλλά και οι διαφορετικοί τύποι στους οποίους μπορεί να υπάγεται μία δομή blockchain.

Στο τέταρτο κεφάλαιο γίνεται αναφορά των εργαλείων που χρησιμοποιούνται καθώς επίσης και γίνεται η σύνδεση του SDN & του blockchain δικτύου που υλοποιούνται στην συγκεκριμένη εργασία.

Έπειτα παρουσιάζεται αναλυτικότερα το σενάριο της διπλωματικής, καθώς επίσης αναλύονται οι εφαρμογές του μέσω μελέτης της πειραματικής υποδομής που υλοποιείται.

Τέλος τα δύο τελευταία κεφάλαια που ακουλουθούν αφορούν την αξιολόγηση, την συνεισφορά, καθώς και τα συμπεράσματα της παρούσας εργασίας, χωρίς να λείπουν βέβαια οι μελλοντικές επεκτάσεις.

Κεφάλαιο 2

Συγγενικές Εργασίες και Θεματικές Περιοχές

2.1 Εισαγωγή

Οι θεματικές περιοχές στις οποίες εουμε ανακαλύψει συγγενικές εργασίες, αφορούν τις τεχνολογίες του Blockchain και των SDN δικτύων, τα οποία άλλωστε είναι και οι βασικές θεματικές περιοχές της παρούσας εργασίας. Επίσης για την παρούσα εργασία, αναφορές γίνονται σε εργασίες από συγγενικές θεματικές περιοχές όπως τα διομότιμα δίκτυα, τα οποία αποτελούν τη βασική δομή και ιδέα μίας υλοποίησης blockchain.

Σημαντική επιπλέον, είναι η αναφορά σε εργασίες που αφορούν και σχετίζονται με την ανεκτικότητα σφαλμάτων σε διομότιμα δίκτυα καθώς επίσης και πως οι υλοποιήσεις blockchain αρχιτεκτονικών, καταλλήγουν σε αυτό που ονομάζεται και ορίζεται ως αποφάσεις ομοφωνίας δικτύου. Επιπρόσθετα συνοπτικά μελετάται και η διαχείριση εμπιστοσύνης σε SDN δίκτυα, καθώς άλλωστε σκοπός της παρούσας εργασίας είναι η μελέτη και η πρόταση ενός πιο ασφαλούς διαχειριστικού μοντέλου που να συνδυάζει το blockchain με τα SDN δίκτυα. Τέλος, αναφορά γίνεται σε εργασίες που εξετάζουν το βασικό αλλά και πρόσφατο πρωτόκολλο διαχείρισης και λειτουργίας των SDN αρχιτεκτονικών, το οποίο ονομάζεται OpenFlow, καθώς αυτό αποτελεί τόσο κομμάτι της υλοποίησής μας, αλλά και τον πυρήνα μιας SDN αρχιτεκτονικής.

2.2 Διαχείριση Εμπιστοσύνης στα Δίκτυα

Η διαχείριση εμπιστοσύνης και η διαχείριση φήμης στο διαδίκτυο και στα ομοσπονδιακά δίκτυα είναι ένα ανοιχτό ερευνητικό πρόβλημα. Με την ανάπτυξη των διαδικτυακών συστημάτων και τις εφαρμογές IoT, η εμπιστοσύνη σε τέτοιου είδους συστήματα είναι ένα ζήτημα που κλιμακώνεται διαρκώς. Ο τρόπος με τον οποίο διάφορες εφαρμογές και τεχνολογίες χειρίζονται την έννοια της εμπιστοσύνης, συνεχώς χρειάζεται νέες τροποποιήσεις. Με την καθιέρωση των τεχνολογιών 5G και του Internet of Things, μεγάλο κομμάτι της επιστημονικής κοινότητας έχει επικεντρωθεί στις τεχνολογίες Network Function Virtualization (NFV) και στα Software Defined Networks (SDN). Η εμπιστοσύνη σε τέτοιου είδους δίκτυα είναι

ένα σημαντικό πρόβλημα και αποτελεί μία ανοιχτή πρόκληση στον τεχνολογικό τομέα των δικτύων νέας γενιάς [34]. Για παράδειγμα σε ένα μοντέλο μίας έξυπνης πόλης, δεδομένα που παράγονται από πολλαπλές πηγές, όπως αισθητήρες, κινητά τηλέφωνα, RFIDs και άλλα, χρειάζονται αυτόνομα επαληθεύσιμες ταυτότητες για την αυτο-επαλήθευση και την καθιέρωση εμπιστοσύνης στα δεδομένα που έχουν συλλεχθεί [7].

Στην εργασία σχετικά με την διαχείριση φήμης σε υποδομες υπολογιστικού νέφους [8], προτείνεται ένα σύστημα διαχείρισης εμπιστοσύνης για υποδομές υπολογιστικού νέφους (cloud computing) με σκοπό την επιλογή παρόχου και υποδομών από μελλοντικούς χρήστες. Το σύστημα διαχείρισης φήμης και εμπιστοσύνης επεξεργάζεται σε αντικειμενικές δικτυακές μετρήσεις και υποκειμενικές αξιολογήσεις χρηστών με τη χρήση της μεθοδολογίας Fuzzy Analytical Hierarchical Process (AHP). Για την διόρθωση κακόβουλων αξιολογήσεων, ένας μηχανισμός αξιοπιστίας συγκρίνει την βαθμολογία των χρηστών με SLA δεδομένα.

Επίσης στην σχετική εργασία για την ασαφή διαχείριση πολλαπλών κριτηρίων βασιζόμενη στην εμπιστοσύνη, για το μέλλον στο Διαδίκτυο [9], οι συγγραφείς προτείνουν μια μέθοδο διαχείρισης της εμπιστοσύνης σε ομόσπονδες ετερογενείς πειραματικές υποδομές (federated testbeds). Η εξαγωγη της φήμης μιας υποδομής στηρίζεται σε αντικειμενικές δικτυακές μετρήσεις και σε υποκειμενικές μετρήσεις της ποιότητας της υπηρεσίας από τους χρήστες. Η επεξεργασία των δεδομένων γίνεται με βάση τη μεθοδολογία Fuzzy VIKOR. Για την μείωση της επίδρασης μεροληπτικών χρηστών, ένας αλγόριθμος μέτρησεις της αξιοπιστίας του χρήστη έχει αναπτυχθεί και βασίζεται στην διαφορά μεταξύ αντικειμενικών μετρήσεων των δικτυακών μεγεθών και της βαθμολογίας του χρήστη.

Ορισμός: Η εμπιστοσύνη ορίζεται ως η υποκειμενική αντίληψη της οντότητας A, ότι η οντότητα B εκτελεί μία ορισμένη λειτουργία ή πράξη [10]. Αυτό σημαίνει ότι η εμπιστοσύνη είναι εξ ορισμού, προσωπική άποψη και αλλάζει αναλόγως την εκάστοτε οντότητα και το εκάστοτε σύστημα το οποίο την διαχειρίζεται.

Σχετικά με την διαχείριση της εμπιστοσύνης, υπάρχουν αρκετά πρότυπα που έχουν οριστεί από οργανισμούς και ινστιτούτα όπως το European Telecommunications Standards Institute [11]. Το ινστιτούτο ETSI συνιστά την χρήση πρωτοκόλλων SSL/TLS και πιστοποιητικών (certificates) για την εγκατάσταση ασφαλούς επικοινωνίας και την ανάπτυξη εμπιστοσύνης σε δικτυακές οντότητες. Όμως τέτοιου είδους πρότυπα και κατευθυντήριες γραμμές ορίζονται και λειτουργούν κατά βάση, για τα καθιερωμένα μοντέλα κεντρικοποιημένων συστημάτων και απαιτούν την προϋπάρχουσα εμπιστοσύνη σε συγκεκριμένη, πιστοποιημένη οντότητα.

Μία ενδιαφέρουσα κατηγορία συστημάτων εμπιστοσύνης είναι όσα υλοποιούνται στα πλαίσια διομότιμων (P2P) δικτύων όπου οι χρήστες αλληλεπιδρούν μεταξύ τους συνήθως χωρίς προκαθορισμένο τρόπο. Για παράδειγμα, ο αλγόριθμος EigenTrust [12], σχεδιάστηκε για να προστατέψει τους χρήστες των p2p δικτύων από το να κατεβάζουν κακόβουλα ή μη αυθεντικά αρχεία που διακινούνται στο δίκτυο. Αυτό γίνεται με τον υπολογισμό και την απόδοση μίας παγκόσμιας τιμής εμπιστοσύνης σε κάθε διομότιμο χρήστη του δικτύου, χρησιμοποιώντας μία αναδρομική μέθοδο και τις απόψεις των άλλων χρηστών. Σημαντικό είναι ότι, ο αλγόριθμος

αυτός είναι βασισμένος στην επανάληψη ισχύος και μπορεί να υλοποιηθεί τόσο σε μία κεντρική αλλά και σε μία κατανεμημένη υπηρεσία.

Επιπλέον ο μηχανισμός Reputation, Opinion, Credibility and Quality (ROCQ) [13] πρότεινε ένα σύστημα διαχείρισης εμπιστοσύνης βασισμένο στη φήμη των οντοτήτων, που παρήγαγε τιμές φήμης για να αντιπροσωπεύει την αξιοπιστία των χρηστών σε δίκτυα P2P. Η αξιολόγηση των χρηστών παρέχεται μετά το τέλος κάθε συναλλαγής μεταξύ τους. Το σύστημα ROCQ θα μπορούσε να υλοποιηθεί με κατανεμημένο τρόπο και η αξία της φήμης βασίστηκε στη γνώμη του χρήστη, την αξιοπιστία του από τις προηγούμενες εκτιμήσεις του και την ποιότητα που αντιπροσώπευε την εμπιστοσύνη ενός άλλου χρήστη στην ακρίβεια της αξιολόγησής του. Αυξάνεται έτσι η εμπορευσιμότητα στο δίκτυο και κακόβουλες επιθέσεις τρίτων καθίστανται πιο δύσκολες στην υλοποίηση.

Ακόμα στο [14], οι συγγραφείς προτείνουν το Trust List, μία υλοποίηση η οποία διασυνδέει τις τεχνολογίες του blockchain και των SDN, με σκοπό την αποφυγή επιθέσεων σε συσκευές IoT σε Εδγε δίκτυα. Το Trust List, διαχειρίζεται την εμπιστοσύνη μεταξύ IoT οντοτήτων και παρέχει αυτόνομη εφαρμογή στη διαχείριση της κίνησης του Edge δικτύου με τις παραπάνω τεχνολογίες. Με αυτό το μοντέλο κάθε συσκευή IoT, παρέχει το προφίλ της σε ένα blockchain δίκτυο. Το προφίλ κάθε υπηρεσίας (service profile), αποτελείται από ορισμένες πληροφορίες όπως η διεύθυνση IP και η πόρτα και όλα μαζί αποτελούν μία λίστα εμπιστοσύνης. Αφού τα προφίλ ληφθούν και αποκωδικοποιηθούν από SDN controllers, εφαρμόζονται στους εικονικούς δρομολογητές του SDN δικτύου (virtual switches-vs), ως κανόνες δρομολόγησης. Έτσι προκύπτει ένα δίκτυο εμπιστοσύνης μέσω αυτών των κανόνων εφόσον οι πληροφορίες που υπάρχουν σε αυτό το δίκτυο έχουν προέλθει μέσα από ένα blockchain και έχουν γίνει έγκυρες.

Το πρόβλημα της εμπιστοσύνης και της ασφάλειας, είναι μείζονος σημασίας στις σύγχρονες υποδομές δικτύων. Σε αυτό το πρόβλημα για τα διομότιμα δίκτυα, μία σχετικά νέα τεχνολογία έρχεται να δώσει μία διαφορετική λύση στο πρόβλημα της εμπιστοσύνης. Η τεχνολογία του blockchain προσφέρει ένα P2P δίκτυο στο οποίο χωρίς κάποια ενδιάμεση οντότητα εμπιστοσύνης, δικτυακές οντότητες οι οποίες δεν έχουν εμπιστοσύνη η μία με την άλλη, μπορούν να αλληλεπιδρούν με έναν επαληθεύσιμο τρόπο μεταξύ τους [15]. Το blockchain εξ αρχής και με στην πρώτη υλοποίησή του, δηλαδή το bitcoin [5], είχε ως σκοπό να κάνει τις πληρωμές μεταξύ οντοτήτων να απαιτούν λιγότερη εμπιστοσύνη και ταυτόχρονα περισσότερη ασφάλεια, έτσι ώστε να δημιουργήσει ένα πιο ανθεκτικό στις παραβιάσεις σύστημα. Πλέον που η τεχνολογία αυτή μπορεί να επεκταθεί και σε άλλα συστήματα εκτός από τα χρηματοοικονομικά, το ίδιο πλεονέκτημα μεταβιβάζεται και σε άλλα ήδη πληροφορίας πέραν αυτής των εικονικών νομισμάτων. Συγκεκριμένα οι αναφορές οι οποίες ακολουθούν στην ενότητα αυτή, υποδεικνύουν ότι η χρήση του blockchain για δικτυακούς σκοπούς είναι σε πολύ πρώιμο στάδιο αλλά ταυτόχρονα η μελέτη της έχει να προσφέρει πάρα πολλά στον επιστημονικό κόσμο.

2.3 Το Διαδίκτυο των Πραγμάτων – IoT

Το Διαδίκτυο των πραγμάτων (IoT) προσελκύει τον κόσμο της τεχνολογίας ως αναπόσπαστο μέρος για την υλοποίηση έξυπνων συσκευών, έξυπνων κτιρίων, έξυπνων πόλεων και κατά

βάση ενός έξυπνου κόσμου. Αυτός ο έξυπνος κόσμος είναι ο προορισμός της τρέχουσας τάσης της τεχνολογίας και έχει ως στόχο τη δικτυακή σύνδεση των πάντων, με τα πάντα. Ωστόσο, ορισμένα μείζονα προβλήματα όπως ζητήματα ασφάλειας ή έλλειψης επαρκούς επεξεργασίας της πληροφορίας, εμποδίζουν την ελεύθερη και χωρίς φραγμούς ανάπτυξη της ερευνητικής κοινότητας καθώς και των μεγάλων εταιρειών στον χώρο της τεχνολογίας. Ένα από τα βασικότερα αυτά προβλήματα είναι η συναίνεση για την ολοκλήρωση μιας εξαιρετικά ασφαλούς και αποδοτικής αρχιτεκτονικής για έξυπνες πόλεις, όπως περιγράφεται στην αντίστοιχη δημοσίευση με τίτλο SDIoBoT: A Software-Defined Internet of Blockchains of Things Model [16].

Στο παρόν στάδιο τεχνολογικής ανάπτυξης, υπάρχουν ήδη επιστημονικά έγγραφα τα οποία συνδέουν την έννοια του IoT με τις τεχνολογίες SDN και Blockchain [15, 16]. Υπάρχουν επίσης επιστημονικές εργασίες, οι οποίες παρουσιάζουν ως αναπόσπαστο κομμάτι ενός νέου μοντέλου δικτύωσης, την χρήση των SDN δικτύων και των NFV υλοποιήσεων σε εφαρμογές του Internet of Things [17, 18, 19, 20, 21]. Πρόσφατα μάλιστα, έχουν προταθεί μοντέλα τα οποία βελτιώνουν την ασφάλεια των συσκευών IoT μέσω της χρήσης των SDN δικτύων όπως στο [22].

Προκύπτει λοιπόν ως άμεσο συμπέρασμα, η τεχνολογική κατεύθυνση της ακαδημαικής και της επιχειρησιακής κοινότητας στην σύνδεση των πάντων, με τα πάντα. Επομένως είναι σημαντική η μελέτη των τεχνολογιών που μπορούν να διασυνδέσουν το τεράστιο πλήθος συσκευών με μοντέλα τα οποία προωθούν και σέβονται την ασφάλεια της πληροφορίας. Την θέση για αυτές τις τεχνολογίες έρχονται να πάρουν δύο έννοιες. Αρχικά, η έννοια των SDN δικτύων, έτσι ώστε η δικτύωση ολοένα και αυξανόμενου πλήθους συσκευών, να μην αποτελεί πρόβλημα στην επεκτασιμότητα των δικτύων αυτών [23]. Έπειτα, η τεχνολογία του blockchain, η οποία παρέχει μία λογικού είδους ασφάλεια σε δίκτυα στα οποία υπάρχει διαμοιρασμός κρίσιμης πληροφορίας [24, 25].

2.4 Κατανεμημένες Αλυσδες Κοινοποιήσεων - Blockchain

Το blockchain επιτρέπει τη συναίνεση στην αποθήκευση δομών δεδομένων που πληρούνται από μη εμπιστευτικές κατανεμημένες οντότητες. Τα ενωμένα μεταξύ τους blocks που τοποθετούνται στην αλυσίδα, περιέχουν σύνολα υπογεγραμμένων δομών δεδομένων (π.χ. συναλλαγές ή συμβόλαια), συμπεριλαμβανομένων διαφανών τρόπων για την επαλήθευση των εσωτερικών τους πληροφοριών [26].

Σην εργασία [26] οι συγγραφείς θέτουν τα θεμέλια για την ανάπτυξη αποκεντρωμένων εφαρμογων (Decantralized Applications -DApps) αναπτύσσοντας και χρησιμοποιώντας την τεχνολογία του blockchain (bc) για την διαχείριση (administration) και ενορχήστρωση (orchestration) δικτυακών υπηρεσιών πολλών τομέων (Multi-domain) [27, 28]. Το multi administrative networking θεωρείται μια πολλά υποσχόμενη προσέγγιση για την παροχή των επερχόμενων πρωτοποριακών υπηρεσιών 5G που περιλαμβάνουν χρήσεις με ποικίλες επιχειρησιακές ανάγκες και μια σταδιακή αποσύνδεση από το σημερινό μονολιθικό μοντέλο δικτύωσης συσκευών, όπως αυτό του πελάτη – εξυπηρετητή.

Στην εργασία [29], προτείνεται ένα μοντέλο για το τρόπο με τον οποίο διάφορες αρχιτεκτονικές IoT, μπορούν να εκμεταλλευτούν τις τεχνικές που προσφέρει το bc όταν αυτό συνδυάζεται με τα SDN δίκτυα. Με αυτόν τον τρόπο, κατά το [29], μπορούν να αυτοματοποιηθούν διάφορες επιχειρισιακές λειτουργίες, καθώς δεν θα υπάρχει η ανάγκη για κοστοβόρες και πολύπλοκες κεντρικοποιημένες IT υποδομές. Επίσης, με τον τρόπο αυτό, η οικοδόμηση εμπιστοσύνης μεταξύ συσκευών στο δίκτυο θα είναι πιο άμεση και εύκολη, καθώς επίσης μειώνεται το κόστος της υποδομής, ο κίνδυνος πλαστογράφησης ταυτοτήτων του δικτύου, εξαλείφονται οι μεσάζοντες και ελαχιστοποιείται ο χρόνος διακανονισμού των δικτυακών συναλλαγών. Παράλληλα μαζί με τις τεχνολογίες SDN δικτύων, αυξάνεται όπως προαναφέραμε και προηγουμένως η επεκτασιμότητα της υποδομής και σε συνδυασμό με το χαμηλό κόστος οι υλοποιήσεις τέτοιων μοντέλων στον επιχειρησιακό τομέα είναι οικονομικά πιο αποδοτικές αλλά και πιο ασφαλείς.

Επίσης στην εργασία για θέματα και προκλήσεις της χρήσης του blockchain στο IoT [30], παρουσιάζεται μία ολοκληρωμένη έρευνα των υφιστάμενων πρωτοκόλλων blockchain για τα δίκτυα IoT καθώς επίσης γίνεται και αναφορά στα SDN δίκτυα, τα οποία όπως αναφέρεται διευκολύνουν κατά πολύ τέτοιου είδους εφαρμογές. Επιπλέον, παρέχεται μια ταξινόμηση των μοντέλων απειλής, για το blockchain σε μια τέτοιου είδους εφαρμογή, σε πέντε βασικές κατηγορίες:

1) Επιθέσεις με βάση την ταυτότητα 2) Επιθέσεις με βάση τη χειραγώγηση 3) Κρυπτοαναλυτικές επιθέσεις 4) Επιθέσεις με βάση τη φήμη 4) Επιθέσεις με βάση την υπηρεσία.

Στην ίδια εργασία, αναφέρεται ακόμα, πώς τα αποκεντρωμένα ΣΔΝ δίκτυα βασισμένα στο blockchain [15] μπορούν να μετριάσουν τις συνέπειες αλλά και να αντιμετωπίσουν πρόωρα επιθέσεις DdoS/DoS (Distributed Denial of Service).

Αναφορικά ακόμα με την ασφάλεια στα SDN δίκτυα, οι συγγραφείς στο [31] προτείνουν το BSS, (Blockchain Security over SDN) χρησιμεύει στην ενίσχυση της ασφάλειας των διαμοιρασμένων αρχείων στα SDN δίκτυα, αποθηκεύοντας τα σε μία πλατφόρμα OpenStack Cloud, προστατεύοντας έτσι την ιδιωτικότητα και την διαθεσιμότητα των πόρων του δικτύου, κατά των μελών που δεν είναι έμπιστα στο υπόλοιπο δίκτυο. Έτσι στην ουσία, αρχεία που διαμοιράζονται σε ένα SDN δίκτυο, αφού εισέλθουν σε μία δομή βασισμένη στο blockchain και επαληθευθεί η ταυτότητά τους, αποθηκεύονται στο Cloud. Για τα παραπάνω οι συγγραφείς χρησιμοποίησαν την γλώσσα προγραμματισμού Serpent [32], βασισμένη στην γνωστή python και μία από τις αρχικές γλώσσες στον πηγαίο κώδικα του Ethereum Blockchain [33], για τα έξυπνα συμβόλαια (smart contracts), που όμως τώρα δεν χρησιμοποιείται πλέον. Επίσης τα εργαλεία που χρησιμοποιήθηκαν στην εργασία των συγγραφέων και θα δούμε και εμείς στα επόμενα κεφάλαια, στην παρούσα εργασία, είναι ονομαστικά το mininet καθώς και το λογισμικό OpenDayLight.

Μία από τις μεγαλύτερες προκλήσεις σχετικά με τα σημερινά SDN δίκτυα είναι η ενίσχυση της ασφάλειας σε αυτά. Τα αρχεία τα οποία διαμοιράζονται σε τέτοια δίκτυα, μπορούν να προστατευθούν πολύ περισσότερο, με την χρήση της τεχνολογίας του blockchain. Επεκτείνοντας το κρίσιμο πρόβλημα της ασφάλειας των ταυτοτήτων σε ένα δίκτυο και των πληροφοριών που ανταλλάσσονται στο δίκτυο αυτό, ένα σημαντικό πρόβλημα είναι όπως αναφέραμε και παρα-

πάνω οι επιθέσεις DDoS. Παρόλο που η πρώτη επίθεση DDoS, εν ονόματι Trin00, έγινε 20 χρόνια πριν [35], είναι ακόμή καίριο ζήτημα για δίκτυα μεγάλης κλίμακας ανεξαρτήτου εύρους ζώνης, τόσο σε πανεπιστήμια όσο και εταιρείες. Οι απώλειες για τα δίκτυα που επιδέχονται μία DDoS είναι σημαντικές, ειδικά στην σημερινή εποχή όπου οι τεχνολογικές υποδομές και υπηρεσίες τήνουν προς ένα μοντέλο XaaS [36], το οποίο απαιτέι την ύπαρξη και την συνεχή διαθεσιμότητα δικτυακού εύρους. Το XaaS περιγράφει οποιοδήποτε μοντέλο ή λειτουργία (έστω X) που μπορεί να προσφερθεί και να λειτουργήσει ως υπηρεσίας σε τρίτους, δηλαδή as a service).

Με την αρχή που έκανε το βιτςοιν και την ανάπτυξη των blockchain εφαρμογών, σήμερα βλέπουμε αναφορές στην επιστημονική κοινότητα οι οποίες προτείνουν λύσεις σχετικά με τις DDoS επιθέσεις μέσω της χρήσης του blockchain μοντέλου [37]. Τα υπάρχοντα πρωτόκολλα σηματοδοσίας DDoS αποτελούν εμπόδιο για την πλήρη λειτουργία της συντονισμένης και κατανεμημένης υπεράσπισης έναντι των επιθέσεων αυτών. Η τεχνολογία Blockchain προσφέρει μια out-of-the-box λύση, η οποία μειώνει την πολυπλοκότητα της επεξεργασίας της πληροφορίας στις επιθέσεις DDoS. Στο [37] παρουσιάζεται το BloSS, ένα τέτοιο σύστημα σηματοδοσίας μπλοκαρίσματος (Blocking Signaling System). Η αρχιτεκτονική που χρησιμοποιείται για το σύστημα αυτό είναι βασισμένη σε SDN δίκτυο και συγκεκριμένα χρησιμοποιεί Ryu controllers [38] για την παρακολούθηση και την επιβολή κανόνων δρομολόγησης στα Openflow switches.

Εκτός από τις εφαρμογές του blockchain σε περιβάλλοντα SDN δικτύων όπως είδαμε παραπάνω, η ασφάλεια που παρέχει και η εξάλειψη των οντοτήτων εμπιστοσύνης στις εφαρμογές του, μπορεί να αξιοποιηθεί και σε άλλα συστήματα και μηχανισμούς. Χαρακτηριστικό παράδειγμα του παραπάνω είναι η πρόταση για το μοντέλο που παρουσιάζεται στην εργασία [39]. Τα βασικά πλεονεκτήματα μιας τέτοιας προσέγγισης είναι όπως αναφέρουν οι συγγραφείς, η εξάλειψη οποιουδήποτε μηχανισμού εμπιστοσύνης που έχει χαρακτηριστικά PKI, ένα καταγεγραμμένο και εγκυροποιημένο ιστορικό συναλλαγών, εξουσιοδοτήσεις πολλαπλών υπογραφών για βελτιωμένη ασφάλεια, εύκολη επεκτασιμότητα και δυνατότητα προγραμματισμού με scripts για την εξασφάλιση νέων τύπων πόρων Διαδικτύου και δυνατότητες για ενσωματωμένη κρυπτογράφηση. Η υποδομή δημόσιου κλειδιού (PKI), είναι ένας τρόπος παροχής μέτρων ασφαλείας με την εφαρμογή ζευγών κλειδιών μεταξύ των χρηστών ενός δικτύου [40].

Όμως οι εφαρμογές του blockchain και κατά επέκταση και τα πλεονεκτήματα του δεν σταματάνε μόνο εκεί. Όταν γίνονται αναφορές σαν αυτή των Hari, Lakshman, για την χρήση του bc και την εξέλιξη βασικών δομικών στοιχείων της αρχιτεκτονικής του διαδικτύου, έτσι όπως αυτή ορίζεται σήμερα, δομικά στοιχεία όπως το σύστημα DNS (Domain Name System), τότε προκύπτει ως άμεσο συμπέρασμα ότι η τεχνολογία του bc, έρχεται στο προσκήνιο του διαδικτύου για να θέσει νέους όρους για την ασφάλεια και την προστασία των δεδομένων και της πληροφορίας στα δίκτυα. Το παραπάνω σε συνδυασμό με την ολοένα και μεγαλύτερη εξέλιξη των SDN δικτύων, μας φέρνει σε θέση να αναρωτηθούμε πόσο επικερδείς θα ήταν περαιτέρω επιστημονικές έρευνες για την από κοινού μελέτη των δύο αυτών τεχνολογιών και την εφαρμογή τους σε πραγματικές εφαρμογές και επιχειρησιακά περιβάλλοντα παροχής υπηρεσιών.

2.5 Τα SDN Δίκτυα

Η δικτύωση που καθορίζεται από το λογισμικό (SDN) είναι μια αναδυόμενη τεχνολογία, που χωρίζει φυσικά τα δεδομένα και τα επίπεδα ελέγχου στις συσκευές δικτύου. Από την άποψη της ασφάλειας, το SDN έχει δύο πλευρές. Πρώτον, επιτρέπει τις λειτουργίες ασφάλειας δικτύου εκ σχεδιασμού του, επειδή οι ροές κυκλοφορίας μπορούν να δρομολογηθούν εκ νέου ή να φιλτραριστούν με βάση το περιεχόμενο των πακέτων ή τη λειτουργικότητα του επιπέδου εφαρμογής, η οποία μέχρι σήμερα απαιτεί πρόσθετες συσκευές ασφάλειας δικτύων όπως τείχη προστασίας, συστήματα ανίχνευσης εισβολών ή φίλτρα ανεπιθύμητης αλληλογραφίας σε δίκτυα παλαιού τύπου. Από την άλλη πλευρά, λόγω του φυσικού διαχωρισμού των δύο προαναφερθέντων επιπέδων, η SDN αρχιτεκτονική, ενδεχομένως εγκυμονεί επιπλέον κενά ασφαλείας σε σύγκριση με τις παραδοσιακές αρχιτεκτονικές δικτύων. Αυτά τα κενά, ενδέχεται να επηρεάσουν σοβαρά τη συνολική διαθεσιμότητα τέτοιων δικτύων σε πραγματικές εφαρμογές, καθώς και την εμπιστευτικότητα, την αυθεντικότητα, την ακεραιότητα και τη συνέπεια των δεδομένων κίνησης δικτύου αλλά και ελέγχου. Στην εργασία [41], υπάρχει μία εκτενής ανάλυση και σύγκριση σχετικά με την ασφάλεια που παρέχεται από τα SDN δίκτυα, σε σχέση με τα παραδοσιακά. Η ανάλυση αυτή καταλήγει στο συμπέρασμα ότι παρόλο που η νέα αυτή αρχιτεκτονική δικτύου, δεν εξαλείφει όλες τις πιθανές παραβιάσεις σε θέματα ασφαλείας, είναι σαφώς προτιμότερη από τις παραδοσιακές δικτυακές αρχιτεκτονικές.

Οι αρχιτεκτονικές SDN μπορούν να συνδυαστούν και να λειτουργήσουν μαζί με άλλες τεχνολογίες σχετικές με τη δικτύωση, οι οποίες εξασφαλίζουν περισσότερη ασφάλεια και εμπιστευτικότητα στα δεδομένα και στον έλεγχο του δικτύου. Μία από αυτές τις τεχνολογίες που αξίζει να μελετηθεί σε συνδυασμό με τα SDN δίκτυα, είναι το blockchain και μέχρι σήμερα τόσο στον ακαδημαϊκό αλλά και στον επιχειρησιακό κόσμο έχει γίνει ορισμένη πρόοδος, αλλά η έρευνα πάνω σε αυτό το αντικείμενο είναι ακόμα σε πρώιμο στάδιο.

Στην εργασία [42] οι συγγραφείς κάνουν μία ανασκόπηση της χρήσης των SDN αρχιτεκτονικών στην αρχιτεκτονική των NFV δομών (Network Functions Virtualization). Δύο έννοιες που συχνά χρησιμοποιούνται μαζί, SDN και NFV, αλληλοσυμπληρώνονται μεταξύ τους και προσφέρουν μία αφαιρετική λογική ως προς τις υποδομές που βασίζονται στο υλικό (hardware). Όπως μπορεί να διαπιστωθεί και από την εργασία στο [43], υπάρχει ήδη ερευνητική μελέτη για το πώς μπορούν να διαμορφωθούν και να διαχειριστούν λειτουργίες NFV με την χρήση του blockchain. Η ανάγκη ασφαλούς παροχής υπηρεσιών δικτύου καθίσταται κρίσιμη, καθώς απλές τροποποιήσεις στον πυρήνα του δικτύου μπορούν να επηρεάσουν πολλούς χρήστες του δικτύου. Έτσι προτάθηκε μία αρχιτεκτονική που εξασφαλίζει την αμεταβλητότητα, και την ελεγκτικότητα στις VNF/SDN διαμορφώσεις, διατηρώντας επίσης την ανωνυμία της πληροφορίας και μετριάζοντας τις πιθανότητες μίας στοχευμένης επίθεσης σε τέτοιου είδους υλοποιήσεις.

Επίσης σχετικά με μία πιο ασφαλή και συνεπή αρχιτεκτονική στα SDN δίκτυα υπάρχουν δημοσιεύσεις οι οποίες προτείνουν κατανεμημένα μοντέλα ελεγκτών στο επίπεδο ελέγχου της αρχιτεκτονικής. Μία από αυτές, είναι και η εργασία [44], σχετικά με την προσαρμοστική κατάσταση συνοχής, σε ένα κατανεμημένο επίπεδο ελέγχου SDN. Ο συγχρονισμός καταστάσεων

σε ομάδες SDN ελεγκτών είναι μία πρόταση που παρέχει ένα ισχυρό μοντέλο συνέπειας και συνοχής σε τρέχουσες SDN εφαρμογές, όσον αφορά την πληροφορία την οποια διαχειρίζονται οι ελεγκτές αυτοί, βασικό κομμάτι της οποίας είναι οι κανόνες δρομολόγησης στο δίκτυο. Βέβαια για μεγάλης κλίμακας εφαρμογές και υλοποιήσεις η διαδικασία αυτή αυξάνει την καθυστέρηση του συγχρονισμού των καταστάσεων στους ελεγκτές του συμπλέγματος. Ως εκ τούτου έχει επιζήμια επίδραση στις λειτουργίες ενός SDN δικτύου, οι οποίες απαιτούν γρήγορη απόκριση, σταθερότητα και μία βασική τουλάχιστον ποιότητα υπηρεσιών. Κατά συνέπεια, παρόλο που το παραπάνω μοντέλο κατανεμημένων ελεγκτών λύνει προβλήματα όπως, η ανθεκτικότητα των κανόνων δρομολόγησης ή όπως οι τρωτότητες ενός μόνο σημείου στην αποτυχία (μοναδικός ελεγκτής που για οποιονδήποτε λόγο δεν μπορεί να εξυπηρετήσει το δίκτυο), υπάρχουν ζητήματα τα οποία δεν επιδέχονται λύση με το μοντέλο αυτό. Τέτοια ζητήματα για παράδειγμα είναι η ασφάλεια ή η πηγή των κανόνων δρομολόγησης, οι οποίοι μπορεί να παραποιηθούν από τρίτες οντότητες και η ακεραιότητά τους να είναι αμφισβητήσιμη. Επίσης λόγω της δομής ενός τέτοιου μοντέλου, εφαρμογές οι οποίες απαιτούν μεγάλη κλιμάκωση, δεν είναι εύκολα υλοποιήσιμες, και αυτό αποτελεί επίσης ένα ζήτημα.

Τροποποιώντας το κεντρικοποιημένο μοντέλο διαχείρισης των SDN δικτύων, τα δίκτυα αυτά μπορούν και είναι θεμιτό να συνδυαστούν με ένα peer-to-peer (p2p) δίκτυο blockchain, το οποίο με την σωστή χρήση του, βελτιώνει την ασφάλεια [45, 46], την διαθεσιμότητα, την αξιοπιστία αλλά και την ευκαμψία της πληροφορίας που διαχειρίζεται το δίκτυο. Για παράδειγμα οι ερευνητές στο [15], κατασκεύασαν και προτείνουν μία κατανεμημένη, στο επίπεδο ελέγχου, SDN αρχιτεκτονική δικτύου για εφαρμογές IoT, η οποία βασίζεται στο bc, και προσφέρει όλα τα παραπάνω πλεονεκτήματα καθώς επίσης σύμφωνα με τους συγγραφείς είναι και εύκολα κλιμακούμενο.

Σε αντίστροφη λογική με την προαναφερθείσα, πάλι για την ασφάλεια των δεδομένων σε ένα SDN δίκτυο, έχει προταθεί το ChainGuard [47], το οποίο ενσωματώνει ένα SDN δίκτυο και το πρωτόκολλο OpenFlow, χρησιμοποιώντας ως τοίχος προστασίας σε οποιονδήποτε τύπο blockchain εφαρμογών. Ως αποτέλεσμα η τεχνολογία της SDN αρχιτεκτονικής αυτή την φορά, αξιοποιείται για να προστατεύσει κόμβους που υπάρχουν στο p2p δίκτυο του bc, από DoS επιθέσεις και μη εξουσιοδοτημένες προσβάσεις τρίτων κόμβων στο p2p δίκτυο. Συμπεραίνουμε λοιπόν πως η ασφάλεια και η εμπιστευτικότητα στα δεδομένα των σύγχρονων δικτυακών αρχιτεκτονικών, δεν επωφελείται μόνο από την ενσωμάτωση του bc σε SDN αρχιτεκτονικές, αλλά και από το αντίστροφο. Η χρήση των δύο αυτών τεχνολογιών μαζί, μπορεί να μετριάζει αποτελεσματικά διάφορους τύπους επιθέσεων και στις δύο αυτές αρχιτεκτονικές, όπως επιθέσεις πλημμύρας, επιθέσεις Sybil αλλά και διάφορες επιθέσεις τύπου botnet [49, 48].

Επιπρόσθετα τα παραπάνω, όπως μπορούμε να δούμε από το Trust List [14], το SDN σε συνδυασμό με το bc, μπορεί να δημιουργήσει ένα μοντέλο το οποίο αυτοματοποιεί τη διαδικασία αμφισβήτησης, επαλήθευσης και εμπιστοσύνης σε υπηρεσίες IoT, για την αποτελεσματική πρόληψη επιθέσεων και καταχρήσεων στο δίκτυο. Όπως έγινε γνωστό από την DDoS στο DNS πάροχο Dyn το 2016 [50], ο συντονισμός πολλών δικτυακών συσκευών περιορισμένης υπολογιστικής ισχύος όπως συσκευές IoT, μπορεί να αποτελέσει απειλή ακόμα και για οργανωμένες και προετοιμασμένες άμυνες κρίσιμων για το διαδίκτυο υπηρεσιών, όπως οι υπηρεσίες

DNS. Με συνδυαστικά μοντέλα blockchain και SDN, όπως το Trust List, για την έξυπνη διαχείριση της δικτυακής κίνησης, μπορούν να αποφευχθούν τέτοιου είδους επιθέσεις.

Για να λειτουργήσουν όμως τέτοιοι μηχανισμοί αποτροπής επιθέσεων σε κατανεμημένα συστήματα και για να υπάρχουν ασφαλείς υλοποιήσεις χωρίς παραβιάσεις από κακόβουλες οντότητες, είναι θεμιτό και απαραίτητο να υπάρχει η δυνατότητα επαληθεύσιμων ταυτοτήτων για του χρήστες του δικτύου με αξιοπιστία και ανωνυμία. Το VeidBlock είναι ένας blockchain μηχανισμός, ο οποίος μπορεί να ενσωματωθεί σε SDN δίκτυα και επιτυγχάνει το αποτέλεσμα αυτό.

Κεφάλαιο 3

Αρχιτεκτονική

3.1 Τα Καθοριζόμενα από το Λογισμικό Δίκτυα

3.1.1 Η ανάγκη για νέες αρχιτεκτονικές δικτύων

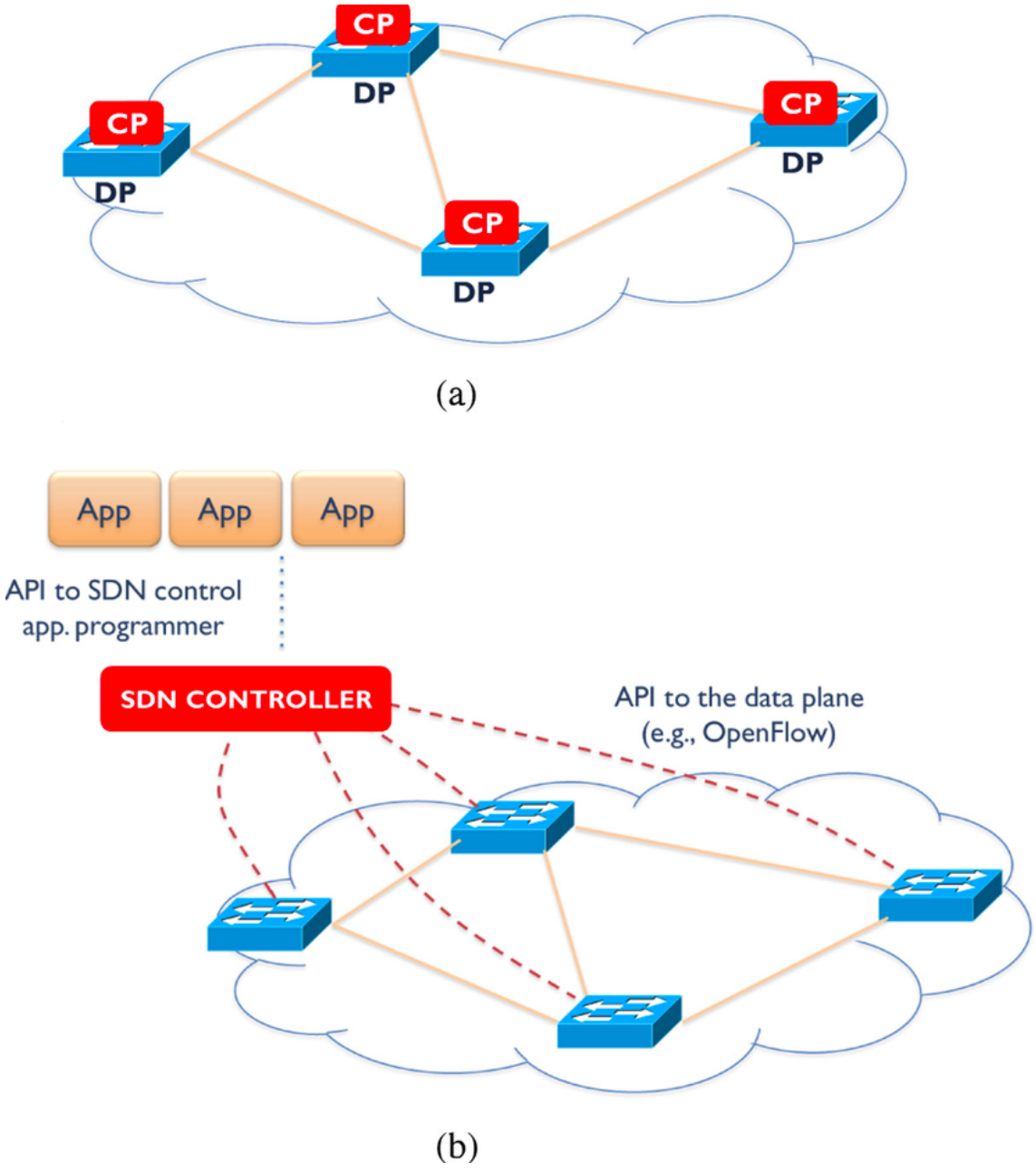
Η εκρηκτική αύξηση στις κινητές συσκευές και στο περιεχόμενο τους, η εικονικοποίηση διακομιστών και η εμφάνιση των υπηρεσιών cloud συγκαταλέγονται στις τάσεις που οδηγούν τη βιομηχανία δικτύωσης να επανεξετάσει τις παραδοσιακές αρχιτεκτονικές δικτύων και να στραφεί προς νέες αρχιτεκτονικές. Πολλά συμβατικά δίκτυα είναι ιεραρχικά, χτισμένα με βαθμίδες Ethernet και δρομολογητές τοποθετημένους σε δομή δέντρου. Αυτός ο σχεδιασμός είχε νόημα όταν κυριαρχούσε το μοντέλο πελάτη-διακομιστή, αλλά μια τέτοια στατική αρχιτεκτονική δεν ανταποκρίνεται στις δυναμικές ανάγκες της πληροφορικής και της αποθήκευσης των σημερινών κέντρων δεδομένων, πανεπιστημιακών συγκροτημάτων και φορέων. Οι βασικές τάσεις υπολογιστικής οι οποίες περιλαμβάνουν μοντέλα δικτυακής κίνησης που συνεχώς αλλάζουν, περιλαμβάνουν μεταξύ άλλων αυξημένη χρήση συσκευών τεχνολογίας από τους χρήστες, άνοδο των υποδομών σύννεφου (cloud services), καθώς και μεγάλο όγκο δεδομένων (big data) τα οποία χρειάζονται ολοένα και αυξανόμενο εύρος ζώνης (bandwidth) [52]. Συνεπώς με την εξέλιξη των δικτύων και των πρωτοκόλλων που τα συνοδεύουν, αυξήθηκε παράλληλα και η ανάγκη για μεγαλύτερη ευελιξία στην διαχείριση αλλά και καλύτερη απόδοση, με λιγότερες καθυστερήσεις και δυναμικό προγραμματισμό στις ροές της κίνησης. Όλα τα παραπάνω καθώς και η ανάγκη για μία πιο απλοποιημένη παροχή υπηρεσιών, οδήγησαν σε ένα νέο μοντέλο υπολογιστικής δικτύωσης, τα Δίκτυα Καθοριζόμενα από το Λογισμικό - SDN.

Η βασική ιδέα των δικτύων αυτών είναι ο διαχωρισμός της διαχείρισης του δικτύου σε επίπεδα. Οι αρχιτεκτονικές SDN, αποσυνδέουν και διαχωρίζουν τις λειτουργίες ελέγχου (control plane), με τις λειτουργίες προώθησης των πακέτων (forwarding or data plane), δύο λειτουργίες που στα δίκτυα παλαιού τύπου συνυπήρχαν, σε διαφορετικά επίπεδα. Αυτό επιτρέπει και κάνει πιο άμεσο τον προγραμματισμό του ελέγχου στο δίκτυο. Επίσης η υποκείμενη εκάστοτε δικτυακή υποδομή, αποσυνδέεται και λειτουργεί αφαιρετικά ως προς τις εφαρμογές και τις δικτυακές υπηρεσίες. Η ευελιξία που προκύπτει από την αποσύνδεση του επιπέδου ελέγχου

με το επίπεδο των δεδομένων ή αλλιώς επίπεδο προώθησης, επιτρέπει στους διαχειριστές να προσαρμόζουν δυναμικά τις ροές της κυκλοφορίας των πακέτων στο δίκτυο και αυτό έχει ως αποτέλεσμα το δίκτυο να προσαρμόζεται δυναμικά στις ανάγκες των εκάστοτε υπηρεσιών που υπάρχουν πάνω σε αυτό. Συνεπώς τα SDN πρωτόκολλα, επιτρέπουν στους διαχειριστές του δικτύου, την βελτιστοποίηση των πόρων του, πράγμα σημαντικό όταν υπάρχουν υπηρεσίες αλλά και δίκτυα που ο φόρτος τους αλλάζει δυναμικά και συνεχώς μεταβάλλεται.

Λόγω του παραπάνω διαχωρισμού των δύο επιπέδων, τα SDN Δίκτυα έχουν κεντρική διαχείριση και η νοημοσύνη του δικτύου είναι συγκεντρωμένη σε οντότητες οι οποίες ονομάζονται ελεγκτές (controllers) όπως αναφέραμε και στο προηγούμενο κεφάλαιο. Χαρακτηριστικό των ελεγκτών είναι ότι διατηρούν μία συνολική εικόνα του δικτύου, έχοντας έτσι ρόλο επόπτη, δίνοντας στο δίκτυο χαρακτηριστικά κεντρικοποιημένου συστήματος. Ρόλος των ελεγκτών είναι ο καθορισμός των κανόνων δρομολόγησης στους μεταγωγείς (switches) του δικτύου, από τους οποίους διέρχονται τα πακέτα πληροφορίας και η κίνηση του δικτύου.

3.1.2 Δομή ενός τυπικού SDN



Σχήμα 3.1: Σύγκριση παραδοσιακών δικτυακών αρχιτεκτονικών (α) και δικτύων SDN (β). [73] (α) Στην παραδοσιακή δικτύωση, το επίπεδο ελέγχου (Control Plane - CP) και το επίπεδο δεδομένων (Data Plane - DP) συντονίζονται σε συσκευές για να διασφαλιστεί ο αποκεντρωμένος έλεγχος δικτύου. (β) Στα SDNs, τα DP και τα CP είναι χωρισμένα, και υπάρχει ένας κεντρικός ελεγκτής που ελέγχει πολλαπλά DPs υποστηρίζοντας ένα API προς τις Dps και ένα API προς τις SDN εφαρμογές. (Πηγή: [73]])

Τα στοιχεία ενός δικτύου καθοριζόμενου από το λογισμικό μπορούν να ενταχθούν σε τρία επίπεδα, που το απαρτίζουν και από κάτω προς τα πάνω (bottom-up) είναι:

- το επίπεδο δεδομένων ή προώθησης,
- το επίπεδο ελέγχου,
- το επίπεδο εφαρμογών.

Οι εφαρμογές είναι προγράμματα που κοινοποιούν τις απαιτήσεις τους στους ελεγκτές μέσω μίας διεπαφής που ονομάζεται northbound interface (NBI). Αποτελούνται από την λογική της εφαρμογής και έναν ή περισσότερους NBI drivers, δηλαδή προγράμματα διασύνδεσης με τον controller.

Στο επόμενο επίπεδο ο ςοντρολλερ, αποτελεί μία λογικά συγκεντρωμένη οντότητα η οποία είναι υπεύθυνη για:

1. 1. Να μεταφράζουν τις απαιτήσεις του επιπέδου εφαρμογών στο επίπεδο δεδομένων.
2. 2. Να παρέχουν στις SDN εφαρμογές μία αφηρημένη άποψη του δικτύου.

Ένας SDN controller αποτελείται από έναν ή περισσότερους πράκτορες για την επικοινωνία με το επίπεδο εφαρμογών μέσω του NBI, την λογική ελέγχου στο δίκτυο και τέλος την διεπαφή για την διασύνδεση του controller με το επίπεδο δεδομένων (Control to Data Plane Interface - CDPI). Παρόλο που τα SDN προκαθορίζουν μία λογικά συγκεντρωμένη οντότητα με κεντρικοποιημένα χαρακτηριστικά, αυτό δεν αποκλείει SDN αρχιτεκτονικές με ομοσπονδία πολλών ελεγκτών, που έχουν ιεραρχική σύνδεση μεταξύ τους και ίσως πραγματοποιούν επίσης τεμαχισμό των πόρων στο δίκτυο. Ένα τέτοιο παράδειγμα SDN αρχιτεκτονικής είναι η υλοποίηση Clustering σε μία εφαρμογή για SDN Δίκτυα γραμμένη σε Java που ονομάζεται OpenDayLight (ODL), για την οποία θα μιλήσουμε εκτενώς παρακάτω.

Το κατώτατο επίπεδο, δηλαδή των δεδομένων ή προώθησης απαρτίζεται από από στοιχεία του δικτύου που ονομάζονται SDN Datapaths. Τα datapaths είναι λογικές συσκευές στο δίκτυο οι οποίες εκθέτουν την ορατότητα και επιτρέπουν τον έλεγχο των διαφημιζόμενων δυνατοτήτων προώθησης και επεξεργασίας δεδομένων. Τα Datapaths αποτελούνται από CDPI διεπαφές (CDPI Agents) και ένα σύνολο από έναν ή παραπάνω κανόνες προώθησης της κυκλοφορίας καθώς και ορισμένες λειτουργίες επεξεργασίας της τελευταίας (forwarding engines / processing functions).

Η διεπαφή CDPI (Control to Data Plane Interface), είναι η οντότητα που ενώνει τον SDN Controller με τα SDN Datapaths. Παρέχει προγραμματικό έλεγχο όλων των λειτουργιών προώθησης, διαφήμιση δυνατοτήτων, αναφορά στατιστικών στοιχείων και ειδοποίηση συμβάντων.

Η διεπαφή NBI, (NorthBound Interface), είναι η οντότητα που ενώνει τον SDN Cotnroller με το επίπεδο εφαρμογών. Όπως είπαμε με τα NBIs επιτυγχάνονται αφηρημένες προβολές δικτύου και έτσι επιτρέπεται η απευθείας έκφραση της συμπεριφοράς και των απαιτήσεων του δικτύου.

Συνεπώς και όπως προκύπτει ως άμεσο συμπέρασμα των παραπάνω, η κύρια ιδιότητα των SDN Δικτύων, είναι ότι επίπεδο δεδομένων και επίπεδο ελέγχου είναι ξεχωριστά προσφέροντας έτσι μία πιο ομαλή και αποτελεσματική επεξεργασία της επικοινωνίας μεταξύ των οντοτήτων του δικτύου.

3.1.3 Το βασικό πρωτόκολλο SDN - OpenFlow Interface

Σημαντικό είναι να αναλύσουμε το πρωτόκολλο OpenFLow το οποίο συνδέεται άμεσα με τα SDN δίκτυα, καθώς ήταν και το πρώτο πρότυπο που σχεδιάστηκε για την υλοποίηση αυτών.

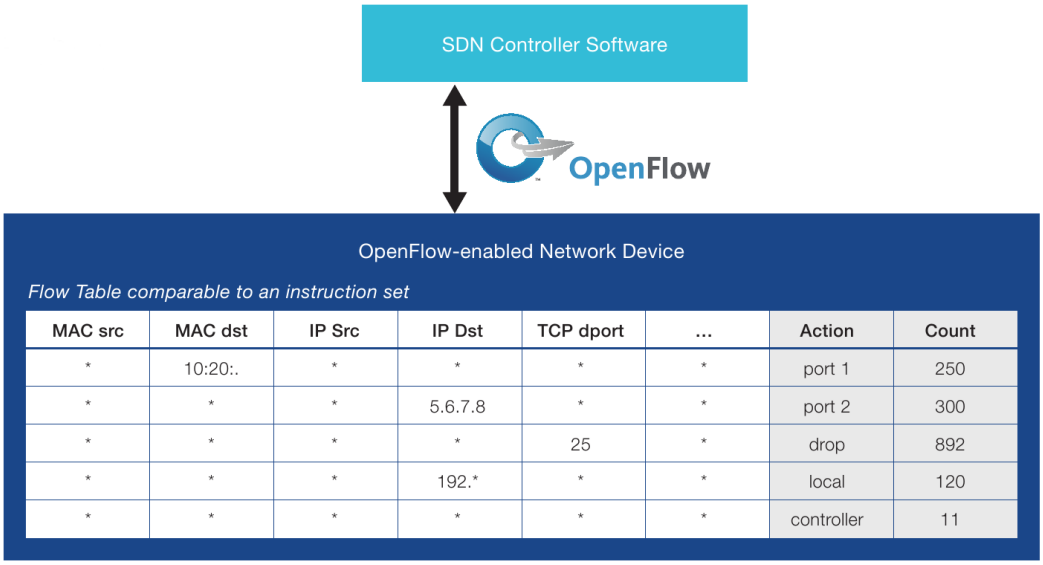
Το OpenFlow είναι ένα πρωτόκολλο επικοινωνίας το οποίο δίνει πρόσβαση στο επίπεδο προώθησης ενός δρομολογητή του δικτύου. Ουσιαστικά, δίνει την δυνατότητα στους ελεγκτές του SDN δικτύου, να καθορίσουν το μονοπάτι ή την διαδρομή που θα ακολουθήσουν τα πακέτα στο δίκτυο μέσω των δρομολογητών του. Μέσω του πρωτοκόλλου OpenFlow επιτρέπεται μία πιο εξελιγμένη διαχείριση στην κυκλοφορία της πληροφορίας στο δίκτυο, από ό,τι ήταν εφικτό προηγουμένως με τις λίστες ελέγχου πρόσβασης (ACLs) και με τα πρωτόκολλα δρομολόγησης που υπήρχαν πριν από τα SDN.

Αποτελεί την πρώτη τυπική διεπαφή επικοινωνιών που ορίζεται μεταξύ των επιπέδων ελέγχου και προώθησης μιας αρχιτεκτονικής SDN [51]. Το OpenFlow επιτρέπει την άμεση πρόσβαση και το χειρισμό του επιπέδου προώθησης των συσκευών δικτύου, όπως τα σωιτσηες και οι ρουτερς, τόσο φυσικά όσο και εικονικά (βασισμένα σε κάποιον hypervisor) και μετακινήσει τον έλεγχο δικτύου από τους φυσικούς διακόπτες δικτύωσης (physical switches) σε ένα λογικά κεντρικό λογισμικό ελέγχου, τους ελεγκτές (controllers).

Πιο συγκεκριμένα, επιτρέπει την απομακρυσμένη διαχείριση των πινάκων προώθησης ενός layer 3 switch, τροποποιώντας κανόνες και ενέργειες αντιστοίχισης πακέτων (matching rules). Οι αποφάσεις δρομολόγησης λαμβάνονται, προγραμματισμένα ή μη, από τον controller και μεταφράζονται σε κανόνες και ενέργειες για τους μεταγωγείς, με συγκεκριμένη διάρκεια ζωής. Έπειτα, εγκαθίστανται στον πίνακα κανόνων (flow table) ενός switch, που βρίσκεται σε μία βάση δεδομένων. Ο switch με την σειρά του είναι υπεύθυνος για την προώθηση των πακέτων με βάση αυτούς τους κανόνες. Την εφαρμογή δηλαδή των κανόνων στα πακέτα του δικτύου. Στην περίπτωση όπου έχουμε πακέτα που δεν κάνουν match με κανέναν κανόνα, τότε αυτά προωθούνται στον controller, ο οποίος είτε τροποποιεί, αναλόγως του υπάρχοντες κανόνες στα switches, είτε δημιουργεί νέους κανόνες, αν κρίνεται απαραίτητο, είτε δρομολογεί την κίνηση ο ίδιος αν έχει προσυμφωνηθεί με τα switches να προωθούνται ολόκληρα τα πακέτα σε αυτόν και όχι μόνο οι επικεφαλίδες τους (headers), που είναι και η προκαθορισμένη λειτουργία.

Επομένως το OpenFlow χρησιμοποιείται και έχει ως κύριο σκοπό την επικοινωνία μεταξύ των controller και των switches σε ένα SD Δίκτυο, ώστε να γίνεται δυνατή η εγκατάσταση και η παραμετροποίηση των κανόνων προώθησης ή όπως ονομάζονται OpenFlow κανόνες.

Το OpenFlow μπορεί να συγκριθεί με το σετ εντολών μιας CPU. Όπως φαίνεται στο σχήμα 2, το πρωτόκολλο καθορίζει βασικές αρχές που μπορούν να χρησιμοποιηθούν από μια εξωτερική εφαρμογή για τον προγραμματισμό του επιπέδου προώθησης των συσκευών δικτύου, ακριβώς όπως το σετ εντολών μιας CPU καθορίζει ένα σύστημα υπολογιστή [52].



Σχήμα 3.2: Παράδειγμα Συνόλου Εντολών OpenFlow (Πηγή: [113]

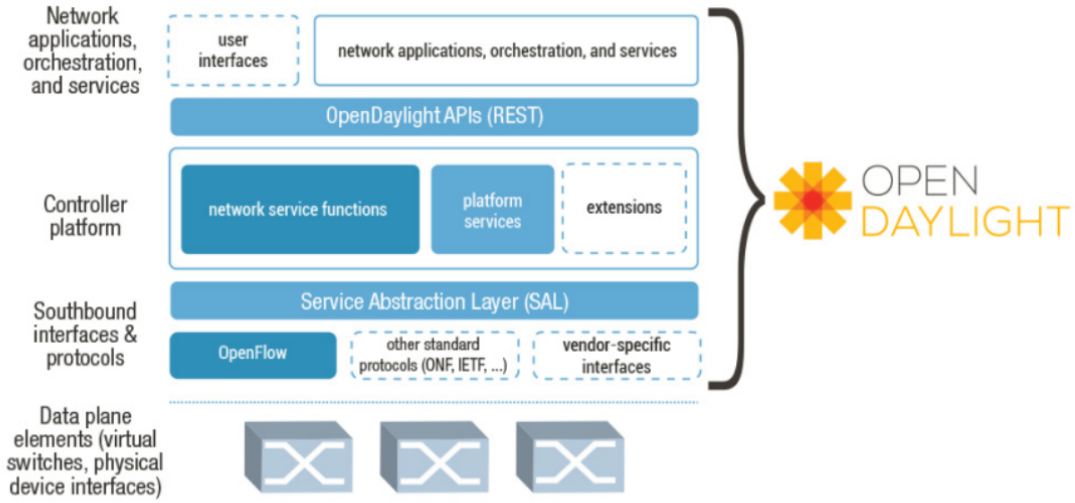
Το πρωτόκολλο OpenFlow εφαρμόζεται και στις δύο πλευρές της διασύνδεσης μεταξύ των συσκευών υποδομής δικτύου και του λογισμικού ελέγχου SDN. Το OpenFlow χρησιμοποιεί την έννοια των ροών για τον προσδιορισμό της επισκεψιμότητας των κόμβων (node connectivity) και της κίνησης (traffic) του δικτύου βάσει προκαθορισμένων κανόνων αντιστοίχισης που μπορούν να προγραμματιστούν στατικά ή δυναμικά από το λογισμικό ελέγχου SDN. Επιτρέπει επίσης στην διαχείριση να καθορίσει τον τρόπο ροής, μέσω συσκευών δικτύου με βάση παραμέτρους όπως τα πρότυπα χρήσης, τις εφαρμογές και τους πόρους σε υποδομές σύννεφου. Δεδομένου ότι το OpenFlow επιτρέπει στο δίκτυο να προγραμματίζεται με βάση τη ροή, μια αρχιτεκτονική SDN βασισμένη σε OpenFlow παρέχει εξαιρετικά λεπτομερή έλεγχο, επιτρέποντας στο δίκτυο να ανταποκρίνεται στις αλλαγές σε πραγματικό χρόνο σε επίπεδο εφαρμογής, χρήστη και συνεδρίας (session). Η τρέχουσα δρομολόγηση βάσει IP δεν παρέχει αυτό το επίπεδο ελέγχου, καθώς όλες οι ροές μεταξύ δύο τελικών σημείων πρέπει να ακολουθούν την ίδια, προκαθορισμένη διαδρομή μέσω του δικτύου, ανεξάρτητα από τις διαφορετικές απαιτήσεις τους.

Στην αρχιτεκτονική μας, το πρωτόκολλο αυτό αποτελεί βασικό κομμάτι καθώς σε αυτό βασίζεται όλο το SDN δίκτυο που ακολουθεί στην συνέχεια. Ο ρόλος του είναι η επικοινωνία του controller με τα virtual switches καθώς και στην εφαρμογή των κανόνων δρομολόγησης από τα switches στους τυπικούς ηοστς του SDN δικτύου.

3.1.4 OpenDaylight Controller

Για την υλοποίηση του SDN δικτύου μας, κάναμε χρήση του OpenDayLight project [53, 54], το οποίο παρέχει και προωθεί την χρήση λογισμικού ανοιχτού κώδικα για SDN αρχιτεκτονικές. Το Opendaylight είναι επίσης project του Linux Foundation. Στην παρακάτω εικόνα φαίνεται η γενική αρχιτεκτονική στην οποία βασίζεται το OpenDayLight - ODL [55].

Επί της ουσίας, είναι μία πλατφόρμα για SDN, και όπως πολλοί άλλοι ελεγκτές για SDN [56], υποστηρίζει επίσης το πρωτόκολλο OpenFlow, καθώς και έτοιμες προς εγκατάσταση δικτυακές λύσεις [52].



Σχήμα 3.3: Η Δομή της πλατφόρμας OpenDayLight (Πηγή: [114]

Ο ελεγκτής OpenDayLight, λειτουργεί ως καθαρό λογισμικό και μπορεί να τρέξει σε οποιοδήποτε λειτουργικό σύστημα ως εικονική μηχανή Java (JVM), αλλά και σε φυσικό μηχάνημα το οποίο υποστηρίζει Θαα, στην οποία είναι γραμμένο και το λογισμικό. Στα πλαίσια της παρούσας εργασίας, το λογισμικό ODL για τον SDN controller ενσωματώθηκε για να τρέχει μέσα σε docker container. Αυτό έγινε έτσι ώστε να υπάρχει διαχωρισμός μεταξύ των πορτς και των σεριες της εικονικής μηχανής, στην οποία έγινε η διεξαγωγή των πειραμάτων και η υλοποίηση της εργασίας. Θα αναφερθούμε εκτενώς σε αυτό, στο κεφάλαιο της υλοποίησης. Για τις ανάγκες του παρόντος κεφαλαίου μπορούμε να δούμε τον ODL controller ως ένα μαύρο κουτί που επιτελεί λειτουργίες SDN ελεγκτή.

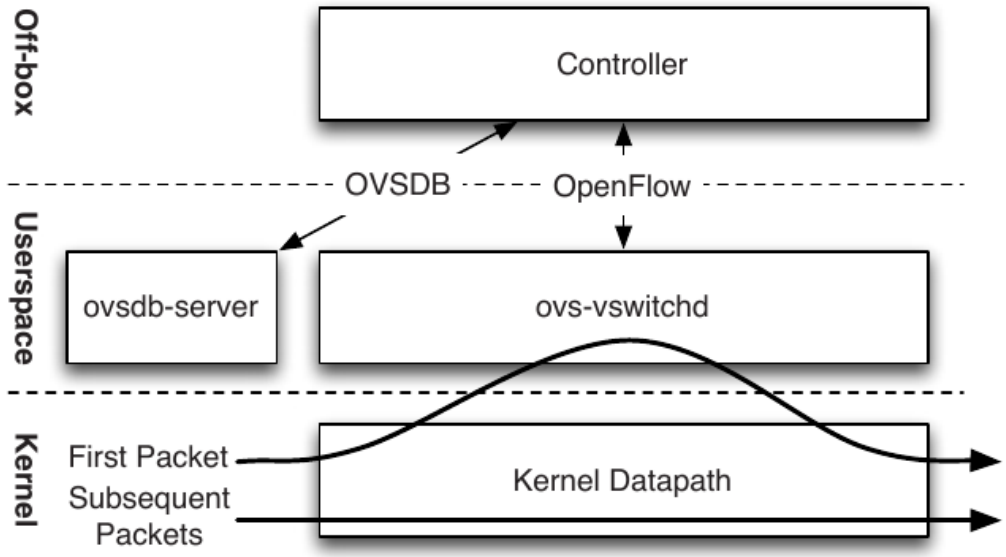
3.1.5 OpenVirtualSwitch – OVSDB

Ένας ιρτυαλ switch (εικονικός μεταγωγέας πακέτων) είναι η λογική υλοποίηση ενός φυσικού μεταγωγέα σε λογισμικό, και αποτελεί ένα ξεχωριστό επίπεδο λογισμικού σε έναν υπολογιστή server, ο οποίος επίσης φιλοξενεί εικονικές μηχανές (VMs) ή containers με εικονικές Ethernet πόρτες (vNICs). Ο ρόλος ενός virtual switch (VS) είναι η δρομολόγηση κίνησης και πακέτων, στις εικονικές υπολογιστικές υποδομές, είτε μεταξύ των εικονικών μηχανών είτε μεταξύ κέντρων δεδομένων που υποστηρίζουν εικονικά περιβάλλοντα [57].

Το OpenVirtualSwitch (OvS) είναι ένα ανοιχτού κώδικα εργαλείο που υλοποιεί ένα vSwitch, ο οποίος μπορεί να λειτουργήσει ως εικονικός διακόπτης σε εικονικά περιβάλλοντα, ή ως μεταγωγέας σε εφαρμογές που είναι βασισμένες σε λογισμικό (όχι εικονικό περιβάλλον), ή ακόμα και ως κέντρο ελέγχου για μεταγωγείς υλικού (hardware switches) [57]. Παίζει

σημαντικό ρόλο σε αρκετές εφαρμογές SDN και NFV όπως το OpenStack, OpenNebula, OpenDayLight [57].

Στο OvS, υπάρχουν δύο βασικά στοιχεία που αποτελούν την αρχιτεκτονική του και τον τρόπο λειτουργίας του, με τον οποίο προωθούνται τα πακέτα. Το πρώτο είναι ο ovs-vswitchd δαίμονας, και το δεύτερο το datapath kernel module, αλλά στην παρούσα εργασία δεν θα μπούμε σε τόση λεπτομέρεια και αναφέρονται μόνο για καλύτερη κατανόηση της παρακάτω εικόνας. Επίσης σημαντικό είναι να αναφέρουμε την βάση την οποία διαχειρίζεται τοπικά κάθε OvS, η οποία ονομάζεται OVSDB και είναι εκεί όπου αποθηκεύονται οι κανόνες δρομολόγησης για την κίνηση του δικτύου. Στο σχήμα που ακολουθεί φαίνεται πως ο OvS συνδέεται με τον SDN controller και την βάση [58]. Επίσης, φαίνεται ξεκάθαρα η λειτουργία της διεπαφής OpenFlow η οποία αναλαμβάνει όλη την επικοινωνία μεταξύ OvS και controller.

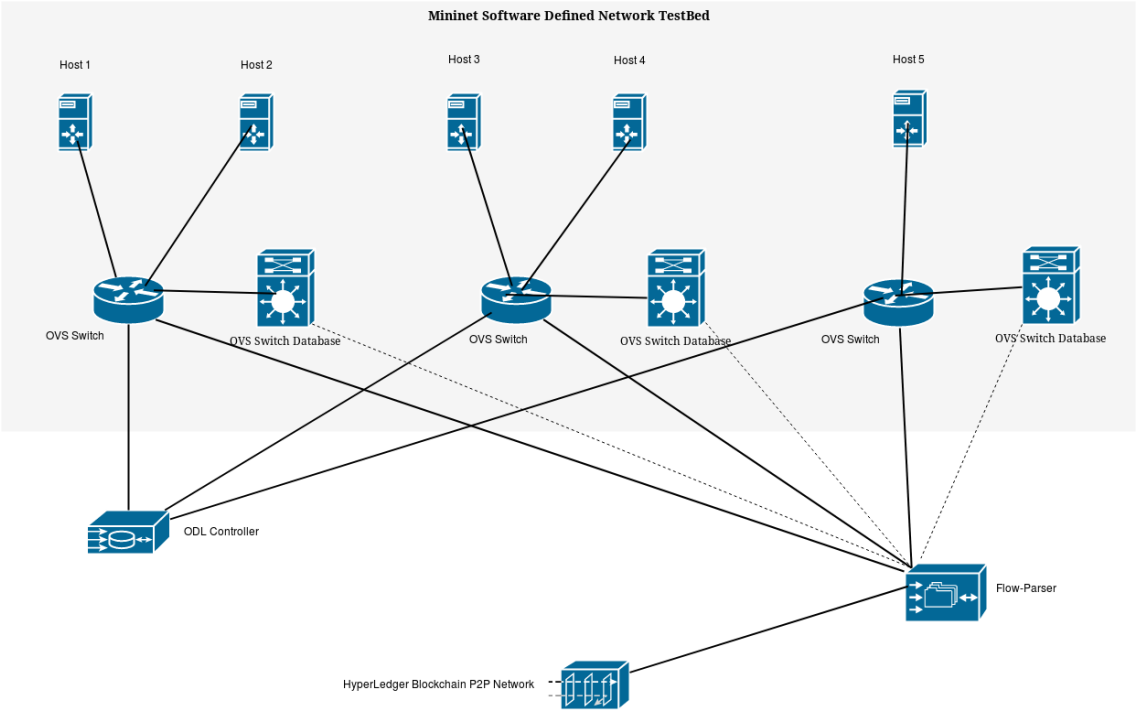


Σχήμα 3.4: Τα Στοιχεία & οι Διεπαφές του OVS (Πηγή: [58]

3.1.6 Οντότητες και στοιχεία SDN Δικτύου

Το SDN δίκτυο που υλοποιήσαμε αποτελείται από 3 σωιτσηες με τις αντίστοιχες βάσεις στις οποίες αποθηκεύεται η πληροφορία για τους κανόνες δρομολόγησης και τις αποφάσεις για την κίνηση του SDN δικτύου από τον controller. Επίσης κάθε VS συνδέεται ενδεικτικά σε έναν ή δύο hosts και πιο συγκεκριμένα οι δύο πρώτοι VS έχουν από δύο hosts, ενώ ο τρίτος από έναν όπως φαίνεται στο παρακάτω σχήμα 5. Στο σχήμα 5 φαίνεται η δομή του δικτύου που υλοποιήσαμε για την παρούσα εργασία. Συγκεκριμένα φαίνεται αναλυτικά το SDN δίκτυο που βασίζεται στο mininet και πως αυτό συνδέεται με τα υπόλοιπα στοιχεία τα οποία θα αναλυθούν εκτενώς στην συνέχεια. Όπως ορίζεται από ένα SDN δίκτυο για την λογική αποσύνδεση του επιπέδου ελέγχου με το επίπεδο αποφάσεων, οι τρείς εικονικοί δρομολογητές συνδέονται με

τον Opendaylight ελεγκτή, ρόλος του οποίου είναι οι αποφάσεις δρομολόγησης κίνησης για την βέλτιστη χρήση πόρων στο δίκτυο. Επίσης υπάρχει ένα ακόμα στοιχείο το οποίο συνδέει το P2P δίκτυο του blokchain, με το SDN δίκτυο. Ρόλος της οντότητας αυτής, την οποία ονομάσαμε flow-parser [120], είναι να αναλύει τους κανόνες δρομολόγησης που έχουν οριστεί από τον SDN controller και έχουν αποσταλεί στις βάσεις των VS, και έπειτα να τους μεταφέρει στο blockchain για να καταγράφονται στο ledger του Hyperledger δικτύου.



Σχήμα 3.5: Δομή αρχιτεκτονικής και στοιχεία δικτύου.

Τα παραπάνω στοιχεία τα οποία απαρτίζουν την SDN τοπολογία είναι ενδεικτικά στον αριθμό και αναπαριστούν ένα απλό δίκτυο για τους σκοπούς της εργασίας. Μπορούν εύκολα και με την ίδια σχεδίαση να κλιμακωθούν είτε προς τα κάτω είτε προς τα πάνω, στην οποία περίπτωση η λογική του δικτύου θα λειτουργεί με τον ίδιο ακριβώς τρόπο. Επίσης στην παραπάνω εικόνα φαίνεται και ως μεμονωμένο στοιχείο το P2P δίκτυο για το blockchain (Hyperledger Blockchain P2P Network), που φαίνεται στο κάτω μέρος της εικόνας και του οποίου την αρχιτεκτονική θα αναλύσουμε στην παράγραφο 3.3.3 και του οποίοι η δομή φαίνεται στο σχήμα 6.

3.1.7 Δίκτυα και ανεκτικότητα σφάλματος

Η πιο απλή υλοποίηση σε ένα SDN δίκτυο είναι μία αρχιτεκτονική της οποίας η τοπολογία περιλαμβάνει έναν μόνο controller, ο οποίος συνδέεται και ελέγχει όλους του μεταγωγείς του δικτύου, έχοντας έτσι συνολική εικόνα των στοιχείων του δικτύου, και λαμβάνοντας ως μεμονωμένη οντότητα όλες τις αποφάσεις για την δρομολόγηση των πακέτων. Αυτό το σενάριο όμως εγείρει ορισμένους προβληματισμούς στην περίπτωση όπου ο ένας και μοναδικός

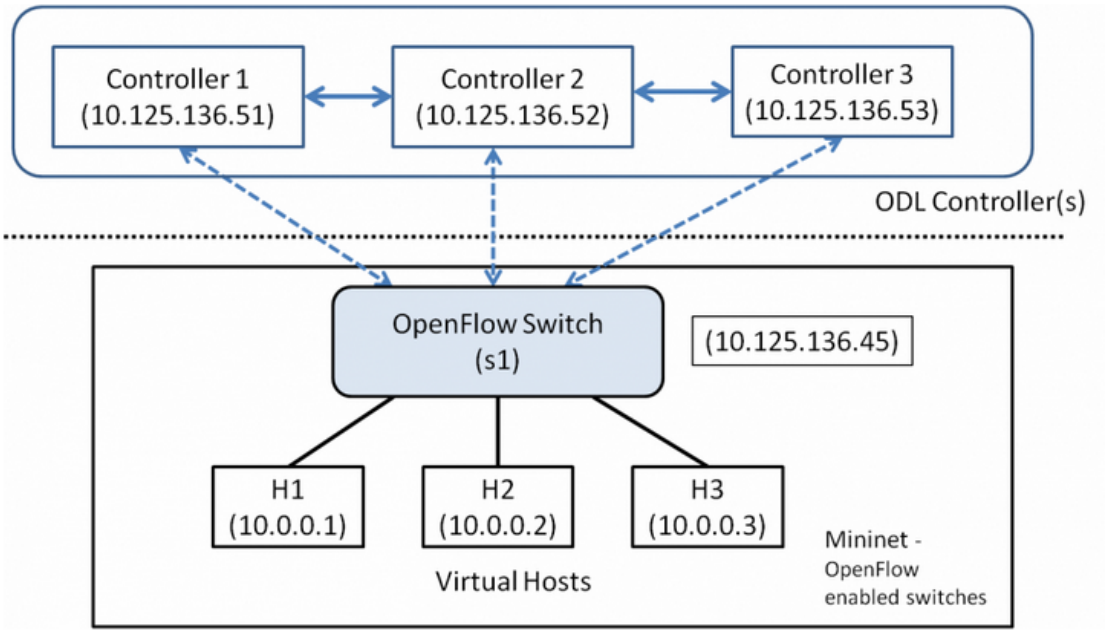
controller δεχθεί μία επίθεση και μεταδίδει κακόβουλους κανόνες στους μεταγωγείς με σκοπό την εκμετάλλευση του δικτύου της εφαρμογής. Σε αυτήν την περίπτωση, λέμε ότι δεν υπάρχει η ανεκτικότητα των ελεγκτών σε σφάλματα καθώς η δυσλειτουργία του ενός και μοναδικού ελεγκτή στο επίπεδο ελέγχου, μεταφράζεται αυτομάτως σε πιθανή και άμεση δυσλειτουργία στο επίπεδο δεδομένων.

Με βάση το παραπάνω, διάφοροι μηχανισμοί έχουν προταθεί μέχρι σήμερα για την επίλυση του παραπάνω ζητήματος [58, 60]. Μία λογική σκέψη είναι να εισαχθούν περισσότεροι ελεγκτές στο επίπεδο ελέγχου, με σκοπό την αυξημένη διαθεσιμότητα και την ανοχή τους σε κακόβουλες επιθέσεις και σφάλματα. Μπορεί να προκύψει με αυτόν τον τρόπο, ένα ελαφρώς διαφορετικό μοντέλο στο οποίο περισσότεροι από έναν controllers, διαχειρίζονται έναν ή περισσότερους μεταγωγείς. Προς απλούστευση των περιγραφών θα θεωρήσουμε ότι στο υποθετικό μας δίκτυο έχουμε έναν και μόνο μεταγωγέα πακέτων (switch). Έτσι σχηματίζεται ένα σύνολο από κόμβους ελεγκτές, εφ εξής σύμπλεγμα ή cluster, το οποίο παρέχει μία ανεκτική σε σφάλματα, αποκεντρωμένη συμμετοχή από ομότιμους κόμβους χωρίς μοναδικό σημείο αποτυχίας. Για να λειτουργήσει όμως αυτό το σενάριο, δεν θα πρέπει όλοι οι ελεγκτές να λειτουργούν ταυτόχρονα για την διαχείρηση του δικτύου. Έτσι εκλέγεται μεταξύ των controller, ένας κόμβος ηγέτης (leader) και οι υπόλοιποι κατέχουν ρόλο οπαδού (follower) και ακολουθούν τις αποφάσεις του leader, αποθηκεύοντας την πληροφορία σε τμήματα από αντίγραφα, τα λεγόμενα shards, υλοποιώντας έτσι έναν μηχανισμό αντιγραφής που ονομάζεται sharding, σε μία ευρύτερη λογική μίας κατανεμημένης βάσης εγγραφών (Distributed Data Store). Ο leader, έχει πλήρη πρόσβαση στον μεταγωγέα και διαχειρίζεται την το φορτίο του δικτύου, το fail-over, δηλαδή ποιός θα πάρει την θέση του σε περίπτωση που αυτός "πέσει" καθώς και την σύγκλιση του clustering, δηλαδή τον τρόπο με τον οποίο αντιγράφεται η πληροφορία στους followers controllers. Με αυτόν τον τρόπο, αν ο leader για κάποιο λόγο δεν μπορεί να επιτελέσει τον ρόλο του ή δεν έχει επικοινωνία με τον μεταγωγέα, δίνει την θέση του σε κάποιον από τους followers η επιλογή του οποίου γίνεται με βάση κάποια κριτήρια. Όλα τα παραπάνω έχουν ως αποτέλεσμα την ύπαρξη ενός μηχανισμού που επιτρέπει τα σφάλματα στο λογισμικό και την συνδεσιμότητα των ελεγκτών και λέμε ότι το δίκτυο είναι ανεκτικό σε σφάλματα.

Υπάρχουν αρκετές δημοσιεύσεις για την βέλτιστη λειτουργία ενός συστήματος SDN με κατανεμημένο επίπεδο ελέγχου (Distributed Control Plane – DCP) [61, 62]. Όλες οι παραπάνω βελτιώσεις αφορούν την βελτιστοποίηση της συνέπειας στο συγχρονισμό καταστάσεων μεταξύ των διαφορετικών controllers και την βελτιστοποίηση της διαθεσιμότητας αυτών, πράγμα σημαντικό για τις συνεχείς ανάγκες ενός δικτύου ή μίας εφαρμογής που βρίσκεται πάνω στο επίπεδο δεδομένων. Παρ' όλα αυτά το ζήτημα της αλλοίωσης των εγγραφών και των κανόνων στους ελεγκτές παραμένει ακόμα και σε αυτές τις αρχιτεκτονικές, αν και γίνεται λίγο πιο περίπλοκο καθώς πλέον δεν υπάρχει μία κεντρική οντότητα ελέγχου. Για αυτό τον λόγο προτείνουμε την ενσωμάτωση του blockchain σε μία τέτοιου είδους εφαρμογή, προσφέροντας επιπλέον ασφάλεια, εμπιστοσύνη και σταθερότητα στο σύστημα, και μειώνοντας τις πιθανότητες στην τροποποίηση των κανόνων δρομολόγησης.

Στο παρακάτω σχήμα φαίνεται μία τέτοια ενδεικτική τοπολογία με ένα σύμπλεγμα αποτε-

λούμενο από 3 controllers:



Σχήμα 3.6: Τοπολογία SDN δικτύου με κατανεμημένο επίπεδο ελέγχου (DCP) (Πηγή: [115]

3.2 Blockchain

Η ιδέα για τις κρυπτογραφικά ασφαλισμένες αλυσίδες κοινοποιήσεων εφεξής blockchains - bc, περιγράφηκαν για πρώτη φορά το 1991, με σκοπό ένα σύστημα στο οποίο δεν θα μπορούσαν να αλλοιωθούν οι χρονικές σφραγγίδες ηλεκτρονικών εγγράφων. Αυτό το σύστημα ήταν σχεδιασμένο με βάση τα δέντρα κρυπτογράφησης (Merkle tree).

Παρόλα αυτά, η πρώτη ολοκληρωμένη υλοποίηση του BlockChain πήρε μορφή το 2008 με το πρώτο κρυπτονόμισμα, το ονομαζόμενο Bitcoin. Η καθοριστική βελτίωση που έκανε το blockchain υλοποιήσιμο, και κατ' επέκταση την δημιουργία του πρώτου κρυπτονομίσματος, ήταν προσθήκη μπλοκ στην αλυσίδα εγγραφών χωρίς την απαίτηση για την υπογραφή της συναλλαγής που έλαβε μέρος, από μία αξιόπιστη τρίτη οντότητα (trusted third party).

Η ανάγκη για μία τρίτη οντότητα η οποία θα επικύρωνε κάθε συναλλαγή σε ένα δίκτυο, καθιστούσε οποιοδήποτε ηλεκτρονικό νόμισμα ή υπηρεσία μέχρι τότε επιρρεπή, καθώς υπήρχε η απαίτηση η οντότητα να είναι πλήρως αξιόπιστη και κεντρικοποιημένου χαρακτήρα.

Όλη η ιδέα και η υλοποίηση του πρώτου BlockChain συστήματος, όπως αυτό παρουσιάζεται στο αντίστοιχο whitepaper [5] (BitCoin: A Peer-to-Peer Electronic Cash System), προήλθε από το πρόβλημα της διπλής δαπάνης (double spending), ελάττωμα στα έως τότε συστήματα ηλεκτρονικών πληρωμών. Το πρόβλημα της διπλής δαπάνης, περιγράφει την κατάσταση στην οποία ένα ψηφιακό τεκμήριο μπορεί να δαπανηθεί περισσότερο από μία φορές. Για παράδειγμα, όπως και στο φυσικό χρηματικό σύστημα με τα πραγματικά νομίσματα έτσι και στα εικονικά, ένα νόμισμα μπορεί να πλαστογραφηθεί ή να αντιγραφεί, δημιουργώντας

έτσι έναν πληθωρισμό στις χρηματικές μονάδες και μειώνει την εμπιστοσύνη των χρηστών στο σύστημα.

Η λύση είναι ίσως προφανής. Όπως και στα φυσικά χρηματοοικονομικά συστήματα, έτσι και σε ένα ηλεκτρονικό σύστημα, μπορούμε να εισάγουμε μία αξιόπιστη τρίτη οντότητα η οποία μπορεί να επιβεβαιώσει δεδομένης μίας συναλλαγής ποια τεκμήρια (tokens), έχουν δαπανηθεί. Αυτό όμως αντιπροσωπεύει επίσης και μοναδικό σημείο αποτυχίας (single point of failure), τόσο από πλευράς διαθεσιμότητας αλλά και από άποψης εμπιστοσύνης των χρηστών στο σύστημα. Θα ήταν επομένως λογικό να αφαιρεθεί από την εξίσωση ο παράγοντας αυτός, και θα ήταν το πλέον λογικό για κάποιον να σκεφτεί τα κατανεμημένα και αποκεντρχικοποιημένα συστήματα. Η μοναδική δυσκολία ήταν το πώς μπορούν να εξακριβωθούν οι συναλλαγές σε ένα δίκτυο χωρίς την παρουσία μίας τρίτης οντότητας.

Η παραπάνω λύση και υλοποίηση δόθηκε και περιγράφηκε ολοκληρωμένα, στο προαναφερθέν whitepaper για το Bitcoin. Οι έννοιες smart contract και ledger, δηλαδή οι έξυπνες συμβάσεις και οι κατανεμημένες εγγραφές, άλλαξαν όλο τον τρόπο με τον οποίο θα μπορούσε να γίνει πλέον, μία επιτυχής ηλεκτρονική συναλλαγή οι οποία να βασίζεται πλήρως σε ένα ομότιμο δίκτυο από χρήστες. Επομένως, η ανάγκη για μία τρίτη εξουσιοδοτημένη και αξιόπιστη οντότητα η οποία θα επικυρώνει τις συναλλαγές στο δίκτυο εξαλείφεται, καθώς οι ίδιοι οι χρήστες του δικτύου λειτουργούν και ως επικυρωτές των συναλλαγών που έχουν μεταξύ τους.

Το Blockchain όπως καταλαβαίνουμε και από την λέξη, είναι μία αλυσίδα από μπλοκ με χρονικές σφραγίδες, τα οποία συνδέονται με κρυπτογραφικά hashes. Επί της ουσίας το Blockchain θα μπορούσε να χαρακτηριστεί ως μία κατανεμημένη βάση η οποία διατηρεί μία συνεχώς αυξανόμενη λίστα από αμετάβλητες εγγραφές οι οποίες προκύπτουν ομόφωνα με βάση έναν αλγόριθμο (consencus) από τους χρήστες του δικτύου και απαρτίζουν τα λεγόμενα μπλοκ από αλυσίδες.

Το πρώτο πείραμα για ένα αποκεντρικοποιημένο ηλεκτρονικό νόμισμα όχι μόνο πέτυχε αλλά οι έννοιες και η υλοποίηση του άνοιξε νέους ορίζοντες για τον τρόπο με τον οποίο μπορεί να διαχειριστεί οποιαδήποτε είδους πληροφορία σε ένα ομότιμο δίκτυο. Η τεχνογνωσία που παρείχαν οι κρυπτογραφημένες κατανεμημένες εγγραφές, δεν σταμάτησε μόνο σε ένα απλό ηλεκτρονικό νόμισμα που υλοποιεί ένα αποκεντρωμένο σύστημα χρηματικών συναλλαγών. Στην πορεία μπόρεσαν να ενσωματωθούν μέσα σε ένα τέτοιο σύστημα, και διάφορα άλλα περιουσιακά στοιχεία τα οποία περιγράφουν έννοιες όπως δάνεια, ορισμένες αρχικές χρηματικές καταστάσεις, και διάφορα άλλα συμβόλαια, τα οποία αναφέρονται ως έξυπνα συμβόλαια (smart contracts). Όμως το blockchain και πάλι δεν σταμάτησε εκεί. Οι κατανεμημένες εγγραφές κρυπτογράφησης, παρέχουν μία υποκείμενη τεχνολογία η οποία μπορεί να αποκοπεί από τα κρυπτονομίσματα και να χρησιμοποιηθεί σε όλυ του είδους τις συναλλαγές πληροφορίας και αρχείων μεταξυ των μελών ενός ομότιμου δικτύου. Τα κεντρικοποιημένα συστήματα αποτυγχάνουν στην παροχή ασφάλειας και ιδιωτικότητας των δεδομένων αν το εκάστοτε κεντρικό σύστημα υποστεί βλάβη. Δεδομένου αυτού αλλά και της ανάγκης των συστημάτων για μεγάλη διαθεσιμότητα και επεκτασιμότητα, δύο χαρακτηριστικά που υπάρχουν εξ ορισμού στο blockchain, μπορεί να γίνει λόγος για πολλές εφαρμογές που μπορούν να υλοποιηθούν

και να ενσωματωθούν ένα blockchain και να εκμεταλλεύονται τόσο τα παραπάνω όσο και την ασφάλεια του blockchain στην παραβίαση δεδομένων, επιτυγχάνοντας έτσι μία ασφαλή διαλειτουργικότητα μεταξύ των εφαρμογών.

Το blockchain θα μπορούσε να χαρακτηριστεί ως μία κατανεμημένη βάση η οποία διατηρεί μία συνεχώς αυξανόμενη λίστα από εγγραφές, τα ονομαζόμενα μπλοκ (blocks). Η αρχική πρακτική εμφάνιση αυτής της τεχνολογίας ήταν όπως προαναφέρθηκε ένα πετυχημένο πείραμα ηλεκτρονικού νομίσματος, το Bitcoin [5]. Παρόλα αυτά το βασικό κομμάτι του blockchain, δεν ήταν η εφαρμογή και η λειτουργία του στα ηλεκτρονικά συστήματα νομισμάτων. Κατά βάση καθιέρωσε έναν διαφορετικό τρόπο με τον οποίο μπορεί να λειτουργήσει με ασφάλεια ένα διομότιμο δίκτυο και τα πορίσματα αυτής της εφαρμογής έγιναν εμφανή λίγο χρόνια αργότερα από την εμφάνιση του πρώτου κρυπτονομίσματος.

Το blockchain δεν είναι ένα απλό p2p δίκτυο το οποίο παρέχει μόνο την δυνατότητα καταγραφής νομισματικών συναλλαγών μεταξύ των ομότιμων χρηστών. Κατά την εξέλιξή του, οι χρήστες μπορούσαν να επιπλέον να καταγράφουν στους ledgers και άλλες έννοιες όπως δάνεια ή συμβόλαια, πέραν των δυνατοτήτων που προσέφερε το bitcoin. Έτσι προέκυψε η ιδέα των έξυπνων συμβολαίων (smart contracts). Αλλά οι τεχνολογίες που μπορούσε να υλοποιήσει το blockchain και πάλι δεν σταμάτησαν εκεί. Οι κατανεμημένες εγγραφές μας δίνουν μια υποκείμενη τεχνολογία η οποία μπορεί να αποκοπεί από τα κρυπτονομίσματα και να χρησιμοποιηθεί σε όλου του είδους τις ηλεκτρονικές συναλλαγές μεταξύ των μελών ενός διομότιμου δικτύου. Αυτό συμβαίνει γιατί τα κεντρικοποιημένα συστήματα αποτυγχάνουν στην παροχή ασφάλειας και ιδιωτικότητας των δεδομένων στην περίπτωση που το κεντρικό εκάστοτε σύστημα υποστεί οποιαδήποτε βλάβη. Όλα τα παραπάνω και λόγω της ανάγκης των υπολογιστικών συστημάτων για μεγάλη διαθεσιμότητα και επεκτασιμότητα, υπάρχουν πολλές εφαρμογές οι οποίες θα ωφελούνταν από την ενσωμάτωσή τους σε μία αρχιτεκτονική βασισμένη στο blockchain. Αυτό συμβαίνει γιατί εξ ορισμού το βλοςκςηαιν προσφέρει διαθεσιμότητα πληροφορίας και επεκτασιμότητα στην υποδομή της. Επίσης οι εφαρμογές μπορούν με αυτόν τον τρόπο να εκμεταλλεύονται και την ασφάλεια του blockchain στην παραβίαση δεδομένων, επιτυγχάνοντας έτσι μία διαλειτουργικότητα μεταξύ των εφαρμογών και των χρηστών του.

Για αυτό το λόγο και σε αυτή την εργασία, διασυνδέουμε τον πυρήνα ενός κεντρικοποιημένου συστήματος, δηλαδή τους κανόνες προώθησης ενός SDN δικτύου με κεντρικοποιημένους ελεγκτές, με ένα σύστημα που λειτουργεί εξ ολοκλήρου και βασίζει την λειτουργία του στην ιδέα των κατανεμημένων εγγραφών σε ένα διομότιμο δίκτυο.

3.2.1 Βασική δομή και λειτουργία του Blockchain

Για την χρήση του Blockchain, αρχικά είναι απαραίτητη η δημιουργία ενός P2P δικτύου, στο οποίο όλοι οι κόμβοι να κάνουν χρήση του Blockchain. Κάθε κόμβος του δικτύου, λαμβάνει και έχει στην κατοχή του δύο κλειδιά, ένα δημόσιο, το οποίο χρησιμοποιείται από τους ομότιμους κόμβους για την κρυπτογράφηση των μηνυμάτων που στέλνουν στους άλλους κόμβους, και ένα ιδιωτικό το οποίο επιτρέπει σε κάθε κόμβο να αποκρυπτογραφεί και να διαβάζει αυτά τα μηνύματα, όπως ακριβώς και στην ασύμμετρη κρυπτογραφία. Πρακτικά τα κλειδιά αυτά έχουν έναν ελαφρώς διαφορετικό ρόλο στις blockchain υλοποιήσεις. Το ιδιωτικό κλειδί χρησιμοποιείται για την υπογραφή των συναλλαγών που λαμβάνουν μέρος στο δίκτυο, ενώ το δημόσιο κλειδί χρησιμοποιείται για να καθορίσει μία μοναδική διεύθυνση για κάθε κόμβο στο δίκτυο. Όταν έναν κόμβος πραγματοποιεί μία συναλλαγή στο δίκτυο, την υπογράφει και έπειτα την δημοσιεύει στους γειτονικούς κόμβους του (one-hop peers). Το γεγονός ότι η συναλλαγή υπογράφεται με ένα μοναδικό ιδιωτικό κλειδί, πιστοποιεί αυτόματα την αυθεντικότητα της συναλλαγής (μόνο ο κόμβος με το κλειδί μπορεί να βάλει αυτή την υπογραφή, προφανώς με το κλειδί του). Επίσης αυτομάτως προσφέρει ακεραιότητα καθώς σε περίπτωση που υπάρξει ένα λάθος κατά την μεταφορά των δεδομένων στο δίκτυο και τα δεδομένα της συναλλαγής αλλοιωθούν, η πληροφορία δεν θα μπορεί να αποκρυπτογραφηθεί και να χρησιμοποιηθεί. Καθώς έπειτα οι γειτονικοί κόμβοι παραλαμβάνουν την συναλλαγή του παραπάνω κόμβου, πιστοποιούν ότι είναι έγκυρη, πριν την διαφημίσουν με την σειρά τους στους επιμέρους γειτονικούς κόμβους, συμβάλλοντας έτσι στην εξάπλωση της πληροφορίας σε όλο το δίκτυο.

Έτσι οι συναλλαγές οι οποίες διαδίδονται σε όλο το δίκτυο και θεωρούνται έγκυρες, ταξινομούνται με βάση την χρονική τους σφραγίδα και συγκεντρώνονται από μία ειδική κατηγορία κόμβων (miners) σε ένα μπλοκ το οποίο έχει επίσης χρονική σφραγίδα. Βασικό κομμάτι ενός blockchain, είναι ο τρόπος με τον οποίο εκλέγονται οι παραπάνω κόμβοι, καθώς και τα δεδομένα που εν τέλει καταγράφονται στο τελικό μπλοκ της αλυσίδας. Τα δύο παραπάνω, αναφέρονται ως ο αλγόριθμος ομοφωνίας (consencus algorithm) του Blockchain. Τέλος όσα μπλοκ εισέρχονται στην αλυσίδα, γίνονται broadcast πάλι πίσω στο δίκτυο και στο οποίο όλοι οι κόμβοι επικυρώνουν οτι το κάθε μπλοκ περιέχει έγκυρες συναλλαγές καθώς και ότι κάθε μπλοκ αναφέρεται και συνδέεται με το προηγούμενό του, κάνοντας χρήση του αντίστοιχου hash που περιέχει το νέο μπλοκ. Αν οι παραπάνω συνθήκες δεν επαληθευθούν, τότε το μπλοκ απορρίπτεται.

Αν από την άλλη, οι συνθήκες εγκυροποιηθούν επιτυχώς, οι κόμβοι εντάσσουν το μπλοκ στην αλυσίδα, ανανεώνοντας ο καθένας το αρχείο των συναλλαγών τους (ledger). Με αυτόν τον τρόπο, σε μία τελική κατάσταση, όλοι οι κόμβοι έχουν μία κατάσταση, για μία έγκυρη εικόνα των συναλλαγών σε ένα δίκτυο. Επιπλέον η πληροφορία που μπορεί να ενταχθεί στις αλυσίδες από μπλοκ μπορεί να είναι οποιουδήποτε τύπου καθιστώντας έτσι το blockchain, ικανό να ενσωματωθεί σε οποιουδήποτε είδους υπηρεσία η οποία μπορεί να έχει ανάγκη για ακεραιότητα, εμπιστευτικότητα, και ασφάλεια των δεδομένων της.

3.2.2 Οι Διαφορετικοί Τύποι Blockchain

Υπάρχουν διαφορετικές υποκατηγορίες σε μία υλοποίηση ενός Blockchain, με βάση τα δεδομένα που διαχειρίζεται, την διαθεσιμότητα αυτών καθώς και τις πράξεις που μπορούν να πραγματοποιηθούν από τους χρήστες-κόμβους του ομότιμου δικτύου. Ένα Blockchain μπορεί να είναι δημόσιο ή ιδιωτικό (public/private) καθώς και με άδεια ή αδέσμευτο (permissioned/permissionless).

Όσον αφορά τα κρυπτονομίσματα, ο επιπλέον διαχωρισμός μεταξύ σε public / private και permissionless/permissioned μπορεί να είναι περιττός καθώς τα public σημαίνουν ταυτόχρονα και permissionless και τα πριατε αντίστοιχα permissioned, αλλά αυτό δεν συμβαίνει σε υλοποιήσεις του blockchain σε άλλου είδους εφοαρμογές πέρα από τα κρυπτονομίσματα. Είναι σημαντικό να γίνει ο διαχωρισμός μεταξύ, εξουδιοδότησης (authorization), που μεταφράζεται στο τι έχει δικαίωμα να κάνει ένας κόμβος στο δίκτυο (permissioned vs permissionless) και ελέγχου ταυτότητας (authentication), το οποίο μεταφράζεται στο ποιός κόμβος μπορεί να εισέλθει και να χρησιμοποιήσει το blockchain δίκτυο (private vs public).

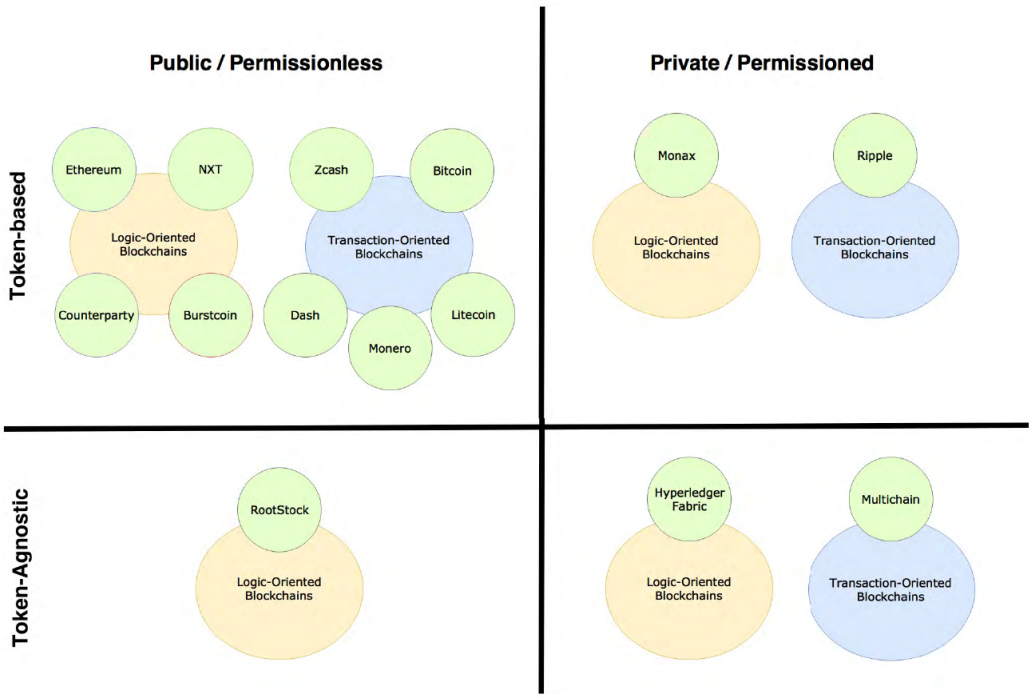
Στα δημόσια (public) Blockchains οποιοσδήποτε μπορεί να συμμετέχει στο δίκτυο, χωρίς την έγκριση από τρίτες οντότητες, και μπορεί να κατέχει ρόλο απλού κόμβου ή ως κόμβου έγκρισης συναλλαγών (validator) ή ως miner. Στα ιδιωτικά δίκτυα, ο ιδιοκτήτης επιτ ρέπει ή απορρίπτει την πρόσβαση ενός κόμβου στο δίκτυο. Πολλά ιδιωτικά δίκτυα είναι permissioned, αλλά το ότι ένα δίκτυο είναι ιδιωτικό δεν σημαίνει αυτόματα ότι είναι και permissioned. Για παράδειγμα ένας οργανισμός μπορεί να κάνει χρήση ενός ιδιωτικού δικτύου βασισμένο στο Ethereum Blockchain το οποίο όμως είναι permissionless. Παράδειγμα permissioned Blockchain δικτύου είναι το Hyperledger Fabric, με το οποίο θα ασχοληθούμε και εμείς στην παρούσα διπλωματική.

Επιπλέον τα Blockchains μπορούν να διαχωριστούν σε όσα στοχεύουν μόνο στην καταγραφή των ηλεκτρονικών συναλλαγών, όπως το BitCoin, ή σε όσα μπορούν να υποστηρίξουν και συγκεκριμένες λογικές όπως τα έξυπνα συμβόλαια (smart contracts). Επίσης υπάρχουν συστήματα που κάνουν χρήση τοκενς (όπως το Ripple), και άλλα που δεν τα χρησιμοποιούν (όπως το Hyperledger). Τέτοιου είδους tokens μπορεί να μην μεταφράζονται σε απαραίτητα σε κάποιο κρυπτονόμισμα αλλά αποδείξεις που πιστοποιούν ότι ορισμένα γεγονότα συνέβησαν σε ορισμένες χρονικές στιγμές.

Στο παρακάτω σχήμα φαίνεται μία συνολική εικόνα των διαφόρων τύπων Blockchain, μαζί με παραδείγματα υλοποιήσεων τους.

3.2.3 Κριτήρια για την χρήση του Blockchain στις σύγχρονες εφαρμογές

Παρόλο που οι κατανεμημένες εγγραφές αλυσίδας, προσφέρουν σημαντικά πλεονεκτήματα για τα δεδομένα που διαχειρίζεται ένα σύστημα σε σχέση με άλλα συστήματα, θα πρέπει να αναφερθεί ότι η χρήση τους δεν είναι πάντα θεμιτή και η καλύτερη για κάθε σενάριο. Αρκετές φορές ίσως είναι καλύτερη η χρήση μίας παραδοσιακής βάσης δεδομένων ή κάποιας άλλης τεχνολογίας κατανεμημένων εγγραφών (Distributed Ledger Technology – DLT), όπως οι



Σχήμα 3.7: Οι βασικοί τύποι BlockChain (Πηγή: [116])

εγγραφές Κατευθυνόμενων Ακυκλικών Γράφων (Directed Acyclic Graph – DAG Ledgers). Για να καθορίστεί κατά πόσο είναι κατάλληλη ή όχι η χρήση ενός Blockchain δικτύου για μία εφαρμογή, τα παρακάτω κριτήρια πρέπει να ληφθούν υπόψιν.

- ***Αποκεντροποίηση.*** Εφαρμογές που απαιτούν ένα αποκεντρωμένο σύστημα, ή δουλεύουν καλύτερα σε ένα τέτοιο σύστημα, ειδικά όταν δεν είναι εύκολη η εφαρμογή ενός αξιόπιστου κεντρικοποιημένουν συστήματος. Παρόλα αυτά σήμερα, πολλοί οργανισμοί και εφαρμογές, εμπιστεύονται τυφλά τρίτες οντότητες, όπως εταιρίες, κυβερνήσεις ή τράπεζες. Οπότε στην περίπτωση που υπάρχει αμοιβαία εμπιστοσύνη, το blockchain δεν είναι απαραίτητο.

- ***Peer-to-Peer συναλλαγές.*** Σε ορισμένες εφαρμογές, ειδικά στις εφαρμογές του IoT (Internet of Things), η πλειονότητα των συναλλαγών δεν συμβαίνει μεταξύ ομότιμων χρηστών, αλλά μεταξύ κόμβων (nodes) και πυλών (gateways), οι οποίες προωθούν την οποιαδήποτε επικοινωνία σε απομακρυσμένες οντότητες. Επομένως σε τέτοιου είδους εφαρμογές όπως δεν υπάρχει επικοινωνία και ανταλλαγή πληροφορίας σε ομότιμο επίπεδο χρηστών επίσης η εφαρμογή και η χρήση του blockchain πρέπει να εξετασθεί αναλόγως την περίπτωση.

- ***Συστήματα πληρωμών.*** Πολλές εφαρμογές χρειάζονται για την λειτουργία τους οικονομικές συναλλαγές, με τρίτες οντότητες, ενώ άλλες όχι. Παρόλο που τέτοιου είδους οικονομικές συναλλαγές μπορούν εύκολα να πραγματοποιηθούν με ένα παραδοσιακό σύστημα πληρωμών, τα τέλη συναλλαγών που επιβάλουν οι τρίτες οντότητες σε παραδοσιακές πληρωμές καθώς και η εμπιστοσύνη που πρέπει να υπάρχει σε αυτές τις οντότητες, καθιστούν τα παραδοσιακά συστήματα πιο επιρρεπή σε θέματα ασφαλείας από την εναλλακτική χρήση ενός blockchain δικτύου.
- ***Δημόσια καταχώρηση ακολουθιακών συναλλαγών.*** Πολλές εφαρμογές διαχειρίζονται δεδομένα τα οποία είναι απαραίτητο να φέρουν χρονική σήμανση και να αποθηκεύονται διαδοχικά. Στην περίπτωση αυτή η ενσωμάτωση του blockchain ταιριάζει επίσης απόλυτα. Βέβαια, τέτοιου είδους ανάγκες μπορεί να εκπληρούνται εύκολα και από παραδοσιακές βάσεις δεδομένων, ειδικά όταν η ασφάλεια δεν είναι μείζονος σημασίας ή είναι με κάποιον άλλο τρόπο εγγυημένη ή σε περίπτωση που η πιθανότητα για μία κακόβουλη επίθεση δεν είναι μεγάλη ή δεν αποτελεί παράγοντα για το σύστημα.
- ***Εύρωστο κατανεμημένο σύστημα.*** Παρόλο που η ανάγκη για ένα εύρωστο κατανεμημένο σύστημα δεν είναι αρκετή για να δικαιολογήσει την χρήση του blockchain σε αυτό, αν υπάρχει απουσία εμπιστοσύνης σε όποια οντότητα διαχειρίζεται το σύστημα αυτό, το blockchain καθίσταται πάλι μία υλοποίηση που προσφέρει λύση σε αυτο το σενάριο.
- ***Συλλογές συναλλαγών.*** Ορισμένες εφαρμογές κατά την εκτέλεση των συναλλαγών τους χρειάζεται να τηρούν αρχεία είτε για λόγους ανιχνευσιμότητας, είτε για λόγους ελέγχου. Τέτοιου είδους προβλήματα μπορούν εύκολα να λυθούν με την χρήση ενός μηχανισμού που επεκτείνει το Blockchain και ονομάζεται sidechain. Ουσιαστικά ένα sidechain επιτρέπει την καταγραφή πληροφορίας από ένα ή πολλά συστήματα σε ένα για κάθε σύστημα sidechain, το οποίο στην συνέχεια αποτελεί είτε το δικό του blockchain είτε γίνεται κομμάτι μιάς ευρύτερης αλυσίδας από διαφορετικά sidechains.

3.3 Hyperledger Fabric

Με την ανάπτυξη του blockchain και την αξιοποίηση του σε ένα ευρύτερο πεδίο εφαρμογών πέρα από την εξόρυξη δεδομένων και τα κρυπτονομίσματα, το Linux Foundation ανακοίνωσε πριν 4 χρόνια την δημιουργία του Hyperledger Project [63]. Στόχος του έργου αυτού είναι η προώθηση της διαεπαγγελματικής συνεργασίας με την ανάπτυξη πιο αξιόπιστων αλυσίδων εγγραφών και κατανεμημένων λεδγερ, με την ενσωμάτωση ανεξάρτητων ανοιχτών πρωτοκόλλων και προτύπων, ώστε να είναι σε θέση να υποστηριχτούν παγκόσμιες επιχειρηματικές συναλλαγές τύπου blockchain σε δίκτυα επιχειρήσεων. Με σκοπό να θεμελιώσει την ανάπτυξη εφαρμογών και λύσεων με ρυθμιζόμενη αρχιτεκτονική, το Fabric, ένα από τα projects του Hyperledger, επιτρέπει στα στοιχεία του όπως η ομοφωνία των χρηστών του ομότιμου δικτύου ή τα έξυπνα συμβόλαια (smart contracts) να είναι plug-and-play με άλλες υπηρεσίες, προσφέροντας έτσι διαλειτουργικότητα.

Συγκεκριμένα χρησιμοποιεί την τεχνολογία των κιβωτίων σε Docker (Docker contrainers) για φιλοξενήσει την λογική που περιλαμβάνει η εφαρμογής του συστήματος, η οποία ονομάζεται chaincode, δηλαδή αλυσίδα κώδικα. Για το Hyperledger Fabric, τα chaincodes είναι τα αντίστοιχα έξυπνα συμβόλαια (smart contracts) και απαρτίζουν τον πυρήνα του [64].

Ένα δίκτυο στο Hyperledger Fabric απαρτίζεται από ένα σύνολο κόμβων, όπου κάθε ένας επιτελεί διαφορετικό ρόλο στο δίκτυο. Αρχικά υπάρχουν οι ομότιμοι κόμβοι (peer nodes) [65], οι οποίοι εκτελούν τα έξυπνα συμβόλαια γνωστά ως chaincodes, έχουν πρόσβαση στα δεδομένα του ledger, καθώς επίσης εγκρίνουν τις συναλλαγές στο δίκτυο και διασυνδέουν τις εφαρμογές επάνω σε αυτό. Ακόμα το Fabric διαθέτει τους κόμβους διαχειριστών (orderer nodes), οι οποίοι εξασφαλίζουν την συνοχή και τη συνέπεια του blockchain, προσθέτοντας τις εγκριθείσες συναλλαγές στο ledger με μια καθορισμένη σειρά. Η υπηρεσία παραγγελιών (ordering service), η οποία αποτελείται από τους orderer nodes, δεν εκτελεί τις ίδιες τις συναλλαγές (endorsed transactions) με τους ομότιμους χρήστες του δικτύου, αλλά τις ταξινομεί κατά σειρά στο εκάστοτε μπλοκ και τις περνάει στο ledger [66]. Τέλος διαθέτει τις υπηρεσίες MSP (Membership Service Provider), που υλοποιούνται ως αρχή πιστοποίησης (Certificate Authority - CA), και διαχειρίζονται πιστοποιητικά που χρησιμοποιούνται για την εξακρίβωση ταυτότητας και των ρόλων των μελών [67]. Οι MSPs είναι στοιχεία τα οποία προσφέρουν μία αφαιρετική άποψη από όλους τους μηχανισμούς κρυπτογράφησης και τα πρωτόκολλα που επιτελούν την λειτουργία της έκδοσης και της επικύρωσης πιστοποιητικών και τον έλεγχο της ταυτότητας του χρήστη.

3.3.1 ChainCode

Τα chaincode (CC) είναι προγράμματα σε GoLang, Node.js ή Java που υλοποιούν μία καθορισμένη διεπαφή, με τα υπόλοιπα στοιχεία του blockchain. Ο αλυσιδωτός αυτός κώδικας, όπως μπορεί αλλιώς να ονομαστεί, τρέχει όπως είπαμε σε ένα Docker container, απομονωμένος από την διαδικασία που επικυρώνει τους ομότιμους χρήστες στο δίκτυο (endorsing peer process). Το chaincode αρχικοποιεί και διαχειρίζεται εν συνεχεία, την κατάσταση του ledger μέσω των συναλλαγών που υποβάλλονται από τις εφαρμογές. Επομένως ένα CC μπορεί να χρησιμοποιηθεί για την κατάθεση (invocation) μία συναλλαγής, την ενημέρωση (update) ή την ερώτηση (query) στον ledger για μία προτεινόμενη συναλλαγή (transaction). Σημαντικό είναι ότι τα CC λειτουργούν μέσα σε ορισμένα κανάλια (channels) [68], και ένας τέτοιος κώδικας μπορεί να επικαλεστεί κάποιον άλλο cc, είτε στο ίδιο κανάλι με αυτόν είτε σε διαφορετικό κανάλι, με σκοπό να αποκτήσει πρόσβαση στην κατάστασή του, δηλαδή την κατάσταση ενός ξεχωριστού blockchain. Στην περίπτωση αυτή, δεν υπάρχει δικαίωμα εγγραφής και ελέγχων επικύρωσης κατάστασης (state validation checks) και επιτρέπονται μόνο ερωτήσεις στον ledger. Δηλαδή ένα ςς δεν μπορεί να τροποποιήσει ένα άλλο. Οι καταστάσεις οι οποίες δημιουργούνται από ένα cc καλύπτονται αποκλειστικά από τον ίδιο τον κώδικα και κάθε άλλο cc μπορεί μόνο να έχει πρόσβαση στην κατάσταση του πρώτου, φυσικά με την κατάλληλη άδεια. Με πιο απλά λόγια, το παραπάνω μας επιτρέπει να υπάρχουν πολλοί διαφορετικοί ledgers στο δίκτυο, που λειτουργούν για διαφορετικό σκοπό. Κατά μία έννοια μπορούμε να πούμε ότι ένα

cc είναι υπεύθυνο για την κατάσταση ενός ledger και μόνο, όπου κάθε ledger ορίζει και από ένα διαφορετικό blockchain. Προκύπτει ως συμπέρασμα πώς τα cc διαχειρίζονται μία επιχειρηματική λογική η οποία έχει προσυμφωνηθεί από τα μέλη του δικτύου. Επομένως η λογική των chaincodes αντικαθιστά στο Hyperledger Fabric τα έξυπνα συμβόλαια όπως αυτά υπάρχουν στο Ethereum [69].

3.3.2 Chaincode Lifecycle

Το API του Hyperledger, επιτρέπει την αλληλεπίδραση με τους διάφορους κόμβους σε ένα δίκτυο blockchain το οποίο όπως αναφέρθηκε αποτελείται από τους ομότιμους χρήστες, τους διαχειριστές, και τους MSP's. Επίσης επιτρέπει τη συσκευασία (packaging), την εγκατάσταση (installation), τη δημιουργία (instantiation) και την αναβάθμιση στιγμιότυπων (upgrade) στο CC των αναγνωρισμένων ομότιμων κόμβων, όπως τα ονομάζει το HLF [70], τις οποίες θα εξηγήσουμε εν συνεχεία. Το Hyperledger API, μπορεί να προσπελαστεί απευθείας μέσω της γραμμής εντολών, την οποία και θα χρησιμοποιήσουμε.

Οι τέσσερις εντολές που χρησιμοποιούνται για την διαχείριση του κύκλου ζωής ενός chaincode είναι οι παρακάτω:

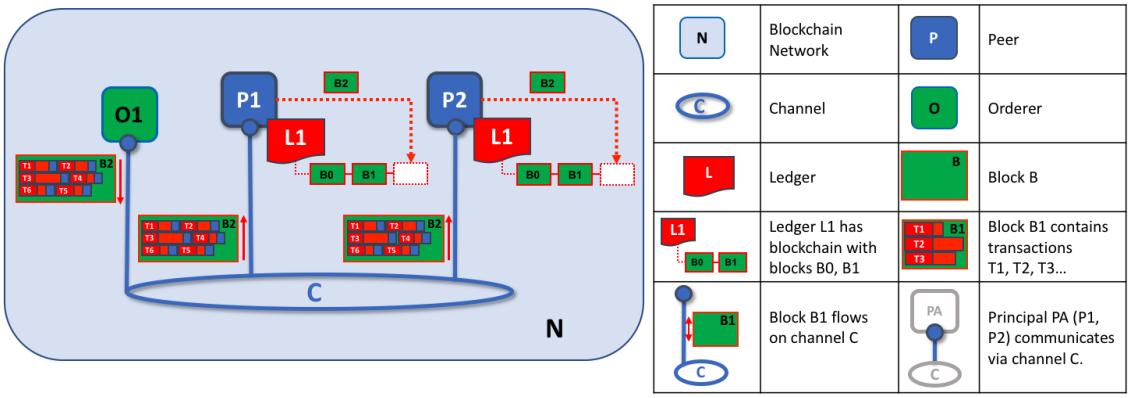
- Το packaging του κώδικα,
- Η εγκατάσταση (installation),
- Η δημιουργία ή αρχικοποίηση με μία αρχική κατάσταση (instantiation) που ορίζεται από το εκάστοτε δίκτυο,
- Η αναβάθμιση για την οποιαδήποτε αλλαγή κατάστασης του (upgrade)

Αφού εγκατασταθεί το chaincode και αρχικοποιηθεί ο τύπος του με επιτυχία, στην συνέχεια είναι ενεργός (δηλαδή λειτουργεί) και μπορεί να επεξεργαστεί συναλλαγές των κόμβων του blockchain μέσω της συναλλαγής ινοκε. Ένας αλυσιδωτός κώδικας μπορεί να αναβαθμιστεί οποιαδήποτε στιγμή μετά την εγκατάσταση του. Τα παραπάνω θα αναλυθούν λεπτομερώς με παραδείγματα στην υλοποίησή που θα παρουσιάσουμε στην συνέχεια.

3.3.3 Οντότητες και στοιχεία δικτύου Hyperledger Fabric

Το δίκτυο που έχει σχεδιαστεί και υλοποιηθεί παρουσιάζει την καταγραφή των κανόνων δρομολόγησης (flow rules) του πρωτοκόλλου OpenFlow σε ένα δίκτυο blockchain βασισμένο στο OpenProject του Hyperledger. Η ενσωμάτωση μίας blockchain υποδομής στον πυρήνα ενός δικτύου καθορισμένου από το λογισμικό, προσφέρει μία κατανεμημένη καταγραφή των κανόνων δρομολόγησης, η οποία με την σειρά της προσφέρει εγκυρότητα και εμπιστοσύνη στον/στους διαχειριστές του SDN δικτύου. Επίσης με την χρήση των αλυσιδωτών κατανεμημένων εγγραφών, γίνεται εφικτή η ακεραιότητα κάθε κανόνα OpenFlow που εισάγεται σε αυτό και γίνεται validate από τους ομότιμους χρήστες μέσω του chaincode. Έτσι επιτυγχάνεται μία πιο ασφαλή υλοποίηση ενός SDN δικτύου και λαβάνεται υπόψιν η εμπορευσιμότητα των

Openflow κανόνων από τους δρομολογητές του SDN δικτύου. Στις μέχρι τώρα υλοποιήσεις οι κανόνες γινόντουσαν propagated από τους controllers στα switches, με τους τελευταίους να δέχονται οποιαδήποτε αλλαγή χωρίς κάποιο validation των κανόνων. Το ίδιο εξακολουθεί να συμβαίνει και στην υλοποίηση μας, με την διαφορά ότι πλέον οι κανόνες σημειώνονται στους ledgers των ομότιμων χρηστών του Hyperledger δικτύου. Ένα σχεδιάγραμμα με τις οντότητες του δικτύου που απαρτίζουν το κομμάτι του blockchain, φαίνεται και αναλύεται παρακάτω:



Σχήμα 3.8: Δομή P2P blockchain δικτύου [66] (Πηγή: [65]

Όπως φαίνεται στην παραπάνω εικόνα το Blockchain δίκτυο, αποτελείται από τις παρακάτω οντότητες:

- Δύο peer κόμβους οι οποίοι διατηρούν την κατάσταση του δικτύου και κρατούν από ένα αντίγραφο του ledger. Υπάρχουν δύο διαφορετικού τύποι από peers σε ένα blockchain δίκτυο όπως αυτό ορίζεται από το HLF [65].

  1. **Endorsing peers** ή endorsers. Είναι κόμβοι οι οποίοι προσομοιώνουν και υποστηρίζουν τις συναλλαγές στο δίκτυο.
  2. **Commiting peers** ή commiters. Είναι κόμβοι οι οποίοι επαληθεύουν τα αποτελέσματα των συναλλαγών και τα επικυρώνουν πριν αυτές ταξινομηθούν και μπουν στο blockchain από τους orderers.

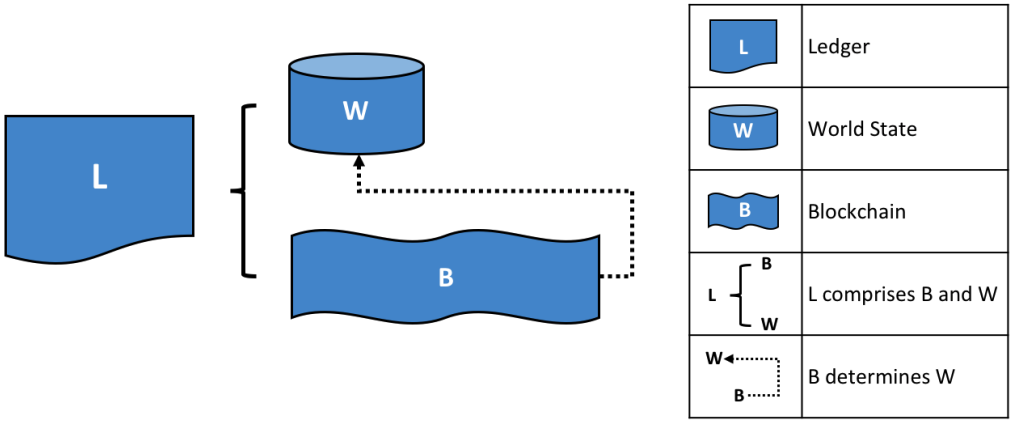
  Οι endorsing peers είναι μία ειδική υποκατηγορία των commiters και υπάρχει μία επικάλυψη μεταξύ των δύο με την έννοια ότι και τα δύο είδη κόμβων κάνουν commit blocks στον ledger. Στην περίπτωσή μας έχουμε έναν endorser και έναν commiter, αλλά όπως αναφέρθηκε και παραπάνω και οι δύο peers κάνουν commit transactions στο δίκτυο.

- Την υπηρεσία παραγγελιών (Ordering Service) η οποία μπορεί να αποτελείται από έναν κόμβο, όπου στην περίπτωση αυτή λέμε ότι έχουμε solo ordering service, είτε από περισσότερους (σε Kafka ή Simplified Byzantine Fault Tolerant υλοποιήσεις) οι οποίοι παρέχουν μία υπηρεσία παραγγελιών που είναι ανθεκτική σε σφάλματα. Στην περίπτωσή

μας, η ordering υπηρεσία για λόγους απλούστευσης αποτελείται από έναν μόνο κόμβο, τον O1 όπως φαίνεται στο παραπάνω σχήμα.

- Τα κανάλια επικοινωνίας. Στην περίπτωσή μας είναι μόνο ένα αλλά όπως είπαμε και σε προηγούμενη ενότητα μπορεί να είναι και παραπάνω καθώς και να επικοινωνούν μεταξύ τους. Η χρήση του είναι για την επικοινωνία των υπόλοιπων κόμβων του blockchain δικτύου, όπως φαίνεται άλλωστε και στο σχήμα. Το κανάλι συνδέει τον orderer με τους peers, και αποτελεί το μέσο πάνω στο οποίο μεταφέρεται η πληροφορία των block των συναλλαγών.

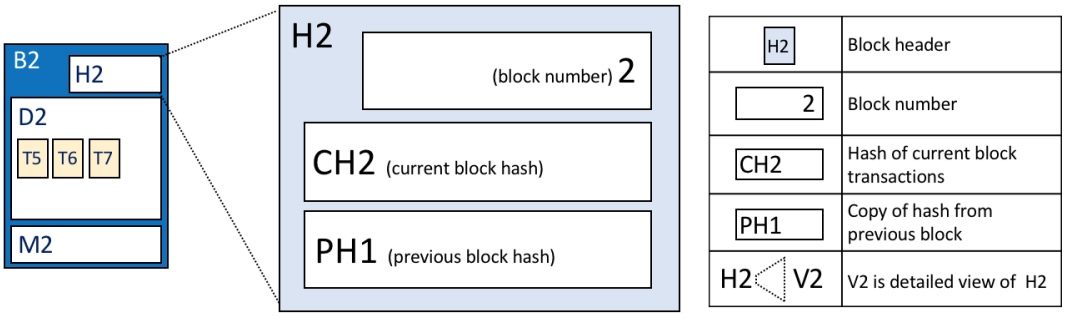
- Τον ledger (L1). Οι ledgers είναι αρχεία, παρόμοιας λογικής με αυτή μιας βάσης δεδομένων, στα οποία καταγράφονται οι συναλλαγές του δικτύου [71]. Αποτελούνται από δύο επιμέρους στοιχεία, το ίδιο το Blockchain και το λεγόμενο World State, που είναι η τρέχουσα κατάσταση του Blockchain πριν αυτό ανανεωθεί σε κάθε transaction. Ουσιαστικά το world state αναφέρεται και καθορίζεται από το ίδιο το Blockchain, αμέσως όταν οι νέες εγγραφές γίνουν commit. Για να υπάρχει στο δίκτυο μία αποκεντρωμένη λογική, κάθε κόμβος έχει στην κατοχή του ένα αντίγραφο του ledger, το οποίο ανανεώνεται κάθε φορά που ο orderer ταξινομεί συναλλαγές στα block και τις προσθέτει στο blockchain. Μετά την ανανέωση αυτή, οι αλλαγές γίνονται commit και εισέρχονται πλέον στο world state όπως επεξηγεί το παρακάτω σχήμα.



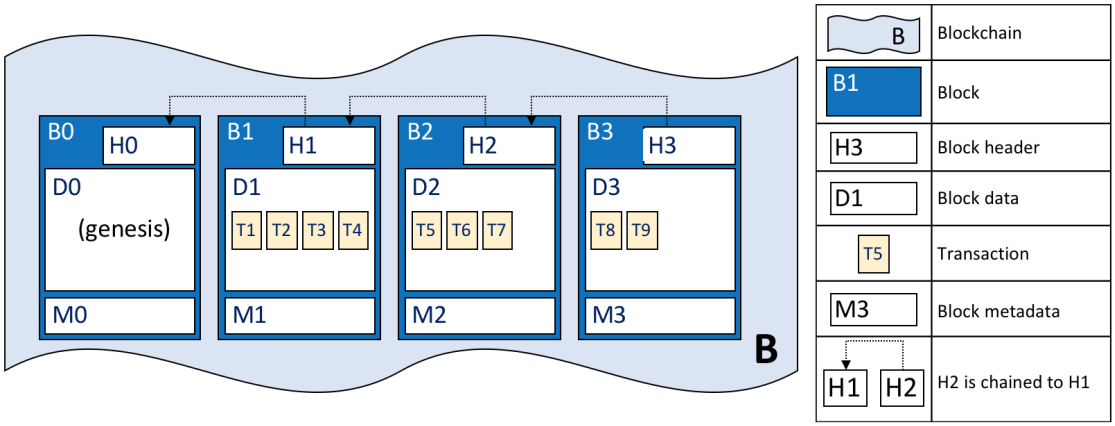
Σχήμα 3.9: Δομικά στοιχεία του Ledger [71] (Πηγή: [71]

- Τα blocks (B1, B2, etc), τα οποία όπως αναφέραμε περιέχουν τις χρονικά ταξινομημένες συναλλαγές του δικτύου. Αξίζει να αναφερθεί ότι, κάθε block εκτός από την πληροφορία των συναλλαγών, την αντίστοιχη χρονική σφραγίδα του, έναν αύξοντα αριθμό και τις επικεφαλίδες του , περιέχει ένα ακόμα σημαντικό για την λειτουργία του στοιχείο, όπως φαίνεται στο παρακάτω σχήμα 8. Το στοιχείο αυτό είναι ένα hash από τις υπάρχουσες στο block συναλλαγές, το οποίο συνδέεται άμεσα και περιέχει το αντίστοιχο hash του προηγούμενου block όπως ακριβώς ένα Merkle Tree [72], καθώς προφανώς και τον

αύξοντα αριθμό του block. Αυτή η αλυσίδα από hashes είναι που παρέχει στο blockchain την αμεταβλητότητα και σταθερότητα της πληροφορίας που περιέχει, και το καθιστά πρακτικά απαραβίαστο (tamper-proof).



Σχήμα 3.10: Ενδεικτικά δομικά στοιχεία ενός Block στο Blockchain (Πηγή: [71]



Σχήμα 3.11: Τα πρώτα blocks στο Blockchain (Πηγή: [71]

Επίσης σημαντικό στοιχείο του HLF Blockchain είναι, οι λεγόμενοι πελάτες (clients) όπως τους ονομάζει το HLF. Οι πελάτες είναι οποιουδήποτε είδους εφαρμογές οι οποίες προτείνουν συναλλαγές στο blockchain δίκτυο και επωφελούνται από αυτό και την λογική του, αποκτώντας έτσι ασφάλεια, σταθερότητα (immutability) και συνέχεια στην οποιαδήποτε πληροφορία συναλλάσουν με το Blockchain δίκτυο. Στην περίπτωσή μας η εφαρμογή αυτή είναι το SDN δίκτυο ως οντότητα και η πληροφορία που προτείνει ως συναλλαγές στο blockchain είναι οι OpenFlow κανόνες που ορίζουν οι controllers για την προώθηση των πακέτων στο SDN δίκτυο.

Κεφάλαιο 4

Υλοποίηση και Εργαλεία

4.1 Εισαγωγή

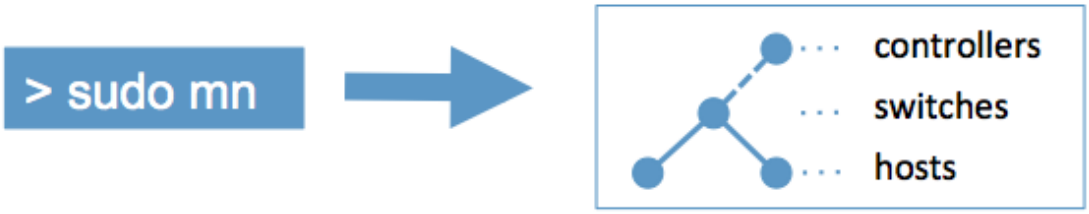
Τα εργαλεία τα οποία χρησιμοποιήθηκαν στην παρούσα διπλωματική για την υλοποίηση περιλαμβάνουν ολοκληρωμένες ad-hoc υπηρεσίες (όπως το Hypeledger) και software που επιτελεί τις ζητούμενες και επιθυμητές λειτουργίες, με έναν διαφορετικό σκοπό ο οποίος είναι και ο στόχος την διπλωματικής, όπως αυτός αναφέρθηκε στην ενότητα δύο.

4.2 Εργαλεία για το SDN

4.2.1 Mininet Software

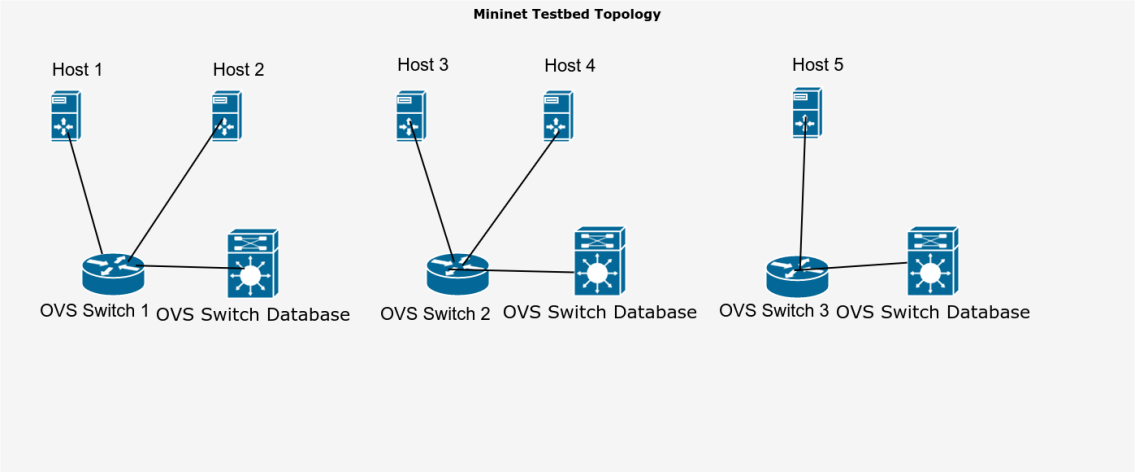
Αρχικά χρησιμοποιήθηκε το εργαλείο mininet [74]. Το mininet (mn) είναι ολοκληρωμένο λογισμικό ανοιχτού κώδικα γραμμένο σε Python2.7. Υλοποιεί ένα εικονικό περιβάλλον δικτύου, και ο κώδικάς του τρέχει σε τρία διαφορετικά στρώματα. Επομένως τρέχει κώδικα τόσο στον πυρήνα, όσο και κώδικα για τους δρομολογητές, αλλά και κώδικα εφαρμογών σε ένα μεμονωμένο μηχάνημα το οποίο μπορεί να είναι είτε Virtual Machine ή Cloud ή Native Machine) [75]. Η αλληλεπίδραση με το δίκτυο του προσομοιωτή γίνεται είτε μέσω της γραμμής εντολών (mininet CLI) αλλά υπάρχει και η δυνατότητα του Python API το οποίο παρέχεται.

Το mn αποτελείται για τον χρήστη του, από όλα τα βασικά στοιχεία οντοτήτων ενός SDN Δικτύου, όπως αυτά αναφέρθηκαν παραπάνω και είναι οι controllers, τα virtual switches (εικονικοί δρομολογητές) και οι hosts. Αποτελεί ένα πολύ ισχυρό εργαλείο στην ανάπτυξη εφαρμογών και στον πειραματισμό με το πρωτόκολλο OpenFlow, καθώς επί της ουσίας υλοποιεί ένα αρκετά παραμετροποιήσιμο SDN δίκτυο. Ακόμα παρόλο που έχει ενσωματωμένα όλα τα τρία στοιχεία που απαρτίζουν ένα SDN, παρέχει επίσης συνδεσιμότητα με άλλα εξωτερικά εργαλεία όπως για παράδειγμα ο ODL controller.



Σχήμα 4.1: Δομικά στοιχεία προσομοιωτή mininet (Πηγή: [117]

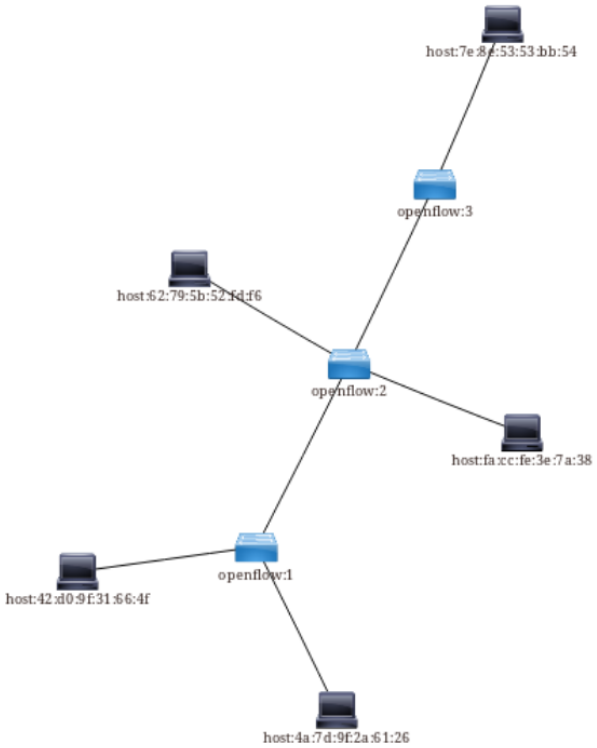
Στην περίπτωσή μας, η συμβολή του mininet στην παρούσα εργασία, έγινε στο SDN κομμάτι το οποίο απαρτίζεται από τους hosts και στα virtual switches, όπως αυτά φαίνονται στην εικόνα που ακολουθεί:



Σχήμα 4.2: SDN δίκτυο βασισμένο στο mininet (Πηγή: [117]

Ο τρόπος με τον οποίο δημιουργήθηκαν οι παραπάνω οντότητες είναι με χρήση της python, στην οποία δημιουργήθηκε μία custom τοπολογία δικτύου που συνδέει τις οντότητες μεταξύ τους όπως αναφέραμε στο κεφάλαιο 3. Εναλλακτικά θα μπορούσαμε να εκκινήσουμε το SDN δίκτυο, χωρίς την χρήση της python κατευθείαν από την γραμμή εντολών με την χρήση του προσομοιωτή και τις ανάλογες παραμέτρους για το δίκτυό.

Στο σχήμα 11 που ακολουθεί μπορούμε να επιβεβαιώσουμε για το περιβάλλον του mininet από το οποίο δημιουργήσαμε το SDN δίκτυό μας, μέσω του Web Service που μας παρέχει ο ODL controller, ότι όντως η τοπολογία μας είναι η προβλεπόμενη. Μπορούμε να δούμε τους κόμβους (nodes), τις συνδέσεις (links) και το δίκτυο (net). Οι παρενθέσεις περιλαμβάνουν επίσης εντολές στην γραμμή εντολών του mininet και έχουν αντίστοιχη λειτουργία με αυτήν που αναφέραμε. Φαίνεται λοιπόν παρακάτω η τοπολογία του SDN δικτύου όπως την καταλαβαίνει ο SDN controller που χρησιμοποιήσαμε.



Σχήμα 4.3: Τοπολογία SDN Δικτύου

4.2.2 OpenDayLight Controller

Όπως αναφέραμε και στο προηγούμενο κεφάλαιο, για το έλεγχο του SDN δικτύου, έγινε χρήση του ODL software. Το λογισμικό αυτό τρέχει ως ανεξάρτητη οντότητα η οποία συνδέεται με το SDN mininet δίκτυο μέσω του OpenFlow πρωτοκόλλου [76]. Συγκεκριμένα έγινε χρήση της έκδοσης 0.8.4 [77] και η σύνδεση με το SDN δίκτυο στο API του mininet έγινε όπως περιγράφεται στο [78], με την διαφορά ότι τα χαρακτηριστικά, features όπως τα ονομάζει το λογισμικό του controller, στην περίπτωσή μας είναι τα παρακάτω:

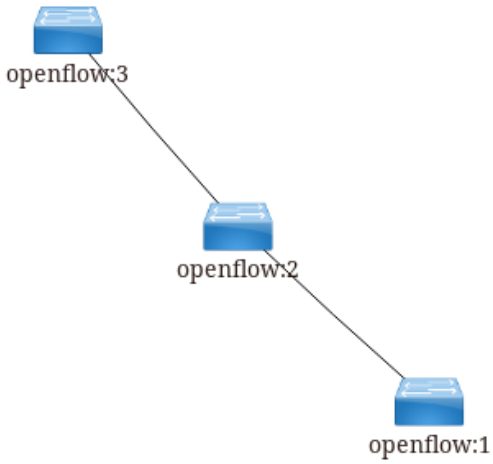
OpenDayLight Features
opendaylight-user@root >feature:install
odl-restconf-all
odl-l2switch-switch
odl-mdsal-apidocs
odl-dlux-core
odl-ovsdb-library
odl-mdsal-clustering
odl-openflowplugin-flow-services
odl-dluxapps-nodes
odl-dluxapps-topology
odl-dluxapps-yangui
odl-dluxapps-yangvisualizer
odl-dluxapps-yangutils
odl-dluxapps-yangman
odl-ovsdb-utils

Λόγω των παραπάνω χαρακτηριστικών μπορούμε να δούμε ορισμένα στοιχεία του SDN ελεγκτή από το Web interface το οποίο "σηκώνει" στην πόρτα 8181, και όπου περιέχονται όλα τα Yang Data Models τα οποία και χρησιμοποιεί το OpenDayLight [79] για το data store, δηλαδή την βάση του. Το παραπάνω interface μπορούμε να το προσπελάσουμε από τον σύνδεσμο http://83.212.173.124:8181/index.html με default credentials admin:admin, όπου η παραπάνω IP είναι αυτή του μηχανήματος στο οποίο υπάρχει η υπηρεσία του controller Επίσης χαρακτηριστικό του controller, με την βοήθεια των LLDP πρωτοκόλλου [80, 81], η γνώση που έχει για το SDN δίκτυο αρχικά και η γνώση που αποκτάει για την τοπολογία καθώς αρχίζουν να ανταλλάσσονται μηνύματα στο δίκτυο μεταξύ των switches. Συγκεκριμένα στην παρακάτω εικόνα βλέπουμε την τοπολογία όπως την γνωρίζει ο controller όταν συνδέεται αρχικά στο δίκτυο και δεν έχει γίνει κάποια ανταλλαγή πακέτων. Σε αντίθεση με την εικόνα 2, βλέπουμε ότι αρχικά ο ελεγκτής έχει γνώση μόνο για τους εικονικούς δρομολογητές (οι οποίοι απεικονίζονται με το όνομα openflow:X, όπου X αύξων αριθμός του switch), χωρίς να γνωρίζει ποιοί ή πόσοι hosts είναι συνδεδεμένοι σε αυτούς.

Ο ODL controller στην περίπτωσή μας τρέχει, όπως και τα στοιχεία του blockchain που θα δούμε στην συνέχεια, μεσα σε ένα docker container. Αυτό γίνεται με την βοήθεια ενός Dockerfile το οποίο περιέχει τις κατάλληλες παραμέτρους για να δημιουργηθεί η υπηρεσία του controller. Το container που δημιουργείται το ονομάζουμε odl_controller [Παράρτημα].

4.2.3 Πρωτόκολλο OpenFlow και αποθήκευση των κανόνων δρομολόγησης

Το πρωτόκολλο OpenFlow είναι βασικό κομμάτι της υλοποίησης καθώς οι κανόνες δρομολόγησης με τους οποίους θα ασχοληθούμε αποτελούν και τον πυρήνα του OpenFlow. Καθιστά



Σχήμα 4.4: Τοπολογία Δικτύου πριν τα Pings

δυνατή την αμοιβαία αλληλεπίδραση του ή των controllers, με το επίπεδο προώθησης των συσκευών του δικτύου, δηλαδή τα virtual switches (vs). Έτσι η δικτυακή κίνηση μπορεί να προσαρμόζεται καλύτερα στις μεταβαλλόμενες απαιτήσεις και ανάγκες του δικτύου. Οι δρομολογητές και τα switches μπορεί να είναι είτε εικονικές είτε φυσικές συσκευές (βασισμένες σε hypervisor).

Μέσω της OpenFlow διεπαφής ο ελεγκτής (controller) ωθεί τις επιθυμητές αλλαγές στον πίνακα ροής του ς επιτρέποντας στους διαχειριστές του δικτύου να διαχωρίζουν την κυκλοφορία, όπως επίσης και να ελέγχουν τις ροές για βέλτιστη απόδοση στο δίκτυο.

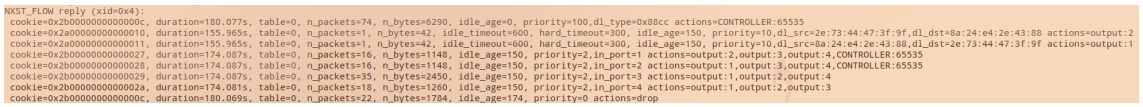
Όπως είδαμε και στην ενότητα 3.1.2 ο πυρήνας το OpenFlow διαχειρίζεται τους κανόνες δρομολόγησης οι οποίοι μπορούν να αναπαρασταθούν σε έναν πίνακα με βασικά πεδία τα παρακάτω:

- **Τα πεδία αντιστοίχισης (match fields):** για την αντιστοίχιση με πακέτα της κίνησης. Αυτά αποτελούνται από τη θύρα εισόδου και τις κεφαλίδες πακέτων, και προαιρετικά μεταδομένα που καθορίζονται από έναν προηγούμενο πίνακα.
- **Την Προτεραιότητα (priority):** για την προτεραιότητα της καταχωρημένης ροής έναντι των άλλων.
- **Τους μετρητές (counters):** οι οποίοι ενημερώνονται όταν αντιστοιχίζονται τα πακέτα με την αντίστοιχη ροή στο δίκτυο.
- **Τις οδηγίες (instructions):** για να τροποποιούν το σύνολο των ενεργειών ή την επεξεργασία των πακέτων.

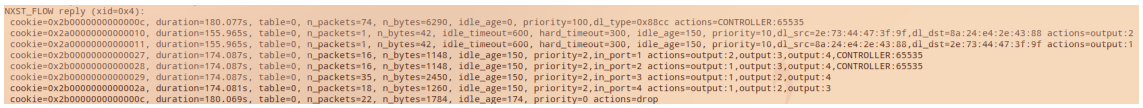
- **Τα χρονικά όρια (timeouts):** μέγιστο χρονικό διάστημα ή χρόνος αδράνειας πριν τη λήξη της ροής από την βάση των ς.

- **Τα cookies:** τα οποία είναι αδιαφανές τιμές δεδομένων που επιλέγονται από τον ελεγκτή. Μπορεί να χρησιμοποιηθεί από τον ελεγκτή για να φιλτράρει στατιστικές ροές, να τροποποιήσει και διαγράψει ροές. Δεν χρησιμοποιείται κατά την επεξεργασία των πακέτων.

Για το SDN δίκτυό μας παρακάτω φαίνονται δύο στιγμιότυπα ενδεικτικά για την βάση του virtual switch 2, πριν και μετά την εφαρμογή κίνησης στο δίκτυο (με την εντολή pingall). Προφανώς και στις δύο περιπτώσεις η σύνδεση με τον controller έχει γίνει, και τα παραπάνω πεδία των κανόνων δρομολόγησης φαίνονται όπως αυτά έχουν καταγραφεί στον πίνακα δρομολόγησης του VS 2.



Σχήμα 4.5: Κανόνες Δρομολόγησης πριν γίνει ανταλλαγή ping πακέτων μεταξύ των hosts στο δίκτυο



Σχήμα 4.6: Κανόνες Δρομολόγησης αφού γίνει ανταλλαγή ping πακέτων μεταξύ των hosts στο δίκτυο

Όλα τα παραπάνω καταγράφονται σε μία βάση η οποία έχει αντιστοιχία ένα προς ένα με κάθε δρομολογητή και η λειτουργία της βασίζεται στο μοντέλο client-server. Η βάση αυτή ονομάζεται OVSDB. Η OVSDB είναι ένα σύστημα βάσης δεδομένων προσβάσιμο από το δίκτυο και αποτελεί μέρος του λογισμικού Open-VirtualSwitch (OVS), το οποίο λειτουργεί ως εξυπηρετητής για τους εικονικούς δρομολογητές του SDN δικτύου [82]. Οι OVS switches για την διαχείριση τους διαθέτουν ένα μεγάλο μέρος εντολών αλλά οι βασικές είναι οι ακόλουθες τέσσερις [83]:

- **ovs-vsctl**: Χρησιμοποιείται για τη διαμόρφωση της βάσης δεδομένων (ovs-db) του ovs-vswitchd δαίμονα.

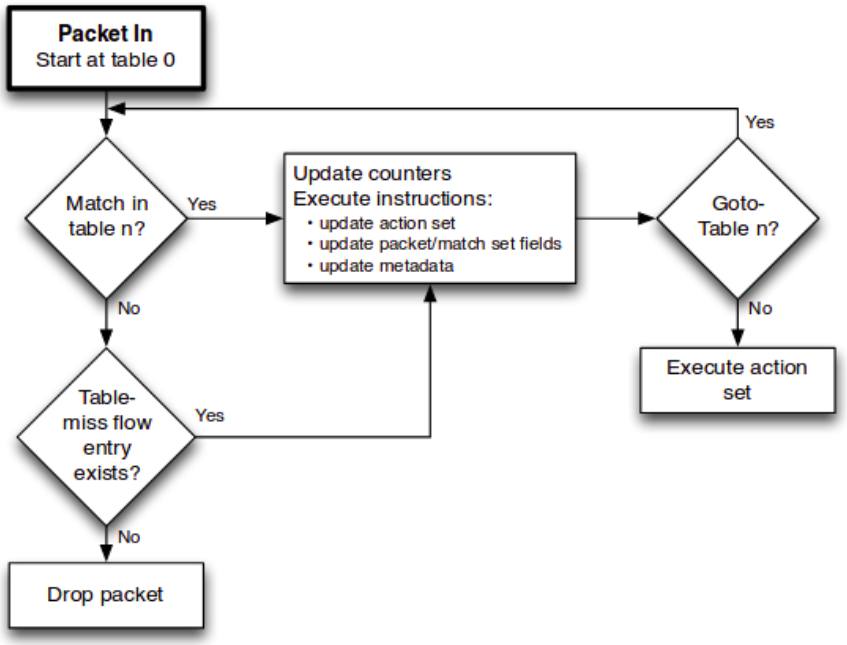
- **ovs-ofctl:** Ένα εργαλείο για την παρακολούθηση και τη διαχείριση των OpenFlow switches.

- **ovs-dpctl:** Χρησιμοποιείται για τη διαχείριση των Open-vSwitch datapaths.

- **ovs-appctl:** Χρησιμοποιείται για την αναζήτηση και τον έλεγχο των Open-vSwitch daemons.

Εν προκειμένω, η εντολή "ovs-ofctl dump-flows [switch]" χρησιμοποιήθηκε παραπάνω για την εξέταση των κανόνων δρομολόγησης του [switch].

Στην παρακάτω εικόνα φαίνεται το διάγραμμα ροής για κάθε πακέτο που εισέρχεται σε έναν εικονικό δρομολογητή. Κάθε πακέτο, αφού εισέλθει στον δρομολογητή, αντιστοιχίζεται με τον πρώτο κανόνα δρομολόγησης που θα βρεθεί, αν υπάρχει, σε όσους πίνακες δρομολόγησης έχει ο εικονικός δρομολογητής, αρχίζοντας από τον που έχει αύξοντα αριθμό μηδέν. Αν το πακέτο δεν αντιστοιχηθεί και υπάρχει κανόνας ο οποίο να επιβάλλει το δροπ των πακέτων, τότε το πακέτο γίνεται drop. Αν το πακέτο αντιστοιχηθεί, τότε γίνονται υπδατε οι μετρητές και εκτελείται το πεδίο των εντολών του κανόνα δρομολόγησης. Συνεπώς με αυτόν τον τρόπο, το πακέτο προωθείται κατάλληλα από τον δρομολογητή και φτάνει στον προορισμό του.



Σχήμα 4.7: Διάγραμμα Ροής που περιγράφει τη ροή πακέτων μέσω ενός OpenFlow (Πηγή: [118])

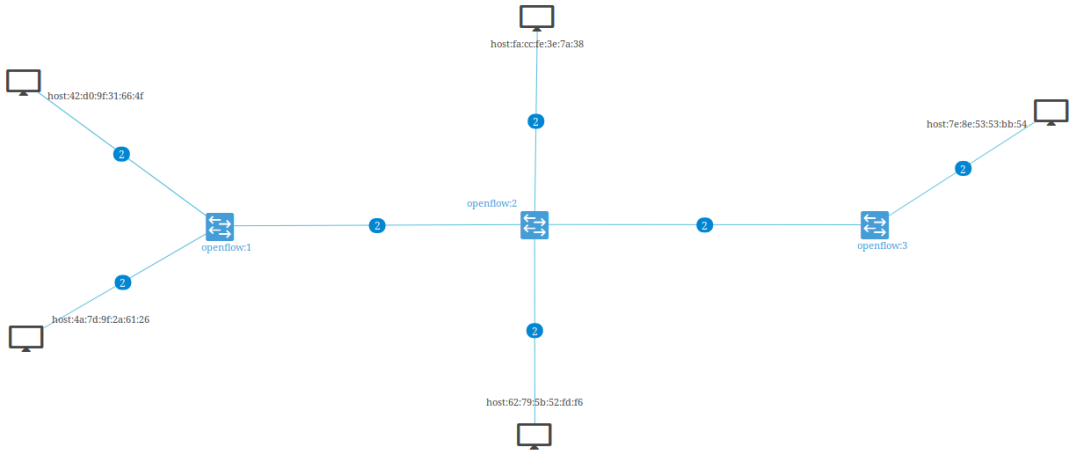
4.2.4 Openflow Manager

Για να έχουμε μία καλύτερη κατανόηση και μία πιο σφαιρική εικόνα σχετικά με το SDN δίκτυο, χρησιμοποιήθηκε η εφαρμογή OpenFlow Manager (OFM) [84]. Η εφαρμογή αυτή αναπτύχθηκε για να τρέχει πάνω από το ODL για να συλλέγει στοιχεία και να απεικονίζει τις τοπολογίες του OpenFlow δικτύου, να προγραμματίζει τις διαδρομές των πακέτων και να συγκεντρώνει στατιστικά για το SDN δίκτυο. Η εφαρμογή αυτή λειτουργεί μέσω του εργαλείου Grunt [85]. Ακούει ως εξυπηρετητής στην πόρτα 9000, στην οποία μεταφέρονται δεδομένα από το datastore του ODL controller [86]. Η εκκίνηση την εφαρμογής είναι απλή τροποποιώντας απλά το πεδίο baseURL στο αρχείο env.module.js, με την IP του μηχανήματος στο οποίο τρέχει η εφαρμογή. Το αρχείο αυτό βρίσκεται στο path ofm/src/common/config.

Οι βασικές επιλογές που προσφέρει το OFM όπως είναι διατεταγμένες στο Web User Interface της εφαρμογής, αποτελούνται από:

- Η προεπιλεγμένη καρτέλα “Βασική προβολή”, η οποία εμφανίζει την τοπολογία που αντιστοιχίζει τις συσκευές OpenFlow στο SDN δίκτυό και τους υπολογιστές (hosts) που είναι συνδεδεμένοι με αυτές.
- Η καρτέλα ¨Διαχείριση ροών¨, η οποία μπορεί να χρησιμοποιηθεί για τον προσδιορισμό του αριθμού των ροών για κάθε OpenFlow συσκευή στο SDN δίκτυό καθώς και για την προβολή μιας λίστας όλων των ροών που έχουν ρυθμιστεί αυτήν τη στιγμή, την προσθήκη, την τροποποίηση και τη διαγραφή των ροών.
- Η καρτέλα ¨Στατιστικά¨, η οποία παρέχει στατιστικά στοιχεία τόσο για τις ροές που έχουν διαμορφωθεί στο δίκτυό, όσο και για τις αντίστοιχες θύρες συσκευών.
- Η καρτέλα Ὑποδοχές’ παρέχει συνοπτικές πληροφορίες για τις συσκευές OpenFlow που διαχειρίζεται τοOFM.

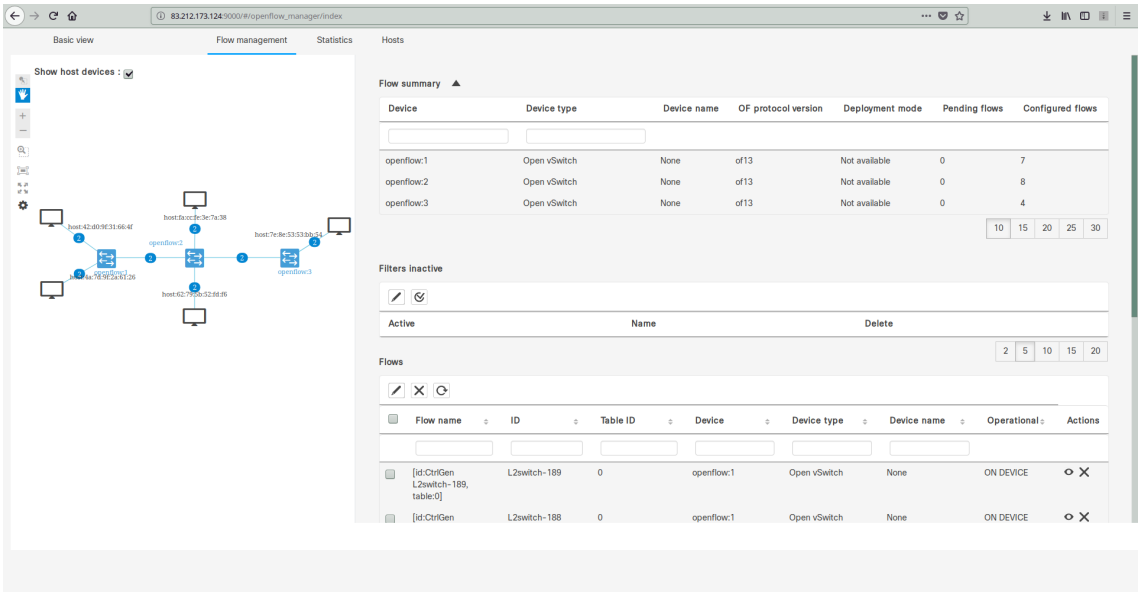
  Στις επόμενες δύο εικόνες φαίνεται αρχικά η ίδια τοπολογία με την εικόνα 2, αλλά μέσα από το διαχειριστικό περιβάλλον του OpenFlow Manager καθώς και το ίδιο το περιβάλλον του εξυπηρετητή που περιλαμβάνει όλα τα παραπάνω.



Σχήμα 4.8: Τοπολογία SDN δικτύου μέσω της εφαρμογής OFM

4.3 Εργαλεία για το Blockchain

Για την κατασκευή του Blockchain δικτύου χρησιμοποιήσαμε την Open-Source υλοποίηση που προσφέρει το HLF project, το οποίο είναι ένα framework για permissioned distributed ledger υλοποιήσεις [87]. Το blockchain δίκτυο που δημιουργήθηκε σε παρούσα εργασία, βασίστηκε στο βασικό δίκτυο (basic network) όπως το περιγράφει το HLF στο reposiroty



Σχήμα 4.9: Διαχειριστικό περιβάλλον OFM

fabric-samples [88]. Με βάση το basic-network, έγινε προσθήκη ενός ακόμα peer host στον ίδιο οργανισμό (organization) και έτσι στο δίκτυο μας έχουμε ένα πολύ απλό p2p blockchain δίκτυο που αποτελείται από δύο διομότιμους χρήστες. Ακόμα όπως θα δούμε στην συνέχεια, προστέθηκαν οι κατάλληλες εντολές για την προσθήκη και του δεύτερου πεερ κόμβου στο κανάλι (channel) του blockchain δικτύου, με εντολές που θα δούμε στην συνέχεια. Επίσης έγιναν οι απαραίτητες αλλαγές στα Docker files για να δημιουργείται ένα cli docker contrainer, έτσι ώστε να είναι πιο έυκολη η διαχείρηση και η δημιουργία των chaincodes και της συνολικής λειτουργίας του δικτύου. Μέσα από το cli container, εκτελούνται για παράδειγμα λειτουργίες του HLF όπως το installation και το instantiation του chaincode στο ordering service και στο κανάλι επικοινωνίας του διομότιμου blockchain δικτύου.

Από εδώ και στο εξής θα γίνεται αναφορά σε υλικό που υπάρχει στο directory fabric=samples/basic-network στο repository [88], εκτός και αν αναφέρεται κάτι διαφορετικό. Όπως αναφέραμε και στο προηγούμενο κεφάλαιο, το blockchain δίκτυο μας, αποτελείται συνοπτικά από τις παρακάτω ξεχωριστές οντότητες μαζί με τα ονόματά τους, οι οποίες ορίζονται με όλες τις απαραίτητες παραμέτρους τους στο YAML αρχείο docker-compose.yml:

- Την **Αρχή Πιστοποίησης (CA)** του Fabric (ca.example.com)
- Το **Orderering Service**, το οποίο στην περίπτωσή μας αποφασίσαμε να αποτελείται μόνο από έναν κόμβο για λόγους απλότητας. (orderer.example.com)
- Τους **διομότιμους χρήστες (peers)**, οι οποίοι ανήκουν σε οργανισμούς. Στην περίπτωσή μας επιλέξαμε μόνο έναν οργανισμό (Org1) ο οποίος περιέχει δύο peers. (peer0.org1.example.com & peer1.org1.example.com)

Το παραπάνω δίκτυο είναι το ελάχιστο και πιο απλουστευμένο Blockchain δίκτυο βασισμένο στο HLF, που περιλαμβάνει και υλοποιεί όλες τις λειτουργίες ενός κανονικού blockchain

δικτύου. Συνεπώς μπορεί πολύ εύκολα να επεκταθεί και να χρησιμοποιηθεί και για μεγαλύτερες εφαρμογές με την κατάλληλη τροποποίηση των αρχείων που θα αναλύσουμε στην συνέχεια.

Σημαντικό είναι να αναφέρουμε ότι στο αρχείο docker-compose.yml ορίζονται ακόμα τρία στο σύνολο docker containers, εκτός από τα παραπάνω, τα οποία είναι τα ακόλουθα:

- Δύο **CouchDB containers**, τα οποία αντιστοιχούν κάθε ένα σε έναν peer του δικτύου. Η λειτουργία τους είναι να αποθηκεύουν τις καταστάσεις του blockchain δικτύου [89] σε μία βάση δεδομένων βασισμένη στην couchDB [90]. Σε αυτά τα containers αποθηκεύεται το world state του blockchain [71].
- Ένα **CLI (Command Line Interface)**, που Χρησιμοποιείται για την αλληλεπίδραση του chaincode με τα στοιχεία του δικτύου, και προσφέρει εύκολη αποσφαλμάτωση σε επίπεδο ανάπτυξης εφαρμογών στο HLF δίκτυο. Επί της ουσίας, δεν είναι απαραίτητο για την λειτουργία της εφαρμογής, αλλά διευκολύνει αρκετά την ανάπτυξη και αποτελεί ένα κεντρικό σημείο από το οποίο μπορούν να εκτελεστούν πολλές εντολές για την λειτουργία του δικτύου.

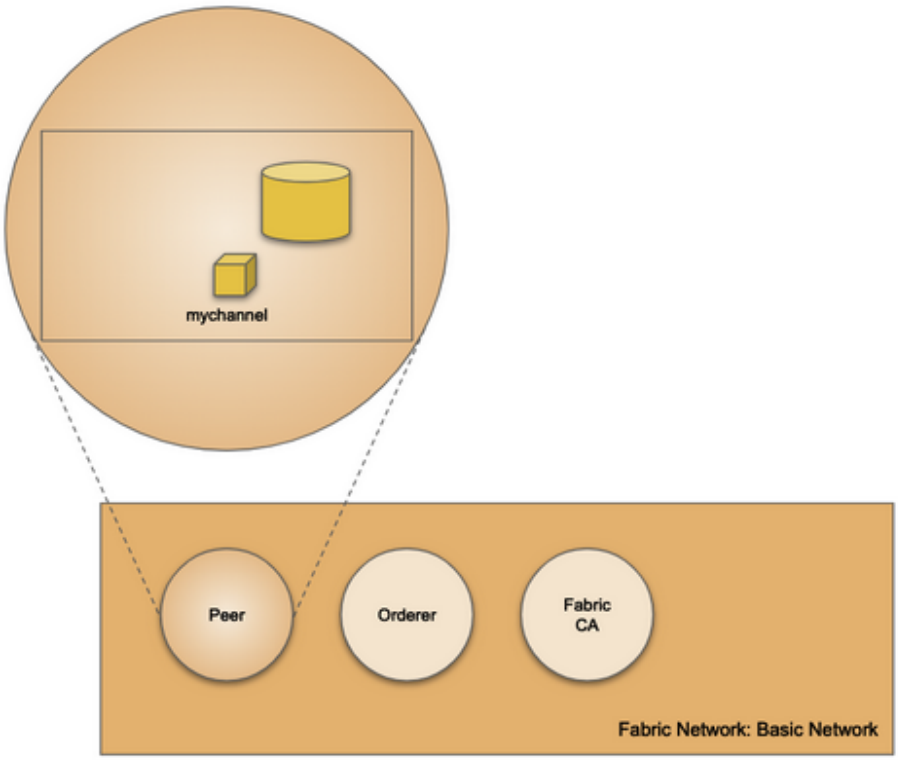
Εξετάζοντας τα περιεχόμενα του φακέλου φαβρις-σαμπλες/βασις-νετωορκ υπάρχουν δύο βασικά YAML αρχεία το crypto-config.yaml και το configtx.yaml. Τα δύο αυτά αρχεία είναι υπεύθυνα για την ρύθμιση του organization, των πιστοποιητικών της κάθε οντότητας που συμμετέχει στο δίκτυο και την διαμόρφωση των συναλλαγών στο δίκτυο του blockchain. Επίσης υπάρχουν οι φάκελοι config/ και crypto-config/, στους οποίους υπάρχει ένα σύνολο από προεγκατεστημένα και προκαθορισμένα αρχεία που εν γένει δημιουργούνται από το blockchain δίκτυο κατά την εκκίνησή και αρχικοποίηση του, την οποία θα αναλύσουμε στην συνέχεια. Στον φάκελο crypto-config υπάρχουν τα private/public keys για τους πεερς και τους orderers, καθώς επίσης και τα πιστοποιητικά τους. Ο φάκελος config/ περιέχει ύστερα από την αρχικοποίηση του blockchain δικτύου (blockchain network – bn), τρία αρχεία, τα οποία το HLF ονομάζει artifacts. Τα artifacts είναι αρχεία ρυθμίσεων για το κανάλι ή κανάλια και δημιουργούνται με το εργαλείο configtxgen [91] κατά το πρώτο στάδιο δημιουργίας και αρχικοποίησης των παραμέτρων του δικτύου. Όλες οι εντολές για την δημιουργία των artifacts βρίσκονται στο αρχείο generate.sh, το οποίο εκτελείται και ως πρώτο για την κατασκευή του bn. Στην παρούσα εργασία τα artifacts είναι τα παρακάτω:

- **Channel.tx:** Χρησιμοποιείται για την ρύθμιση συναλλαγών στο κανάλι του blockhain που επεξεργάζεται τα transactions.
- **genesis.block:** Αποτελεί το πρώτο block της αλυσίδας, το οποίο αρχικοποιεί το blockchain.
- **Org1MSPanchors.tx:** Ορίζει έναν peer ως Anchor Peer [92] στο δίκτυο, για την εφαρμογή του πρωτοκόλλου gossip.

Το βασικό δίκτυο έρχεται επίσης με κάποια ακόμα έτοιμα scripts τα οποία διευκολύνουν την δημιουργία του blockchain δικτύου (bn). Μετά από την εκτέλεση του generete.sh για την ρύθμιση των παραπάνω παραμέτρων, εκτελείται το start.sh αρχείο.

Το αρχείο αυτό δημιουργεί τα εφτά docker containers (CA, Orderer, Peer0.Org1, Peer1.Org1 και CouchDB x2, CLI), τα οποία αναφέραμε προηγουμένως και απαρτίζουν το δίκτυο. Το παραπάνω κάνει η εντολή docker-compose στο YAML αρχείο "docker-compose.yml" και γίνεται εκκίνηση των containers. Στην συνέχεια εκτελείται η εντολή docker exec -it [93] για την δημιουργία του καναλιού από το channel.tx artifact και την ένταξη των peers στο κανάλι αυτό με την HLF εντολή peer channel και την αντίστοιχη υποεντολή (create, fetch, join) [94].

Έτσι με την εγκατάσταση των βασικών στοιχείων στην υποδομή του blockchain, το bn είναι έτοιμο για οποιαδήποτε εκτέλεση και λειτουργία του chaincode που πρόκειται να τρέξει σε αυτό.



Σχήμα 4.10: Δομικά στοιχεία του bn (απεικονίζεται ο ένας από τους 2 peers) (Πηγή: [119])

Στο bn υπάρχουν επίσης δύο αρχεία τα οποία ονομάζονται connection.json και connection.yaml και παραμετροποιούν τις διευθύνσεις και τις πόρτες στις οποίες ακούν κάθε μία από τις οντότητες του bn. Ακόμα υπάρχουν δύο bash αρχεία, stop.sh και teardown.sh, τα οποία χρησιμοποιούνται, όπως υποδηλώνουν τα ονόματα για να σταματήσουν το δίκτυο και για την διαγραφή όλων των οντοτήτων και συνδέσεων που έχουν δημιουργηθεί σε αυτό.

Τέλος, χρειάζεται να αρχικοποιηθεί και να εγκατασταθεί το chaincode στο κανάλι (mychannel) που ορίστηκε κααι δημιουργήθηκε παραπάνω. Αυτό γίνεται με τις δύο ακόλουθες εντολές:

Chaincode installation & Instantiation in the channel
1) docker exec -it cli peer chaincode install -n mychainc -p github.com/ -v v0 2) docker exec -it cli peer chaincode instantiate -o orderer.example.com:7050 -C mychannel -n mychainc -v v0 -c '"Args":["init"]'‘

Οι εντολές εκτελούν το installation και instantiation του chaincode το οποίο ονομάζουμε mychainc. Αρχικά το chaincode που έχουμε υλοποιήσει για την καταγραφή των κανόνων δρομολόγησης του SDN δικτύου, και έχει υλοποιηθεί με Golang κώδικα, εγκαθίσταται στο κατάλληλο μονοπάτι που έχουμε ορίσει με την επιλογή -p, και στο οποίο βρίσκονται όλες οι κατάλληλες βιβλιοθήκες για το compile και την εκτέλεση του .go αρχείου που υλοποιεί το chaincode (cc). Το αρχείο αυτό βρίσκεται σε ειδικό docker volume [95], έτσι ώστε να είναι προσβάσιμο από τους peers και το ordering service μέσω του CLI container από το οποίο και εκτελούμε τις παραπάνω εντολές. Στην συνέχεια γίνεται το instantiation, το οποίο περιλαμβάνει την εφαρμογή του ςς στο bn δίκτυο και την αρχικοποίησή αυτού με μία Init, δηλαδή αρχική, τιμή η οποία έχει οριστεί μέσα στον πηγαίο κώδικά που αναπτύχθηκε. Πλέον το blockchain δίκτυο είναι έτοιμο να δεχθεί συναλλαγές και να ανανεώσει την κατάσταση των ledger που έχουν οι peers, μέσω του orderer.

4.4 Σύνδεση SDN και Blockchain Δικτύων

Το SDN δίκτυο συνδέεται με τη παραπάνω υλοποίηση του blockchain και το τελευταίο καταγράφει την πληροφορία από τους OpenFlow κανόνες δρομολόγησης, στον ledger του διομότιμου δικτύου. Με αυτόν τον τρόπο η βασική πληροφορία που προκύπτει από ένα κεντρικοποιημένο σύστημα, κατανέμεται σε ένα ανεξάρτητο p2p σύστημα, το οποίο προσφέρει την απαραίτητη ασφάλεια στην αλλοίωσή των δεδομένων από κακόβουλες τρίτες οντότητες.

Όπως έχουμε ήδη αναφέρει και στα προηγούμενα κεφάλαια, ο SDN controller είναι υπεύθυνος για την μεταβίβαση των κανόνων δρομολόγησης στους εικονικούς δρομολογητές, όταν σε αυτούς, είναι άγνωστη η τοπολογία του δικτύου. Για παράδειγμα όταν ένα πακέτο φτάνει στον εικονικό δρομολογητή και ο ίδιος δεν ξέρει που να το προωθήσει, τότε κατόπιν αιτήματος, ο SDN controller μαθαίνει και γνωστοποιεί την δικτυακή τοπολογία στον δρομολογητή, ο οποίος με την σειρά του προωθεί κατάλληλα το ή τα πακέτα στο ανάλογο μονοπάτι της δικτυακής τοπολογίας. Με την χρήση ενός blockchain chaincode το οποίο υλοποιήσαμε, το p2p blockchain δίκτυο το οποίο επικοινωνεί με το SDN, έχει γνώση της μέχρι τώρα δικτυακής τοπολογίας, και έιναι σε θέση να επέμβει σε μία πιθανή αποτυχία του controller έτσι ώστε να η λειτουργία των εικονικών δρομολογητών να επιρεάζεται κατά το δυνατόν ελάχιστο από την αποτυχία στο κεντρικοποιημένο σύστημα που απαρτίχεται από το SDN δίκτυο.

Το παραπάνω είναι εφικτό, λόγω των βάσεων (Open-VirtualSwitch-Database) που έχουν οι εικονικοί δρομολογητές (σε εντιστοιχία ένα προς ένα), στις οποίες αποθηκεύεται όλη η πληροφορία σχετικά με τους κανόνες δρομολόγησης.

Στην παρούσα φάση αξίζει να αναφερθεί ξανά, ότι οι κανόνες δρομολόγησης (flow rules), χωρίζονται για την περίπτωση και το σενάριο που εξετάζουμε σε δύο βασικές κατηγορίες:

- Τους Κανόνες δρομολόγησης οι οποίοι έχουν πεδίο timeout,
- Τους κανόνες δρομολόγησης οι οποίοι δεν έχουν πεδίο timeout.

Αυτό σημαίνει ότι όταν ένας κανόνας δρομολόγησης ενταχθεί στην βάση δεδομένων του αντίστοιχου εικονικού δρομολογητή, υπάρχει το αντίστοιχο πεδίο που δηλώνει πόσο χρονικό διάστημα θα παραμείνει ο κανόνας αυτός μέσα στην βάση. Με τον τρόπο αυτό, ο εκάστοτε δρομολογητής γνωρίζει που χρειάζεται να προωθήσει τα αντίστοιχα πακέτα που καταυθάνουν σε αυτόν και για πόσο χρονικό διάστημα αναμένεται να υπάρχει ο αντίστοιχος κανόνας. Όπως είναι λογικό, σε περίπτωση που ορισμένοι κανόνες έχουν λήξει και δεν βρίσκονται πλέον στην αντίστοιχη βάση δεδομένων, και ταυτόχρονα ο εικονικός δρομολογητής δεν μπορεί να επικοινωνήσει με τον SDN controller, τότε τα αντίστοιχα πακέτα γίνονται drop και δεν δρομολογείται με τον επιθυμητό τρόπο η κίνηση στο SDN δίκτυο. Σε αυτήν την περίπτωση, η συνδυαστική υλοποίηση του blockchain chaincode, έρχεται να λύσει το πρόβλημα αυτό. Τα προβλήματα τα οποία προέρχονται από τον συγκεντρωτισμό της πληροφορίας στο επιπέδο του ελέγχου ενός SDN δικτύου, όπως αυτά αναλύθηκαν και στα προηγούμενα κεφάλαια μπορούν να συνοψιστούν στις εξής τρεις εργασίες [96, 97, 98].

Όλα τα παραπάνω έχουν υλοποιηθεί με ένα bash script. Το script αυτό τρέχει, αφού έχουν εκκινήσει και τα δύο δίκτυα (blockchain δίκτυο & ενΣΔΝ δίκτυο). Μέσω του κώδικα chaincode περνάει η απαραίτητη πληροφορία των OpenFlow κανόνων στο blockchain δίκτυο και καταγράφεται στους ledgers του τελευταίου. Αυτό γίνεται με την συνάρτηση invoke η οποία υλοποιείται μέσα στον κώδικα του chaincode. Η συνάρτηση αυτή, παίρνει ως παραμέτρους τα κατάλληλα πεδία από τους αντίσοτιχους πίνακες της ενότητας 4.1.3, και επιστέφει το κατάλληλο μήνυμα αναλόγως αν ολοκληρώθηκε με επιτυχία ή όχι η καταγραφή των κανόνων δρομολόγησης στον ledger του p2p δικτύου.

Ο τρόπος με τον οποίο λειτουργεί το συγκεκριμένο script που υλοποιήσαμε είναι ο εξής. Αρχικά τα δύο δίκτυα που προαναφέραμε δημιουργούνται και το κάθε ένα είναι έτοιμο να εκτελέσει την αντίστοιχη λειτουργία του. Το SDN δίκτυο μπορεί να δρομολογήσει κανονικά δικτυακή κίνηση μεταξύ των hosts και των switches, με την βοήθεια του SDN controller, ο οποίος αναλαμβάνει την διαχείριση των κανόνων δρομολόγησης. Από την άλλη, το blockchain δίκτυο είναι σε ετοιμότητα να δεχθεί τα δεδομένα, τα οποία θα περάσουν από τους διομότιμους κόμβους του, θα επικυρωθούν και θα καταγραφούν τελικά στον ledger. Το πιο απλο σενάριο το οποίο υλοποιήσαμε είναι η χειροκίνητη εκτέλεση του συγκεκριμένου script, από την αντίστοιχη γραμμή εντολών του συστήματος μας.

Σε κάθε εκτέλεση λοιπόν του script που υλοποιήσαμε, οι κανόνες δρομολόγησης που υπάρχουν στις βάσεις των εικονικών δρομολογητών (όπως αυτές φαίνονται στα σχήματα 4.2, 4.5, 4.6), αρχικά λαμβάνονται μέσω των κατάλληλων εργαλείων που μας παρέχουν οι βάσεις των εικονικών δρομολογητών. Έπειτα αποθηκέυονται στις κατάλληλες μεταβλητές για κάθε πίνακα και για κάθε OVS βάση και με μία συγκεκριμένη σειρά από εντολές, εισέρχονται σειριακά και έπειτα καταγράφονται στο p2p δίκτυο του blockchain. Κάθε τέτοιου είδους καταγραφή

γίνετια με την εντίσοτιχη εντολή invoke του Hyperledger Fabric εργαλείου, και ο τρόπος με τον οποίο γίνεται, καθώς και τα ανάλογα μηνύματα που επιστρέφονται κατά την εκτέλεση της invoke, θα αναλυθούν περαιτέρω στην ενότητα 5.3.

Τα πλεονεκτήματα της συγκεκριμένης υβριδικής αρχιτεκτονικής αφορούν τόσο την διασφάλιση των δεδομένων του SDN δίκτυου, όσο και την ακεραιότητα και την ασφάλεια αυτών ύστερα από την καταγραφή τους στο blockchain. Αυτό συμβαίνει και ισχύει για πάνω από μία περιπτώσεις. Για παράδειγμα, όπως αναφέραμε και σε προηγούμενη ενότητα, εάν υπάρξει κάποια ανωμαλία ή βλάβη στην ομαλή λειτουργία του SDN δίκτυου και οι κανόνες δρομολόγησης αλλοιωθούν/χαθούν τότε η ανάκτηση τους από μία τελευταία κατάσταση τους είναι δυνατή λόγω του ledger στο p2p δίκτυο. Ένα άλλο παράδειγμα είναι η κακόβουλη εγγραφή κανόνων δρομολόγησης με σκοπό την παραβίαση και την αθέμιτη χρήση του εύρους ζώνης στο SDN δίκτυο. Και σε αυτήν την περίπτωση η καταγραφή των Openflow rules στο blockchain δίκτυο διασφαλίζει μία αλληλουχία από εγγραφές οι οποίες δεν μπορούν να αλλοιωθούν και παρέχουν ένα ιστορικό από καταγεγραμμένους κανόνες δρομολόγησης. Προκύπτει επομένως ως ένα αρχικό συμπέρασμα ότι η μελέτη μίας υβριδικής, ως προς την διαχείριση της πληροφορίας (συγκεκριμένα τους SDN κανόνες δρομολόγησης) αρχιτεκτονικής έχιε πολλά να προσφέρει.

Κεφάλαιο 5

Σενάριο, Πειράματα & Αξιολόγηση

5.1 Εισαγωγή

Για να αντιληφθούμε την σύνδεση των SDN δικτύων και του blockchain, θα υλοποιήσουμε και θα αναπαραστήσουμε ένα σενάριο. Στο σενάριο αυτό, με συγκεκριμένη δικτυακή τοπολογία η οποία θα υλοποιηθεί στο mininet, θα αναπαράγουμε ένα πείραμα. Το πείραμα αυτό, θα δείχνει την λειτουργία του blockchain και τον τρόπο που το SDN δίκτυό μας, αλλά και κατέπέκταση κάθε SDN δίκτυο, χειρίζεται τους κανόνες δρομολόγησης από τις βάσεις δεδομένων των δρομολογητών.

Το πείραμα-σενάριο που θα δείξουμε, θα γίνει σε μία τοπολογία σαν αυτή που είδαμε στην παράγραφο 4.2.1 και συγκεκριμένα στο σχήμα 4.2.

5.2 Επικοινωνία, Προσθήκη & Αφαίρεση των Κανόνων Δρομολόγησης

Για να υπάρξει καλύτερη κατανόηση σχετικά με τον τρόπο που λειτουργεί το σενάριό μας και κατά επέκταση η υλοποίησή, θα περιγράψουμε σύντομα και θα συνοψίσουμε τρία βασίκα πράγματα που αφορούν την λειτουργία ενός SDN δικτύου. Αυτά σχετίζονται με την προκαθορισμένη λειτουργία ενός SDN δικτύου και αφορούν τα εξής παρακάτω:

- Την επικοινωνία των v-switches με τον SDN controller.
- Την συμπλήρωση του πίνακα δρομολόγησης στις βάσεις των v-switches Flow table population.
- Την αφαίρεση των κανόνων δρομολόγησης από τις βάσεις των εικονικών δρομολογητών.

5.2.1 Επικοινωνία Openflow V-Switch & SDN Controller

Τα βήματα τα οποία ακολουθούνται για την επικοινωνία ενός Openflow Switch με τον SDN Controller περιγράφονται παρακάτω στον αλγόριθμο 1 [99].

Algorithm 1 OpenFlow Switch and SDN Controller Default Communication

1: **Procedure** *OpenFlow v-switch connects to SDN Network with some initial settings (i.e. IP address, Default Gateway, Openflow Controller*
2: *Controller places proactively openflow rules in the v-switch by creating LLDP packets which are sent to the v-switch, which is instructed to sent LLDP packet out of all ports and repeat for all ports (**if ether_type=LLDP, actions=output.controller**)*
3: *LLDP packets are then sent from all ports, to the controller, which determines the position of v-switch in the SDN network topology*
4: **End Procedure**

5.2.2 Προσθήκη Κανόνων Δρομολόγησης

Οι κανόνες δρομολόγησης προστίθενται στις βάσεις των V-Switches, με δύο τρόπους.

1. **Προληπτική Προσθήκη (Proactive).**
2. **Δυναμική Προσθήκη (Reactive).**

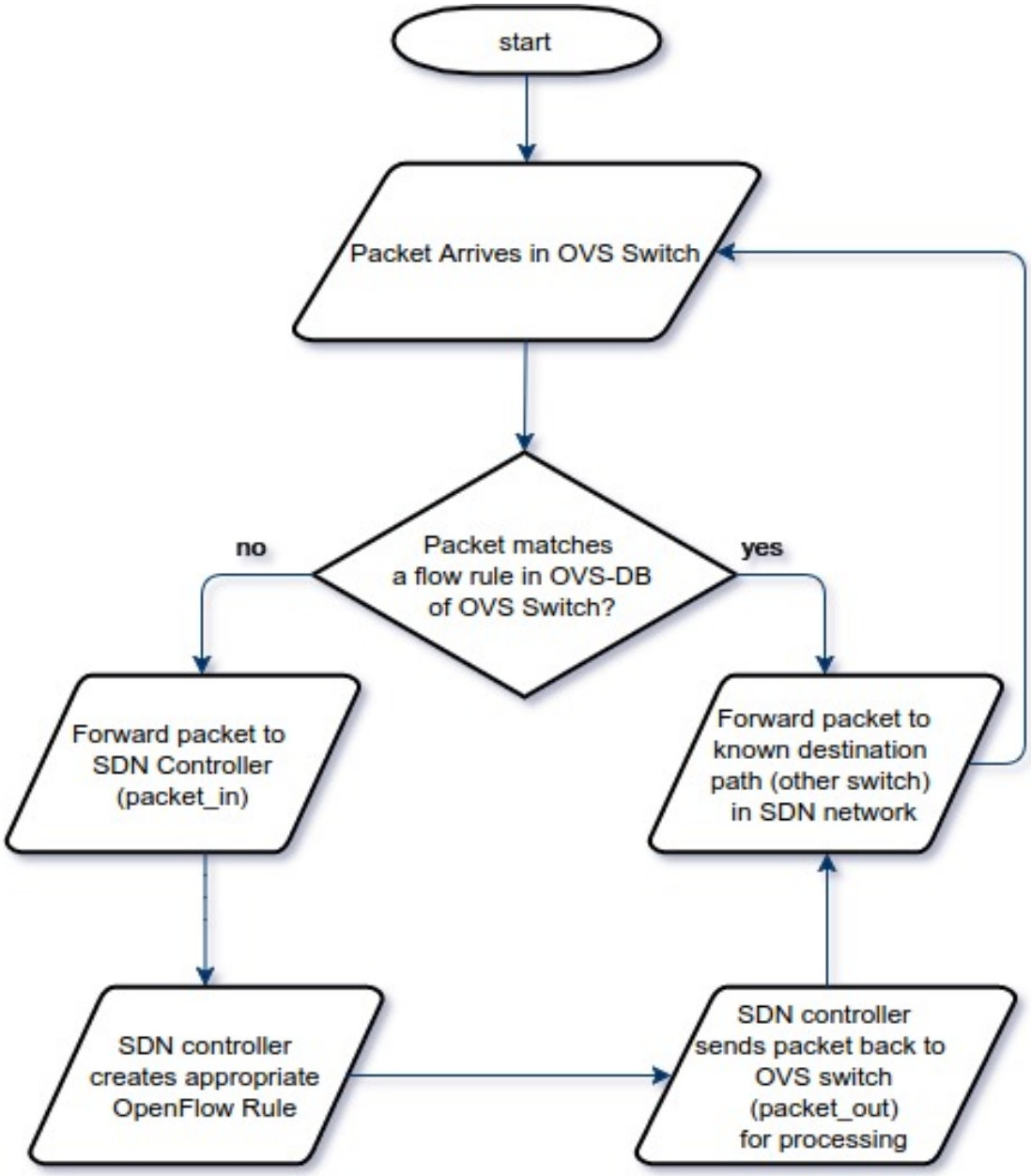
Προληπτική προσθήκη των κανόνων δρομολόγησης σημαίνει ότι αυτοί προστίθενται στην βάση του εικονικού δρομολογητή, πριν αυτοί χρειαστούν για οποιαδήποτε δικτυακή κίνηση στο SDN δίκτυο. Αντιθέτως η δυναμική προσθήκη σημαίνει ότι οι κανόνες προστίθενται δυναμικά όποτε και εφόσον χρειαστούν για την δικτυακή κίνηση. Τα πακέτα τα οποία καταφθάνουν σε έναν εικονικό δρομολογητή και δεν αντιστοιχούν σε κάποιο ήδη υπάρχων κανόνα, στέλνονται στον SDN controller. Ο τελευταίος με την σειρά του, δημιουργεί τον απαραίτητο κανόνα και τον στέλνει μαζί με τα αντίστοιχα πακέτα πίσω στον εικονικό δρομολογητή για να πραγματοποιηθεί η απαραίτητη δικτυακή κίνηση.

Η προκαθορισμένη διαδικασία για ένα SDN δίκτυο, με την οποία συμπληρώνεται ο πίνακας δρομολογήσεων σε κάθε μία από τις βάσεις δεδομένων των εικονικών δρομολογητών, περιγράφεται όπως φαίνεται παρακάτω στον αλγόριθμο 2 [99].

Algorithm 2 Flow Table Population of an OpenVirtual Switch Database

1: **Procedure** *Openflow Packet arrives in v-switch*
2: **if** Packet matches a flow rule in OVS-Database **then**
3: *Forward packet to known destination in the network topology*
4: **else if** Packet does not match a flow rule in the OVS-Database **then**
5: *v-Switch forwards packet to SDN Controller (packet_in)*
6: *Controller creates appropriate Openflow rule, which populates the OVS-Database*
7: *Controller sends packet back (packet_out) for processing by the v-switch*
8: **end if**
9: **End Procedure**

Η παραπάνω διαδικασία φαίνεται και διαγραμματικά στο διαγράμμα ροής που ακολουθεί:



Σχήμα 5.1: Προσθήκη OpenFlow Κανόνων Δρομολόγησης (Πηγή: [99])

5.2.3 Αφαίρεση Καταχωρήσεων από τους Πίνακες Ροής

Οι καταχωρίσεις ροής (flow rules) αφαιρούνται από τους πίνακες ροής (flow tables) με δύο τρόπους. Είτε κατόπιν αιτήματος του ελεγκτή, είτε μέσω του μηχανισμού που ελέχγει αν η καταχωρημένη ροή έχει λήξει (hard/idle timeout). Ο μηχανισμός για την λήξη της ροής, λειτουργεί στον openflow v-switch, ανεξάρτητα από τον ελεγκτή και βασίζεται στην κατάσταση και τη διαμόρφωση των καταχωρημένων κανόνων στους πίνακες ροής. Κάθε είσοδος ροής έχει ένα idle_timeout και ένα hard_timeout. Εάν οποιαδήποτε τιμή από αυτές είναι μηδενική, ο v-switch πρέπει να σημειώσει την ώρα άφιξης της συγκεκριμένης εγγραφής κανόνα,

καθώς ενδέχεται να χρειαστεί να αποβάλει την εγγραφή αργότερα. Ένα μη μηδενικό πεδίο hard_timeout προκαλεί την κατάργηση της καταχώρησης ροής μετά τον δεδομένο αριθμό δευτερολέπτων, ανεξάρτητα από τον αριθμό των πακέτων που έχει αντιστοιχιστεί στον κανόνα αυτό. Ένα μη μηδενικό πεδίο idle_timeout προκαλεί την κατάργηση του συγκεκριμένου κανόνα, όταν δεν έχει αντιστοιχιστεί κανένα πακέτο στον δεδομένο αριθμό δευτερολέπτων. Ο v-switch πρέπει να εφαρμόζει τη λήξη του κανόνα ροής και να τον καταργεί από τον πίνακα της αντίστοιχης βάσης όταν ξεπεραστεί ένα από τα χρονικά περιθώρια.

Στην περίπτωση που το idle_timeout είναι ρυθμισμένο και το hard_timeout είναι μηδέν, η καταχώρηση πρέπει να λήξει μετά από idle_timeout δευτερόλεπτα χωρίς να έχει ληφθεί διαδρομή. Εάν το idle_timeout είναι μηδέν και έχει οριστεί το hard_timeout πεδίο, η καταχώρηση πρέπει να λήγει σε hard_timeout δευτερόλεπτα, ανεξάρτητα από το εάν τα πακέτα χτυπούν την καταχώρηση. Εάν έχουν οριστεί και τα δύο πεδία, η καταχώριση ροής θα λήξει μετά από idle_timeout δευτερόλεπτα χωρίς κίνηση ή μετά από hard_timeout δευτερόλεπτα, όποιο από τα δύο προηγείται. Εάν και τα δύο αυτά πεδία είναι μηδέν, η καταχώρηση θεωρείται μόνιμη και δεν θα λήξει ποτέ. Μπορεί όμως να καταργηθεί με ένα μήνυμα flow_mod το οποίο αλλάζει η καταγρεί τον κανόνα της συγκεκριμένης ροής

5.3 Απώλεια Συνδέσμου μεταξύ Switch-Controller

Στο σενάριο, το οποίο θα υλοποιήσουμε και θα παρουσιάσουμε, το blockchain network (bn), δηλαδή το SDN δίκτυο το οποίο λειτουργεί σε συνδυασμό με το blockchain. Το δίκτυο αυτό αποτελείται από μία τοπολογία παρόμοια με αυτήν του κεφαλαίου 3.1.6, όπως αυτή φαίνεται στο σχήμα 3.5.

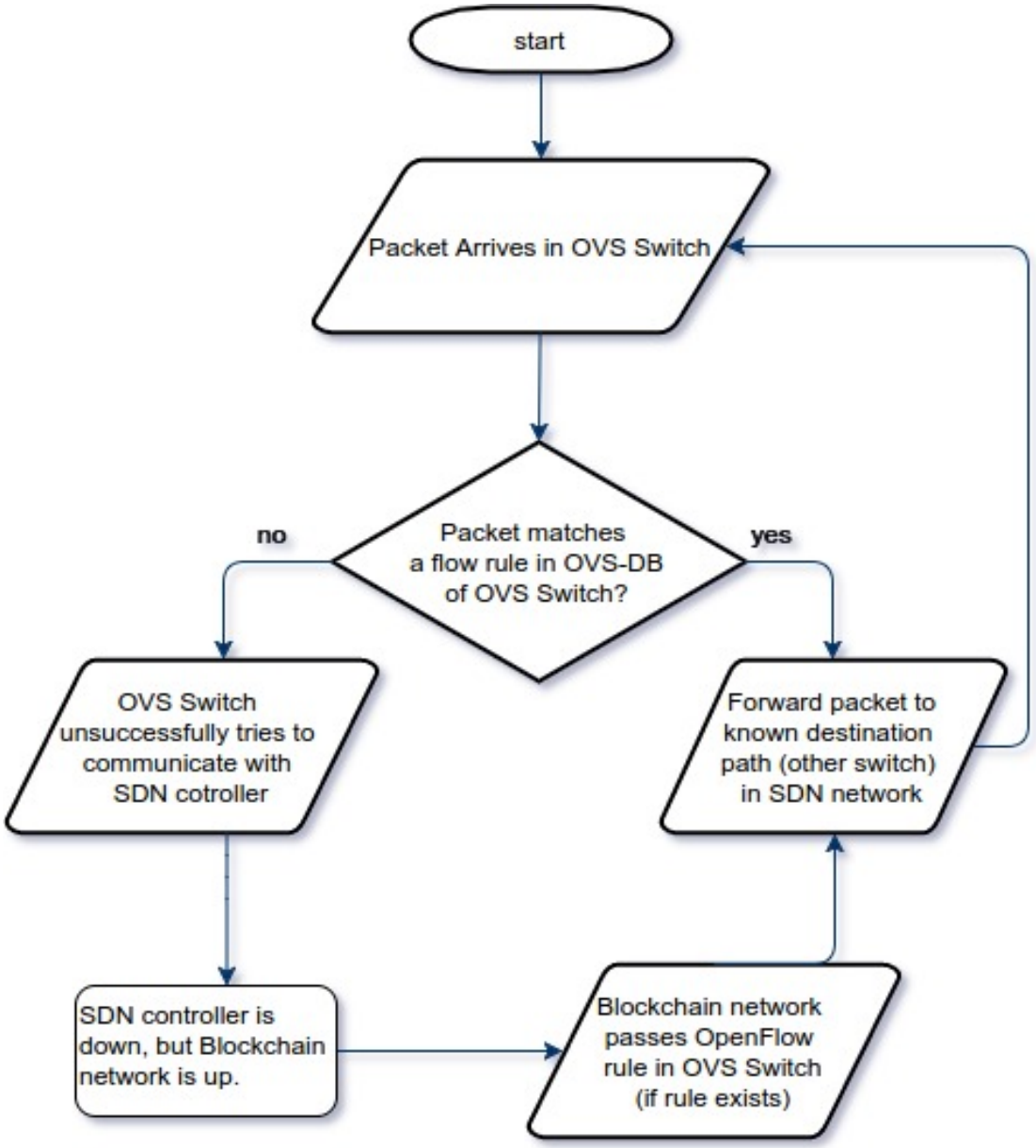
Για λόγους απλούστευσης και προκειμένου να απλοποιηθεί η διαμόρφωση του δικτύου το οποίο θα χρησιμοποιήσουμε για το πείραμά μας, η τοπολογία για το κομμάτι του SDN δικτύου θα διαφέρει ως προς αυτήν που φαίνεται στο σχήμα 4.2 του κεφαλαίου 4.2.1. Επομένως, αντί για τρεις εικονικούς μεταγωγείς εκ των οποίων οι δύο πρώτοι συνδέονται με δύο εικονικούς hosts και ο τρίτος με έναν ,θα αφαιρέσουμε τον τρίτο εικονικό δρομολογητή μαζί με τον host που είναι συνδεδεμένος με αυτόν. Έτσι θα έχουμε μία τοπολογία η οποία είναι πιο συμμετρική και θα είναι ίδια με την παρακάτω τοπολογία (όπως αυτή φαίνεται μέσα από την γραμμή εντολών του mininet, με τα αντίστοιχα στοιχεία:

Mininet	
Dump	***Links***
Host h1: h1-eth0:10.0.0.1 pid=2431	h1-eth0 - s1-eth1 (OK OK)
Host h2: h2-eth0:10.0.0.2 pid=2434	h2-eth0 - s1-eth2 (OK OK)
Host h3: h3-eth0:10.0.0.3 pid=2437	h3-eth0 - s2-eth1 (OK OK)
Host h4: h4-eth0:10.0.0.4 pid=2440	h4-eth0 - s2-eth2 (OK OK)
OVSSwitch s1: s1-eth1:None,s1-eth2:None,s1-eth3:None pid=2446	s1-eth3 - s2-eth3 (OK OK)
OVSSwitch s2: ,s2-eth1:None,s2-eth2:None,s2-eth3:None pid=2449	s2-eth3 - s1-eth3 (OK OK)
RemoteController'ip': '83.212.174.56', 'port': 6633 c0: 83.212.174.56:6633 pid=2425	

Η αποτυχία και αποκατάσταση (failover) σε ένα δίκτυο επικοινωνιών είναι η διαδικασία της άμεσης μεταφοράς εργασιών από ένα αποτυχημένο στοιχείο σε ένα παρόμοιο στοιχείο το οποίο εκ πρώτης όψεως είναι περιττό (στην ομαλή λειτουργία του συστήματος), για να αποφευχθεί η διαταραχή και η διατήρηση των λειτουργιών του δικτύου [100]. Η αυτοματοποίηση αυτής της ¨αποτυχίας & αποκατάστασης¨, είναι η δυνατότητα αυτόματης αναδρομολόγησης των δεδομένων από ένα αποτυχημένο στοιχείο, σε ένα δεύτερο λειτουργικό στοιχείο, με σκοπό την συνέχεια της ομαλής λειτουργίας του συστήματος.

Όπως είναι λογικό μία τέτοια διαδικασία είναι σημαντική σε κρίσιμης σημασίας υποδομές και συστήματα. Σε μία τέτοιου είδους κατάσταση, ανακατεύθυνσης της πληροφορίας όπως μπορεί αλλιώς να χαρακτηριστεί, οι συσκευές οι οποίες μπορούν να υποστούν βλάβη δεδομένων μπορεί να ποικίλουν, και να είναι από δικτυακές συσκευές, μέχρι δίσκοι, βάσεις δεδομένων και τείχοι προστασίας. Η λογική όμως για κάθε τύπο και είδος συσκευής είναι η ίδια. Δεδομένα μεταφέρονται από την προβληματική συσκευή ή σύστημα, στον ίδιο τύπο πλεονάζοντος εξαρτήματος, για να διασφαλιστεί ότι υπάρχει όσο το δυνατόν περιορισμένη διακοπή στη ροή δεδομένων και στη συνολική λειτουργία αρχικά της συσκευής και συνολικά του συστήματος, στο οποίο αυτή ανήκει. Συνεπώς αν ένα πρωτεύον εξάρτημα δεν είναι διαθέσιμο λόγω είτε αποτυχίας είτε προγραμματισμένου χρόνου διακοπής, το δευτερεύον στοιχείο χρησιμεύει ως αντίγραφο ασφαλείας και αναλαμβάνει εκ μέρους του άλλου. Παραδείγματος χάριν, το σχέδιο αναδιάρθρωσης (failover) για μία οποιαδήποτε βάση δεδομένων είναι η πλήρης αποκατάσταση ή ανάκτηση δεδομένων από μία αποτυχία στην βάση αυτή, με τρόπο που να είναι ανεκτά λειτουργικός και πάλι για το σύστημα που δέχθηκε την αποτυχία αυτή [101].

Προκύπτει λοιπόν ως άμεσο συμπέρασμα ότι κρίσιμες υποδομές, χρειάζονται ένα αξιόπιστο και έμπιστο σύστημα ανακατεύθηνσης της πληροφορίας το οποίο παρέχει προστασία των δεδομένων και της λειτουργίας ενός συστήματος, είτε αυτό είναι μία βάση δεδομένων, είτε μία υπηρεσία, είτε ένα δίκτυο που επιτελεί έναν συγκεκριμένο στόχο. Έτσι επιτυγχάνεται μία σημαντική ανοχή σφάλματος (fault tolerance) και υψηλή διαθεσιμότητα (high availability), εφόσον έχουν γίνει ξεκάθαρα και έχουν υλοποιηθεί τα συστατικά μέρη που συμμετέχουν σε ένα σετ αποτυχίας (Failover Set). Τα συστατικά αυτά μέρη, μπορεί να περιλαμβάνουν τον τρόπο με τον οποίο ορίζεται η αποτυχία για ένα σύστημα, ποια είναι η πολιτική που ακολουθείται, καθώς και ποια είναι τα βήματα που εκτελεί ένα σύστημα όταν αποτυγχάνει ένα στοιχείο του (failing-over an element) [102].



Σχήμα 5.2: Λειτουργία OpenFlow με το Blockchain δίκτυο

Η διαδικασία με την οποία λειτουργεί το παραπάνω σενάριο στην περίπτωσή μας φαίνεται αναλυτικά στο παραπάνω διάγραμμα ροής. Εφόσον το blockchain δίκτυο έχει αποθηκεύσει τον κατάλληλο κανόνα στον ledger του, τότε μπορεί να λειτουργήσει ως συνδετικός κρίκος για την ομαλότερη λειτουργία του SDN δικτύου. Έτσι κοινοποιείται ο κατάλληλος κανόνας δρομολόγησης στον κατάλληλο εικονικό δρομολογητή για την ομαλή αναπόκριση του τελευταίου στην δικτυακή κίνηση του τελευταίου στις απαιτήσεις του SDN δικτύου.

Στο συγκεκριμένο σενάριο λοιπόν, της παραπάνω δικτυακής τοπολογίας, θα δείξουμε τι γίνεται στο σύστημά μας όταν χάνεται η σύνδεση μεταξύ ενός εικονικού δρομολογητή (virtual switch) και του αντίστοιχου controller του SDN δικτύου.

Αρχικά, εκκινούμε όλα τα επιμέρους στοιχεία του Blockchain δικτύου. Τα στοιχεία αυ-

τά είναι όπως έχουμε προαναφέρει είναι εικονικές μηχανές οι οποίες υλοποιούνται σε docker containers. Στον παρακάτω πίνακα βλέπουμε το όνομα κάθε container. Επίσης στον πίνακα φαίνεται η εντολή με την οποία έχει ξεκινήσει κάθε μηχάνημα που απαρτίζει το blockchain δίκτυο. Όπως φαίνονται, οι εντολές chaincode, peer & orderer αποτελούν κομμάτι του Hyperledger Fabric και αναφέρονται στα αντίστοιχα επιμέρους στοιχεία του. Επίσης το cli container είναι για την αλληλεπίδραση με τα υπόλοιπα δοχεία. Ακόμα τα δοχεία με όνομα couchdb1, couchdb αναφέρονται στις βάσεις δεδομένων των αντίστοιχων κόμβων (peer0, peer1) του p2p blockchain δικτύου και υπάρχουν για την αποθήκευση των δεδομένων του decentralized ledger, για το λόγο αυτό έχουν και ένα-προς-ένα αντιστοίχιση με τους peers. Αυτά χρησιμοποιούνται με την βοήθεια του εργαλείου tini, το οποίο είναι μία διαδικασία δαίμονα (όπως η init), αλλά για δοχεία. Τέλος υπάρχει και ο ca-server (Certificate Authority Server), ο οποίος επίσης είναι στοιχεία απαραίτητο για την λειτουργία του Hyperledger Blockchain δικτύου.

BlockChain Network Components	
Name	*Command*
dev-peer0.org1.example.com-mychainc-v0-[Unique hash] (dev-peer0.org1.example.com-mychainc-v0)	chaincode -peer.address= peer0.org1.example.com:7052
cli (hyperledger/fabric-tools)	/bin/bash
peer0.org1.example.com (hyperledger/fabric-peer)	peer node start
peer1.org1.example.com (hyperledger/fabric-peer)	peer node start
orderer.example.com (hyperledger/fabric-orderer)	orderer
ca.example.com (hyperledger/fabric-ca)	sh -c 'fabric-ca-server start -b admin:adminpw -d'
couchdb1 (hyperledger/fabric-couchdb)	tini – /docker-entrypoint.sh /opt/couchdb/bin/couchdb
couchdb (hyperledger/fabric-couchdb)	tini – /docker-entrypoint.sh /opt/couchdb/bin/couchdb

Επόμενο βήμα, είναι η εκκίνηση του SDN δικτύου. Αυτό γίνεται με την χρήση του προσομοιωτή mininet. Το SDN δίκτυο, δηλαδή τα Openflow v-Switches συνδέονται με τον εξωτερικό OpenDayLight Controller.

Αφού εκκινήσουμε το παραπάνω δίκτυό, με την κατάλληλη δικτυακή τοπολογία, σιγουρευόμαστε ότι υπάρχει συνδεσιμότητα μεταξύ των εικονικών κόμβων, δηλαδή των virtual hosts του mininet. Αυτό γίνεται με την βοήθεια της ενσωματωμένης εντολής pingall, η οποία όπως και η γνωστή ping, ελέγχει την συνδεσιμότητα μεταξύ όλως των κόμβων της τοπολογίας, και μας επιστρέφει, αναλόγως αν αυτή υπάρχει ή όχι, το κατάλληλο μήνυμα.

Με την χρήση της εντολής αυτής, εύκολα βλέπουμε ότι στο αρχικό SDN δίκτυο, υπάρχει συνδεσιμότητα μεταξύ όλων των εικονικών κόμβων και η λειτουργία του SDN δικτύου είναι η αναμενόμενη. Αυτό σημαίνει ότι όσα πακέτα φτάνουν από κάθε έναν virtual host (h0,h1,h2,h3) σε ένα από τους εικονικούς δρομολογητές (OVSs1,s2) του δικτύου, δρομολογούνται κατάλληλα, αναλόγως με τον προορισμό τους σε έναν από τους τέσσερεις hosts.

Το παραπάνω επιτυγχάνεται και με την βοήθεια του SDN OpenDayLight Controller. Αυτό γίνεται κατά τον τρόπο λειτουργίας που ορίζουν οι SDN τεχνολογίες. Για κάθε πακέτο που καταφθάνει σε έναν v-switch, ο ίδιος για να προωθήσει το πακέτο αυτό στον κατάλληλο προορισμό, ανατρέχει στην βάση στην οποία είναι αποθηκευμένοι οι κανόνες δρομολόγησής του. Επομένως, για όσα πακέτα καταφθάνουν και ο ίδιος δεν γνωρίζει που να τα προωθήσει, ανατρέχει στην βάση που είναι αποθηκευμένοι οι κανόνες δρομολόγησης (OVS-DB). Στην OVS-Database, κατά την σύνδεση του SDN δικτύου με τον εξωτερικό controller, αρχικοποιούνται και ορισμένοι κανόνες δρομολόγησης. Ένας από αυτούς, υποδεικνύει στον εκάστοτε virtual switch, την προώθηση των πακέτων στον controller, για κάθε πακέτο, για το οποίο δεν γνωρίζει τον κατάλληλο προορισμό του. Αυτό καθορίζεται από το πεδίο actions όταν αυτό παίρνει την τιμή actions=CONTROLLER:65535. Παρακάτω βλέπουμε τους κανόνες δρομολόγησης στην βάση του εικονικού δρομολογητή s1 πριν την εκτέλεση της εντολής pingall. Όπως είναι αναμενόμενο, υπάρχουν ήδη από την σύνδεση του SDN δικτύου με τον remote controller, αρχικοποιημένοι, ορισμένοι κανόνες στην βάση του εικονικού δρομολογητή. Όπως φαίνεται από τους παρακάτω κανόνες, ο s1 δεν έχει γνώση ούτε για την τοπολογία του δικτύου αλλά ούτε και για τις MAC διευθύνσεις των virtual hosts στους οποίους είναι συνδεδεμένος. Η γνώση που υπάρχει στην βάση του s1 αφορά μόνο τα in & out ports, καθώς και την προώθηση της δικτυακής κίνησης στον ODL controller.

- cookie=0x2b0000000000000e, duration=17.136s, table=0, n_packets=0, n_bytes=0, idle_age=17, priority=100, dl_type=0x88cc, actions=CONTROLLER:65535
- cookie=0x2b00000000000022, duration=15.135s, table=0, n_packets=5, n_bytes=378, idle_age=3, priority=2, **in_port=1 actions=output:2, CONTROLLER:65535**
- cookie=0x2b00000000000023, duration=15.135s, table=0, n_packets=8, n_bytes=628, idle_age=3, priority=2, **in_port=2 actions=output:1, CONTROLLER:65535**
- cookie=0x0, duration=17.146s, table=0, n_packets=0, n_bytes=0, idle_age=17, priority=1, tcp, tp_src=1 actions=drop
- cookie=0x2b0000000000000e, duration=17.137s, table=0, n_packets=7, n_bytes=586, idle_age=15, priority=0, actions=drop

Αφού εφαρμόζουμε δικτυακή κίνηση πάνω στο SDN δίκτυό, βλέπουμε ότι ο πίνακας με τους κανόνες ροής έχει αλλάξει για την βάση του s1, καθώς έχουν προστεθεί νέοι κανόνες δρομολόγησης. Αυτή η αλλαγή γίνεται για κάθε βάση εικονικού δρομολογητή που έχει λάβει μέρος στην δικτυακή αυτή κίνηση. Για τους σκοπούς της παρούσας εργασίας δεν έχει καμία σημασία τι είδους δικτυακή κίνηση εφαρμόζουμε στο δίκτυό μας. Αυτό συμβαίνει για τον

απλό λόγο ότι οποιαδήποτε είδους κίνηση μεταξύ των δρομολογητών, προωθείται με βάση όσα είπαμε, στον controller. Επομένως για τον λόγο αυτό, και για λόγους απλούστευσης, η δικτυακή κίνηση που εφαρμόζουμε είναι η πιο βασική και δεν είναι άλλη από την εντολή pingall. Η ανανεωμένη βάση με τους νέους κανόνες ροής φαίνεται παρακάτω. Η βάση αυτή αφορά επίσης τον εικονικό δρομολογητή s1, καθώς όλες οι αλλαγές που εξετάζουμε φαίνονται ξεκάθαρα σε αυτήν την περίπτωση. Έτσι δεν χρειάζεται αναφορά και στις υπόλοιπες βάσεις κανόνων ροής.

- cookie=0x2b0000000000000e, duration=26.958s, table=0, n_packets=0, n_bytes=0, idle_age=26, priority=100, dl_type=0x88cc, actions=CONTROLLER:65535
- cookie=0x2a00000000000016, duration=8.734s, table=0, n_packets=1, n_bytes=42, **idle_timeout=600, hard_timeout=300**, idle_age=8, **priority=10, dl_src=de:98:24:b8:2c:0b, dl_dst=a2:b4:62:12:0b:5c**, actions=output:1
- cookie=0x2a00000000000017, duration=8.734s, table=0, n_packets=0, n_bytes=0, **idle_timeout=600, hard_timeout=300**, idle_age=8, priority=10, **dl_src=a2:b4:62:12:0b:5c, dl_dst=de:98:24:b8:2c:0b**, actions=output:2
- cookie=0x2b00000000000022, duration=24.957s, table=0, n_packets=6, n_bytes=420, idle_age=8, **priority=2**, in_port=1, actions=output:2, CONTROLLER:65535
- cookie=0x2b00000000000023, duration=24.957s, table=0, n_packets=8, n_bytes=628, idle_age=13, **priority=2**, in_port=2, actions=output:1, CONTROLLER:65535
- cookie=0x0, duration=26.968s, table=0, n_packets=0, n_bytes=0, idle_age=26, priority=1, tcp,tp_src=1, actions=drop
- cookie=0x2b0000000000000e, duration=26.959s, table=0, n_packets=7, n_bytes=586, idle_age=25, priority=0, actions=drop

Είναι εμφανές από την παραπάνω λίστα κανόνων δρομολόγησης η προσθήκη δύο επιπλέον κανόνων ροής στους οποίους βλέπουμε δύο νέα πεδία. Αυτά τα πεδία είναι τα dl_src & dl_dst και αφορούν τις MAC διευθύνσεις των εικονικών host οι οποίοι είναι συνδεδεμένοι σε κάθε εικονικό δρομολογητή. Επίσης φαίνεται ότι οι συγκεκριμένοι κανόνες έχουν μεγαλύτερη προτεραιότητα, έναντι εκείνων που υποδεικνύουν την προώθηση των πακέτων στον controller. Ένα ακόμα σημαντικό πεδίο είναι το idle_timeout & hard_timeout.

5.3.1 Το Σενάριο - Πείραμα

Για τα πλαίσια της παρούσας εργασίας, μπορούμε να υποθέσουμε το εξής ακόλουθο σενάριο. Έστω ότι το SDN δίκτυο, λειτουργεί με τον τρόπο που είδαμε πιο πάνω. Ο controller συνδέεται κανονικά στο δίκτυο επικοινωνεί με τους εικονικούς δρομολογητές για την ανάθεση κανόνων ροής κατόπιν αιτήματος στις βάσεις των κανόνων δρομολόγησης. Η λειτουργία του δικτύου είναι ομαλή, μέχρι ο ελεγκτής να χάσει την συνδεσιμότητά του με το SDN δίκτυο.

Έστω ότι η σύνδεση λοιπόν έχει χαθεί, ή ο controller είναι εκτός λειτουργίας για οποιονδήποτε άλλο λόγο, επειδή έχει παρουσιάσει κάποιο πρόβλημα. Αυτό όπως είναι λογικό αποτελεί εμπόδιο στην σωστή λειτουργία ενός SDN δικτύου. Η σύνδεση για παράδειγμα μπορεί να παραμείνει χαμένη για παραπάνω χρονικό διάστημα από όσο θα μπορούσε το δικτυο να ανταπεξέλθει. Κάτι τέτοιο για παράδειγμα θα μπορούσε να σημαίνει ότι οι κανόνες δρομολόγησης που είδαμε παραπάνω, έχουν λήξει και έχουν διαγραφεί από τον πίνακα δρομολόγησης.

Με την βοήθεια του blockchain δικτύου που υλοποιήσαμε, μπορεί να αποφευχθεί ένα μεγάλο ποσοστό της αποτυχίας της λειτουργίας του SDN δικτύου. Έχοντας λοιπόν υλοποιήσει όλα τα παραπάνω, θα δούμε στην συνέχεια πώς οι κανόνες δρομολόγησης εισέρχονται μέσα στο p2p δίκτυο. Έτσι, ως κατανεμημένες εγγραφές σε ένα αποκεντρωμένο p2p σύστημα, υπάρχουν μέσα στο ledger κάθε διομότιμου χρήστη του δικτύου αυτού. Όσο μεγαλύτερη επομένως γίνεται η κλίμακα των δικτύων αυτών, και όσο αυξάνει ο αριθμός των διομότιμων χρηστών, αυξάνεται και πιθανότητα ομαλής λειτουργίας, καθώς το δίκτυο γίνεται πιο ανθεκτικό στις αποτυχίες των κόμβων [103], όπως αυτές προκύπτουν από το πρόβλημα των Βυζαντινών Στρατηγών [104].

Παρακάτω φαίνονται δύο παραδείγματα κανόνων δρομολόγησης, όπως αυτά εισέρχονται στο blockchain. Όπως βλέπουμε από το πρώτο αποτέλεσμα, εφόσον και άν η συγκεκριμένη ροή, έχει ξαναεισέλθει στο ledger του p2p δικτύου, τότε εμφανίζεται ένα μήνυμα με status code 500, όπως αυτό φαίνεται με έντονη γραμματοσειρά παρακάτω. Το μήνυμα αυτό προειδοποιεί ότι η συγκεκριμένη ροή υπάρχει ήδη στο ledger, και ότι η επικύρωση της συνναλαγής στο blockchain απέτυχε.

This is now the flow: 1 in bridge: s1

2019-12-12 13:11:58.284738: flow: flow1 |cookie: 0x2b0000000000001c |duration: 884.846 |table: 0 |n_packets: 178 |n_bytes: 15130 |idle_age: 4 |actions: CONTROLLER:65535
The chaincode is going to run now to add flow1

2019-12-12 13:11:58.975 UTC [chaincodeCmd] InitCmdFactory ->INFO 001 Retrieved channel (mychannel) orderer endpoint: orderer.example.com:7050

Error: endorsement failure during invoke. response: status:500 message:"This flow already exists: flow1"

Αντιθέτως, στην κανονική λειτουργία μίας επιτυχημένης συνναλαγής, τα πράγματα είναι λίγο διαφορετικά. Σε αυτήν την περίπτωση, κατά την κλήση της συνάρτησης initFlow για μία ροή δρομολόγησης που δεν έχει ακόμα εισέλθει στο blockchain δίκτυο, λαμβάνουμε μήνυμα με status code 200. Το μήνυμα αυτό, δηλώνει ότι η κλήση του chaincode αλγορίθμου ήταν

επιτυχής και η συγκεκριμένη ροή εισήλθε στο κατανεμημένο δίκτυο. Εκεί πλέον η συγκεκριμένη εγγραφή, είναι ασφαλής από αποτυχίες ενός σημείου, οι οποίες όπως προαναφέραμε μπορούν να συμβούν εύκολα, σε κεντρικοποιημένα συστήματα όπως τα SDN δίκτυα, καθώς τα τελευταία βασίζουν την λειτουργία τους σε κεντρικές οντότητες που αποφασίζουν για την δρομολόγηση των πακέτων στο δίκτυο. Παρακάτω φαίνεται η επιτυχής εισαγωγή ενός κανόνα δρομολόγησης στο p2p δίκτυο, και με έντονη γραμματοσειρά φαίνεται το αντίστοιχο μήνυμα επιτυχίας της συναλλαγής.

This is now the flow: 13 in bridge: s2

2019-12-12 13:11:58.284738: flow: flow13 |cookie: 0x2b00000000000044
|duration: 884.751 |table: 0 |n_packets: 12 |n_bytes: 868 |
idle_age: 31 |actions: output:1,output:3,CONTROLLER:65535

The chaincode is going to run now to add flow13

2019-12-12 13:12:07.971 UTC [chaincodeCmd] InitCmdFactory ->INFO 001
Retrieved channel (mychannel) orderer endpoint:
orderer.example.com:7050

2019-12-12 13:12:07.985 UTC [chaincodeCmd] chaincodeInvokeOrQuery ->
INFO 002 Chaincode invoke successful. result: status:200

the chaincode finished now with invocation of initFlow

Με τον τρόπο αυτό, και με τον κατάλληλο chaincode κώδικα που υλοποιήθηκε, επιτυγχάνεται ενός είδους αποκεντροποίηση στον τρόπο με τον οποίο λειτουργούν τα SDN δίκτυα. Οι κανόνες αποθηκεύονται στην p2p αρχτεκτονική που προσφέρει το blockchain. Έτσι είναι αυτομάτως κάθε στιγμή προσβάσιμοι, για επαλήθευση ή τροφοδοσία στην βάση δρομολήγησης του κατάλληλου δρομολογητή, του SDN δικτύου.

Τα βασικά αρχεία για την υλοποίηση του παρόντος σεναρίου, με βάση όσα περιγράψαμε παραπάνω, βρίσκονται διαθέσιμα στο github repository που βρίσκεται στο τέλος της παρούσας εργασίας [120].

5.4 Αξιολόγηση

Για την υλοποίηση του παραπάνω πειράματος, χρησιμοποιήθηκαν ολοκληρωμένες υπηρεσίες και λογισμικά. Τα εργαλεία αυτά παραμετροποιήθηκαν κατάλληλα, έτσι ώστε ο συνδυασμός τους να προσφέρει το επιθυμητό για την εργασία αποτέλεσμα το οποίο περιλάμβανει τον συσχετισμό των SDN αρχιτεκτονικών με την τεχνολογία του Blockchain. Με αυτόν

τον τρόπο επιτυγχάνεται ένας από τους στόχους της παρούσας εργασίας, ο οποίος είναι να συνδυαστούν δύο νέα τεχνολογικά αντικείμενα, διαμετρικά αντίθετων αρχιτεκτονικών. Αυτό συμβαίνει καθώς η αποκεντροποίηση των p2p blockchain οντοτήτων και η κεντρικοποιημένη υλοποίηση των SDN αρχιτεκτονικών δεν συμβαδίζουν εκ πρώτης όψεως. Ένας άλλος στόχος, που καθόρισε την υλοποίηση είναι η προσφορά που μπορεί να έχει μία τεχνολογία βασισμένη στο blockchain, όταν αυτή εφαρμόζεται σε SDN αρχιτεκτονικές. Ο παραπάνω συσχετισμός δασφαλίζει την ασφάλεια και την ακεραιότητα των δεδομένων, όπως αυτά ανταλλάσονται σε δικτυακά περιβάλλοντα.

Για την λειτουργία και την σωστή δομή των SDN δικτύων, αλλά και για την ομαλή και έυκολη διαχείρισή τους, απαραίτητη είναι η ύπαρξη των κανόνων δρομολόγησης στις βάσεις δεδομένων των εικονικών δρομολογητών. Όπως εκτενώς αναλύσαμε στα προηγούμενα κεφάλαια, τα SDN δίκτυα για να λειτουργήσουν χρειάζονται κεντρικές οντότητες, τους controllers, για την λήψη αποφάσεων στην δρομολόγηση των πακέτων. Ο τρόπος με τον οποίο λειτουργεί μια SDN δικτύωση επομένως, βασίζεται σε σε ένα κεντρικοποιημένο μοντέλο συστήματος, καθώς οι δρομολογήσεις των πακέτων καθορίζονται από τις κεντρικοποιημένες αρχές των ελεγκτών.

Με την παραπάνω υλοποίηση, του μοντέλου που αναλύσαμε στο προηγούμενο κεφάλαιο, το ερώτημα το οποίο θέσαμε στην ενότητα 1.2.1 μπορεί πλέον να απαντηθεί. Η κεντρικοποιημένη δομή των SDN δικτύων μπορεί και συνδυάζεται με κατανεμημένες δικτυακές δομές όπως τα διομότιμα δίκτυα. Το αποτέλεσμα είναι να διαθέτει έτσι στοιχεία και πλεονεκτήματα και των δύο αυτών δομών. Για παράδειγμα, οποιαδήποτε είδους δυσλειτουργία σε ένα SDN δίκτυο η οποία οφείλεται σε αποτυχία ενός μόνο σημείου, μπορεί πλέον να μετριαστεί με την χρήση κατάλληλου κώδικα. Το ονομάζόμενο chaincode, μπορεί και υλοποιεί ένα κατανεμημένο δίκτυο blockchain. Το p2p δίκτυο, συμβάλλει στην διατήρηση της πληροφορίας (εν προκειμένω οι κανόνες δρομολόγησης του SDN δικτύου). Αυτό συμβαίνει, διότι ο αποκεντρωμένος χαρακτήρας των p2p δικτύων αυξάνει την ευρωστία, καταργώντας το ενιαίο σημείο αποτυχίας. Το p2p δίκτυο το οποίο υλοποιεί ένα blockchain, δεν εξαφανίζει, εξ ορισμού, μόνο τις αποτυχίες ενός σημείου, αλλά μπορεί να φανεί χρήσιμο και στην περίπτωση όπου κάποιος δίαυλος επικοινωνίας μεταξύ ελεγκτή και εικονικού δρομολογητή χαθεί. Έτσι για παράδειγμα, στην περίπτωση μίας επανεκκίνησης του δικτύου, η απώλεια των κανόνων δρομολόγησης, μπορεί να συνεχίζει να συμβαίνει, αλλά εφόσον είναι καταγεγραμμένοι στο blockchain, είναι άμεσα διαθέσιμοι για την εισαγωγή τους, εκ νέου, στο SDN δίκτυο.

Στην προηγούμενη ενότητα παρουσιάσαμε, και αναλύσαμε, καθ΄ όλα τα παραπάνω, έναν μηχανισμό ο οποίος, στην περίπτωση μίας ασυνέχειας στην διαχείριση ή στην λειτουργία του, μπορεί να επαναφέρει την τελευταία καταγεγραμμένη διαχειριστική κατάσταση δικτύου στο ελεγκτή και στους εικονικούς δρομολογητές. Η καταγραφή των κανόνων δρομολόγησης στο blockchain προσφέρει ένα είδος ιστορικού εγγραφών οι οποίες είναι αμετάβλητες στον χρόνο, καθώς και αμετάβλητες από οποιαδήποτε κακόβουλη οντότητα. Το τελευταίο φυσικά συμβαίνει εφόσον πληρούνται οι κατάλληλες προϋποθέσεις οι οποίες καθιστούν ένα blockchain p2p δίκτυο ανθεκτικό σε αλλοιώσεις από τρίτους [105, 106].

Το πρόβλημα που εξετάσαμε στην παρούσα εργασία επομένως, σχετίζεται με την καταγραφή και την αποθήκευση των κανόνων δρομολόγησης ενός SDN δικτύου. Η καταγραφή αυτή

έγινε σε έναν decentralized ledger ενός blockchain p2p δικτύου, το οποίο βασίστηκε στο Hyperledger Project. Οι κανόνες δρομολόγησης καθορίζουν το μονοπάτι που ακολουθεί ένα πακέτο δικτυακά. Ορίζονται από τον SDN ελεγκτή και προωθούνται στους εικονικούς δρομολογητές. Από του εικονικούς δρομολογητές και συγκεκριμένα από την βάση στην οποία έχουν αποθηκευτεί, προωθούνται κατά βούληση, στο peer-to-peer δίκτυο για να αποθηκευτούν με την σειρά τους, στο ledger της κατανεμμημένης αλυσίδας εγγραφών. Όπως εξηγείσαμε και στην προηγούμενη παράγραφο, αν για οποιονδήποτε αποτύχει η ομαλή λειτουργία του SDN δικτύου με τον προβλεπόμενο τρόπο υπονομεύει την ακεραιότητα του δικτύου.

Κεφάλαιο 6

Συμπεράσματα & Μελλοντικές Επεκτάσεις

6.1 Συμπεράσματα και Συνεισφορά

Σε αυτή τη εργασία, εξετάσαμε μέρος του προβλήματος το οποίο αφορά την ασφάλεια και την ακεραιότητα των δεδομένων στα SDN δίκτυα και πως αυτό μπορεί να βελτιωθεί με την χρήση δομών blockchain. Μία από τις κύριες συμβολές της δουλειάς στην παρούσα διπλωματική είναι να εκφραστεί η τεχνολογία του blockchain ως συνδυαστικός παράγοντας βελτιστοποίησης στα δεδομένα τα οποία επεξαργάζεται ένα SDN δίκτυο καθώς και να προτείνει μία νέα ιδέα για την καλύτερη διαχείριση τους με βάση το p2p δίκτυο στο οποίο μεταφέρονται τα δεδομένα αυτά. Συγκεκριμένα τα τελευταία αφορούν του κανόνες δρομολόγησης του Openflow πρωτοκόλλου σε ένα SDN δίκτυο.

Έχει δοθεί υλικό για διαφορετικές, αλλά παρόμοιες, εργασίες άλλων ατόμων που ασχολούνται με το αντικείμενο. Συγκεκριμένα, γίνανε αναφορές άλλων εργασιών, τόσο από την πλευρά των SDN δικτύων, όσο και από την θεματική περιοχή των blockchain τεχνολογιών και τα διομότιμα δίκτυα. Η συμβολή της παρούσας εργασίας είναι διττή, καθώς συνδυάζει τα δύο διαφορετικά μοντέλα συστημάτων τα οποίο περιγράφηκαν στις πρώτες παραγράφους.

Αρχικά, παρουσιάστηκαν οι έννοιες και οι οντότητες μίας βασικής αρχιτεκτονικής για τα SDN δίκτυα, η οποία κατηγοριοποιείται ως κεντρικοποιημένη αρχιτεκτονική. Έπειτα, περιγράφηκαν οι βασικές έννοιες που χαρακτηρίζουν ένα p2p δίκτυο και πιο ειδικά τα δομικά στοιχεία, οι λειτουργίες και οι διαφορετικοί τύποι ενός blockchain.

Ο κύριος στόχος την εργασίας μας ήταν η δημιουργία ενός μηχανισμού και μία αρχιτεκτονικής η οποία θα συνέδεε τα SDN δίκτυα με ένα blockchain, με βασικό στόχο την συμβολή, της ακεραιότητας και της ασφάλειας των δεδομένων του blockchain, στην SDN αρχιτεκτονική. Έτσι, χρησιμοποιώντας δύο εργαλεία όπως OpenFlow [99] και Hyperledger Fabric [63], προέκυψε μία νέα αρχιτεκτονική προσέγγιση. Η συμβολή μας σε αυτό το μέρος ήταν να προσαρμόσουμε τους κανόνες δρομολόγησης του SDN δικτύου έτσι ώστε να γίνει καταχώριση αυτών στο blockchain, το οποίο μέχρι στιγμής δεν έχει αξιοποιηθεί σε εφαρμογές που σχε-

τίζονται το κομμάτι αυτό την τεχνολογίας. Η ένταξη του blockchain στα SDN δίκτυα, αίρει τους περιορισμούς που έχουν αυτά στην ασφάλεια και την καταχώρηση των κανόνων δρομολόγησης, και ανοίγει νέους ορίζοντες στην διαχειριση μεγάλων δικτυακών υποδομών με πιο ενισχυμένο και ασφαλή τρόπο. Η χρήση της τεχνολογίας του blockchain σε κεντρικοποιημένα περβάλλοντα δικτύων όπως οι υλοποιήσεις SDN θεωρούμε, με βάση την παρούσα εργασία, ότι συμβάλει σε μία πιο λειτουργική και διαχειρισιμη κατάσταση.

Ενσωματώνοντας μία τεχνολογία βασισμένη στο blockchain, ένα SDN δίκτυο μπορεί να λειτουργήσει με αυτό, ακόμα και αν υπάρξει αποτυχία ενός μόνο σημείου. Χωρίς την χρήση blockchain, το παραπάνω συνέβαινε μόνο σε μεμονομένες περιπτώσεις SDN δικτύακών υλοποιήσεων, όπως για παράδειγμα οι κατανεμημένες αρχιτεκτονικές στο επίπεδο ελέγχου [60, 107].

Ο συνδυασμός των δύο αυτών τεχνολογιών, θεωρούμε ότι προσφέρει στην υλοποίηση μιας πιό στιβαρής λειτουργίας των SDN δικτύων. Αυτό συμβαίνει, όπως αναφέρθηκε και στο προηγούμενο κεφάλαιο, διότι μία blockchain δομή, μπορεί να λειτουργήσει και να εξασφαλίσει την ακεραιότητα των δεδομένων που φιλοξενεί, χωρίς την ανάγκη για μία κεντρική αρχή πιστοποίησης [108, 109], που ήταν εώς τώρα, βασικό στοιχείο των μοντέλων και δομών πιστοποίησης. Με αυτόν τον τρόπο μειώνεται το ρίσκο της απώλειας ή της αλλοίωσης των δεδομένων, από τρίτες κακόβουλες οντότητες. Ως αποτέλεσμα οι κανόνες δρομολόγησης, οι οποίοι μπορούν ανά πάσα στιγμή να αλλάξουν ή να τροποποιηθούν στο SDN δίκτυο, καταγράφονται και στο ledger του p2p δικτύου, δημιουργώντας έτσι μία αλυσίδα από τις συναλλαγές με όλες τις εγγραφές των κανόνων δρομολόγησης που έχουν υπάρξει. Η αλυσίδα συναλλαγών είναι ασφαλή, αξιόπιστη, ελέγξιμη και αμετάβλητη, τέσσερα χαρακτηριστικά σημεία τα οποία αποτελούν μέρος των ιδιοτήτων μίας blockchain υλοποίησης [110].

Η τεχνολογία του blockchain ίσως να μην είναι μια μεμονομένη λύση, που μπορεί με ασφάλεια, να αποτρέψει όλα τα πιθανά σημεία αποτυχίας. Όμως γνωρίζουμε ότι είναι ένα πιο προηγμένο σημείο εκκίνησης για υποδομές ασφάλειας, έναντι των κεντρικοποιημένων συστημάτων που έχουν κεντρικά σημεία αποτυχίας [111]. Το ίδιο ισχύει και για την σύζευξη του με τα SDN δίκτυα.

Εν κατακλείδι, η συνεισφορά της παρούσας εργασίας, αφορά τον τρόπο με τον οποίο θα μπορούσε να δομηθεί και να λειτουργήσει ένα SDN δίκτυο κατά έναν πιο ασφαλή και εύρωστο τρόπο. Η αποθήκευση των κανόνων δρομολόγησης ενός SDN, σε αποκεντρωμένα δίκτυα blockchain έχει να προσφέρει πολλά οφέλη. Εκτενέστερη και περαιτέρω μελέτη τέτοιου είδους προβλημάτων, που συνδυάζουν εκ διαμέτρου αντίθετες έννοιες πιστεύουμε ότι μπορεί να προσφέρει λύσεις σε πολλά προβλήματα, που ακόμα υφίστανται και σχετίζονται με την ασφάλεια και την σωστή λειτουργία μίας δικτυακής υποδομής όπως το πρόβλημα αρνησης εξυπηρέτησης και τις blochchain λύσεις που παρουσιάζονται στις εργασίες [37, 112].

6.2 Μελλοντικές Επεκτάσεις

Οι μελλοντικές εργασίες πάνω στο παρόν θέμα, αφορούν βαθύτερη ανάλυση και μελέτη των συγκεκριμένων μεθόδων, νέες προτάσεις για δοκιμή διαφορετικών μοντέλων και αρχιτεκτονικών δικτύωσης, ή και απλώς περισσότερη περιέργεια και φαντασία.

Υπάρχουν ακόμα αρκετές ιδέες και σενάρια, που μπορούν να δοκιμαστούν κατά την ανάπτυξη αρχιτεκτονικών που συνδυάζουν τις έννοιες των SDN δικτύων και του blockchain. Αυτή η εργασία επικεντρώθηκε κυρίως στη καταχώρηση των OpenFlow κανόνων δρομολόγησης ενός SDN δικτύου, σε μία δομή blockchain. Η αρχιτεκτονική που υλοποιήσαμε βασίστηκε και αποκτήθηκε από βιβλιογραφία παρόμοιων αρχιτεκτονικών και ιδεών που συνδυάζουν τις δύο αυτές τεχνολογίες, που είχαν όμως σκοπό την εξαγωγή διαφορετικών συμπερασμάτων καθώς η μελέτη αφορά παραπλήσιες επιστημονικές περιοχές με συναφή αντικείμενα. Περαιτέρω μελέτη της συγκεκριμένης θεματικής επιστημονικής περιοχής έχει νόημα να μελετηθεί περαιτέρω για τον λόγο ότι, συγχονεύοντας τις εφαρμογές που έχει το blockchain με τις SDN τεχνολογίες, επιπλέον στρώματα προστασίας δεδομένων μπορούν να ενορχηστρωθούν και να δημιουργήσουν ένα πιο ολοκληρωμένο αποτέλεσμα.

Η επέκταση της μελέτης, σχετικά με τον συνδυασμό των δύο αυτών τεχνολογιών, για μία πιο ασφαλή διαχείριση δεδομένων και οντοτήτων στα προγραμματιζόμενα και έξυπνα δίκτυα του μέλλοντος, μπορούν να δοκιμαστούν κάποιες από τις ιδέες που αναφέρονται παρακάτω.

Αρχικά, μπορεί να μελετηθεί περαιτέρω η καταγραφή στο blockchain όλων των SDN οντοτήτων, και όχι μόνο των κανόνων δρομολόγησης που εμείς υλοποιήσαμε στην παρούσα εργασία. Τέτοιου είδους καταγραφή μπορεί να αφορά για παράδειγμα όλες τις προσπάθειες δημιουργίας ή διαγραφής δεδομένων, είτε είναι έγκυρες και θεμιτές είτε όχι. Μετά την καταγραφή τους στο blockchain, τα δεδομένα θα είναι διαθέσιμα και απαλλαγμένα από κάθε παραποίηση.

Πιο συγκεκριμένα σχετικά με την παραποίηση δεδομένων και την χρήση τους για αποδεικτικούς σκοπούς σε συστήματα ασφαλείας, αξίζει η περαιτέρω μελέτη και υλοποίηση συστημάτων που κάνουν ανάκτηση δεδομένων από το blockchain. Η λειτουργία τέτοιων συστημάτων θα έχει ως βάση ερωτήματα (queries) προς τον ledger, όπως αναφέρθηκαν στην ενότητα 3.3.1. Τετοια συστήματα βέβαια προυποθέτουν την σωστή καταγραφή διτκυακών αλλά και άλλων δεδομένων στο blockchain.

Ένα άλλο ενδιαφέρον παραπλήσιο αντικείμενο μελέτης είναι ο έλεγχος (whitelist/blacklist) των SDN οντοτήτων ενός δικτύου, όπως διαχειριστές, ή εγκεκριμένες αξιόπιστες οντότητες όπως αυτές των SDN controllers. Φυσικά το παραπάνω πιστεύουμε θα ήταν δυνατόν να ελεγχθεί μέσα από ένα blockchain δίκτυο. Το αποτέλεσμα θα ηταν όχι μόνο η αύξηση αξιοπιστίας και εμπιστοσύνης στις οντότητες του SDN δικτύου αλλά και εν τέλει, ένα πιο καθαρό από κακόβουλες οντότητες δίκτυο, καθώς δικτυακά στοιχεία με τέτοιου είδους χαρακτήρα θα μπορούσαν να απομονωθούν πιο εύκολα.

Εξίσου ενδιαφέρον αντικείμενο μελέτης αποτελεί η αξιοποίηση του blockchain στις τεχνολογίες SDN δικτύων για την διαχείριση και τον έλεγχο των ταυτοτήτων και την πιστοποίηση ή μη των δικτυακών στοιχείων. Αυτό ισχύει και μπορεί να υλοποιηθεί για την εξασφάλιση εμπιστευτικότητας και ακεραιότητας, όχι μόνο των κανόνων δρομολόγησης αλλά όλων των δεδομένων που βρίσκονται σε κίνηση σε ένα SDN δίκτυο.

Σε συνδυασμό με το παραπάνω, είναι εφικτή η πιστοποίηση κάθε χρήστη, συσκευής ή εφαρμογής που βρίσκεται πάνω σε ένα SDN δίκτυο, έτσι ώστε να επιτρέπεται ή να αποτρέπεται

η πρόσβαση στο δίκτυο μέσω αυθεντικοποίησης από το blockchain.

Μέσα από την παρούσα εργασία, το κύριο ερώτημα που τίθεται είναι το εξής: Πως γίνεται να υλοποιήσουμε πρόσθετα επίπεδα ασφαλείας, κλιμάκωσης και ελεγκτικής ικανότητας στα SDN δίκτυα, επιτρέποντας ελέγχους και χρησιμοποιώντας το blockchain ως ¨πύλη' για τις επερχόμενες SDN τεχνολογίες.

Βιβλιογραφία

[1] Jesús San-Miguel-Ayanz, Hugo Costa, Daniele de Rigo, Giorgio Libertá, Tomas Artés Vivancos, Tracy Durrant, Daniel Nuijten, Peter Löffler, Peter Moore. Basic criteria to assess wildfire risk at the pan-European level. *JRC Technical Reports*, pages 5-9, 2018.

[2] Greek Research and Technology Network. High Performance Computing HPC Services. `https://hpc.grnet.gr/en/` 2016. Last accesed on 08/11/2019

[3] Karl Worthmann, Christopher M. Kellett, Senior Member, Philipp Braun, Lars Grüne, and Steven R. Weller. Distributed and Decentralized Control of Residential Energy Systems Incorporating Battery Storage. *IEEE TRANSACTIONS ON SMART GRID*, pages 1-10, 2015.

[4] Yi Seung, Kravets Robin. MOCA : MObile Certificate Authority for Wireless Ad Hoc Networks. page 1, 2004.

[5] Satoshi Nakamoto. Bitcoin:A peer-to-peer electronic cash system. pages 1-9, 2008.

[6] Alan T. Sherman, Farid Javani, Haibin Zhang, Enis Golaszewski. On the Origins and Variations of Blockchain Technologies. In *IEEE Security & Privacy*, (Volume 17, Issue 1, Jan.-Feb.2019).

[7] Zhu, Xiaoyang and Badr, Youakim. Identity Management Systems for the Internet of Things: A Survey Towards Blockchain Solutions. In *Sensors, December 2018*, (Volume 18).

[8] Papadakis-Vlachopapadopoulos Konstantinos, González Román, Dimolitsas Ioannis, Dechouniotis Dimitrios, Ferrer Ana, Papavassiliou Symeon. Collaborative SLA and reputation-based trust management in cloud federations. In *Future Generation Computer Systems*, (Volume 100, pages 498-512).

[9] Dechouniotis, Dimitrios and Dimolitsas, Ioannis and Papadakis-Vlachopapadopoulos, Konstantinos and Papavassiliou, Symeon Fuzzy Multi-Criteria Based Trust Management in Heterogeneous Federated Future Internet Testbeds In *Future Internet*.

[10] Gambetta, Diego. Can We Trust Trust? `http://www.sociology.ox.ac.uk/papers/gambetta213-237.pdf` In *Trust: Making and Breaking Cooperative Relations, Department of Sociology, University of Oxford*, chapter 13, pp. 213-237, 1988.

[11] European Telecommunications Standards Institute 2016. /https://www.etsi.org Last accesed on 08/11/2019

[12] Kamvar, Sepandar and Schlosser, Mario and Garcia-molina, Hector The eigentrust algorithm for reputation management in p2p networks. In *Proceedings of the ACM International Conference on World Wide Web*, pp. 640–651, Budapest, Hungary,20–24 May 2003.

[13] Garg Anurag, Battiti Roberto. The Reputation, Opinion, Credibility and Quality (ROCQ) Scheme

[14] Kotaro Kataoka, Saurabh Gangwar, Prashanth Podili Trust list: Internet-wide and distributed IoT traffic management using blockchain and SDN In *2018 IEEE 4th World Forum on Internet of Things (WF-IoT)*, February 2018.

[15] Pradip Kumar Sharma, Saurabh Singh, Young-Sik Jeong, and Jong Hyuk Park DistBlockNet : A distributed blockchains-based secure SDN architecture for IoT networks In *IEEE Communications Magazine* (Volume: 55, Issue: 9, Sept. 2017).

[16] Navid Rajabi and Jihad Qaddour SDIoBoT: A Software-Defined Internet of Blockchains of Things Model In *International Journal of Internet of Things*, (Volume 8, Issue 1, pages 17-26, 2019) http://article.sapub.org/10.5923.j.ijit.20190801.03.html.

[17] Thubert, Pascal and Palattella, Maria and Engel, Thomas 6TiSCH centralized scheduling: When SDN meet IoT In *2015 IEEE Conference on Standards for Communications and Networking (CSCN), Tokyo, Japan*.

[18] Li, Jie and Altman, Eitan and Touati, Corinne A General SDN-based IoT Framework with NVF Implementation. *September 2015*.

[19] Valdivieso, Leonardo and Benito Peral, Alberto and Barona, Lorena and García Villalba, Luis SDN: Evolution and Opportunities in the Development IoT Applications In *International Journal of Distributed Sensor Networks 2014(10):1-10, May 2014*.

[20] Nolot, Florent and Flauzac, Olivier and Carlos Javier, Gonzalez and Hachani, Abdelhak SDN Based Architecture for IoT and Improvement of the Security In *IEEE 29th International Conference on Advanced Information Networking and Applications Workshops*.

[21] Vilalta, Ricard and Mayoral, Arturo and Pubill, David and Casellas, Ramon and Martinez, Ricardo and Serra, Jordi and Verikoukis, Christos and Muñoz, Raul End-to-End SDN Orchestration of IoT Services Using an SDN/NFV-enabled Edge Node. In *Optical Fiber Communication Conference*.

[22] cp, Vandana, New Horizon College of Engineering. Security improvement in IoT based on Software Defined Networking (SDN). In *International Journal of Science, Engineering and Technology Research (IJSETR), January 2016*.

[23] Cui, Laizhong and Yu, F. and Yan, Qiao, When big data meets software-defined networking: SDN for big data and big data for SDN. In *IEEE Network 30(1) pages: 58-65, January 2016*.

[24] Kang, Jiawen and Huang, Xumin and Wu, Maoqiang and Maharjan, Sabita and Xie, Shengli and Zhang, Yan, Blockchain for Secure and Efficient Data Sharing in Vehicular Edge Computing and Networks. In *IEEE Internet of Things Journal, October 2018*.

[25] Bozic, Nikola and Pujolle, Guy and Secci, S., A tutorial on blockchain and applications to secure network control-planes. In *2016 3rd Smart Cloud Networks & Systems (SCNS), December 2016*.

[26] Rosa, Raphael and Esteve Rothenberg, Christian, Blockchain-Based Decentralized Applications for Multiple Administrative Domain Networking. In *ACM SIGCOMM 2018 Conference, August 2018*.

[27] Sonkoly, Balázs and Czentye, János and Szabo, Robert and Jocha, Dávid and Elek, Janos and Sahhaf, Sahel and Tavernier, Wouter and Risso, Fulvio, Multi-Domain Service Orchestration Over Networks and Clouds. In *ACM SIGCOMM Computer Communication Review, 45(5):377-378, August 2015*.

[28] Kostas, Katsalis and Nikaein, Navid and Edmonds, Andy, Multi-Domain Orchestration for NFV: Challenges and Research Directions. In *2016 15th International Conference on Ubiquitous Computing and Communications and 2016 International Symposium on Cyberspace and Security (IUCC-CSS), December 2016*.

[29] Sartor, Nadeem and Javaid, Nadeem, Hybrid Network Architecture using Blockchain for the Smart Cities.

[30] Ferrag, Mohamed Amine and Derdour, Makhlouf and Mukherjee, Mithun and Derhab, Abdelouahid and Maglaras, Leandros and Janicke, Helge, Blockchain Technologies for the Internet of Things: Research Issues and Challenges.

[31] Basnet, Sadhu and Shakya, Subarna, BSS: Blockchain security over software defined network.

[32] Roger B. Dannenberg. An Introduction to Serpent. `https://www.cs.cmu.edu/~music/serpent/doc/serpent.htm` Last accessed on 21/11/2019

[33] A Next-Generation Smart Contract and Decentralized Application Platform. `https://github.com/ethereum/wiki/wiki/White-Paper` Last accessed on 21/11/2019

[34] Martìn Casado, Michael J. Freedman, Justin Pettit, Jianying Luo, Nick McKeown, Scott Shenker Ethane: Taking Control of the Enterprise. In *ACM SIGCOMM Computer Communication Review*, (Volume 37, Issue 4, pages 1-12, October 2007).

[35] Distributed Denial of Service: Trin00, Tribe Flood Network, Tribe Flood Network 2000, and Stacheldraht CIAC-2319. `https://apps.dtic.mil/dtic/tr/fulltext/u2/a396999.pdf` Last accessed on 21/11/2019

[36] Duan, Yucong and Fu, Guohua and Zhou, Nianjun and Sun, Xiaobing and Narendra, Nanjangud and Hu, Bo, Everything as a Service (XaaS) on the Cloud: Origins, Current and Future Trends. April 2016.

[37] Rodrigues, Bruno and Bocek, Thomas and Stiller, Burkhard, Enabling a Cooperative, Multi-domain DDoS Defense by a Blockchain Signaling System (BloSS). In *42nd IEEE Conference on Local Computer Networks 2017 (LCN 2017)*.

[38] Salman, Ola and Elhajj, Imad and Kayssi, Ayman and Chehab, Ali, SDN controllers: A comparative study. In *2016 18th Mediterranean Electrotechnical Conference (MELECON)*.

[39] Hari, Adiseshu and Lakshman, T., The Internet Blockchain: A Distributed, Tamper-Resistant Transaction Framework for the Internet. In *the 15th ACM Workshop, November 2016*.

[40] Albarqi, Aysha and Alzaid, Ethar and Ghamdi, Fatimah and Asiri, Somaya and Kar, Jayaprakash, Public Key Infrastructure: A Survey. In *Journal of Information Security, pages 31-37, January 2015*.

[41] Schehlmann, Lisa and Abt, Sebastian and Baier, Harald, Blessing or curse? Revisiting security aspects of Software-Defined Networking. In *Proceedings of the 10th International Conference on Network and Service Management, CNSM 2014*.

[42] Network Functions Virtualisation (NFV); Ecosystem; Report on SDN Usage in NFV Architectural Framework. `https://www.etsi.org/deliver/etsi_gs/NFV-EVE/001_099/005/01.01.01_60/gs_NFV-EVE005v010101p.pdf` Last accessed on 21/11/2019

[43] Alvarenga, Igor and Rebello, Gabriel and M. B. Duarte, Otto Carlos, Securing configuration management and migration of virtual network functions using blockchain. In *NOMS 2018-2018 IEEE/IFIP Network Operations and Management Symposium, April 2018*.

[44] Sakic, Ermin and Sardis, Fragkiskos and Guck, Jochen and Kellerer, Wolfgang, Towards Adaptive State Consistency in Distributed SDN Control Plane. In *IEEE International Conference on Communications, May 2017, Paris*.

[45] Singh, Irish and Lee, Seok-Won, Comparative Requirements Analysis for the Feasibility of Blockchain for Secure Cloud. In *Requirements Engineering for Internet of Things, pp.57-72, January 2018*.

[46] Gaetani, Edoardo and Aniello, Leonardo and Lombardi, Federico and Margheri, Andrea and Sassone, V, Blockchain-based database to ensure data integrity in cloud computing environments. In *Italian Conference on Cybersecurity, January 2017*.

[47] Steichen, Mathis and Hommes, Stefan and State, Radu, ChainGuard — A firewall for blockchain applications using SDN with OpenFlow. In *2017 Principles, Systems and Applications of IP Telecommunications (IPTComm), September 2017*.

[48] Bozic, Nikola and Pujolle, Guy and Secci, S., A tutorial on blockchain and applications to secure network control-planes. In *2016 3rd Smart Cloud Networks & Systems (SCNS)*.

[49] Tuncer, Daphné and Koch, Robert and Badonnel, Rémi and Stiller, Burkhard, Security of Networks and Services in an All-Connected World (AIMS 2017).

[50] Scott Hilton. Dyn Analysis Summary Of Friday October 21 Attack. `https://dyn.com/blog/dyn-analysis-summary-of-friday-october-21-attack/` 2016. Last accesed on 21/11/2019

[51] Saurabh Kumar Agarwa. Understanding OpenFlow. `https://sdngeeks.files.wordpress.com/2014/08/understanding-openflow.pdf` June 2014. Last accesed on 21/11/2019

[52] Open Networking Foundation. Software-Defined Networking: The New Norm for Networks. `https://www.opennetworking.org/images/stories/downloads/sdn-resources/white-papers/wp-sdn-newnorm.pdf` 13 April 2012. Last accesed on 21/11/2019

[53] Badotra, Sumit and Singh, Japinder., Open Daylight as a Controller for Software Defined Networking. In *International Journal of Advanced Computer Research 8(5), May 2017*.

[54] Linux Foundation Collaborative Projects. OpenDaylight - An Open Source Community & Meritocracy for SoftwareDefined Networking. `https://www.opendaylight.org/wp-content/uploads/sites/14/2018/03/opendaylight_open_community_and_meritocracy_for_sdn_v3.pdf` Last accesed on 21/11/2019

[55] David Meyer, OpenDaylight Introduction and Overview. `http://www.1-4-5.net/~dmm/talks/interop_nyc_2013.pdf` In *Interop Annual Conference, New York City, 2013*.

[56] Khondoker, Rahamatullah and Zaalouk, Adel and Marx, Ronald and Bayarou, Kpatcha, Feature-based Comparison and Selection of Software Defined Networking (SDN) Controllers. In *ICCSA, January 2014*.

[57] Intel's Collaboration with the Open vSwitch Community, Open-vSwitch Enables SDN & NFV Transformation. `https://networkbuilders.intel.com/docs/open-vswitch-enables-sdn-and-nfv-transformation-paper.pdf`.

[58] Ben Pfaff, Justin Pettit, Teemu Koponen, Ethan Jackson, Andy Zhou, Jarno Rajahalme, Jesse Gross, Alex Wang, Joe Stringer, and Pravin Shelar, Keith Amidon, Martín Casado, The Design and Implementation of Open vSwitch. `https://www.usenix.org/system/files/conference/nsdi15/nsdi15-paper-pfaff.pdf` In *12th USENIX Symposium on Networked Systems Design and Implementation (NSDI 2015). May 4–6, 2015, Oakland, USA*.

[59] Zhou, Yuanhao and Zhu, Mingfa and Xiao, Limin and Ruan, Li and Duan, Wenbo and Li, Deguo and Liu, Rui, A Load Balancing Strategy of SDN Controller Based on Distributed Decision.

[60] Phemius, Kévin and Bouet, Mathieu and Leguay, Jeremie, DISCO: Distributed multi-domain SDN controllers. August 2013.

[61] Dixit, Advait and Hao, Fang and Mukherjee, Sarit and Lakshman, T. and Kompella, Ramana, Towards an Elastic Distributed SDN Controller. In *ACM SIGCOMM Computer Communication Review 43(4):7-12 · September 2013*.

[62] Blial, Othmane and Ben Mamoun, Mouad and Redouane, Benaini, An Overview on SDN Architectures with Multiple Controllers. In *Journal of Computer Networks and Communications 2016(2):1-8 · January 2016*.

[63] Linux Foundation Hyperledger Project, An Intoduction to Hyperledger, Whitepaper. `https://www.hyperledger.org/wp-content/uploads/2018/07/HL_Whitepaper_IntroductiontoHyperledger.pdf`.

[64] Hyperledger Project Documentation, Key Concepts, Chaincode. `https://hyperledger-fabric.readthedocs.io/en/release-1.4/chaincode.html`.

[65] Hyperledger Project Documentation, Key Concepts, Peers. `https://hyperledger-fabric.readthedocs.io/en/release-1.4/peers/peers.html`.

[66] Hyperledger Project Documentation, Key Concepts, Ordering Service. `https://hyperledger-fabric.readthedocs.io/en/release-1.4/orderer/ordering_service.html`.

[67] Hyperledger Project Documentation, Key Concepts, Membership Service Providers. `https://hyperledger-fabric.readthedocs.io/en/release-1.4/msp.html`.

[68] Hyperledger Project Documentation, Key Concepts, Channels. https://hyperledger-fabric.readthedocs.io/en/release-1.4/channels.html.

[69] Ethereum Collabaration Project, Learn about Ethereum. https://ethereum.org/learn/\#ethereum-basics.

[70] Hyperledger Project Documentation, Key Concepts, Peer Lifecycle Cahincode. https://hyperledger-fabric.readthedocs.io/en/latest/commands/peerlifecycle.html.

[71] Hyperledger Project Documentation, Key Concepts, Ledger. https://hyperledger-fabric.readthedocs.io/en/release-1.4/ledger/ledger.html.

[72] Brian Curran, What is a Merkle Tree? Beginner's Guide to this Blockchain Component. https://blockonomi.com/merkle-tree.

[73] Qadir, Junaid and Ahmed, Nadeem and Ahad, Nauman, Building Programmable Wireless Networks: An Architectural Survey. In *EURASIP Journal on Wireless Communications and Networking 2014(1) · October 2013*.

[74] Mininet Team, Emulator for rapid prototyping of Software Defined Networks. https://github.com/mininet/mininet.

[75] Mininet Team, A Mininet Overview. http://mininet.org/overview/.

[76] Mininet Team, Mininet Walkthrough. http://mininet.org/walkthrough/\#using-a-remote-controller.

[77] OpenDayLight Linux Foundation Project Software, Karaf version 0.8.4. https://nexus.opendaylight.org/content/repositories/public/org/opendaylight/integration/karaf/0.8.4/karaf-0.8.4.tar.gz.

[78] Brian Linkletter, Using the OpenDaylight SDN Controller with the Mininet Network Emulator. http://www.brianlinkletter.com/using-the-opendaylight-sdn-controller-with-the-mininet-network-emulator/.

[79] Jan Medved, Lukas Sedlak, Martin Vitez, Robert Varga, Tony Tkacik , YANG Tools Project. https://wiki.opendaylight.org/view/YANG_Tools:Main.

[80] Attar, Vahida and Piyush, Chandwadkar, Network Discovery Protocol LLDP and LLDP-MED. In *International Journal of Computer Applications 1(9), February 2010*.

[81] Srinivasan, Manikantan, EE Times - Tutorial on the Link Layer Discovery Protocol.

[82] Linux Foundation Collaborative Projects, OpenvSwitch Database (OVSDB). http://docs.openvswitch.org/en/latest/ref/ovsdb.7/.

[83] Linux Foundation Collaborative Projects, Open vSwitch Reference Guide. http://docs.openvswitch.org/en/latest/ref/.

[84] Jan Medved, Daniel Malachovsky, Juraj Sebin, Vijay Kannan, Chris Metz, Niklas Montin, Daniel Kuzma, Stanislav Jamrich, Zdenko Krnac, Sergey Madaminov, Andrej Vanko, Bimal Grewal, OpenDaylight OpenFlow Manager (OFM) App. https://github.com/CiscoDevNet/OpenDaylight-Openflow-App.

[85] Ben Alman, Tyler Kellen, Kyle Robinson Young, Vlad Filippov, Sindre Sorhus, Isaac Durazo, Jarrod Overson, Tim Branyen, Jörn Zaefferer, James Smith, Dave Geddes, Grunt, The Javascript Task Runner. https://gruntjs.com/.

[86] OpenDayLight Documentation, ODL OpenFlowPlugin, Release Master. https://buildmedia.readthedocs.org/media/pdf/odl-openflowplugin/latest/odl-openflowplugin.pdf.

[87] The Linux Foundation Projects, Hyperledger Fabric. https://www.hyperledger.org/projects/fabric.

[88] The Linux Foundation Projects, Hyperledger Fabric – Code Samples. https://github.com/hyperledger/fabric-samples/.

[89] The Linux Foundation Projects, Hyperledger Fabric – Using CouchDB. https://hyperledger-fabric.readthedocs.io/en/release-1.4/couchdb_tutorial.html.

[90] Apache, Couch Database. http://couchdb.apache.org/.

[91] The Linux Foundation Projects, Hyperledger Fabric – Commands Reference - configtxgen. https://hyperledger-fabric.readthedocs.io/en/release-1.4/commands/configtxgen.html.

[92] The Linux Foundation Projects, Hyperledger Fabric – Anchor Peer. https://hyperledger-fabric.readthedocs.io/en/release-1.4/glossary.html\#anchor-peer.

[93] Docker Inc., Docker Commands - Exec. https://docs.docker.com/engine/reference/commandline/exec/.

[94] The Linux Foundation Projects, Hyperledger Fabric – Commands Reference – Peer Channel. https://hyperledger-fabric.readthedocs.io/en/release-1.4/commands/peerchannel.html.

[95] Docker Inc., Docker Application Data - Volumes. https://docs.docker.com/storage/volumes/.

[96] Benton, Kevin and Camp, L. and Small, Chris. OpenFlow Vulnerability Assesment. http://www.ljean.com/files/vulnerability_analysis.pdf.

[97] Mutaher, Hamza. OpenFlow Controller-Based SDN: Security Issues and Countermeasures.

[98] Benabbou, Jaouad and Elbaamrani, Khalid and Idboufker, Noureddine. Security in OpenFlow-based SDN, opportunities and challenges.

[99] Steven Wallace, Uwe Dahlmann, Ron Milford, Chris Small, Indiana Center for Network Translational Research and Education. OpenFlow in a day. https://archive.nanog.org/sites/default/files/mon.tutorial.wallace.openflow.31.pdf.

[100] Ecessa. Everything You Need to Know About Network Failover, page 3. http://www.ecessa.com/wp-content/uploads/2015/02/Everything-You-Need-To-Know-About-Network-Failover.pdf.

[101] Curtis P. Kolovson, Hewlett-Packard Company. Fast Database Failover, page 13. https://patentimages.storage.googleapis.com/9c/b7/c0/d02b13eace9c37/US5951695.pdf.

[102] Richard Meyer, Kumar Gajjar, Chan Ng, Andrey Gusev, NetApp Inc . Failover processing in a storage system. https://patents.google.com/patent/US7039827B2/en.

[103] Zhang, Rui and Xue, Rui and Liu, Ling. Security and Privacy on Blockchain. https://arxiv.org/pdf/1903.07602.pdf.

[104] Pease, Marshall C. and Shostak, Robert E. and Lamport, Leslie. Reaching Agreement in the Presence of Faults. https://lamport.azurewebsites.net/pubs/reaching.pdf.

[105] Beck, Roman and Czepluch, Jacob and Lollike, Nikolaj and Malone, Simon. Blockchain - The Gateway to Trust-Free Cryptographic Transactions. https://aisel.aisnet.org/cgi/viewcontent.cgi?article=1145&context=ecis2016_rp.

[106] Zyskind, Guy and Zekrifa, Djabeur and Alex, Pentland and Nathan, Oz. Decentralizing Privacy: Using Blockchain to Protect Personal Data. https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=7163223.

[107] Canini, Marco and Kuznetsov, Petr and Levin, Dan and Schmid, Stefan A distributed and robust SDN control plane for transactional network updates.

[108] Puthal, Deepak and Malik, Nisha and Mohanty, Saraju and Kougianos, Elias and Yang, Chi. The Blockchain as a Decentralized Security Framework [Future Directions].

[109] Beck, Roman. Beyond Bitcoin: The Rise of Blockchain World.

[110] Underwood, Sarah. Blockchain beyond Bitcoin.

[111] Rick Conklin, Ben Murray. Securing Enterprise Blockchain with SDN. https://cdn2.hubspot.net/hubfs/5196751/D19_WP_Blockchain_091119.pdf.

[112] Rodrigues, Bruno and Bocek, Thomas and Lareida, Andri and Hausheer, David and Rafati niya, Sina and Stiller, Burkhard. A Blockchain-Based Architecture for Collaborative DDoS Mitigation with Smart Contracts.

[113] Satriana, Chandra and Sadov, Oleg and Grudinin, Vladimir. Development of An Ecology-oriented SDN Framework.

[114] Jared Haynes. OpenDayLight Hydrogen Release. https://www.slideserve.com/jared-haynes/opendaylight-hydrogen-release.

[115] OpenDayLight Project. OpenDayLight Wiki - Controller Cluster Deployment. https://wiki.opendaylight.org/view/CrossProject:Integration_Group:Controller-Cluster_Deployment.

[116] Fernández-Caramés, Tiago and Fraga-Lamas, Paula, A Review on the Use of Blockchain for the Internet of Things. In *May 2018 IEEE Access 6:32979-33001*.

[117] Mininet Team, Mininet. http://mininet.org/.

[118] Open Networking Foundation. OpenFlow Switch Specification. https://www.opennetworking.org/wp-content/uploads/2014/10/openflow-spec-v1.3.0.pdf 25 June 2012. Last accesed on 21/11/2019

[119] Hyperledger Project Documentation. https://hyperledger-fabric.readthedocs.io/en/release-1.4.

[120] Flow parser & chaincode files https://github.com/sebek88/netmode_thesis_code.