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

Δημιουργία Blockchain με χρήση DHT

Διπλωματικη Εργασια

του

ΦΟΙΒΟΥ ΒΑΡΘΑΛΙΤΗ

Επιβλέπων: Νεκτάριος Κοζύρης
Καθηγητής Ε.Μ.Π.

Εργαστηριο Υπολογιστικων Συστηματων
Αθήνα, Οκτώβριος 2019

Εθνικό Μετσόβιο Πολυτεχνείο
Σχολή Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών
Τομέας Τεχνολογίας Πληροφορικής και Υπολογιστών
Εργαστήριο Υπολογιστικών Συστημάτων

Δημιουργία Blockchain με χρήση DHT

Διπλωματικη Εργασια

του

ΦΟΙΒΟΥ ΒΑΡΘΑΛΙΤΗ

Επιβλέπων: Νεκτάριος Κοζύρης
Καθηγητής Ε.Μ.Π.

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την 9η Οκτωβρίου 2019.

(Υπογραφή)	*(Υπογραφή)*	*(Υπογραφή)*
.....	.....	.....
Νεκτάριος Κοζύρης	Γεώργιος Γκούμας	Δημήτριος Τσουμάκος
Καθηγητής Ε.Μ.Π.	Επ. Καθηγητής Ε.Μ.Π.	Αν. Καθηγητής Ιόνιο Παν.

Αθήνα, Οκτώβριος 2019

(Υπογραφή)

.....

Φοιβος Βαρθαλιτης

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

Εθνικό Μετσόβιο Πολυτεχνείο
Σχολή Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών
Τομέας Τεχνολογίας Πληροφορικής και Υπολογιστών
Εργαστήριο Υπολογιστικών Συστημάτων

Ευχαριστίες

Θα ήθελα να ευχαριστήσω τον επιβλέποντα καθηγητή κ. Νεκτάριο Κοζύρη καθώς και τη κα. Κατερίνα Δόκα που με εμπιστεύτηκαν και μου έδωσαν την ευκαιρία να ανακαλύψω έναν τόσο ενδιαφέροντα κλάδο.

Θέλω επίσης να ευχαριστήσω την οικογένεια μου για την στήριξη τους, την αγάπη τους και την εμπιστοσύνη τους στην αποφάσεις μου.

Τέλος θα ήθελα να ευχαριστήσω τους συμφοιτητές-φίλους μου που έκαναν το ταξίδι στη γνώση μια πραγματικά πολύτιμη εμπειρία για εμένα στο σύνολο των ακαδημαικών μου χρόνων, τόσο λόγω της αλληλοβοήθειας όσο και λόγω των δυνατών φιλιών που αναπτύξαμε.

Περίληψη

Το διαδίκτυο αναμφισβήτητα αποτελεί μία από τις πιο σπουδαίες εφευρέσεις και έχει φτάσει πλέον σε σημείο να προσδιορίζει σε μεγάλο βαθμό τη ζωή μας. Εχει ανατρέψει πλέον τα δεδομένα στον τρόπο που οι άνθρωποι αντιλαμβάνονται τον κόσμο και διαχειρίζονται τη γνώση, καθώς και καθορίζει τη λειτουργία του επιχειρηματικού κόσμου. Χαρακτηριστικό των υπηρεσιών που είναι διαθέσιμες στο διαδίκτυο είναι η αρχιτεκτονική client-server, δηλαδή ένα μοντέλο το οποίο καθιστά απαραίτητη τη χρήση μιας κεντρικής αρχής. Την τελευταία δεκαετία ωστόσο επικρατεί μία τάση αποκεντροποίησης της διαχείρισης των "ελεύθερων" διαδικτυακών εφαρμογών. Η ανάπτυξη της τεχνολογίας του Blockchain έχει συμβάλει σε μεγάλο βαθμό προς αυτή την κατεύθυνση.

Το Blockchain, το οποίο σαν ιδέα δεν συλλήφθηκε πρόσφατα αλλά παρουσιάστηκε ολοκληρωμένα το 2008, με την υλοποίηση ενός ηλεκτρονικού συστήματος διαχείρισης πληρωμών - του κρυπτονομίσματος Bitcoin. Το Bitcoin προτάθηκε κατά την περίοδο της οικονομικής κρίσης που ξέσπασε στην Αμερικανική ήπειρο και τελικά επηρέασε όλο τον κόσμο. Η βάση της σύλληψης του διαφαινόμενου πρωτεργάτη του Bitcoin (Satoshi Nakamoto) ήταν η προσφορά ενός συστήματος διαχείρισης πληρωμών, όπου θα υπήρχε διαφάνεια στις συναλλαγές και θα είχε αποκεντρωμένη φύση. Στόχος ήταν η παράκαμψη των Τραπεζικών ιδρυμάτων που, κατα τα φαινόμενα, με τις πρακτικές τους είχαν οδηγήσει την παγκόσμια οικονομία σε όξυνση.

Το Bitcoin καθώς και άλλες υλοποιήσεις κρυπτονομισμάτων με βάση των Blockchain, παρά την καινοτόμα φύση τους, έχουν και πολλά μειονεκτήματα. Η αύξηση του ενδιαφέροντος προς τα κρυπτονομίσματα από μεγάλους οργανισμούς με κύριο στόχο το κέρδος, έχει οδηγήσει στην δημιουργία μεγάλων φαρμών με υπολογιστές-κόμβους που ελέγχονται από λίγα φυσικά πρόσωπα. Συνεπώς ο αποκεντρωμένος χαρακτήρας σε πολλές περιπτώσεις έχει απολεσθεί αφού η μεγάλη πλειοψηφία της ισχύος και πλούτου του δικτύου συγκεντρώνεται σε λίγους. Παράλληλα η ενέργεια που καταναλώνεται απο τα δίκτυα κρυπτονομισμάτων είναι ασύμμετρα μεγάλη για την συνεισφορά τους στο κοινωνικό σύνολο. Τέλος το μέγεθος των δικτύων των δημοφιλών υλοποιήσεων (βλ. Bitcoin, Ethereum κλπ) καθιστά πολύ δύσκολη τη συμμετοχή στο μέσο χρήστη λόγω αποθηκευτικού χώρου και απαιτήσεων υλισμικού. Αυτά και άλλα πολλά είναι τα προβλήματα που αναδείχθηκαν με την ταχεία επέκταση της τεχνολογίας του Blockchain.

Σκοπός της παρούσας διπλωματικής εργασίας είναι η διερεύνηση της τεχνολογίας του Blockchain καθώς και η πρόταση μιας νέας προσέγγισης στο χώρο των κρυπτονομισμάτων, και όχι μόνο. Πιο συγκεκριμένα θα αναζητηθεί τρόπος ώστε να μειωθεί ο δεσμευμένος χώρος

από τις υλοποιήσεις, να ελαττωθεί η καταναλισκόμενη ενέργεια υπό την μορφή υπολογισμών και διαδικτυακών μηνυμάτων. Τέλος η πρόταση θέλει να προσφέρει στον χαρακτήρα αποκεντροποίησης που θέλουν να επιτύχουν τέτοιου είδους ιδέες. Συνοπτικά, λοιπόν, θα προταθούν πρωτόκολλα λειτουργίας που θα επιλύουν μία μερίδα των μειονεκτημάτων που αναφέρθηκαν παραπάνω.

Κατά τη διάρκεια εκπόνησης της διπλωματικής μελετήθηκε σε βάθος η τεχνολογία και τα χρησιμοποιούμενα πρωτόκολλα των δημοφιλών κρυπτονομισμάτων και ειδικότερα του κρυπτονομίσματος Semux. Επίσης διερευνήθηκαν εναλλακτικές μέθοδοι υλοποίησης των λειτουργικών αναγκών τέτοιου είδους αποκεντρωμένων εφαρμογών. Τελικά δημιουργήθηκε μία Proof Of Concept υλοποίηση - το ntuaSemux - με στόχο να επιδείξει βασικές λειτουργικότητες.

Λέξεις Κλειδιά

Blockchain, Consensus, Chord, Distributed hash table, Semux, Κατανεμημένα συστήματα

Abstract

The internet is ,undoubtedly, one of the most important inventions and has now reached the point of largely defining our lives. It has changed the way on how people perceive the world and manage knowledge, as well as determine the functioning of the business world. A characteristic of the services available online is the client-server architecture, that is, a model that requires the use of a central authority. In the last decade, however, there has been a tendency to decentralize the management of "free" web applications. The development of Blockchain technology has greatly contributed to this.

Blockchain is a technology not recently captured but fully introduced in 2008, with the implementation of an electronic payment management system - cryptocurrency Bitcoin. Bitcoin was proposed during the time of the economic crisis that erupted on the American continent and eventually affected the whole world. The basis of the proposal, from the emerging leader of Bitcoin Satoshi Nakamoto was to offer a payment management system that would be transparent based on its decentralized in nature. The aim was to bypass the banking institutions that, by their practices, had apparently led the global economy to a boom.

Bitcoin and other cryptocurrency implementations based on Blockchain, despite their innovative nature, have many disadvantages. The growing interest in cryptocurrencies from large organizations with a primary focus on profit has led to the creation of large farms with computer nodes controlled by a few individuals. As a result, the decentralized nature has in many cases been lost as the vast majority of the power and wealth of the network is concentrated in a few. At the same time the energy consumed by the cryptocurrency networks is asymmetrically large for their contribution to society. Finally, the size of the popular deployment networks (eg. Bitcoin, Ethereum etc) makes it very difficult for the average user to participate due to storage space and hardware requirements. These and many more are the problems brought about by the rapid expansion of Blockchain technology.

The purpose of this thesis is to explore the technology of Blockchain and to propose a new approach in the field of cryptocurrencies, and not only. In particular, we will look for ways to reduce the space reserved for implementations, reduce the energy consumed in the form of calculations and messaging among nodes. Finally, the proposal wants to offer the decentralized character, such ideas want to achieve . In summary, operating protocols will be proposed that will address some of the disadvantages mentioned above.

During the preparation of the thesis, the technology and protocols of the popular cryptocurrencies and in particular cryptocurrency Semux were studied in depth. Alternative methods of implementing the functional needs of such decentralized applications were also explored. Finally a Proof Of Concept implementation - ntuaSemux - was created to demonstrate key functionalities.

Keywords

Blockchain, Consensus, Chord, Distributed hash table, Semux, Distributed Systems

Περιεχόμενα

Κατάλογος Σχημάτων

Κεφάλαιο 1

Εισαγωγή

Ο κόσμος έχει προ πολλού εισέλθει στην εποχή της ψηφιοποίησης και του διαδικτύου. Το διαδίκτυο, με την επανάσταση που έφερε σε πολλές πτυχές, έχει αλλάξει πολλές πτυχές της καθημερινής ζωής και οικονομίας. Η βασική αλλαγή είναι το γεγονός της ταχύτατης διάδοσης της πληροφορίας ανεξαρτήτως γεωγραφικού προσδιορισμού, δηλαδή η παγκοσμιοποίηση της γνώσης. Στην πορεία, το διαδίκτυο άρχισε να χρησιμοποιείται για εμπορικούς σκοπούς με τη δημιουργία ηλεκτρονικών καταστημάτων. Στη συνέχεια δημιουργήθηκαν τα κοινωνικά δίκτυα τα οποία καταφέρνουν να εκμηδενίζουν τις αποστάσεις και να ενώνουν ανθρώπους από όλο τον κόσμο, δίνοντας τη δυνατότητα σε νέες μορφές επικοινωνίας και διαλόγου.

Η τεχνολογία του Blockchain μέσω της υλοποίησης του κρυπτονομίσματος Bitcoin έφερε μία επανάσταση στο τρόπο τον οποίο ο κόσμος αντιλαμβάνεται το χρήμα και τις οικονομικές συναλλαγές. Η ψηφιοποίηση του χρήματος με τη δημιουργία νέων ηλεκτρονικών νομισμάτων τα οποία δεν εποπτεύονται από κάποια κεντρική αρχή, τράπεζα, κυβέρνηση ή οργανισμό, είναι πλέον πραγματικότητα. Αυτά τα νομίσματα ονομάζονται κρυπτονομίσματα καθώς μέσω των κρυπτογραφικών πρωτοκόλλων που χρησιμοποιούν, διασφαλίζεται η λειτουργία τους.

Το Blockchain είναι ένα ψηφιακό, κατανεμημένο, δημόσιο λογιστικό βιβλίο στο οποίο καταγράφονται γεγονότα με τρόπο επαληθεύσιμο και αδιάβλητο. Κάθε νέα συστάδα καταχωρήσεων ονομάζεται μπλοκ και συνδέεται με το αμέσως προηγούμενο ως επόμενο κομμάτι της αλυσίδας. Η τεχνολογία του Blockchain βασίζει τη λειτουργία της σε πολλούς υπολογιστές (κόμβους) ανά τον κόσμο. Κάθε κόμβος διατηρεί συνήθως το πλήρες αντίγραφο όλης της πληροφορίας που είναι καταγεγραμμένη στην αλυσίδα απο την αρχή της δημιουργίας της. Οποιοδήποτε γεγονός γράφεται στην αλυσίδα είναι αδύνατο να διαγραφεί η να τροποποιηθεί. Με αυτό τον τρόπο εξασφαλίζεται ότι ανά πάσα στιγμή παρουσιάζεται μία κοινή αλήθεια σε όλους τους εμπλεκόμενους κόμβους. Επίσης, χάρη στο γεγονός ότι οποιαδήποτε κακόβουλη μεταγενέστερη αλλαγή θα απαιτούσε τη συνεργασία της πλειοψηφίας των κόμβων - γεγονός το οποίο σε φυσικούς πόρους είναι εξαιρετικά δύσκολο - παρέχει ασφάλεια ικανή να παρακάμψει την όποια κεντρική έμπιστη αρχή επικυρώνει που επικυρώνει τις συναλλαγές. Δημιουργηθηκε λοιπόν η τάση της αποκεντροποίησης της διαχείρησης των εφαρμογών και υπηρεσιών, με στόχο το μερικό και υπό περιπτώσεις ολικό εκδημοκρατισμό του διαδικτυακού τοπίου.

Η τεχνολογία του Blockchain έχει τη δυναμική - και έχει σαφώς επηρεάσει - διάφορους

τομείς με κύριο τον χρηματοπιστωτικό, που χάρη στο Bitcoin και των ακολούθων του έχει θορυβηθει. Πλέον υπάρχουν αποκεντρωμένες εφαρμογές που κάνουν δυνατή και εύκολη την εκτέλεση συναλλαγών - και όχι μόνο - ανάμεσα σε άτομα. Παράλληλα η πληροφορία των συναλλαγών είναι ελέγξιμη και παρατηρήσιμη από όλους τους συμμετέχοντες του δικτύου ενώ η διαδικασία της επικύρωσης των γεγονότων συμβαίνει από το ίδιο το δίκτυο. Πλέον το ¨άτομο' από απλός εντολοδόχος, γίνεται πλεόν και ¨ελεγκτης' της εντολής αυτής, παρέχοντας κίνητρα για της διασφάλιση της τιμιότητας.

Οι εκατοντάδες υλοποιήσεις κρυτπονομισμάτων που παρουσιάζονται τα τελευταία χρόνια ακολουθούν το μοντέλο διασύνδεσης P2P, δημιουργώντας κατα βάση αδόμητα ή ημιδομημένα δίκτυα υπολογιστών στο διαδικτυακό ιστό. Κάθε υλοποίηση προτείνει εισάγει τα δικά της χαρακτηριστικά προσπαθώντας να ξεπεράσει τα προβλήματα παλαιότερων η έστω να τα περιορίσει, ωστόσο δεν έχει βρεθεί ακόμα η χρυσή τομή.

1.1 Αρχική ιδέα και κοινωνική αξία της εφαρμογής

Η ιδέα μιας αποκεντρωμένης εφαρμογής με τη τεχνολογία του Blockchain που διαχειρίζεται ηλεκτρονικές οικονομικές συναλλαγές έχει δοκιμαστεί τα τελευταία χρόνια. Η χρήση από χιλιάδες άτομα έχει αναδείξει τις ατέλειες των υλοποιήσεων. Οι διαστάσεις των ελαττωμάτων τους είναι μετρήσιμες και χρήζουν αντιμετώπισης αφού όλο και περισσότεροι τείνουν χρησιμοποιούν τέτοιου είδους εφαρμογές.

Αρχικά, μεγάλη μερίδα των πρωτοκόλλων συμφωνίας(Consensus) των κρυπτονομισμάτων έχει ώς βάση την επίλυση ¨πολύ δύσκολων προβλημάτων¨, μία διαδικασία που απαιτεί υπολογιστικούς πόρους και ενέργεια. Αρκεί να αναλογιστεί κανεις το μέγεθος της ενέργειας που καταναλώνει το δίκτυο του Bitcoin που ισούται με το 0.2 % της παγκόσμιας κατανάλωσης ηλεκτρικής ενέργειας. Αυτή η ενέργεια καταναλίσκεται για την παραγωγή νέων μπλοκ της αλυσίδας, μέσω υπολογιστικών πράξεων. Σε μία άλλη μετάφραση, το Bitcoin παράγει ετησίως 22.9 κυβικούς τόνος διοξειδίου του άνθρακα ετησίως.

Παράλληλα, λόγω της φύσης τους, αυτές οι αποκεντρωμένες εφαρμογές σχηματίζουν P2P δίκτυα μεταξύ των κόμβων , τα οποία πλυμμηρίζουν το διαδίκτυο με μηνύματα. Ο μεγάλος αριθμός μηνυμάτων κάνει δύσκολο το έργο τον παρόχων του διαδικτύου αφού κάθε κόμβος δεσμεύει υψηλό bandwidth κατά την ύπαρξη του στο δίκτυο του κρυπτονομίσματος. Επιπρόσθετα, οι υψηλές προοπτικές κέρδους στα κρυπτονομίσματα, έχει προσελκύσει επενδυτές οι οποίοι δημιουργούν μεγάλες φάρμες από εκατοντάδες η ακόμα και χιλιάδες κόμβους. Σε πολλά κρυπτονομίσματα η πιθανότητα κάποιος κόμβος να επικυρώσει το νέο μπλοκ και κατά συνέπεια να λάβει την αμοιβή αυτής της επικύρωση είναι ανάλογη της υπολογιστικής ισχύς που αφιερώνει. Στη βάση αυτού, χάνεται η έννοια του εκδημοκρατισμού, αφού τελικά η εξουσία - δηλαδή η υπολιστική ισχύς - συγκεντρώνεται στους ¨λιγους' οι οποίοι τελικά καρπώνονται και το κέρδος.

Επίσης, ο χώρος που καταλαμβάνει το τοπικό αντίγραφο της αλυσίδας συνεχώς αυξάνεται, όσο αυξάνεται το μήκος της με την πάροδο του χρόνου. Από τη μία αυτό εξασφαλίζει εν μέρει την κοινή αλήθεια του δικτύου ωστόσο σε κάποιο χρονικό σημείο θα αρχίσει να αποτελεί

περιοριστικό παράγοντα για τη συμμετοχή ενός κόμβου στο δίκτυο.

Τέλος, σύμφωνα με τα παραπάνω και δεδομένου ότι κάθε νέο μπλοκ δημιουργείται ανά τακτά χρονικά διαστήματα, η διάδοση του στο δίκτυο απαιτεί χρόνο συναρτήσει του μεγέθους της πληροφορίας, του δικτύου καθώς και της μέσης απόκρισης του κόμβου. Αυτό συνεπάγεται ότι ένα μπλοκ με συναλλαγές πυ προστίθεται χρειάζεται σημαντικό χρόνο για διαδοθεί σε όλους τους κόμβους.

Παραπάνω αναφερονται ορισμένα απο τα προβλήματα που έχουν παρουσιαστεί και τα οποία δεν αποτελούν, παρά ένα μικρό κομμάτι της λίστας των προβλημάτων.

1.2 Αντικείμενο της διπλωματικής

Στη παρούσα διπλωματική εργασία, όπως έχει γίνει εμφανές, πρωταγωνιστικό ρόλο θα καταλάβει η καινοτόμα τεχνολογία του blockchain.

Σκοπός της διπλωματικής είναι πρόταση μίας ολοκληρωμένης λύσης που αποσκοπεί στην επίλυση των προβλημάτων που περιγράφονται. Στόχος της εφαρμογής είναι να επιτρέπει στο χρήστη της την πραγματοποίηση ηλεκτρονικών χρηματικών συναλλαγών με άλλους χρήστες, όπως κάθε άλλο κρυπτονόμισμα. Η καινοτομία έγκειται στο γεγονός ότι ο κάθε κόμβος δεν θα δεσμεύει χώρο από το τοπικό μέσο αποθήκευσης του για την διατήρηση του πλήρους αντιγράφου της αλυσίδας αλλά μονάχα ένα μέρος του. Ταυτόχρονα λόγω του ότι προτείνεται και εναλλακτικός τρόπος διασύνδεσης των κόμβων μειώνονται τα μηνύματα που αποστέλλονται από κάθε κόμβο στο δίκτυο του. Τέλος μέσω του πρωτοκόλλου συμφωνίας που προτείνεται, προωθείται η δημοκρατία, υπό την έννοια της πιθανότητας επικύρωσης νέου μπλοκ.

Για τα ανωτέρω σκόπο θα γίνει ανάπτυξη της υλοποίησης που θα ακολουθεί τις παραπάνω αρχές.

1.3 Οργάνωση του τόμου

Το κείμενο της διπλωματικής αποτελείται απο 8 κεφάλαια.

Το παρόν κεφάλαιο 1 αποτελεί την εισαγωγή.

Στο κεφάλαιο 2 γίνεται μία ιστορική αναφορά στην τεχνολογία του Blockchain με έμφαση στο κρυπτονόμισμα Bitcoin, το λόγο δημιουργίας του, τα χαρακτηριστικά του καθώς και τα ελλαττώματα που έχει.

Στο κεφάλαιο 3 παρουσιάζεται το θεωρητικό υπόβαθρο της της εργασίας. Πιο συγκεκριμένα, γίνεται σύντομη παρουσίαση του Web3 και της ιστορίας και στη συνέχεια παρουσιάζονται τα δίκτυα ομότιμων κόμβων (P2P) που αποτελούν σημαντικό κομμάτι του Web3 καθώς και του Blockchain. Εν συνεχεία γίνεται εκτενής παρουσίαση του πρωτοκόλλου Chord, το οποίο αποτελεί συστατικό στοιχείο της εργασίας. Τέλος γίνετα μία εποπτική αλλά ακριβής ανάλυση σχετικά με τα βασικά στοιχεία ενός κρυπτονομίσματος.

Στο κεφάλαιο 4 γίνεται αναφορά στο γενικότερο πλαίσιο ένταξης της ερφαρμογής και αναλύονται οι λειτουργικές και μη λειτουργικές απαιτήσεις του συστήματος.

Στο κεφάλαιο 5 γίνεται λεπτομερής παρουσίαση των εργαλειών και τεχνολογιών που χρησιμοποιήθηκαν.

Στο κεφάλαιο 6 περιέχει τη διαδικασία σχεδίασης και τα πρωτόκολλα που θα εφαρμοστούν στην εφαρμογή.

Στο κεφάλαιο 7 παρουσιάζονται τόσο μετρήσεις απο την υλοποίηση, όσο και θεωρητικά αποτελέσματα βασισμένα σε μετρήσεις που αποδεικνύουν την αξία της πρότασης.

Το κεφάλαιο 8 αποτελεί τον επίλογο του κειμένου. Σε αυτό συνοψίζονται παρατηρήσεις και συμπεράσματα του συγγραφέα όσον αφορά την υλοποίηση αλλά και γενικότερα τις αποκεντρωμένες εφαρμογές. Επίσης σημειώνονται μελλοντικές επεκτάσεις του συστήματος. Το κεφάλαιο κλείνει με προσωπικό σχόλιο του συγγραφέα.

Κεφάλαιο 2

Κρυπτονομίσματα και Blockchain

2.1 Εισαγωγή

Στο κεφάλαιο αυτό διατυπώνεται το θεωρητικό υπόβαθρο σχετικά με το Blockchain. Αναλυτικότερα, αναφέρονται τα χαρακτηριστικά της τεχνολογίας του Blockchain καθώς και αναφέρονται γενικές πληροφορίες για την δομή και την λειτουργία της πιο δημοφιλούς υλοποίησης - του Bitcoin. Τέλος παρουσιάζονται τα προβλήματα που έχουν προκύψει στον τομέα των κρυπτονομισμάτων - και όχι μόνο - που κάνουν χρήση της συγκεκριμένης τεχνολογίας. Αν και στην εργασία χρησιμοποιείται η υλοποίηση του Semux κρυπτονομίσματος, το Bitcoin χρησιμοποιείται ως μέσο κατανοήσης των βασικών αρχών που σε μεγαλύτερο η μικρότερο βαθμό τηρούνται από όλες τις υλοποιήσεις.

2.2 Ορισμός

Το Blockchain είναι ένας αυξανόμενος κατάλογος αρχείων ή εναλλακτικά ένα "ημερολόγιο" ηλεκτρονικών ενεργειών, που ονομάζονται μπλοκ, και συνδέονται χρησιμοποιώντας κρυπτογραφία. Κάθε μπλοκ περιέχει μια κρυπτογραφημένη "υπογραφή" του προηγούμενου μπλοκ, ένα χρονικό σήμα(timestamp) και δεδομένα συναλλαγής ή κατ επέκταση ενεργειών (που γενικά αντιπροσωπεύεται ως δέντρο Merkle).

Ένα από τα βασικά χαρακτηριστικά του blockchain είναι ότι αποτρέπει την αλλαγή/τροποποίηση των ήδη αποθηκευμένων πληροφοριών. Ουσιαστικά αποτελεί ένα «ανοικτό, κατανεμημένο βιβλίο "κινήσεων/ενεργειών" που μπορεί να καταγράφει τις συναλλαγές μεταξύ δύο η περισσότερων μερών αποτελεσματικά και κατά τρόπο επαληθεύσιμο και μόνιμο». Για να χρησιμοποιηθεί ως κατανεμημένο ημερολόγιο, ένα blockchain διαχειρίζεται συνήθως ένα δίκτυο peer-to-peer συλλογικά ακολουθώντας ένα πρωτόκολλο για επικοινωνία μεταξύ κόμβων και καθολικό πρωτόκολλο για την επικύρωση νέων μπλοκ. Μόλις καταγραφούν, τα δεδομένα σε οποιοδήποτε μπλοκ δεν μπορούν να τροποποιηθούν αναδρομικά χωρίς αλλοίωση όλων των επόμενων μπλοκ, πράγμα που απαιτεί συναινετική πλειοψηφία του δικτύου. Παρόλο που τα αρχεία των μπλοκ των αλυσίδων δεν είναι αναλλοίωτα, τα blockchain μπορεί να θεωρηθούν ασφαλή από το σχεδιασμό τους και να αποτελέσουν παράδειγμα ενός κατανεμημένου συ-

στήματος υπολογιστών με υψηλή ανοχή σε λάθη (Byzantine fault tolerance). Επομένως, η αποκεντρωμένη συναίνεση (Decentralized consensus) απαιτείται και είναι απαραίτητη στην υλοποίηση ενός blockchain.

2.3 Ιστορία

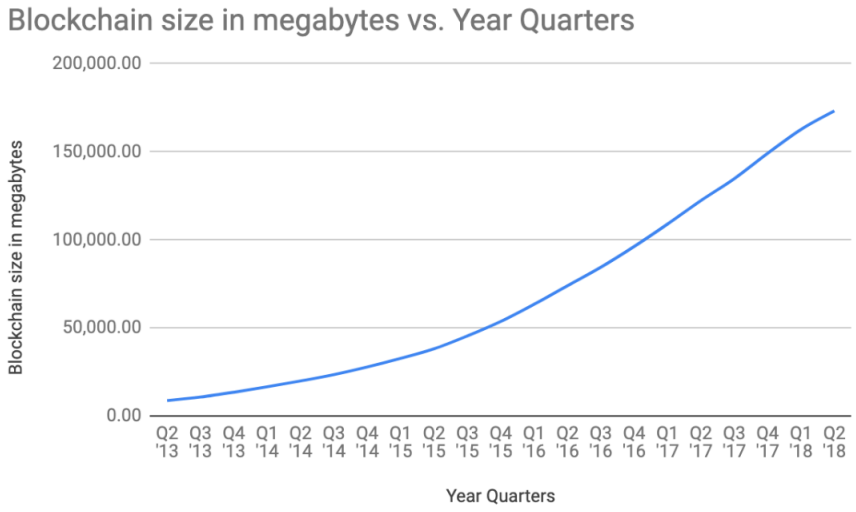
Η υλοποίηση του Blockchain για να χρησιμεύσει ως δημόσιο βιβλίο συναλλαγών του bitcoin cryptocurrency [45][4], επινοήθηκε από ένα άτομο (ή ομάδα ανθρώπων) χρησιμοποιώντας το όνομα Satoshi Nakamoto [21] το 2008. Η ταυτότητα του Satoshi Nakamoto είναι άγνωστη. Η υλοποίηση του blockchain για το bitcoin το κατέστησε το πρώτο ψηφιακό νόμισμα για την επίλυση του προβλήματος των διπλών δαπανών [32] (double-spending problem) χωρίς την ανάγκη μιας αξιόπιστης αρχής ή κεντρικού διακομιστή. Ο σχεδιασμός του βιτςοιν έχει εμπνεύσει άλλες εφαρμογές, και υλοποιήσεις blockchain που είναι προσβάσιμες από το κοινό και χρησιμοποιούνται για την δημιουργία κρυπτονομισμάτων (cryptocurrencies). Το Blockchain θεωρείται ένας τύπος πληρωμής. Ιδιωτικά blockchain έχουν προταθεί για εταιρική χρήση.

Το 1991 ξεκίνησε η εργασία για μια κρυπτογραφημένη, ασφαλή αλυσίδα απο μπλοκ [39] από τους Stuart Haber και W. Scott Stornetta. Στόχος τους ήταν η δημιουργία συστήματος εγγράφων των οποίων η χρονοσήμανση (timestamp) να μην μπορεί να αλλοιωθεί. Το 1992 οι Bayer, Haber και Stornetta συμπεριέλαβαν τα Merkle Trees [29] στο σχεδιασμό, το οποίο οδήγησε στην αύξηση της απόδοσης, αφού επέτρεπε την ενσωμάτωση πολλαπλών πιστοποιητικών των εγγράφων σε ένα μεμονωμένο μπλοκ.

Η πρώτη υλοποίηση blockchain δημιουργήθηκε το 2008 απο τον Satoshi Nakamoto. Το προαναφερθέν όνομα λέγεται ότι αποτελεί ψευδώνυμο για ένα άτομο η μία ομάδα ατόμων, ωστόσο η πραγματική ταυτότητα παραμένει μέχρι σήμερα κρυφή. Ο Satoshi, λοιπόν, βελτίωσε τον μέχρι τότε σχεδιασμό και πλέον δεν ήταν απαραίτητο για κάθε μπλοκ που προστίθεται στην αλυσίδα να "υπογράφεται" από μία αξιόπιστη αρχή. Αυτός ο σχεδιασμός βρήκε πρακτική εφαρμογή τον επόμενο χρόνο με τη δημοσίευση του Bitcoin, το οποίο είναι ένα δημόσιο ηλεκτρονικό "βιβλίο" ηλεκτρονικών συναλλαγών για όλες τις κινήσεις του δικτύου αυτού.

Το Bitcoin πήρε μεγάλη δημοσιότητα και αυτό είναι εμφανές από τον όγκο του blockchain [24], το οποίο μέχρι τον Αύγουστο του 2014 είχε φτάσει τα 20 Gigabytes, ενώ τον Ιανουάριο του 2015 ήταν κατα προσέγγιση 30 Gigabytes. Μέχρι το καλοκαίρι του 2018 και λόγω δημοσιότητας που είχε λάβει το συγκεκριμένο κρυπτονόμισμα ο όγκος των συναλλαγών που ήταν αποθηκευμένες στο blockchain είχε φτάσει τα 180 Gigabytes κατά προσέγγιση. Στο σχήμα 2.1 φαίνεται ο όγκος που καταλαμβάνει η αλυσίδα σε megabytes ανά τετράμηνο χρονιάς.

Το γράφημα αποτυπώνει όχι μόνο την αύξηση της χρήσης του συγκεκριμένου κρυπτονομίσματος, αλλά εκδηλώνει τη γενικότερη τάση για την χρήσης της τεχνολογίας αυτής για εμπορικούς σκοπούς. Το Bitcoin αποτέλεσε το εναρκτήριο λάκτισμα για τη διαδοση της ιδέας και δημιουργία πολλών κρυπτονομισμάτων και εφαρμογών που έχουν βάση το Bitcoin.

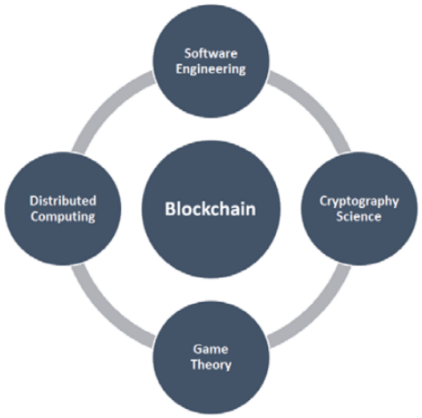


Σχήμα 2.1: Μέγεθος Bitcoin Blockchain

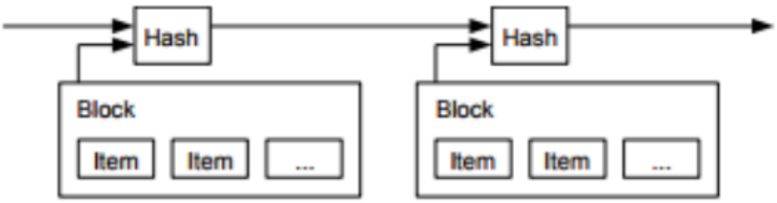
2.4 Αρχιτεκτονική και Χαρακτηριστικά του Blockchain

2.4.1 Blockchain και Bitcoin - Η πρώτη υλοποίηση.

Το Blockchain είναι μια τεχνολογία στην οποία έχουν συμβάλει τομείς όπως το Software Engineering[25], Distributed systems[10],Cryptography Science[8], Game Theory[14] κ.α. Το blockchain υπάρχει στη συμβολή αυτών των πεδίων που παρέχουν σταθερή και με δυνατότητες υψηλής κλιμακωσιμότητας (scalability) υποδομή λογισμικού, με στόχο την παροχή ενός αποκεντρωμένου (decentralized) δικτύου στο παγκόσμιο ιστό με βάση την εξασφάλιση των ηλεκτρονικών περιουσιακων στοιχείων, στο οποίο συμμετέχοντες με κίνητρο οικονομικους (και όχι μόνο) σκοπούς μπορούν να εκτελούν ενέργειες, των οποίων η εγκυρότητα να επικυρώνεται και αναγνωρίζεται από όλους.



Σχήμα 2.2: Ο μύτος πολλών τεχνολογιών το Blockchain



Σχήμα 2.3: Η αλυσίδα των μπλοκς

Οταν αναφερόμαστε στο blockchain, αυτόματα η σκέψη οδηγείται στο κρυπτονόμισμα Bitcoin, το οποίο αποτελεί την πρώτη πρακτική εφαρμογή της τεχνολογίας αυτής. Ουσιαστικά στο white paper του Bitcoin [5]περιγράφεται η δομή που χρησιμοποιήθηκε για να λύσει το πρόβλημα του double-spending, το οποίο αποτελούσε μη αντιμετωπίσιμο ζήτημα στις μέχρι τότε απόπειρες δημιουργίας κρυπτονομισμάτων. Περιγράφεται ο τρόπος δημιουργίας μια αλυσίδας απο μπλοκ, όπου κάθε μπλοκ που παράγεται περιλαμβάνει και τη χρονοσήμανση δημιουργίας (timestamp) και τα δεδομένα του ίδιου καθώς και του προηγούμενο block σε μια κατακερματισμένη απεικόνιση (hash). Αποτελεί μια αλληλουχία καταγραφών στην οποία κάθε καινούργιο κομμάτι της αλυσίδας συνδέεται με το προηγούμενο χρονικά κομμάτι. Συνεπώς τόσο η ακεραιότητα, όσο και η χρονική σειρά των εκάστοτε καταγραφών δεν μπορεί να επηρεαστεί μετά την ένταξη τους στην αλυσίδα καθώς θα οδηγήσει σε μία αλληλουχία που δεν βγάζει νόημα.

Αυτη η λογική από μόνη της δεν αρκεί για να αντιμετωπίσει το πρόβλημα του double-spending κατά το οποίο ο δράστης επιχειρεί να δημιουργήσει race conditions κατα τις οποίες ξοδεύει τους ίδιους εικονικούς πόρους πολλαπλές φορές πριν αυτές οι κινήσεις να επιβεβαιωθούν (να γίνουν validate). Για το λόγο αυτό το Bitcoin εισήγαγε έναν κανονισμό συμφωνίας μεταξύ των συμμετεχόντων του δικτύου[28] (Consensus), το μοντέλο του Proof-of-Work. Σε αυτο το Consensus μηχανισμό οι συμμετέχοντες στο δίκτυο μεμονωμένα προσπαθούν επανειλημμένα, μέχρι να επιτύχουν, να κατακερματιζουν το μπλοκ με έναν τυχαίο αριθμό (nonce),δηλαδή να λύσουν ένα "πολύ δύσκολο" πρόβλημα, μέχρι να καταλήξουν σε ένα αποτέλεσμα το οποίο να είναι μικρότερο από μία συγκεκριμένη τιμή. Οποιος συμμετέχων βρεί πρώτος αυτή την τιμή, την διαδίδει στο δίκτυο, το οποίο ελέγχει την εγκυρότητα της και αν συμφωνεί προστίθεται σαν νέος κρίκος της αλυσίδας.

Το κρυπτονόμισμα Bitcoin είχε σαν απαίτηση μία κατασκευή με την οποία θα ήταν εύκολο να καταγράφεται η σειρά των συναλλαγών , να επαληθεύονται οι ίδιες οι συναλλαγές και τελικά να εξασφαλίζεται η αποθήκευση τους. Το blockchain αποτέλεσε τη προφανή λύση για τις παραπάνω απαιτήσεις με τη παραδοχή ότι κάθε block περιέχει έναν σύνολο συναλλαγών , το οποίο μέσω κρυπτογραφίας συνδέεται με ένα parent μπλοκ. Με δεδομένο ότι το blockchain είναι ένα ελεύθερο αρχείο το οποίο είναι κατανεμημένο, δεν υπάρχει κεντρικό αντίγραφο στο Bitcoin. Βασίζεται στη λογική της αποκεντρωμένης διανομής του αρχείου αυτού σε συνδυασμό με το μηχανισμό συμφωνίας (Consensus) Proof-of-Work με στόχο τον συντονισμό για το ποια μπλοκ προστιθενται στην αλυσίδα και τη διανομή αυτών στους συμμετέχοντες στο δίκτυο.

Δηλαδή το Blockchain του Bitcoin αποτελεί μία κατανεμημένη βάση δεδομένων στην οποία αποθηκεύονται συναλλαγές, χρησιμοποιώντας τη λογική μιας σειράς κινήσεων της μορφής εντολοδόχου- εντολέα , κατά την οποία δεν αποθηκεύονται υπόλοιπα, αλλά μόνο οι ενέργειες. Με τη εξερεύνηση της αλυσίδας είναι γρήγορα υπολογίσιμο το διαθέσιμο υπόλοιπο ενός χρήστη στο Bitcoin. Η προσπάθεια για αλλοίωση της σειράς των συναλλαγών, και του περιεχομένου τους αποτυγχάνει αφού έχει ως συνέπεια να μεταβάλλει την τιμή κατακερματισμου του εκάστοτε block - μια υπολογιστικά πολύ δύσκολη διαδικασία. Παράλληλα η κατανεμημένη φύση της εφαρμογής εξασφαλίζει ότι όλοι οι συμμετέχοντες στο δίκτυο έχουν έγκυρα αντίγραφα της αλυσίδας. Για να καταφέρει να ελέγξει κάποιος του blockchain του bitcoin οφείλει να διαθέτει πάνω από τη μισή υπολογιστική ισχύς του συνολικού δικτύου, αριθμός κάτι που αν μεταφραστεί σε νούμερα, φαντάζει απαγορευτικό σε ένα τόσο μεγάλης κλίμακας δίκτυο.

Η πρώτη συναλλαγή στο δίκτυο του Bitcoin εγινε τον Ιανουάριο του 2009 από το Satoshi Nakamoto, όταν και δημιουργήθηκε το genesis block, το μόνο block της αλυσίδας το οποίο δεν έχει parent block για να συνδεθεί , και κατέθεσε στο λογαριασμό του 50 bitcoins.

Συνοπτικά, ένα blockchain αποτελεί ένα αμετάβλητο ιστορικό αρχείο ενεργειών (στη περίπτωση του Bitcoin ηλεκτρονικών συναλλαγών) από την ημέρα της έναρξης λειτουργίας του, σε ένα decentralized και διαφανές περιβάλλον αποθήκευσης, δηλαδή το δίκτυο του. Η ασφάλεια εξασφαλίζεται από δύο παράγοντες:

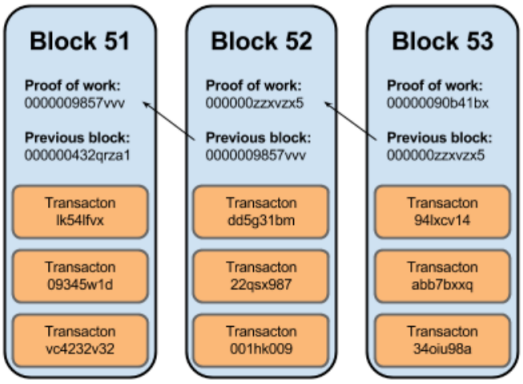
- Απο την κρυπτογραφικό δεσμό μεταξύ δύο συνεχόμενων χρονικά block, που όσο μεγαλύτερο είναι το μήκος της αλυσίδας, τόσο πιο αδύνατη κάνει τη αλλοίωση των περιεχομένων της.
- Απο την διαφάνεια στη διανομή των δεδομένων στο δίκτυο του blockchain, καθώς οι ειλικρινείς κόμβοι που συμμετέχουν αναμένεται πάντα να είναι περισσότεροι από τους κακόβουλους κόμβους που προσπαθούν να επιτεθούν στο δίκτυο.

2.4.2 Βασικά έννοιες που προσδιορίζουν ένα κρυπτονόμισμα-Blockchain

Όπως έχει προαναφερθεί, το blockchain είναι ένα δεςεντραλιζεδ, κατανεμημένο, συνήθως δημόσιο ηλεκτρονικό "ημερολόγιο" συναλλαγών. Τα δομικά χαρακτηριστικά ενός blockchain είναι τα παρακάτω:

Block

Δομική μονάδα του blockchain είναι το block. Το πρώτο block ενός blockchain αποκαλείται genesis block.
Ενα block περιέχει ομάδες από ελεγμένες συναλλαγές οι οποίες είναι κατακερματισμένες και κωδικοποιημένες σε Merkle Trees. Με δεδομένο ότι κάθε block συνδέεται με το χρονικά προηγούμενο block, σε κάθε block υπάρχει το κρυπτογραφημένο hash του προηγούμενου block. Αυτή η σύνδεση σχηματίζει την αδιάσπαστη αλληλουχία που ονομάζουμε αλυσίδα.



Σχήμα 2.4: Αναπαράσταση μιας αλυσίδας κρυπτονομίσματος

Αυτή η διαδικασία χρησιμοποιείται για να εγγυηθεί την ακεραιότητα της αλυσίδας μέχρι και το genesis block.
Στη συνέχεια θα αναλυθεί εκτενώς ο τρόπος δημιουργίας ενός block καθώς και η διαδικασία συμφωνίας (Consensus) μεταξύ των κόμβων του δικτύου ενός blockchain, ωστόσο αξίζει να αναφερθεί σε αυτό το σημείο ότι μπορεί να συμβεί να παραχθούν διαφορετικά block ταυτόχρονα, έχοντας ως αποτέλεσμα ένα προσωρινό fork[13]. Στα ελληνικά θα μπορούσε να βρει ερμηνεία στη λέξη παρακλάδι. Αυτό σημαίνει ότι για μικρό διάστημα ένα blockchain μπορεί να έχει δύο διαφορετικές εκδοχές για ένα μικρό κομμάτι της αλυσίδας. Φυσικά οι υλοποιήσεις των blockchain έχουν βρει τρόπους με συγκεκριμένους αλγόριθμους ώστε να αποφασίζουν ποια εκδοχή της αλυσίδας θα κρατήσουν και ποια θα απορρίψουν. Με το που αποφασιστεί, όλοι οι νοδες του δικτύου ενημερώνονται και συγχρονίζουν τις αλυσίδες τους.
Για παράδειγμα, στο Bitcoin που χρησιμοποιεί ϛονσενσυς προοφ-οφ-ωορκ, όταν συμβαίνει φορκ, τότε κρατιέται το παρακλάδι το οποίο έχει αθροιστικά μεγαλύτερη "ποσότητα" proof-of-work · μία εύκολα υπολογίσιμη μετρική.

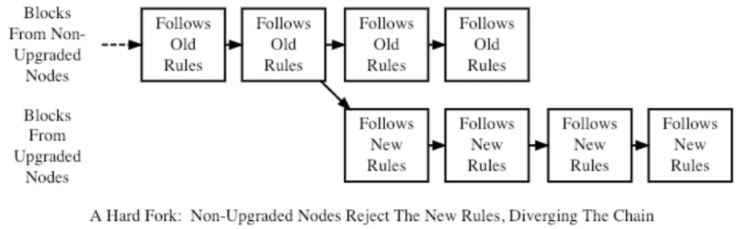
Blocktime

Το blocktime αναφέρεται στο χρόνο που κατά μέσο όρο χρειάζεται το δίκτυο, για να δημιουργήσει ένα νέο block. Στα κρυπτονομίσματα ο μέσος χρόνος δημιουργίας ενός block συνδέεται άμεσα με το χρόνο εφαρμογής συναλλαγών, αφου τα blocks περιέχουν επικυρωμένες συναλλαγές. Συνεπώς όσο μικρότερο blocktime, τόσο πιο "γρήγορες" συναλλαγές. Ο μέσος χρόνος δημιουργίας ενός block στο δίκτυο του Bitcoin είναι περίπου δέκα λεπτά , ενώ στο Ethereum είναι 15 δευτερόλεπτα.

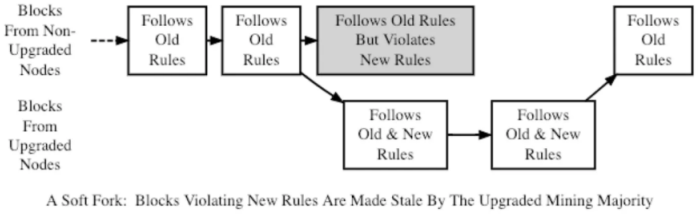
Fork

Σε συνέχεια του είδους fork ενός blockchain, όπως αναφέρθηκε προηγουμένως, έχουν να προστεθούν δύο ακόμα είδη φορκ:

- **Hard Fork**:Πρόκειται για την περίπτωση που αποφασίζεται να αλλάξουν οι κανόνες για



Σχήμα 2.5: Αναπαράσταση ενός Soft Fork



Σχήμα 2.6: Αναπαράσταση ενός Hard Fork

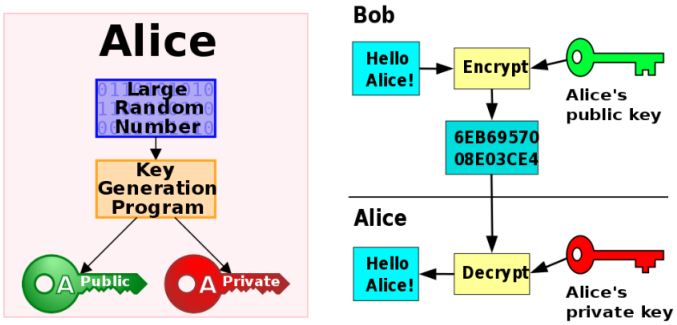
την επικύρωση των block απο τους κόμβους του δικτύου. Σε αυτή τη περίπτωση όλοι οι κόμβοι οφείλουν να αναβαθμίσουν το λογισμικό τους καθώς οι παλιοί κανόνες δεν θα είναι πλέον σε ισχύ. Σε περίπτωση που μία ομάδα κόμβων συνεχίσει να δημιουργεί block χωρίς να αναβαθμίσει το λογισμικό της , ενώ το υπόλοιπο κομμάτι του δικτύου δημιουργεί block σύμφωνα με τους καινούργιους κανόνες τότε μπορούν να προκληθούν παρακλάδια.

- **Soft Fork**:Πρόκειται για την περίπτωση που αποφασίζεται να αλλάξουν οι κανόνες για την επικύρωση των block από τους κόμβους του δικτύου. Σε αυτή τη περίπτωση τα block που παράγονται από τους κόμβους που έχουν αναβαθμίσει το λογισμικό τους θεωρούνται έγκυρα για τους κόμβους που δεν έχουν αναβαθμιστεί ενώ το αντίθετο δεν ισχύει. Και σε αυτή τη περίπτωση υπάρχει πιθανότητα χωρισμού του blockchain.

Cryptography

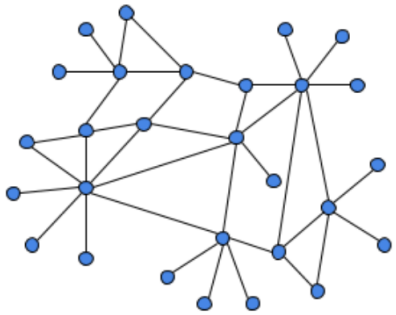
Όντας διαθέσιμο στο διαδίκτυο, μια υλοποίηση blockchain επιστρατεύει την κρυπτογραφία σαν μέθοδο ασφάλειας. Σε αυτές τις μεθόδους περιλαμβάνεται η μέθοδος της ασσύμετρης κρυπτογραφίας[44] .

Συνοπτικά, κάθε κόμβος έχει δύο κλειδιά: ένα δημόσιο, το οποίο αντιπροσωπεύει ένα λογαριασμό στο δίκτυο του blockchain και ένα ιδιωτικό το οποίο είναι γνωστό μόνο από τον κάτοχο του λογαριασμού. Τυπικά το ιδιωτικό κλειδί δίνει πρόσβαση σε όλα τα ηλεκτρονικά περιουσιακά στοιχεία του λογαριασμού αφού ουσιαστικά αποτελεί έναν κωδικό.



Σχήμα 2.7: Η κρυπτογραφία και το Blockchain

Decentralization



Σχήμα 2.8: Αποκεντρωμένος ιστός

Στη βάση της λογικής ότι το blockchain αποθηκεύεται σε όλους τους κόμβους του peer-to-peer δικτύου του , συνεπάγεται ότι δεν υπάρχει κεντρική αρχή η οποία διατηρεί ή διαμοιράζει τα δεδομένα, ή ελέγχει την ακεραιότητα τους. Δηλαδή ένα δίκτυο blockchain δεν έχει single-point of failure αφού δεν διατηρείται ένα κεντρικό αντίγραφο. Αντίθετα ο κάθε κόμβος που συμμετέχει στο δίκτυο, διατηρεί αντίγραφο του blockchain καθώς και διαδίδει κάθε νέα προσθήκη σε αυτό μέσω μηνυμάτων. Παράλληλα είναι σε θέση να ελέγχει και τα δεδομένα τα οποία λαμβάνει. Η λογική που ακολουθείται είναι ότι οι συναλλαγές προς εκπλήρωση γίνονται βροαδςαστ στο δίκτυο του blockchain, και οι κόμβοι οι οποίοι δημιουργούν block (mining/forging) τις προσθέτουν, και στη συνέχεια κάνουν broadcast το νέο block. Στο συνέχεια θα αναλυθεί εκτενώς η λειτουργία ενός κρυπτονομίσματος.

Openness and Transparency

Σχήμα 2.9: Openess and transparency

Όταν αναφερόμαστε σε ένα blockchain , τότε αυτόματα στο μυαλό μας έρχεται η λέξη openness. Openness γιατί όμως·

Αρχικά ο κώδικας είναι open-source στη μεγάλη πλειοψηφία των υλοποιήσεων blockchain (Εξαιρούνται οι υλοποιήσεις που προορίζονται για ιδιωτική/εταιρική χρήσεις) Συγχρόνως μπορεί ο καθένας να συμμετέχει στο δίκτυο ενός ανοιχτού blockchain, αρκεί να έχει πρόσβαση σε υπολογιστή και στο διαδίκτυο. Και το συμμετέχει σημαίνει ότι ο καθένας μπορεί να εξερευνήσει την αλυσίδα με τα περιεχόμενα της, να εξακριβώσει την ακεραιότητα της καθώς και να τεστάρει η να κάνει fork στο πηγαίο κώδικα.

Το διαμοιραζόμενο, δημόσιο blockchain συνεπάγεται τη δυνατότητα του καθένα να παρακολουθεί τι συμβαίνει στο δίκτυο, δηλαδή πόσες συναλλαγές γίνονται, ποιος τις κάνει κλπ. Η διαφάνεια που προσφέρει το blockchain καθώς και η ειλικρίνεια “openness” προς τους χρήστες, είναι από τα κύρια χαρακτηριστικά που ελκύουν τους χρήστες να υιοθετήσουν την συγκεκριμένη τεχνολογία.

(Pseudo) Anonymity and Privacy

Σχήμα 2.10: Privacy

Η ανωνυμία είναι ιδιωτικότητα ή η σιγουριά του ατόμου ότι τα προσωπικά του δεδομένα θα είναι είναι προσβάσιμα μόνο από το ίδιο.

Φυσικά στη γενική περίπτωση δεν καταρρίπτεται ο μύθος της ανωνυμίας στη περίπτωση του blockchain, ωστόσο αποτελεί μία πραγματικότητα ότι είναι πιο δύσκολα εντοπίσημη η ταυτότητα ενός μέσα στο blockchain καθώς και των ενεργειών.

Σε ένα ανοιχτό δίκτυο blockchain , για τη συμμετοχή ενός κόμβου δεν απαιτείται ούτε εγγραφή μέσω e-mail, είτε οποιοδήποτε προσωπικό στοιχείο (διεύθυνση , τηλέφωνο κλπ). Ένας λογαριασμός και κατ' επέκταση ο κάτοχος του που συνδέεται μέσω ενός κόμβου, αντιπροσωπεύεται απο μια μοναδική διεύθυνση (unique identifiers).

Το παραπάνω χαρακτηριστικό δεν αποκλείει την δυνατότητα ενός κακόβουλου δράστη να διασταυρώσει την πραγματική ταυτότητα μιας τέτοιας μοναδικής διεύθυνσης μέσω για παράδειγμα παρακολούθησης της ip η/και των συναλλαγών του λογαριασμού[17]. Υλοποιήσεις blockchain, προσπαθώντας να αντιμετωπίσουν αυτό το "πρόβλημα" , χτίζουν vpn-over-blockchain network ώστε οποιαδήποτε "προσωπική" πληροφορία ενός λογαριασμού να γίνεται mask.

Ενα απο τα αρνητικά της ανωνυμίας είναι ότι προσελκύθηκαν ομάδες που ασχολούνται με παράνομες δραστηριότητες με σκοπό να κρύψουν το πρόσωπο τους καθώς και το εισόδημα τους.

2.4.3 Είδη Blockchain

Αν θέλει κανείς να κατηγοριοποιήσει κάπως τα είδη blockchain μπορεί ως προς το openess τους, ως προς τη μέθοδο consensus που χρησιμοποιούν, ή ως προς τον τρόπο χρήσης τους:

Ως προς το Openness

Δημόσιο blockchain (public): Οι δημοφιλέστερες υλοποιήσεις blockchain-κρυπτονομισμάτων, όπου τόσο ο πηγαίος κώδικας όσο και τα δεδομένα είναι προσβάσιμα από τον καθένα. Υπάρχει διαφάνεια και ένα απο τα χαρακτηριστικά τους είναι ότι με στόχο τη παρότρυνση συμμετοχής σε αυτά τα δίκτυα απονέμονται ανταμοιβές στους χρήστες. (π.χ. Καταφέρνουν να κάνουν mine ένα block.) Είναι επίσης σχεδιασμένα για πλήρη αποκέντρωση (fully decentralized) αφού δεν είναι θεμιτό να ελέγχεται τη πορεία του δικτύου από λίγους, δεδομένου του αριθμού των κόμβων που δυνητικά έχουν τη δυνατότητα να συμμετέχουν.

Ιδιωτικό blockchain (private):
Μια αρκετά δημοφιλής κατηγορία blockchain όπου οι συμμετέχοντες χρειάζονται έγκριση για να γίνουν μέλη του δικτύου, οι συναλλαγές είναι ιδιωτικές και ορατές μόνο από το οικοσύστημα του δικτύου (όχι από εξωτερικούς δράστες) και η λειτουργία τους βασίζεται περισσότερο σε κεντρικές "αρχές" (βλ. Hyperledger fabric [31]. Τέτοιου είδους υλοποιήσεις επιλέγονται κυρίως από οργανισμούς, οντότητες που μέσα στο δίκτυο έχουν αυξημένο έλεγχο.

Υβριδικό blockchain (hybrid):
Όπως γίνεται εύκολα αντιληπτό από τον τίτλο, μία υβριδική υλοποίηση blockchain συνδυάζει χαρακτηριστικά. Δηλαδή μπορεί να καθορίζει ποιες πληροφορίες είναι ιδιωτικές και ποιες αφήνονται προσβάσιμες για το κοινό. Ακόμα επιτυγχάνεται περισσότερη αποκέντρωση (decentralization) απο τα ιδιωτικά blockchain. Για παράδειγμα , κρυπτονομίσματα υποβάλλουν το hash της κάθε συναλλαγής τους σε άλλα blockchain ώστε να είναι πάντα διαθέσιμα. Φυσικά αυτά τα στοιχεία που δίνονται σε τρίτα blockchain πρέπει να γίνονται mask ώστε καμία προσωπική πληροφορία να μην είναι διαθέσιμη.

Ως προς το Consensus:

Σχήμα 2.11: Consensus

Η έννοια του Consensus είναι από τα βασικά χαρακτηριστικά που χαρακτηρίζουν ένα blockchain. Αναφέρεται σε ένα σύνολο κανόνων και μηχανισμών που παίζει καθοριστικό ρόλο στην διατήρηση και επέκταση της αλυσίδας γεγονότων (blockchain). Το είδος αυτών των κανόνων διαφέρει ανά υλοποίηση, με κάθε τέτοιο μηχανισμό να προσφέρει πλεονεκτήματα και μειονεκτήματα αναλόγως των ιδιοτήτων του. Το Consensus συνθέτει το πλαίσιο στο οπο-

ίο γίνεται το validation των block, καθώς και επιλύει ζητήματα που προκύπτουν κατά το validation.

Proof-of-Work

Το πιο διαδεδομένο κρυπτονόμισμα χρησιμοποιεί το Proof-of-Work (PoW) [45] σαν σχέδιο "συμφωνίας" των αποφάσεων μεταξύ των κόμβων ώστε να εξασφαλίζεται η ακεραιότητα των block της αλυσίδας. Με αυτόν το μηχανισμό, οι κόμβοι που αποκαλούνται miners προσπαθούν να κάνουν validate το block που έχει προταθεί προς προσθήκη στο blockchain. Οταν προστεθεί αυτό το block στην αλυσίδα τότε πλέον οι συναλλαγές που περιέχονται θεωρούνται έγκυρες. Το validation απαιτεί την αντίστροφη επίλυση μιας συνάρτησης κρυπτογραφίας που μπορεί να γίνει μόνο "brute-force". Ειδικότερα, οι miners πρεπει να βρουν αποτέλεσμα μιας συνάρτησης κρυπτογραφίας , με το να υπολογίζουν τιμές που βασίζονται στη τιμή hash του προηγούμενου block, στις συναλλαγές που περιέχονται στο block που έχει προταθεί και στο νονςε. Αυτό το αποτέλεσμα αρχίζει με συγκεκριμένο αριθμό μηδενικών.

Σε αυτό το είδος Consensus, η πιθανότητα κάποιος miner να κάνει mine το επόμενο block υπολογίζεται συναρτήσει της υπολογιστικής ισχύος που διαθέτει και της συνολικής υπολογιστικής ισχύος του συνόλου του δικτύου, το οποίο μεταφράζεται σε πολύ μικρή.

Συχνά η παραπάνω μέθοδος απαιτεί ειδικό εξοπλισμό, δηλαδή υλισμικό που είναι dedicated να κάνει γρήγορα τέτοιου είδους υπολογισμούς. και πολύ μεγάλο ενεργειακό αποτύπωμα ηλεκτρικής ενέργειας. Η πρακτική αξία αυτών των δύσκολων υπολογισμών είναι η εξασφάλιση της ακεραιότητας της διαδικασίας δημιουργίας block και της απλής επαλήθευσης των block από όλους τους κόμβους, ωστόσο έχει σαν μειονέκτημα υψηλό blocktime creation. Η αξία των επιβραβεύσεων για το μινινγ ενός block έχει οδηγήσει στο Centralization, δηλαδή πολλοί κόμβοι του δικτύου δημιουργούν μεγάλες mining-pools, γεγονός που αντιβαίνει στις αρχές της τεχνολογίας.

Proof-of-Stake

Το Proof-of-Stake [30] [40] σαν μέθοδος "συμφωνίας" συνδέει τη δημιουργία block με την απόδειξη ιδιοκτησίας ενός ποσού ηλεκτρονικών περιουσιακών στοιχείων. Η πιθανότητα να επιλεχθεί κάποιος κόμβος για να κάνει validate ένα block που έχει προταθεί, μεγαλώνει σε σχέση με το μερίδιο πλούτου που διαθέτει, σε μονάδα κρυπτονομισματος.

Αυτό το σχήμα βασίζεται στην υπόθεση ότι οι κόμβοι που κατέχουν μεγάλο μερίδιο πλούτου του συνόλου του δικτύου είναι πιο "έμπιστοι" αφού δεν τους συμφέρει να "βλάψουν" το δίκτυο, πράξη που θα έχει ως συνέπεια τη μείωση της αξίας του κρυπτονομίσματος. Εχουν αναπτυχθεί διάφορες εκδόσεις του PoS όπως το Randomized block selection -PoS η το Delegated-PoS(D-PoS).

Για παράδειγμα στο D-PoS, με το οποίο θα ασχοληθούμε εκτενέστερα στην πορεία, τα νέα block της αλυσίδας γίνονται forge από προκαθορισμένους κόμβους που έχουν εκλεγεί από τους χρήστες, οι οποίοι και έχουν το τελικό ρίσκο. Αυτοί οι κόμβοι επιβραβεύονται για την τέλεση των καθηκόντων ενώ τιμωρούνται όταν συμμετέχουν σε κακόβουλες ενέργειες. (π.χ. Τη συμμετοχή σε επιθέσεις με σκοπό το double-spending).

Proof-of-Authority

Στο πρωτόκολλο Proof-of-Authority[35], δεν απαιτείται από τους κόμβους του δικτύου η επίλυση μεγάλης δυσκολίας προβλημάτων , αλλα η τήρηση αυστηρά καθορισμένων συγκεκριμένων κανονισμών εξουσίας στο δίκτυο.

Πιο συγκεκριμένα, κάποιοι κόμβοι είναι αποκλειστικά υπεύθυνοι για την δημιουργία block κατά συνέπεια να διασφαλίζουν την ακεραιότητα της αλυσίδας. Αυτός ο μηχανισμός είναι φανερό ότι ταιριάζει σε ιδιωτικά δίκτυα, όπου προκαθορισμένοι κόμβοι είναι υπεύθυνοι και επιφορτισμένοι να επικοινωνουν στη βάση ότι είναι όλοι έμπιστοι, για τον έλεγχο του περιεχομένου που καταγράφεται στα νεα block. Αυτοί οι κόμβοι "υπογράφουν" με τα προσωπικά τους ηλεκτρονικά κλειδιά (private keys) κάθε νέο block, δρώντας συνολικά ως έμπιστη αρχή.

Ως προς τα Permissions:

Permissioned

Τα Permissioned blockchains βασίζουν την ομαλή λειτουργία σε συγκεκριμένους κόμβους κλειδιά/ οντότητες, όπως τους μετόχους ενός οργανισμού. Πρόκειται για ένα συνδυασμό των καλύτερων χαρακτηριστικών απο τα πριατε και τα πυβλις blockchain. Σε αυτό τον τύπο, ο κόμβος που συμμετέχει μπορεί να μη χρειάζεται εξουσιοδότηση για να συνδεθεί στο δίκτυο του blockchain αλλά σίγουρα χρειάζεται συγκεκριμένη άδεια για να συνδιαλλαγεί με τους υπόλοιπους κόμβους, δηλαδή να εγκριθεί από τις "οντότητες" που ελέγχουν το δίκτυο.

Permissionless

Όπως λέει και η λέξη, σε αυτή τη περίπτωση, είναι ελεύθερη τόσο η σύνδεση , όσο και οι συναλλαγές, αφού ο καθένας μπορεί να δημιουργήσει λογαριασμό και στη συνέχεια να μεταφέρει ποσά ή να του μεταφερθούν ποσά. Permissionless υλοποιήσεις είναι το Bitcoin και το Ethereum.

2.4.4 Use Cases

Cryptocurrencies

Η πιο διαδεδομένη περίπτωση χρήσης είναι η ανάπτυξη ηλεκτρονικών νομισμάτων[12] [5] [22] [7], των οποίων η αξιοπιστία βασίζεται στην τεχνολογία του blockchain. Επιτρέπει την εύκολη πραγματοποίηση συναλλαγών μεταξύ λογαριασμών του δικτύου χωρίς την παρουσία μιας κεντρικής ελεγκτικής αρχής και του τραπεζικού συστήματος. Παράλληλα υπάρχει η δυνατότητα απόκρυψης της ταυτότητας των συναλλαγών αυτών.

Smart contracts

Τα έξυπνα συμβόλαια [30] [46] αφορούν συμβολαιογραφικού τύπου ενέργειες οι οποίες εκτελούνται μερικώς ή εξ ολοκλήρου αυτόματα, χωρίς την αλληλεπίδραση με το ανθρώπινο στοιχείο. Για παράδειγμα, δύο άνθρωποι μπορούν να πραγματοποιήσουν ένα ηλεκτρονικό συμβόλαιο το οποίο αυτόματα ανά συγκεκριμένο χρονικό διάστημα και για συγκεκριμένο χρονικό ορίζοντα θα μεταβιβάζει ποσά κρυπτονομίσματος από τον έναν λογαριασμό στον άλλο, για την

εκπλήρωση των υποχρεώσεων του ενοικίου ενός χώρου. Αυτό γίνεται χωρίς την παρέμβαση συμβολαιογράφου καθώς και χωρίς τη διαμεσολάβηση του τραπεζικού συστήματος, απλοποιώντας μια, κατα τα άλλα, γραφειοκρατική διαδικασία που κοστίζει χρόνο και γλιτώνοντας ενδεχόμενες τρίτες χρεώσεις.

Supply Chain Management

[41] Πρόκειται για ένα δίκτυο μεταξύ οργανισμών και των προμηθευτών τους με στόχο την παρακολούθηση των προϊόντων καθώς και την άμεση διαπίστωση προβλημάτων στην γραμμή τροφοδοσίας. Η διασφάλιση της διαθεσιμότητας των πληροφοριών καθώς και της ακεραιότητας τους προέρχεται από την τεχνολογία blockchain.

Other uses

Παραπάνω αναφέρθηκαν οι τρεις πολύ γνωστές περιπτώσεις χρήσης της blockchain τεχνολογίας. Φυσικά, υλοποιήσεις του blockchain έχουν δημιουργηθεί με στόχο:

- Την εξασφάλιση των πνευματικών δικαιωμάτων[3] στη μουσική, στον κινηματογράφο κλπ Την διαπίστωση προέλευσης των τροφών. Για παράδειγμα υπάρχει δίκτυο στο οποίο καταγράφεται η προέλευση ψαριών, η οποία μπορεί να ελεγχθεί με τη βοήθεια της σήμανσης στο καρτελάκι του κάθε ψαριού.

- Την καταγραφή ηλεκτρονικών περιουσιακών στοιχείων μέσα σε ένα ηλεκτρονικό παιχνίδι. Το Crypto-Kitties [9] είναι το διαδικτυακό παιχνίδι το οποίο χρησιμοποιεί το blockchain για να αποθηκεύει ιν-γαμε πληροφορίες καθώς και για in-game αγορές.

- Την εύκολη προσβασιμότητα σε προσωπικά ιατρικά δεδομένα. Στον τομέα της υγείας, η δυνατότητα του θεράποντος ιατρού να έχει πρόσβαση στο συνολικό ιστορικό ενός ατόμου διευκολύνει τόσο τη διαδικασία της διάγνωσης καθώς και μειώνει τη γραφειοκρατία που έχει να αντιμετωπίσει ο ασθενής. Προκλήσεις που μένει να επιλυθούν είναι ζητήματα προστασίας προσωπικών δεδομένων καθώς και η εμπιστοσύνη των αρχών στη συγκεκριμένη τεχνολογία.

- Την αποθήκευση δεδομένων σε cloud. Αντί να χρησιμοποιούνται οι υπάρχουσες ςλουδ πλατφόρμες αποθήκευσης δεδομένων (βλ. Google Drive , Dropbox), υπάρχει η πρόταση για ένα blockchain το οποίο επιτελεί τον ίδιο σκοπό με τη βοήθεια του Proof-of-Existence.

- Την δημιουργία δικτύου "δανεισμού" υπολογιστικής ισχύος [16]. Αναλυτικότερα, αντί να απευθύνεται κάποιος στην Αμαζον για ενοικίαση Virtual Machines, θα μπορεί να απευθύνεται στο δίκτυο του blockchain που έχει τη δυνατότητα να "τρέξει" distributed apps.

2.4.5 Blockchain Technology Considerations

Παραπάνω αναφέρθηκαν βασικές έννοιες και χαρακτηριστικά της blockchain τεχνολογίας χωρίς όμως να διατυπωθούν βασικά προβλήματα.

Waste of energy and computing power

Στο Bitcoin, χρησιμοποιείται το Proof-of-Work σαν πρωτόκολλο συμφωνίας για τη δημιουργία καινούργιων block της αλυσίδας. Δηλαδή εκατοντάδες χιλιάδες κόμβοι ανταγωνίζονται, ώστε να υπολογίζουν το hash του καινούργιου block που θα προστεθεί στην αλυσίδα. Γιατί το Bitcoin πρέπει να έχει ισοδύναμη κατανάλωση ενέργειας με την Ελβετία[43] [48]· Παράλληλα, από όλη την υπολογιστική δύναμη που επιστρατεύεται, σημασία έχει μόνο η προσπάθεια του miner που βρήκε το ζητούμενο hash. Μήπως η ενέργεια που χρησιμοποιείται για την πραγματοποίηση των συγκεκριμένων υπολογισμών θα μπορούσε να μειωθεί η να αξιοποιηθεί αλλου· Το ενεργειακό αποτύπωμα θα έπρεπε να είναι βασικό κριτήριο στην διαδικασία ανάπτυξης τεχνολογιών στα πλαίσια της προστασίας του περιβάλλοντος του πλανήτη μας.

Blockchain is not a distributed system

Υπάρχει η παρερμηνεία, ότι το blockchain αποτελεί ένα κατανεμημένο δίκτυο κόμβων, όπου η συνεργασία των κόμβων οδηγεί στην δημιουργία της αλληλουχίας της αλυσίδας. Στην πραγματικότητα δεν υπάρχει παραλληλία ή συνεργασία μεταξύ των κόμβων. Αντιθέτως κάθε κόμβος του δικτύου κάνει ακριβώς τα ίδια πράγματα. Στην περίπτωση του Bitcoin κάθε κόμβος: Επαληθεύει την εγκυρότητα των συναλλαγών του δικτύου με βάση τους ίδιους κανόνες και πραγματοποιούν ίδιες ενέργειες. Καταγράφει πανομοιότυπη, με τους άλλους κόμβους, πληροφορία ολόκληρου του blockchain πάντα

Αποτελεί μία μη αποδοτική διαδικασία που λαμβάνει χώρα σε κάθε node ξεχωριστά. Εαν και κατανεμημένο, δεν είναι ένα κατανεμημένο υπολογιστικό σύστημα του οποίου η λειτουργία επωφελεί πολλούς.

Scalability issues

Η πιο επιτυχημένη υλοποίηση blockchain, το Bitcoin, ανέδειξε τα προβλήματα της κλιμακωσιμότητας της τεχνολογίας αφού κάθε block που περιέχει συναλλαγές δημιουργείται κάθε , κατά μέσο όρο, 10 λεπτά [26]. Συνεπώς μια συναλλαγή από τη στιγμή της καταχώρισης της χρειάζεται σημαντικό χρονικό διάστημα για να ενταχθεί στην αλυσίδα. Αυτό δεν είναι λειτουργικό για μια μεγάλης κλίμακας enterprise-εφαρμογή, που απαιτεί γρήγορη απόκριση. Παράλληλα μια τέτοια εφαρμογή απαιτεί δυνατότητα υψηλού αριθμού συναλλαγών το δευτερόλεπτο, και στη περίπτωση του Bitcoin κάθε block μπορεί να περιέχει ως και 4000 συναλλαγες, δηλαδή περίπου 6.7 συναλλαγές το δευτερόλεπτο. Αλλες υλοποιήσεις blockchain, προσπαθούν να αυξήσουν το throughput των συναλλαγών μειώνοντας το blockchain, ωστόσο ο ανταγωνισμός είναι σκληρός με την Visa να επεξεργάζεται κατα μέσο όρο 1700 συναλλαγές [27] το δευτερόλεπτο.

Anonymity

Η ανωνυμία σε ένα blockchain [17] θεωρείται μεγάλο πλεονέκτημα αφού η ταυτότητα ενός

λογαριασμου δεν συνδέεται με το λογαριασμό.

Εν μέρει αυτό δεν είναι αληθές αφού με τη παρακολούθηση των συναλλαγών καθώς της δραστηριότητας, είναι δυνατόν να διαπιστωθεί η πιθανή ταυτότητα. Ταυτόχρονα ίσως αποτελεί βραχνά για μία εμπορική. οικονομικού σκοπού, εφαρμογή, αφού ο χρήστης πρέπει να γνωρίζει απο ποιον λαμβάνει χρήματα ή να εξακριβώνει την ταυτότητα του λογαριασμού στον οποίο στέλνει λεφτά. Το να υπάρχει masking της πληροφορίας στην ταυτότητα και στις συναλλαγές δεν είναι λειτουργικό, ενώ η πολλή πληροφορία εμπεριέχει κινδύνους για τους χρήστες.

Blockchain synonymous of Complexity

Η αρχιτεκτονική του blockchain μπορεί να φαίνεται κατανοητή, αλλά η πολυπλοκότητα της είναι μεγάλη και έχει περιορισμούς. Οι υπάρχουσες centralized υποδομές και υπηρεσίες πρέπει να προσαρμοστούν με τις λειτουργικότητες του blockchain, αν θέλουν να επενδύσουν προς αυτό το τομέα. Αυτό συνεπάγεται εκπαίδευση, νέες υποδομές και κοστοβόρες αλλαγές στη λειτουργία των μεγάλων οργανισμών.

Politics

Η τεχνολογία του blockchain προσφέρει την δυνατότητα της ψηφιοποίησης κυβερνητικών μοντέλων καθώς και της "παράκαμψης" υπάρχοντων αρχών όπως του τραπεζικού συστήματος καθώς και της γραφειοκρατίας. Οι κυβερνήσεις ενώ παρακολουθούν την εξέλιξη των πραγμάτων αδυνατούν να υιοθετήσουν τις νέες τεχνολογίες, καθώς πρέπει να δημιουργηθούν / προσαρμοστούν τα υπάρχοντα νομικά πλαίσια[6] [37] για τον έλεγχο η την οριοθέτηση τους, και στη προκειμένη περίπτωση της τεχνολογίας του blockchain. Ο λόγος είναι ότι υπάρχουν πολλές αντιδράσεις από το υπάρχων καθεστώς με τρανταχτό παράδειγμα το Τραπεζικό σύστημα. Αντίφαση· Σίγουρα καθώς ένα σύστημα το οποίο έχει ως στόχο την αποκέντρωση πρέπει να αποκτήσει μία κεντρική αρχή ελέγχου.

Κεφάλαιο 3

Θεωρητικό υπόβαθρο

3.1 Εισαγωγή

Στο κεφάλαιο αυτό διατυπώνεται το θεωρητικό υπόβαθρο το οποίο είναι απαραίτητο για την κατανόηση της πρότασης που θα περιγραφεί σε επόμενο κεφάλαιο. Αρχικά παρουσιάζεται η πορεία του παγκόσμιου ιστού καθως και ο λόγος που επιτάσσεται η αποκεντροποίηση του. Στη συνέχεια αναλύονται οι βασικές αρχές της τεχνολογίας των δικτύων ομότιμων κόμβων P2P πρωτοκόλλου και Chord πρωτοκόλλου - πρωτοκόλλων για την διασύνδεση και επικοινωνία κόμβων στο διαδίκτυο. Τέλος θα γίνει παρουσίαση της ισχύουσας υλοποίησης του κρυπτονομίσματος Semux που βασίζεται στη τεχνολογία του Blockchain και τροποποιήθηκε ο πηγαίος κώδικας για τις ανάγκες της εργασίας.

3.2 Web3 - Ο αποκεντρωμένος ιστός

Στις αρχές της δεκατετίας του 1990, ο παγκόσμιος ιστός έφερε μία επανάσταση στη μετάδοση της πληροφορίας. Ο Παγκόσμιος Ιστός (World Wide Web) είναι μία διαδικτυακή εφαρμογή βασισμένη σε καινοτομίες όπως η γλώσσα HTML, οι διευθύνσεις URL και το πρωτόκολλο μεταφοράς υπερκειμένου ή HTTP.

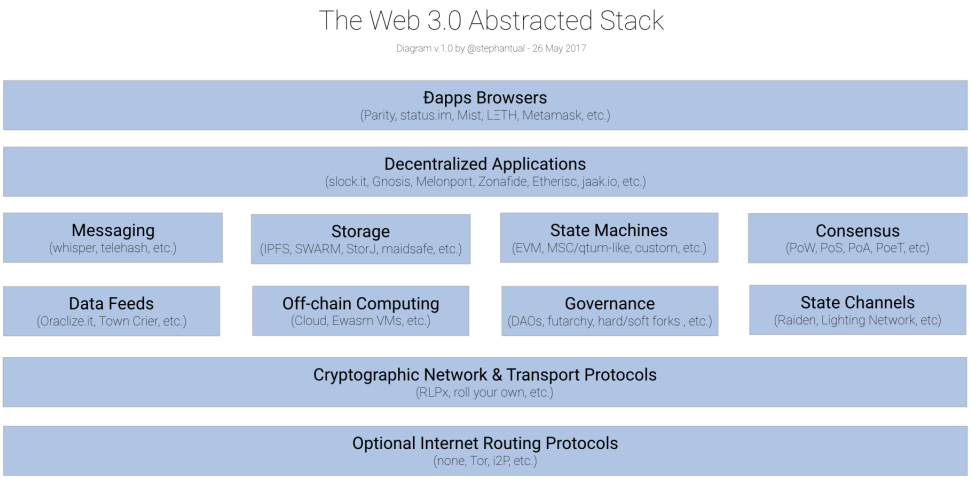
3.2.1 Ιστορία

Το Web 1.0 [15] αποτελεί την πρώτη έκδοση του διαδικτύου που ήταν σε ισχύ εώς και το 2003 και παρείχε κατά βάση read-only ιδιότητες. Έπειτα, ο παγκόσμιος ιστός ωρίμασε και γεννήθηκε το Web 2.0, το οποίο παρείχε δυνατότητες read-write . Με το Web 2.0 μπήκαν στην αρένα μεγάλοι οργανισμοί όπως Google, Facebook, Amazon καθώς και τραπεζικές υπηρεσίες. Χαρακτηριστικό όλων ήταν ότι κάθε εφαρμογή είχε μία κεντρική οντότητα η οποία είχε τον πλήρη έλεγχο της λειτουργίας της. Ακόμα και στις τότε αποκεντρωμένες (P2P) εφαρμογές υπήρχε κάποιου είδους κεντρική οντότητα που εκτελούσε χρέη διαχειριστή. Χαρακτηριστικό εκείνης της εποχής ήταν επίσης ότι επειδή οι υπηρεσίες που παρείχαν οι εφαρμογές ήταν δωρεάν.

"If you're not paying for a product, then you are the product"

"The Times 03/Jan/2009 Chancellor on brink of second bailout for banks."

Η παραπάνω φράση περιέχεται στο genesis block του κρυπτονομίσματος Bitcoin, το οποίο έδωσε το έναυσμα για το Web 3.0/Decentralised Web. Τα κρυπτονομίσματα που χρησιμοποιούν την τεχνολογία του Blockchain παρείχαν τη δυνατότητα οικονομικών συναλλαγών.



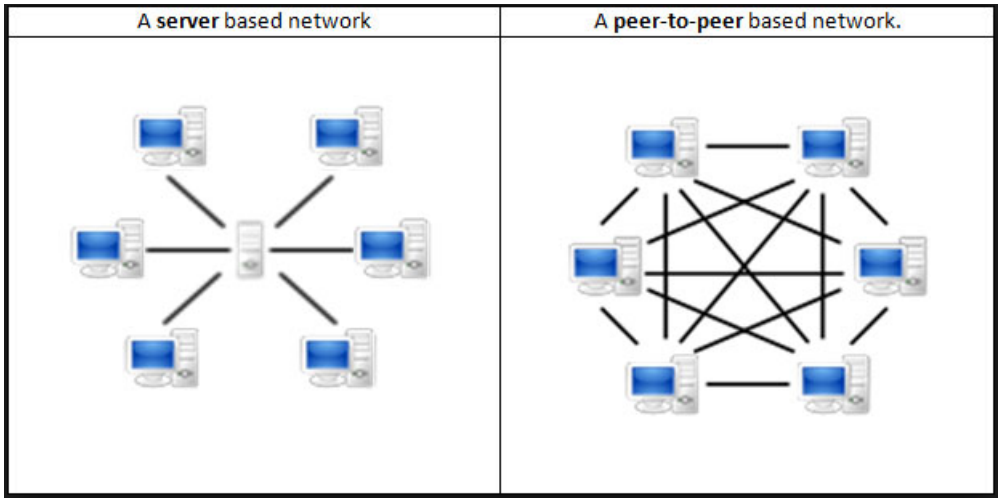
Σχήμα 3.1: Αφηρημένη στοίβα τεχνολογιών του Web 3.0

Διαμορφώνεται με αυτό το τρόπο μία νεά τάση η οποία προτάσει την αποκεντροποίηση του διαδικτύου με την εξάλειψη των κεντρικών εξυπηρετητών. Προτείνεται ως μία λύση, εώς και σήμερα, για την αντιμετώπιση των μειονεκτημάτων του μοντέλου πελάτη-εξυπηρετητή που αφουρούν κυρίως την κατάχρηση προσωπικών δεδομένων, την απώλεια τους καθώς και σε γενικότερο πλαίσιο την ύπαρξη μοναδικού σημείου αποτυχίας(Single Point of Failure).

3.3 Δίκτυα ομότιμων κόμβων

Στην ενότητα αυτή, θα γίνει μία σύντομη παρουσίαση της αρχιτεκτονικής ομότιμων κόμβων, δεδομένου ότι αποτελεί θεμέλιο λίθο της τεχνολογίας του Blockchain.

Η αρχή λειτουργίας των δικτύων ομότιμων κόμβων[33] είναι η απευθείας επικοινωνία ανάμεσα σε ζεύγη συνδεδεμένων υπολογιστών -peers. Αυτοί οι κόμβοι είναι κατά βάση υπολογιστές που δεν ελέγχονται απο κάποιον οργανισμό αλλά ανήκουν σε χρήστες. Οι ομότιμοι κόμβοι παρέχουν ένα μέρος των πόρων τους (υπολογιστικών, αποθηκευτικού χώρου κλπ) για την λειτουργία του δικτύου, καταργώντας έτσι την ανάγκη ύπαρξης μία κεντρικής αρχής για το συντονισμό και τη λειτουργία του δικτύου. Ο ρόλος των ομότιμων κόμβων είναι να λειτουργούν τόσο ώς client που καταναλώνουν πόρους, όσο και ως server προσφέροντας πόρους. Παραδείγματα τέτοιου είδους εφαρμογών είναι το Torrent , Skype. Τα δίκτυα ομότιμων κόμβων χωρίζονται σε 2 κατηγορίες, στα δομημένα και στα αδόμητα.



Σχήμα 3.2: Αρχιτεκτονική client-server και ομότιμων κόμβων

Δομημένα δίκτυα Τα δομημένα δίκτυα ομότιμων κόμβων οργανώνονται, σχηματίζοντας μία ορισμένη τοπολογία σύμφωνα με κάποιο πρωτόκολλο, γεγονός που εξασφαλίζει την αποδοτική αναζήτηση οποιωνδήποτε δεδομένων, από οποιοδήποτε κόμβο. Ενας απο τους πιο διαδεδομένους τύπους δομημένου δικτύου ομότιμων κόμβων είναι το DHT - Distributed Hash Table, το οποίο αναλύεται εκτενώς παρακάτω. Συνοπτικά υλοποιεί έναν κατανεμημένο πίνακα κατακερματισμού με βάση τον οποίο γίνεται η ανάθεση κάθε αρχείου σε συγκεκριμένο κόμβο. Ετσι η αναζήτηση κάθε αρχείο γίνεται αποτελεσματικά από κάθε κόμβο με τη χρησιμοποίηση του κατανεμημένου πίνακα κατακερματισμού.

Αδόμητα δίκτυα Τα αδόμητα δίκτυα, σε αντίθεση με τα δομημένα, δεν επιβάλουν κάποια δομή αλλά πραγματοποιούνται τυχαίες συνδέσεις μεταξύ των κόμβων. Δεν υπάρχει κανένας συσχετισμός μεταξύ των δεδομένων και των κόμβων που τα προσφέρουν.

Τα βασικά πλεονεκτήματα της αρχιτεκτονικής ομότιμων κόμβων είναι·

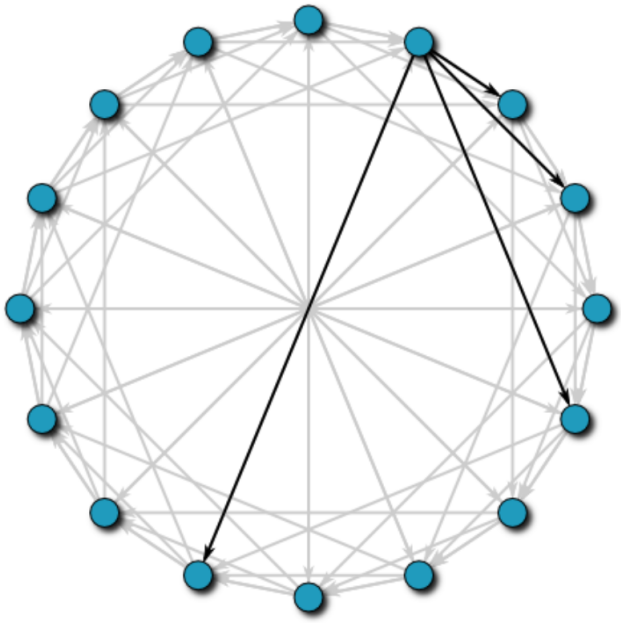
- **Αυτοκλιμακωσιμότητα του δικτύου** Οσο περισσότεροι κόμβοι προστίθενται στο δίκτυο , τόσο αυξάνεται ο φόρτος εργασίας, η απαίτηση δηλαδή σε πόρους. Ταυτόχρονα όμως αυξάνονται και οι διαθέσιμοι πόροι λόγων της φύσεων των συμμετεχόντων κόμβων.
- **Μείωση κόστους** Συνήθως δεν απαιτείται υψηλού κόστους υλισμικό καθώς και ταχεία σύνδεση στο δίκτυο.
- **Μη ύπαρξη μοναδικού σημείου αποτυχίας** Η βλάβη σε έναν κόμβο δεν επηρεάζει απαραίτητα την λειτουργία του υπόλοιπου δικτύου.

Οι προκλήσεις που αντιμετωπίζουν οι εφαρμογές που υλοποιούνται με τη βοηθεία της αρχιτεκτονικής αυτής είναι:

- **Ασφάλεια** Λόγω της κατανεμημένης φύσης τους, οι εφαρμογές μπορουν να δημιουργήσουν προβλήματα ασφαλείας. Σε υλοποιήσεις κρυπτονομισμάτων επιλύεται εύκολα με τη βοήθεια της κρυπτογραφίας.

- **Κίνητρο** Για ποιο λόγο επιθυμούν οι κόμβοι να συμμετέχουν σε ένα τέτοιου είδους δίκτυο και να παρέχουν την υπολογιστική τους ισχύ·
- **Μη φιλικότητα προς τους παρόχους διαδικτυακών υπηρεσιών** Σε τέτοιου είδους δίκτυα παρατηρείται συνήθως ασσύμετρη χρησιμοποίηση του εύρους ζώνης, δηλαδή μεγαλύτερη συρρευματική παρά αντιρρευματική κίνηση, δυσκολεύοντας έτσι την παροχή πόρων προς το σύστημα.

3.4 Chord

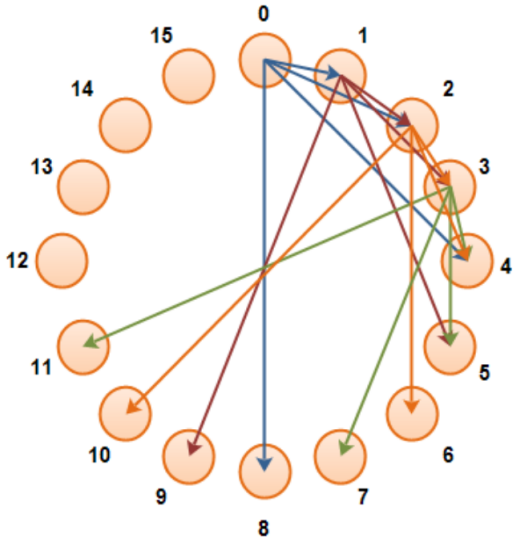


Σχήμα 3.3: Τοπολογία Chord

Τα Peer-to-peer συστήματα είναι κατανεμημένα συστήματα τα οποία δεν ελέγχονται από κάποια κεντρική αρχή ή έχουν ιεραρχική δομή, ενώ ο κώδικας ο οποίος τρέχει σε κάθε κόμβο είναι ίδιος και επιτελεί ίδια λειτουργικότητα. Κάποια χαρακτηριστικά των συστημάτων αυτών αναλύθηκαν σε προηγούμενα κεφάλαια (ανωνυμίας, authentication, redundant storage κλπ). Το chord [47] [34] έχει να προσφέρει σε αυτά ένα αποδοτικότερο πρωτόκολλο για αναζήτηση μέσα σε αυτά τα δίκτυα, όπου οι αφίξεις και οι αποχωρήσεις κόμβων είναι συνεχής.

Η λογική είναι απλή: Δεδομένου ότι υπάρχει ένα κλειδί , τότε αντιστοιχίζεται μόνο σε ένα κόμβο. Αναλόγως την εφαρμογή, ο κόμβος μπορεί να είναι υπεύθυνος για την αποθήκευση της τιμής που σχετίζεται με αυτό το κλειδί. Χρησιμοποιώντας συναρτήσεις για consistent hashing, τείνει να εξισορροπείται ο φόρτος ισόποσα σε όλους τους κόμβους, καθώς σε κάθε μέλος του δικτύου αναλογούν ο ίδιος αριθμός κλειδιών. Παράλληλα, δεν υπάρχει μεγάλη κίνηση όγκου κλειδιών κατά τις αποχωρήσεις η προσθήκες κόμβων στο δίκτυο.

Το πρωτόκολλο αυτό έχει ως στόχο την υψηλή κλιμακωσιμότητα, στη βάση της ιδέας ότι κάθε κόμβος γνωρίζει πληροφορίες και διευθύνσεις μόνο λίγων, συγκεκριμένων κόμβων. Σε



Σχήμα 3.4: Το Chord

κατάσταση ισορροπίας, σε ένα δίκτυο N-κόμβων, ο κάθε κόμβος διατηρεί πληροφορίες μόνο για $O(logN)$ κόμβους , και κάθε αναζήτηση επιλύεται μέσω $O(logN)$ μηνυμάτων σε άλλους κόμβους. Η πληροφορίες του routing ανανεώνονται κατά τις αλλαγές στο δίκτυο. Δηλαδή ένας κόμβος απο τον άλλο είναι "μακριά" εώς και $logN$ μηνύματα.

Η απλότητα του αλγορίθμου με τον οποίο υλοποιείται το πρωτόκολλο του Chord καθώς και η υψηλή απόδοση του, το κάνει να ξεχωρίζει. Στην πραγματικότητα αρκεί να διατηρείται σε κάθε κόμβο για την ομαλή και αποδοτική λειτουργία του πρωτοκόλλου, τα routing tables.

Chord-Το πρωτόκολλο

Μια πρώτη επαφή

Το βασικό feature του Chord, είναι ότι παρέχει γρήγορο κατανεμημένο υπολογισμό της συνάρτησης κατακερματισμού που αντιστοιχεί τα κλειδιά σε υπεύθυνους γι αυτά , κόμβους.

Η χρησιμοποίηση Consistent Hashing έχει ως αποτέλεσμα την εξισορρόπηση του φόρτου που αναλογεί σε κάθε κόμβο καθώς και αποφεύγονται τα collisions μεταξύ των παραγόμενων κλειδιών . Όταν προστίθεται κάποιος κόμβο στο σχηματισμό N-κόμβους, συνεπάγεται ότι αναλογικά το $1/N$ πλήθος κλειδιών χρειάζεται να μετακινηθεί στο δίκτυο, που είναι το ελάχιστο δυνατό.

Οι κόμβοι στο δίκτυο του Chord, οργανώνονται με βάση το αναγνωριστικό τους σε ένα κύκλο, ο οποίος μπορεί να φιλοξενήσει το πολύ $2m$, με το μ να είναι πολύ μεγάλο . Κάθε κόμβος έχει αναγνωριστικό κλειδί με εύρος $(0, 2m - 1)$. Κατά συνέπεια, κάποιοι κόμβοι αντιστοιχίζονται σε ένα κλειδί, ενώ τα περισσότερα κλειδιά (αφού το m είναι πολύ μεγάλο) παραμένουν χωρίς αντιστοίχιση.

Κάθε κόμβος του δικτύου έχει τον πρόγονο (predecessor) και τον απόγονο του (successor) . Ο successor είναι ο επόμενος κόμβος, με βάση το κλειδί που του έχει αποδοθεί , σε κα-

Notation	Definition
finger[k].*start*	$(n + 2^{k-1}) \bmod 2^m$, $1 \leq k \leq m$
.*interval*	[*finger*[k].*start*, *finger*[$k+1$].*start*)
.*node*	first node $\geq n$.*finger*[k].*start*
successor	the next node on the identifier circle; *finger*[1].*node*
predecessor	the previous node on the identifier circle

Σχήμα 3.5: Αναπαράσταση Finger table

τεύθυνση ωρολογιακή. Αντίστοιχα ο predecessor είναι ο προηγούμενος κόμβος, με βάση το κλειδί που του έχει αποδοθεί, σε κατεύθυνση αντιωρολογιακή. Για παράδειγμα, αν το Chord είναι γεμάτο, δηλαδή κάθε δυνατό αναγνωριστικό αντιστοιχίζεται σε κόμβο, τότε ο κόμβος 55 έχει predecessor τον 54 και successor τον 56.

Η λογική με την αντιστοίχιση αναγνωριστικού σε κόμβους μπορεί να ακολουθηθεί και για τον καθορισμό του ποιος κόμβος, θα αποθηκεύσει ένα block πληροφορίας. Εστω ότι υπάρχει ένα κείμενο, τον οποίο σαν unique Id έχει έναν αριθμό και πρέπει να αποφασιστεί ποιος από τους κόμβους του ῆορδ θα αναλάβει να διατηρήσει τη πληροφορία. Ο successor του unique Id του κειμένου είναι ο πρώτος κόμβος του οποίου το αναγνωριστικό ισούται με το unique Id ή είναι αμέσως επόμενο διαθέσιμο μετά το unique Id του κειμένου. Συνεπώς στο συγκεκριμένο παράδειγμα, κάθε κείμενο με unique Id ανατίθεται στο αντίστοιχο κόμβο που είναι successor του unique Id.

Φαίνεται λοιπόν ότι η αναζήτηση ενός κειμένου με βάση το unique Id του είναι μια σχετικά απλή διαδικασία, αφού αρκεί να εντοπιστεί ο successor κόμβος του unique Id.

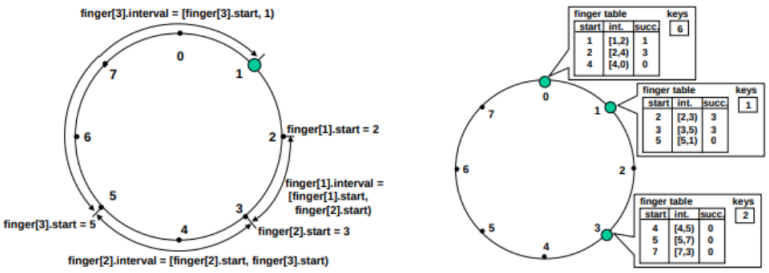
Digging deeper

Η παραπάνω ανάλυση μπορεί είναι κατανοητή, αλλα δεν αρκεί αφού αφήνει αναπάντητα ερωτήματα. Πως είναι αποδοτικό ένα τέτοιο πρωτόκολλο, αφού φαίνεται να χρειάζεται δυνητικά πολλά μηνύματα για να βρεθεί ένα συγκεκριμένο Id σε ένα δίκτυο μεγάλης κλίμακας. Επίσης, είναι δυνατόν αν crash-αρει ένας κόμβος και να χάνεται αυτόματα η πληροφορία που έχει αποθηκεύσει·

Παρακάτω θα αναλυθεί η βάση του πρωτοκόλλου για να καλυφθούν πιθανά κενά. Επίσης θα αναφερθούν πληροφορίες που έχουν ήδη ειπωθεί για λόγους πληρότητας.

Finger Table

Στον πυρήνα της λειτουργίας του chord βρίσκεται το finger table. Το finger table είναι ένας πίνακας δρομολόγησης, στον οποίο καταγράφονται κάποιες “σημαντικές” διευθύνσεις για κάθε κόμβο. Πιο συγκεκριμένα σε κάθε εγγραφή του finger table περιέχεται ένα κλειδί (start) καθώς και ο successor του κλειδιού αυτού (succ) στο δίκτυο. Οπως φαίνεται στην εικόνα 3.6, ο κόμβος 0 έχει πρώτη εγγραφή στο finger table το κλειδί 1 του οποίου successor είναι ο κόμβος 1. Αντίστοιχα successor του κλειδιού 2 είναι ο κόμβος 3 ενώ του κλειδιού 4 είναι ο



Σχήμα 3.6: Παράδειγμα τοπολογίας Chord με τα finger tables

ίδιος ο 0.

Η λογική αυτή έχει ως αποτέλεσμα την μη γραμμική αναζήτηση των κόμβων για τα περιεχόμενα κλειδιά τους αλλά μεγάλα άλματα στο Chord.

Key location-Querying

Η βασική λειτουργία είναι η αναζήτηση από έναν client ενός κλειδιού. Η πρακτική είναι ότι ο client συνδέεται σε ένα κόμβο του δικτύου και στέλνει το request του. Το request μεταδίδεται στο successor του node, αν δεν βρεθεί τοπικά. Η διαδικασία αυτή επαναλαμβάνεται μέχρι να βρεθεί το ζητούμενο αναγνωριστικό. Ωστόσο, εύκολα υπολογίζεται ότι στο worst case scenario θα υπάρχει πολυπλοκότητα $O(N)$, όσοι και οι κόμβοι του δικτύου.
Για την αύξηση των επιδόσεων και τη μείωση των απαιτούμενων μηνυμάτων χρησιμοποιούνται τα finger tables. Ενα finger table , δεν είναι κάτι άλλο από ένα routing table που περιέχει το πολύ μ εγγραφές που αντιστοιχίζονται σε κόμβους του δικτύου.

Η ι-οστή εγγραφή του finger table του κόμβου n, αντιστοιχίζεται στον πρώτο σε σειρά κόμβο s που διαδέχεται το ν κατά τουλάχιστον $2i-1$ με βάση το αναγνωριστικό των κόμβων. Δηλαδή

$$s = successor(n + 2i - 1), 0 < i < m + 1. \tag{3.1}$$

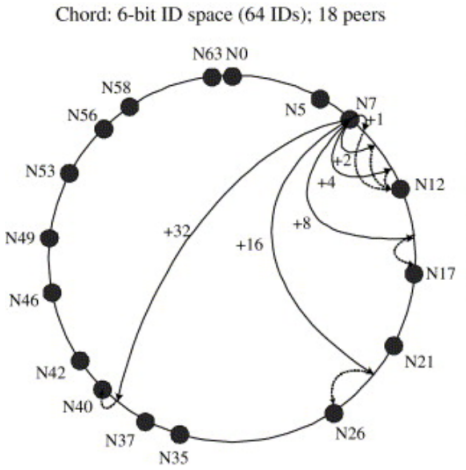
Ο κόμβος s αποκαλείται ως το ι-οστό finger του κόμβου n.

Στην εικόνα 3.7 φαίνεται ότι το finger table του κόμβου 7 του δικτύου που αναπαριστάται. Τα πρώτα 3 fingers είναι ο ίδιος κόμβος , ο κόμβος 12, ενώ το έκτο finger είναι ο κόμβος 40.

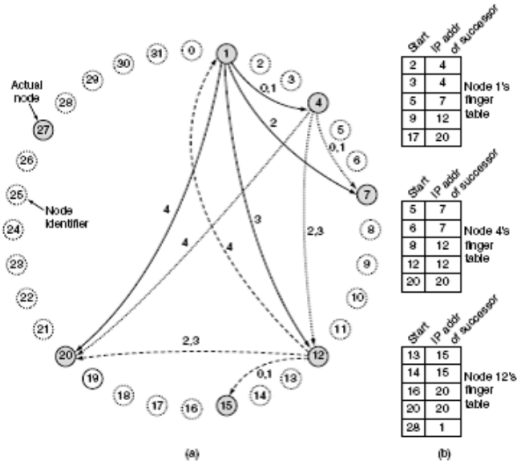
Το σχήμα αυτό κάνει εμφανή 2 χαρακτηριστικά. Πρώτον, κάθε κόμβος αποθηκεύει πληροφορίες για μικρό αριθμό κόμβων του υπολοιπου δικτύου και γνωρίζει περισσότερες πληροφορίες για τους κόμβους που βρίσκονται “κοντά” του. Δεύτερον, το finger table ενός κόμβου δεν είναι αρκετό για καθορίσει το ποιος κόμβος για ένα δεδομένο κλειδί k είναι $successor(k)$.

Τι γίνεται όταν ο κόμβος n δεν γνωρίζει τον successor του κλειδιού k· Ο n έχει την υποχρέωση να αναζητήσει τον κόμβο ο οποίος είναι πιο κοντά από τον ίδιο στο κλειδί k. Αυτός ο κόμβος , έστω j, γνωρίζει περισσότερα από το n για τα αναγνωριστικά του κύκλου για το εύρος του k. Η διαδικασία επαναλαμβάνεται μέχρις ότου να υπάρξει επιτυχία.

Ας υποθέσουμε ότι ο κόμβος 4 επιθυμεί να βρει το successor(18). Δεδομένου του finger table που έχει δημιουργήσει, θα απευθυνθεί στον κόμβο 12 και στη συνέχεια ο κόμβος 12



Σχήμα 3.7: Παράδειγμα finger table ενός κόμβου στο Chord



Σχήμα 3.8: Παράδειγμα χρήσης finger tables

θα επικοινωνήσει με τον 15. Ο 15 γνωρίζει ότι ο $successor(15)$ είναι ο κόμβος 20 και κατά συνέπεια είναι και $successor(18)$. Ουσιαστικά με τα finger tables επιτυγχάνεται η μείωση των αποστάσεων για τον κόμβο-στόχο.

Node joins-leaves

Το χαρακτηριστικό των δυναμικών δικτύων, είναι ότι συνεχώς κόμβοι συνδέονται και αποχωρούν απρόβλεπτα. Αυτό αποτελεί μια πρόκληση για τη διατήρηση των κανόνων που έχουν τεθεί στο πρωτόκολλο. Πιο συγκεκριμένα:

- Κάθε κόμβος συνδέεται κατάλληλα στο δίκτυο ώστε να τον διαδέχεται ο σωστός κόμβος, δηλαδή ο successor του
- Για κάθε κλειδί k , ο κόμβος- $successor(k)$ είναι υπεύθυνος για αυτά.

- Το finger table του κάθε κόμβου μετά από κάθε αλλαγή να διατηρείται ενημερωμένο.

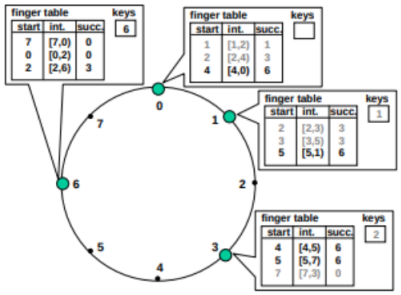
Για την απλοποίηση των λειτουργιών Node join και Node leave κάθε κόμβος στο Chord διατηρεί έναν δείκτη προς τον predecessor του για να μπορεί να επικοινωνεί και προς αντιωρολογιακή φορά, καθώς και ένα δείκτη προς τον successor του.

```
#define successor finger[1].node

// node n joins the network;
// n' is an arbitrary node in the network
n.join(n')
  if (n')
    init_finger_table(n');
    update_others();
    // move keys in (predecessor, n] from successor
  else // n is the only node in the network
    for i = 1 to m
      finger[i].node = n;
    predecessor = n;

// initialize finger table of local node;
// n' is an arbitrary node already in the network
n.init_finger_table(n')
  finger[1].node = n'.find_successor(finger[1].start);
  predecessor = successor.predecessor;
  successor.predecessor = n;
  for i = 1 to m - 1
    if (finger[i + 1].start ∈ [n, finger[i].node))
      finger[i + 1].node = finger[i].node;
    else
      finger[i + 1].node =
          n'.find_successor(finger[i + 1].start);


// update all nodes whose finger
// tables should refer to n
n.update_others()
  for i = 1 to m
    // find last node p whose i^th finger might be n
    p = find_predecessor(n - 2^(i-1));
    p.update_finger_table(n, i);

// if s is i^th finger of n, update n's finger table with s
n.update_finger_table(s, i)
  if (s ∈ [n, finger[i].node))
    finger[i].node = s;
    p = predecessor; // get first node preceding n
    p.update_finger_table(s, i);
```

Σχήμα 3.9: Ψευδοκώδικας εισόδου κόμβου και αρχικοποίησης Finger table

Όταν ένας κόμβος ν συνδέεται στο δίκτυο τότε πρέπει να γίνονται τα ακόλουθα:

1. **Αρχικοποιουνται το finger table και ο predecessor του.** Ο κόμβος n ο οποίος επιθυμεί να συνδεθεί στο δίκτυο μαθαίνει για την ταυτότητα του predecessor του και για τα fingers μέσω ενός κόμβου n' , μέσω του οποίου βρίσκει τη πληροφορία. Ο αφελής τρόπος, δηλαδή ο κόμβος n να βρίσκει successor για κάθε finger που έχει στο finger table του καταλήγει σε μία πολυπλοκότητα $O(mlogN)$. Για να μειωθεί , ο κάθε n κόμβος ελέγχει για κάθε i, αν το i-οστό finger είναι σωστό για το $i+1finger$. Αυτή η πρακτική οδηγεί σε μείωση της πολυπλοκότητας σε $O(log2N)$. Για περαιτέρω βελτίωση, ο κάθε κόμβος που συνδέεται στο δίκτυο, ζητάει από το γείτονα του να μάθει



Σχήμα 3.10: Απεικόνιση δικτύου που βασίζεται στο πρωτόκολλο Chord

για τα περιεχόμενα του finger table του, και για τον predecessor του. Στη βάση ότι το finger table του n θα έχει πολλές ομοιότητες με του άμεσου γείτονα του, ο νεοεισελθων κόμβος το χρησιμοποιεί για να φτιάξει το δικό του. Τελικά αυτή η επιπλέον κίνηση μειώνει τη πολυπλοκότητα κατασκευής του finger table σε $O(logN)$.

2. **Ενημέρωση των finger table των υπαρχόντων κόμβων.** Η σύνδεση ενός νέου κόμβου στο δίκτυο Chord έχει ως συνέπεια απαραίτητες αλλαγές στα finger tables υπαρχοντων nodes. Στην εικόνα (;;) σε σχέση με την εικόνα (;;) έχει συνδεθεί ένας νέος κόμβος,ο 6. Σε σχέση με το (;;) τα fingers των κόμβων 0,1,3 πρέπει να ενημερωθούν με τη προσθήκη του 6 στο δίκτυο. Ο κόμβος n θα γίνει το ι-οστό finger του κόμβου p αν και μόνο αν ο p προηγείται του ν κατά τουλαχιστον $2i-1$ και το ι-οστό finger του p διαδέχεται το n. Ο αλγόριθμος που φαίνεται στον ψευδοκώδικα επιβάλλει την προσπέλαση του chord σε ωρολογιακή φορά ξεκινώντας από το i-οστό finger του κόμβου n μέχρις ότου βρεθεί ο κόμβος του οποίου το i-οστό finger είναι predecessor του n.

3. **Μεταφορά των δεδομένων.** Εφόσον έχουν γίνει τα παραπάνω 2 βήματα και έχει ενταχθεί ο νεοεισελθών κόμβος στο Chord, τέλος χρειάζεται μεταφερθεί η ευθύνη στο νεοεισελθών κόμβο για όλα τα κλειδια για τα οποία είναι υπεύθυνος. Πρακτικά σημαίνει να του λάβει πληροφορίες από τον successor του για τα κλειδια, αφού μόνο αυτός μπορεί να έχει αποθηκεύσει κλειδιά τα οποία τον αφορούν.

```
n.join(n')
    predecessor = nil;
    successor = n'.find_successor(n);

// periodically verify n's immediate successor,
// and tell the successor about n.
n.stabilize()
    x = successor.predecessor;
    if (x ∈ (n, successor))
        successor = x;
    successor.notify(n);

// n' thinks it might be our predecessor.
n.notify(n')
    if (predecessor is nil or n' ∈ (predecessor, n))
        predecessor = n';

// periodically refresh finger table entries.
n.fix_fingers()
    i = random index > 1 into finger[];
    finger[i].node = find_successor(finger[i].start);
```

Σχήμα 3.11: Ψευδοκώδικας για τη διαδικασία Stabilize

Stabilization

Ο αλγόριθμος που ακολουθείται για την σύνδεση ενός νέου κόμβου στο Chord, προσπαθεί να ανανεώσει "επίμονα" τα finger tables του κάθε κόμβου με κάθε νέο συμβάν. Στη πραγματική ζωή όμως μπορεί να υπάρχουν σύγχρονα συμβάντα όπως πολλές συνδέσεις ή συνδέσεις και αποχωρήσεις και ο αλγόριθμος που περιγράφεται παραπάνω μπορεί να αποτυγχάνει σε κάποιες περιπτώσεις η να επιβαρύνει σημαντικά το δίκτυο. Για να εξασφαλιστεί η δυνατότητα σωστής αναζήτησης, τρέχει περιοδικά στο παρασκήνιο το πρωτόκολλο σταθεροποίησης (stabilization protocol) του οποίου στόχος είναι η αναζήτηση τυχόν αλλαγών και διόρθωση του finger table του κάθε κόμβου. Αναλυτικότερα στη μέθοδο stabilize() ο κόμβος ν ρωτάει τον συςεσσορ για τον predecessor p και αποφασίζει αν ο p θα έπρεπε να είναι ο successor του n πρέπει να γίνει αυτή η αλλαγή τότε η μέθοδος $notify()$ "ειδοποιεί" τον successor του n ότι πρέπει να ορίσει το n ως predecessor του. Περιοδικά τρέχει η $fixfinger()$ που ανανεώνει τα finger tables.

Περαιτέρω Βελτιώσεις

Τροποποιήσεις του βασικής ιδέας του πρωτοκόλλου του Chord έχουν προταθεί για την βελτίωση τόσο της απόκρισης όσο και την αποφυγή απώλειας δεδομένων. Μια τέτοια πρόταση είναι η προσθήκη replicas στο Chord. Με την προσθήκη replicas εννοείται η διαδικασία replication όλου του περιεχομένου ενός κόμβους σε έναν άλλο.

3.5 Byzantine's Fault Tolerance

Το πρόβλημα της δημιουργίας ενός πλήρως κατανεμημένου και ταυτόχρονα έμπιστου συστήματος δεν είναι καινούργιο στη επιστήμη των υπολογιστών [42] . Στα κατανεμημένα συ-

στήματα η επιβολή εμπιστοσύνης δημιουργεί την ανάγκη μελέτης των ανοχών στα σφάλματα. Φανταστείτε ένα υπολογιστικό σύστημα με διανεμημένα στοιχεία τα οποία πρέπει να ανταλλάσουν πληροφορίες μεταξύ τους, αλλά αυτές μπορεί να μην είναι ακριβείς (ή και να μην υπάρχουν) λόγω τεχνικών αστοχιών.

Φανταζόμαστε αρκετά τμήματα του Βυζαντινού στρατού που έχουν κατασκηνώσει εξω από τα τείχη της εχθρικής πόλης. Κάθε τμήμα διοικείται από τον δικό του στρατηγό, οι οποίοι μπορούν να επικοινωνήσουν μεταξύ του μόνο με αγγελιαφόρους.

Μετά την παρακολούθηση του εχθρού πρέπει να αποφασίσουν σε ένα κοινό σχέδιο δράσης.

Όμως κάποιοι από τους στρατηγούς μπορεί να είναι προδότες και να προσπαθούν να εμποδίσουν τους άλλους (νομοταγείς) στρατηγούς να έρθουν σε συμφωνία, Οι στρατηγοί πρέπει να έχουν έναν αλγόριθμο που να εγγυάται ότι A - όλοι οι νομοταγείς αποφασίζουν για το ίδιο σχέδιο δράσης B - ο μικρός αριθμός των προδοτών δεν θα μπορούν να προκαλέσουν την υιοθέτηση από τους νομοταγείς στρατηγούς ενός κακού σχεδίου δράσης'

Το πρόβλημα που περιγράφεται, αν και αναφέρεται σε σε Βυζαντινούς στρατηγούς συνδέεται άμεσα με τα κατανεμημένα συστήματα. Ουσιαστικά προτείνει μία λύση - πρωτόκολλο συμφωνίας για τα συστήματα, στα οποία μεταξύ των συμμετεχόντων διεργασιών δεν υπάρχει εμπιστοσύνη.

Σε αυτό το σημείο δεν θα γίνει ανάλυση του αλγορίθμου που προτάθηκε απο τον Lamport ωστόσο πρέπει να σημειωθεί ότι τα κατανεμημένα συστήματα πρέπει να είναι Byzantine Fault Tolerant ώστε να εξασφαλίζουν στους χρήστες τους την απαιτούμενη ασφάλεια. Ετσι το σύστημα θα μπορεί να παρέχει σωστά τις υπηρεσίες που προορίζεται να παρέχει, υποθέτοντας ότι συνεχίζουν να υπάρχουν μέσα σε αυτό τα απαραίτητα λειτουργικά στοιχεία για να το εξασφαλίσουν αυτό.

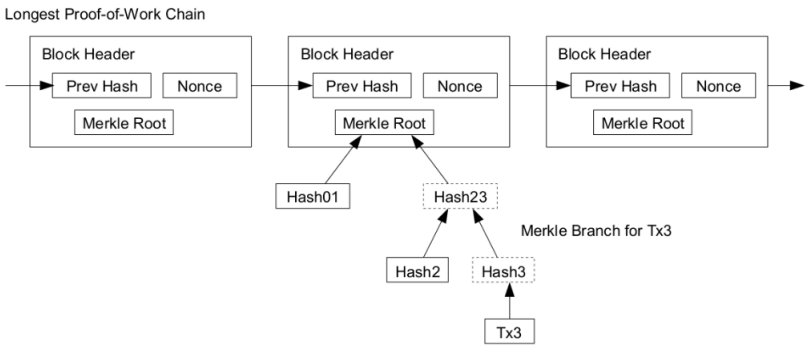
3.6 Αρχιτεκτονική και βασικά χαρακτηριστικά ενός κρυπτονομίσματος με βάση τη τεχνολογία του Blockchain

Σε αυτό το εδάφιο θα περιγραφεί εποπτικά η διαδικασία λειτουργίας του Bitcoin, το οποίο αποτελεί μία από τις piogeneric υλοποιήσεις blockchain και περιέχει όλες τις έννοιες και βασικές λειτουργίες που είναι απαραίτητες για τη διπλωματική εργασία.

Προηγουμένως αναφέρθηκαν αποσπασματικά έννοιες και ορισμοί που αφορούν τα δομικά στοιχεία ενός blockchain. Παρακάτω υπάρχει μία πιο ενδελεχής ανάλυση των εννοιών αυτών, καθώς και πως χρησιμοποιούνται σε μία πραγματική υλοποίηση.

Block

Το block σαν δομική μονάδα, απαραίτητη για τη ύπαρξη του blockchain περιέχει σαν πληροφορία. 3.13



Σχήμα 3.12: Απεικόνιση κομματιού του Blockchain

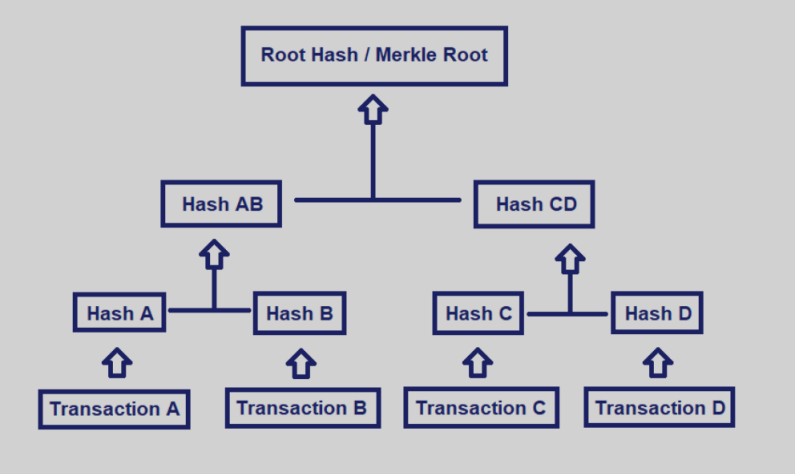
Πεδίο	Περιγραφή
Version	Η έκδοση του Block
Previous block Hash	Η κατακερματισμένη τιμή του προηγούμενου block στην αλυσίδα. Αυτό είναι το στοιχείο που ¨δένει" την αλυσίδα μαζί.
Merkle-root	Όλες οι συναλλαγές του block, κατακερματισμένες. Ουσιαστικά είναι μία περίληψη όλων των συναλλαγών που περιέχονται στο block
Time	Είναι ο χρόνος δημιουργίας του συγκεκριμένου block (χρονοσήμανση) σε χρόνο Unix.
Nonce	Η απόδειξη εργασίας για τη δημιουργία του συγκεκριμένου block. Στη περίπτωση του bitcoin είναι η λύση ενός μεγάλης πολυπλοκότητας προβλήματος.
Target	Ορίζει τη δυσκολία του προβλήματος προς επίλυση.

Σχήμα 3.13: Βασικά πεδία ενός block

Τα Merkle-trees είναι δυαδικά δένδρα που περιέχουν κατακερματισμένες τιμές, στα οποία κάθε κόμβος έχει το πολύ 2 "παιδιά". Κάθε κόμβος-φύλλο περιέχει δεδομένα συναλλαγών ενώ κάθε άλλος κόμβος περιέχει concatenated κατακερματισμένη τιμή των 2 παιδιών του.
Για παράδειγμα στην εικόνα 3.14 έχουμε 4 συναλλαγές. Κάθε μια από αυτές τις συναλλαγές έχει μία τιμή κατακερματισμού η οποία αποθηκεύεται στην σε κάθε κόμβο-φύλλο παράγοντας τους κόμβους Hash A, Hash B, Hash C, Hash D αντίστοιχα. Από τους Hash A- Hash B προκύπτει το Hash AB και αντίστοιχα από τους Hash AB, Hash CD προκύπτει η ρίζα του δέντρου.

Η ρίζα του merkle-tree αποτελεί μια περίληψη των συναλλαγών του block, αποθηκεύεται στο block-header και διασφαλίζει την ακεραιότητα των δεδομένων. Αν η σειρά των συναλλαγών η το περιεχόμενο μιας συναλλαγής αλλάξει τότε αλλάζει και η τιμή της ρίζας.

Η χρησιμοποίηση των merkle-trees παρέχει έναν γρήγορο και αξιόπιστο τρόπο ελέγχου



Σχήμα 3.14: Απεικόνιση Merkle tree

για το αν μία συναλλαγή περιέχεται η όχι σε ένα block. Επίσης δεν απαιτούν πολύ χώρο για την αποθήκευση των δεδομένων και η διαχείριση τους στα πλαίσια μιας εφαρμογής δεν απαιτεί μεταφορά μεγάλου όγκου δεδομένων στο δίκτυο.

Το Nonce 4-bytes είναι το πεδίο το οποίο συμπληρώνεται από τους miners του δικτύου και πρέπει να είναι τέτοιο ώστε το η τιμή κατακερματισμού του block να είναι μικρότερη από το πεδίο target 64-bytes. Ο κανόνας είναι ότι όσο πιο μικρό το πεδίο target, τόσο πιο δύσκολη είναι η εύρεση του κατάλληλου Nonce και κατά συνέπεια της δημιουργίας ενός νέου block.

Η στιγμή της εύρεσης του κατάλληλου Nonce συμπληρώνεται στο πεδίο Time σε Unix-timestamp format.

Consensus Protocol- Proof of Work

Το PoW περιγράφει τη διαδικασία κατα την οποία ο κάθε κόμβος που συμμετέχει στη διαδικασία δημιουργίας block (mining), δεδομένου κάποιων συναλλαγών, δοκιμάζει την τιμή του nonce ώστε να "πετύχει" μία κατακερματισμένη τιμή μικρότερη από το target. Κάθε miner χρησιμοποιώντας μία συγκεκριμένη συνάρτηση κατακερματισμού με input τα πεδία του block, αλλάζει συνεχώς την τιμή του nonce, γεγονός που οδηγεί σε μια τελείως διαφορετική τιμή κατακερματισμού κάθε φορά. Είναι σαν μια λοταρία στην οποία ο νικητής είναι ο πρώτος που θα βρει το κατάλληλο αποτέλεσμα, ενώ μέχρι να βρεθεί ο κάθε κόμβος ξεχωριστά δοκιμάζει συνεχώς νέα νονςε. Κάθε miner που επιτυχώς καταφέρνει τελικά να δημιουργήσει ένα block επιβραβεύεται για τον "κόπο" του, υπό τη μορφή χρηματικών μονάδων του δικτύου.

Process

Ας υποθέσουμε ότι υπάρχει το δίκτυο στο οποίο είναι συνδεδεμένοι ένα πλήθος ομότιμων κόμβων. Τα βήματα που ακολουθούνται είναι τα εξής:

1. Ολες οι συναλλαγές γίνονται broadcast στο δίκτυο.

2. Κάθε miner "συλλέγει" τις συναλλαγές τους δικτύου και τις τοποθετεί σε block.

3. Κάθε μινερ εκτελεί τη διαδικασία PoW.

4. Μόλις κάποιος κόμβος βρει τη λύση του PoW τότε κάνει broadcast το block στο δίκτυο.

5. Το προτεινόμενο block ελέγχεται ως προς τα transactions που περιέχει και την ακεραιότητα των δεδομένων του.

6. Σε περίπτωση που το προτεινόμενο block κριθεί έγκυρο, εντάσσεται το block στην αλυσίδα και εκκινείται ξανά η διαδικασία από το βήμα 1) για το επόμενο κομμάτι της αλυσίδας.

Κεφάλαιο 4

Ανάλυση απαιτήσεων προτεινούμενου συστήματος

4.1 Εισαγωγή

Ακολουθεί ο λόγος της προτεινόμενης ιδέας που δικαιολογεί την προσπάθεια που έγινε. Επίσης σε αυτό το κεφάλαιο θα καταγραφούν λεπτομέρειες που αφορούν τον σκοπό της συγκεκριμένης πρότασης, τις παραδοχές που έγιναν καθώς και τις λειτουργικές και μη λειτουργικές απαιτήσεις του συστήματος. Η διαδικασία αυτή αποτελεί μέρος της προεργασίας που είναι απαραίτητη για την επεξήγηση και θεμελίωση μιας ιδέας προς υλοποίηση. Σε αυτο το στάδιο καθορίζονται με σαφήνεια, ακρίβεια και πληρότητα οι λειτουργίες τις οποίες το λογισμικό πρεπει να υλοποιήσει ώστε να εξυπηρετήσει τις ανάγκες που επιχειρεί να ικανοποιήσει καθώς και να "εισάγει μια διαφορετική λογική που στόχο έχει την επίλυση των προβλημάτων που εντοπίζονται στις μέχρι τώρα πρακτικές.

4.2 Αφορμή

Σε προηγούμενα κεφάλαια αναλύθηκαν η βασική αρχιτεκτονική λειτουργίας ενός κρυπτονομίσματος, οι περιπτώσεις χρήσης της τεχνολογίας καθώς και κάποια προβλήματα ή παρεξηγήσεις που αφορούν τη συγκεκριμένη τεχνολογία. Ο προσωπικός προβληματισμός σχετικά με την τεχνολογία του blockchain και των κρυπτονομισμάτων είναι ότι η πλειοψηφία των υλοποιήσεων ακολουθούν κάποιους βασικούς κανόνες ως προς την διασύνδεση των κόμβων, το πρωτόκολλο συμφωνίας για τη δημιουργία block. Αυτές οι επιλογές προφανώς αποσκοπούν στο να κατοχυρώσουν την ασφάλεια σε ένα ελεύθερο, αποκεντρωμένο και χωρίς εμπιστοσύνη μεταξύ των συμμετεχόντων, δίκτυο. Ενώ στην πράξη προσφέρουν μία λύση για τα παραπάνω, τείνουν να καταναλώνουν πολύ ενέργεια σε "άχρηστους" υπολογισμούς αφού βασίζονται συχνά σε ενεργοβόρα πρωτόκολλα Consensus μεταξύ των κόμβων και οδηγούν στη διάδοση πολλών επαναλαμβανόμενων μηνυμάτων λόγω της δομής τους. Παράλληλα μεγάλο ζήτημα αποτελεί και ο καταναλισκόμενος αποθηκευτικός χώρος αφού σε μία συνηθισμένη υλοποίηση κρυπτο-

νομίσματος, για να συμμετάσχει ένας κόμβος θα πρέπει να συγχρονίσει τοπικά το blockchain του δικτύου που έχει δημιουργηθεί. Μπορεί στην παρούσα φάση να μην αποτελεί πρόβλημα, ωστόσο η μεγάλη αύξηση της χρήσης των δημοφιλών κρυπτονομισμάτων σε συνδυασμό με την συνεχή προσθήκη νέων λειτουργικοτήτων (βλ. Smart contracts, distributed-apps) έχει ως αποτέλεσμα την ανάγκη αποθηκευτικού χώρου της τάξεως εκατοντάδων Gigabytes, μεγαλύτερης υπολογιστικής δύναμης κλπ, γεγονός που αποκλείει τον μέσο κάτοχο υπολογιστή. Τα βασικά ζητήματα είναι η μείωση των μηνυμάτων που διαδίδονται, η αύξηση του throughput και της κλιμακωσιμότητας του δικτύου υπό την έννοια της κατανεμημένης φύσης του. Τέλος, σημαντικό είναι η θεμελίωση μιας διαφορετικής προσέγγισης που μελλοντικά υπάρχει η δυνατότητα να τροποποιηθεί και χρησιμοποιηθεί για να ξεφύγουμε από τη νόρμα. Γύρω από αυτούς τους άξονες κινείται η πρόταση που αναλύεται παρακάτω.

4.3 Σκοπός του συστήματος

Σκοπός του συστήματος είναι η παροχή μιας αποκεντρωμένης πλατφόρμας που θα δίνει τη δυνατότητα συναλλαγών μεταξύ των υπαρχόντων λογαριασμών του δικτύου. Με άλλα λόγια η δημιουργία ενός συστήματος ηλεκτρονικών πληρωμών, στο οποίο να μην υπάρχει μεσολαβητής η κεντρική εξουσία, αλλά να η λειτουργία και η ασφάλεια της πλατφόρμας να ρυθμίζεται από τους συμμετέχοντες της. Φυσικά σαν ιδέα έχει προταθεί και υλοποιηθεί σε πολλαπλές μορφές για εμπορικούς και μη εμπορικούς σκοπούς. Αυτό που στοχεύουμε σε αυτη την περίπτωση είναι η διατήρηση της αναφερόμενης βασικής λειτουργικότητας σε συνδυασμό με τη μείωση της καταναλισκόμενης ενέργειας από το σύνολο του δικτύου και τη μεγάλη κλιμακωσιμότητα του και τη ελαχιστοποίηση τόσο της χρησιμοποιούμενης αποθηκευτικού χώρου και υπολογιστικής ισχύος.

4.3.1 Χρήστες

Όλοι οι χρήστες της εφαρμογής έχουν τα ίδια δικαιώματα. Αυτό σημαίνει ότι κάθε συμμετέχων στο δίκτυο έχει το δικό του λογαριασμό, έχει τη δυνατότητα να πραγματοποιήσει συναλλαγές, να δεχθεί συναλλαγές, να συμμετέχει στη διαδικασία επικύρωσης block καθώς και να αποθηκεύει κομμάτι του blockchain του συστήματος.

4.3.2 Μη Λειτουργικές απαιτήσεις συστήματος

Δεδομένου ότι η υλοποίηση της εφαρμογής βασίζεται στην τεχνολογία του blockchain, και έχει ώς στόχο τις συναλλαγές κρυπτονομισματων μεταξύ χρηστών υπάρχουν οι εξής λειτουργικές απαιτήσεις

Ασφάλεια- Ακεραιότητα: Δεδομένου ότι στην εφαρμογή αυτή πραγματοποιούνται οικονομικές συναλλαγές θα πρέπει να υπάρχει υψηλό επίπεδο ασφάλειας. Τα χρήματα - κρυπτονομίσματα των χρηστών θα πρέπει να είναι ασφαλή απέναντι σε επιθέσεις. Οι συναλλαγές μεταξύ των χρηστών πρέπει να είναι ασφαλείς και να διασφαλίζεται η ακεραιότητα τους

Ευελιξία Η εφαρμογή να μπορεί εύκολα να επεκταθεί χωρίς να απαιτούνται πολλοί επιπλέον πόροι. (scalability) Η υλοποίηση να είναι ευέλικτη ώστε μελλοντικά να μπορεί να φιλοξενήσει πιθανές νέες λειτουργικότητες (π.χ. Έξυπνα συμβόλαια)

4.3.3 Λειτουργικές απαιτήσεις συστήματος

Οι λειτουργικές απαιτήσεις εξασφαλίζουν ότι η εφαρμογή θα έχει τα ζητούμενα χαρακτηριστικά και πως θα παρέχει στους χρήστες της τις κατάλληλες λειτουργίες. Οι λειτουργικές απαιτήσεις λοιπόν της εφαρμογής είναι:

- Η εφαρμογή να συνδέεται με το δίκτυο της πλατφόρμας
- Η εφαρμογή να έχει τη δυνατότητα συγχρονισμού της νεότερης έκδοσης του blockchain
- Να μην υπάρχει συμβατικό λογιν, αλλά να γίνεται διαπίστευση του κάθε χρήστη μόνο μέσω των εργαλείων που προσφέρει η Εφαρμογή.
- Να μην ζητείται ποτέ και από κανέναν χρήστη το ιδιωτικό κλειδί του λογαριασμού του.
- Να δίνεται η δυνατότητα στους λογαριασμούς/χρήστες να πραγματοποιούν συναλλαγές.
- Να δίνεται η δυνατότητα στους λογαριασμούς/χρήστες να εξερευνούν το blockchain υπο την έννοια ότι είναι δημόσιο.
- Μελλοντικά να μπορεί ο κάθε λογαριασμός/χρήστης να υλοποιήσει ένα έξυπνο συμβόλαιο.

4.4 Πρόταση

Η προτεινόμενη ιδέα συνοπτικά έχει ως εξής: Φανταζόμαστε μια πλατφόρμα όπου οι κόμβοι θα συνδέονται διαδικτυακά σε "διάταξη" Chord. Κάθε κόμβος θα διατηρεί ένα μέρος της συνολικής πληροφορίας του blockchain, και όλοι οι συνδεδεμένοι κόμβοι μαζί συνθέτουν τη συνολική πληροφορία του blockchain. Οπως είναι προφανές, πλέον ο κάθε κόμβος δεν θα διατηρεί ένα τοπικό συνολικό αντίγραφο του blockchain, αλλά ένα μικρό κομμάτι του ή αλλιώς διάσπαρτα blocks τα οποία δεν είναι απαραίτητο να συμπληρώνουν μια μεμονωμένη ακολουθία της αλυσίδας. Αυτή η τυχαιότητα στον τρόπο αποθήκευσης αποτελεί και μία επιπλέον δικλείδα ασφαλείας στην περίπτωση προσπάθειας ελέγχου του δικτύου καθώς και βοηθάει στον αποκεντρωμένο χαρακτήρα της πλατφόρμας. Υπό αυτή την έννοια, τελικά δεν παράγεται ένα blockchain με την αυστηρή έννοια του όρου, αλλά ένα distributed blockchain με βάση συγκεκριμένους κανόνες και έτσι θα αναφέρεται από εδώ και πέρα στα πλαίσια της πρότασης. Γίνεται αντιληπτό ότι θεμελιώδης μονάδα λειτουργίας είναι το block μέσα στο οποίο καταγράφονται οι συναλλαγές. Τελικά λοιπόν για να αναπαραχθεί η σειρά δημιουργίας του distributed blockchain χρειάζεται η προσπέλαση σε όλους τους κόμβους του δικτύου. Παρακάτω αναλύονται οι 3 βασικοί άξονες για τη λειτουργία του δικτύου του κρυπτονομίσματος.

4.4.1 Πρωτόκολλο διασύνδεσης των κόμβων του δικτύου.

Η βάση του πρωτοκόλλου διασύνδεσης των κόμβων μεταξύ τους είναι το Chord που αναλύθηκε νωρίτερα. Κατά συνέπεια οι λειτουργίες που αναφέρονται στη βιβλιογραφία και αναλύθηκαν παραπάνω, αποτελούν μέρος της πρότασης. Σε αυτό το σημείο θα περιγραφούν οι βασικές διαδικασίες που οφείλουν να ακολουθούνται.

- Κάθε κόμβος έχει ένα μοναδικό αναγνωριστικό id το οποίο προκύπτει απο την κατακερματισμένη συμβολοσειρά του λογαριασμού του.

- Κάθε κόμβος συνδέεται με 2 άλλους κόμβους άμεσα, με τον προηγούμενο του και με τον επόμενο του. Με την έννοια προηγούμενος εννοείται ο κόμβος ο οποίος έχει το αμέσως μικρότερο node-id απο τον αναφερόμενο κόμβο και με την έννοια επόμενος εννοείται ο κόμβος ο οποίος έχει το αμέσως μεγαλύτερο node-id απο τον αναφερόμενο κόμβο.

- Οπως και στο πρωτόκολλο Chord κάθε κόμβος έχει finger-table το οποίο ανανεώνεται όπως περιγράφεται.

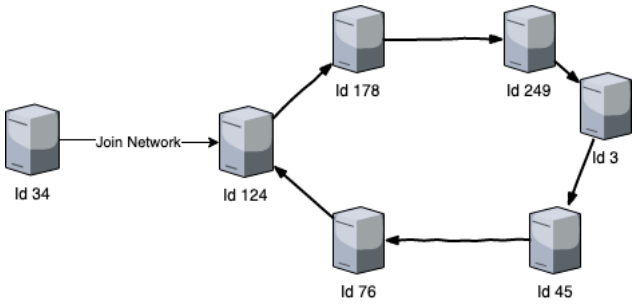
- Στο δίκτυο υπάρχει παράγοντας αναπαραγωγής (replication factor) σχετικά με τα δεδομένα ενός κόμβου. Αν για παράδειγμα το replication factor έχει οριστεί ο αριθμός 10 αυτό συνεπάγεται ότι τα δεδομένα κάθε κόμβου θα αντιγράφονται και στους 9 επόμενους κόμβους του στο Chord. Αυτή η ιδιότητα του δικτύου παίζει σημαντικό ρόλο στη διαδικασία δημιουργίας νέου block από τους κόμβους.

Είσοδος ενός κόμβου στο δίκτυο

Ανεξαρτήτως αν ο κόμβος που θέλει να συνδεθεί στο δίκτυο έχει ξαναυπάρξει η όχι ακολουθείται πάντα η ίδια διαδικασία. Ο κόμβος που θέλει να εισέλθει, απευθύνεται σε έναν τυχαίο κόμβο που είναι ενεργός στο δίκτυο ο οποίος θα τον βοηθήσει να βρεθεί στην προβλεπόμενη γι αυτόν θέση στο Chord. Μόλις βρεθεί η σωστή θέση για τον κόμβο αλλάζει η διάταξη στο συγκεκριμένο κομμάτι του Chord.

Ενα παράδειγμα εισόδου ενός κόμβου στο δίκτυο:

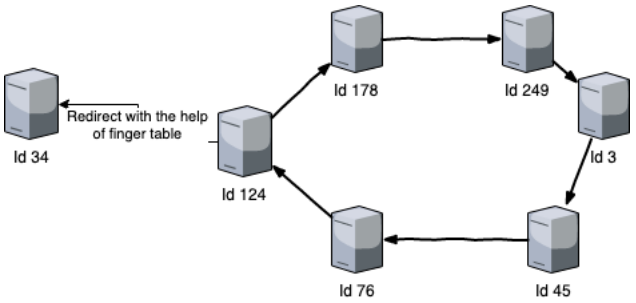
Βήμα 1



Σχήμα 4.1: Αίτημα γι είσοδο του κόμβου στο δίκτυο

Ο κόμβος του οποίου το unique id ισούται με 34 επικοινωνεί με τον κόμβο που έχει unique id 124 με σκοπό να ενταχθεί και αυτός στο δίκτυο. Ο κόμβος με id 124 αναγνωρίζει ότι δεν είναι successor του κόμβου 34 αφού ο predecessor του κόμβου 124 είναι ο κόμβος 76.

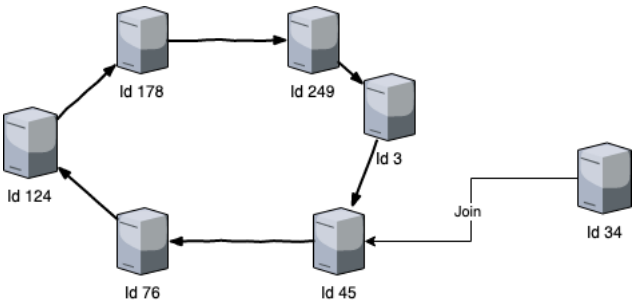
Βήμα 2



Σχήμα 4.2: Απάντηση - ανακατεύθυνση του κόμβου στη σωστή τοπολογικά θέση

Ο κόμβος 34 ανακατευθύνεται από τον κόμβο 124 με τη βοήθεια finger table και αν χρειαστεί με τη βοήθεια και άλλων κόμβων, ώστε τελικά να συνδεθεί στη λάβει τη σωστή θέση στο Chord.

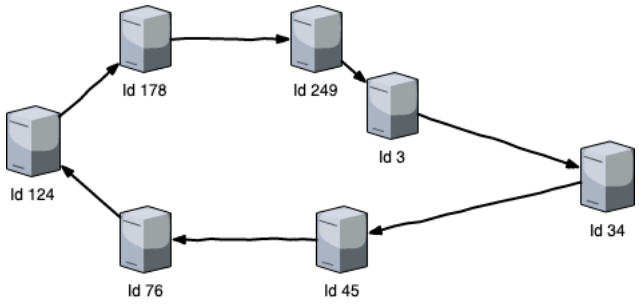
Βήμα 3



Σχήμα 4.3: Εύρεση του successor του κόμβου προς ένταξη στο δίκτυο.

Ο κόμβος 34 βρίσκει τον successor του, δηλαδή τον κόμβο 45 και εκκινείται η διαδικασία για την αλλαγή τοπολογίας του συγκεκριμένου κομματιού του Chord.

Βήμα 4



Σχήμα 4.4: Αλλαγή της τοπικής τοπολογίας για την ένταξη του νεοεισελθέντος κόμβου.

Πρέπει να αλλάξει η τοπολογία του συγκεκριμένου κομματιού του δικτύου αφού πλέον ο κόμβος 34 είναι predecessor του 45 και συςσεσσορ του 3. Κατά συνέπεια η σύνδεση μεταξύ των 3 και 45 διακόπτεται. Ο 3 ορίζει πλέον ως successor του τον 34 και αντίστοιχα ο 34 ορίζει σαν successor του τον 45. Παράλληλα διαδίδεται αυτή η αλλαγή στο δίκτυο ώστε να μεταβληθούν τα finger tables των υπολοίπων κόμβων αν χρειάζεται και φυσικά συγχρονίζει τα δεδομένα που πρέπει να έχει.

Αποχώρηση (οικειοθελής) απο το δίκτυο (όχι λόγω αποτυχίας κόμβου)

Η διαδικασία μιας οικειοθελούς αποχώρησης από το δίκτυο έχει τα εξής βήματα:

Βήμα 1

Ανακοίνωση από τον κόμβο προς αποχώρηση στο successor και στον predecessor του για το γεγονός της αποχώρησης

Βήμα 2

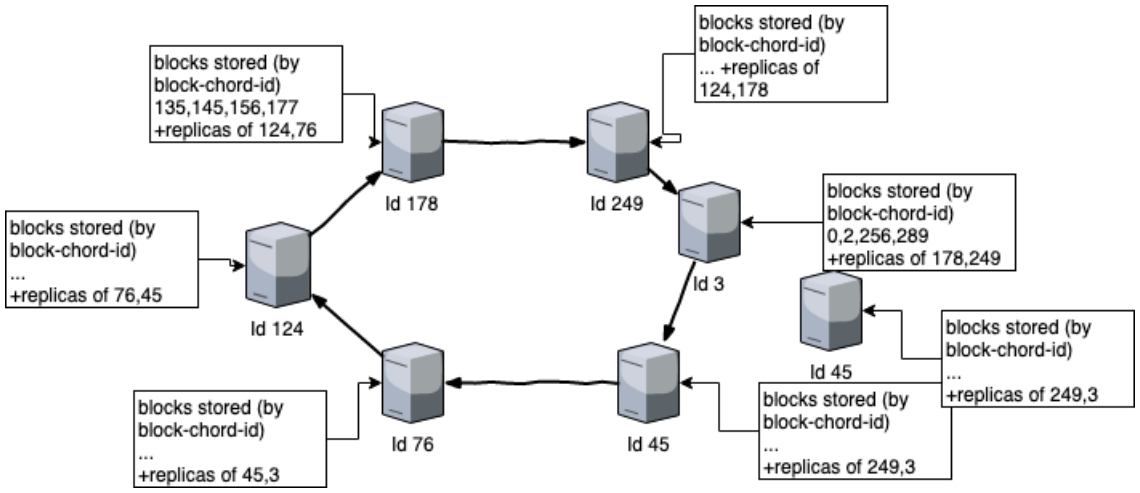
Πραγματοποίηση σύνδεσης μεταξύ τους, των successor και predecessor του κόμβου προς αποχώρηση. Φυσικά διαδίδεται και η αποχώρηση του κόμβου στο δίκτυο ώστε να ανανεωθούν, αν χρειάζεται, τα finger table των κόμβων και τα δεδομένα.

4.4.2 Διαδικασία αποθήκευσης Distributed blockchain

Το distributed blockchain που δημιουργείται συνίσταται από blocks τα οποία αποθηκεύονται στους διάφορους ενεργούς κόμβους του δικτύου. Οπως αναφέρθηκε προηγουμένως κάθε κόμβος έχει μία μοναδική ταυτότητα στο δίκτυο (node-id) που προκύπτει απο τον κατακερματισμό της συμβολοσειράς του λογαριασμού με τον οποίο έχει συνδεθεί ο κόμβος στο δίκτυο. Αν ο ίδιος φυσικός κόμβος συνδεθεί με διαφορετικό λογαριασμό στο δίκτυο, συνεπάγεται διαφορετικό node-id κόμβου και πιθανώς διαφορετική θέση στο Chord. Το node-id του κάθε κόμβου συνδέεται άμεσα με το τι πληροφορία θα αποθηκεύει. Δηλαδή ισχύει:

$$node - chord - id = hashFunction(Peerid) \tag{4.1}$$

Κάθε block που δημιουργείται έχει στο header του απαραίτητα το block hash που αποτελεί μία μοναδική συμβολοσειρά που προκύπτει από τον κατακερματισμό των πεδίων του block header συνήθως. Το πεδίο block hash δίνεται σαν είσοδος στην ίδια συνάρτηση κατακερματισμού



Σχήμα 4.5: Στιγμιότυπο του δικτύου που καταγράφει τοπολογία και αντίγραφα του κάθε κόμβου.

που χρησιμοποιείται και για το Peer Id και σαν έξοδο λαμβάνεται ένας αριθμός (block-id). Το block-id καθορίζει ποιος κόμβος είναι υπεύθυνος για την αποθήκευση του συγκεκριμένου block. Όπως γίνεται ξεκάθαρο στη βιβλιογραφία του Chord, το παραγόμενο block με block-id αποθηκεύεται στον κόμβο x όταν ισχύει :

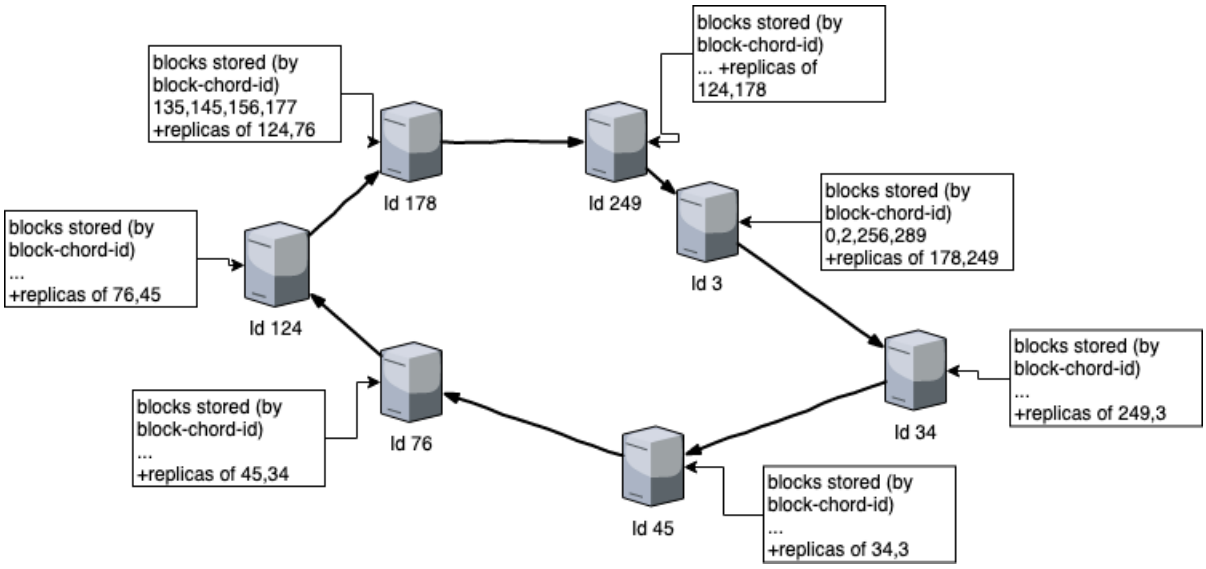
$$node - id.x = successor(block - id), block - id = hash_function(blockhash) \quad (4.2)$$

Παράλληλα είναι απαραίτητο να υπάρχει αντιγραφή των δεδομένων ενός κόμβου σε συγκεκριμένο αριθμό n άλλων κόμβων. Η επιλογή των κόμβων που αποτελούν αντίγραφα δεν γίνεται τυχαία αλλά ουσιαστικά είναι οι n επόμενοι κόμβοι στην τοπολογία του Chord. Η ύπαρξη αντιγράφων είναι απαραίτητη για 2 λόγους

- Πρώτον σε περίπτωση αποτυχίας ενός η πολλαπλών κόμβων είναι σημαντικό να μη χαθεί συνολικά η πληροφορία που περιέχεται σε αυτούς.
- Δεύτερον συμβάλει σε πολύ σημαντικό βαθμό στη διαδικασία δημιουργίας block.

Φυσικά η λειτουργία δημιουργίας αντιγράφων επιβάλλει ένα overhead στην υλοποίηση αφού προσθέτει μηνύματα στο δίκτυο καθώς και αυξάνει τον μεταδιδόμενο όγκο των δεδομένων.

Ας υποθέσουμε ότι έχουμε το προτεινόμενο σύστημα με παράγοντα αντιγραφής 3 δηλαδή τα δεδομένα ενός κόμβου, αντιγράφονται στους επόμενους 2 κόμβους στη διάταξη του Chord. Μέσω του παραδείγματος που φαίνεται στο σχήμα 4.5, γίνεται αντιληπτό ότι ο κόμβος με node-id 178 έχει αποθηκεύσει μεταξύ άλλων τα blocks ,που μετά από τον κατακερματισμό το block hash τους, έχουν block-chord-id 135,145,156,177 αφού ισχύει: $178 = successor(135)$,$178 = successor(145)$,$178 = successor(156)$,$178 = successor(177)$ Παράλληλα ο κόμβος 178 διατηρεί και το πλήρες αντίγραφο που περιέχουν οι κόμβοι 76 και 124 και αντίστοιχα το πλήρες αντίγραφο του κόμβου 178 διαδίδεται και ανανεώνεται στους κόμβους 249 και 3.



Σχήμα 4.6: Είσοδος νέου κόμβου - αλλαγή τοπολογίας και αντιγράφων.

Παρακάτω θα αναλυθούν πως αντιμετω πίζονται οι περιπτώσεις εισόδου και αποχώρησης ενός κόμβου στο δίκτυο.

Είσοδος ενός κόμβου στο δίκτυο

Με το που συνδεθεί ένας κόμβος στο δίκτυο όπως περιγράφεται στον αλγόριθμο παραπάνω, υπάρχει η προϋπόθεση του συγχρονισμού δεδομένων ώστε να μπορέσει να συμμετέχει στη διαδικασία παραγωγής block. Δηλαδή ο νεοεισελθων κόμβος εφόσον βρεθεί στη σωστή θέση, πρέπει να λάβει από τον επόμενο κόμβο στην τοπολογία, τα block για τα οποία είναι successor και φυσικά τα πλήρη αντίγραφα των κόμβων σύμφωνα με το replication factor.

Το παράδειγμα που ακολουθεί συνδέει το πρωτόκολλο διασύνδεσης με τη Διαδικασία αποθήκευσης Distributed blockchain. Εστω ότι ο κόμβος 34 εισέρχεται στο δίκτυο και ο παράγοντας αναπαραγωγής έχει οριστεί 3. Στη συνέχεια της σχήματος 4.5 φαίνεται ότι με την είσοδο του κόμβου με Id 34 στο δίκτυο 4.6 συμβαίνουν τα εξής:

- Συνδέεται ο κόμβος 34 οπότε αλλάζει την τοπολογία του δικτύου και μπαίνει ανάμεσα στους κόμβους με id 3 και 45.
- Ο κόμβος με Id 34 συγχρονίζει τα blocks για τα οποία είναι successor από τον κόμβο 46 και ο κόμβος 46 τα διαγράφει σαν τοπικά του αντίγραφα.
- Ο κόμβος 34 πλέον συγχρονίζει αντίγραφα των κόμβων 249 και 3.
- Οι κόμβοι 45 και 76 συγχρονίζουν αντίγραφα των δεδομένων του κόμβου 34.

Αποχώρηση (οικειοθελής) απο το δίκτυο (όχι λόγω αποτυχίας κόμβου)

Σε αντιδιαστολή με τη περίπτωση εισόδου στο δίκτυο, όταν ένας κόμβος ανακοινώσει την οικειοθελή αποχώρηση του από το δίκτυο πρέπει να συμβούν τα εξής:

Ο επόμενος κόμβος στην τοπολογία του Chord γίνεται successor των blocks του κόμβου που αποχωρεί.

Αλλάζουν τα αντίγραφα των γειτονικών κόμβων λόγω της αποχώρησης του κόμβου. Ο αριθμός των κόμβων που επηρεάζονται εξαρτάται απο το replication factor που εφαρμόζεται.

Σε συνέχεια του σχήματος 4.6 που αποτυπώνει την είσοδο ενός κόμβου στο δίκτυο, όταν ο κόμβος με id 34 αποχωρεί από το δίκτυο τότε επιστρέφουμε στην αρχική κατάσταση 4.5. Αυτό συνεπάγεται ότι :

- Ο κόμβος με id 45 είναι ο πλέον ο successor οσων δεδομένων κατείχε μέχρι πριν την αποχώρηση του ο κόμβος με id 34.
- Οι κόμβοι 45 και 76 συγχρονίζουν τα τοπικα αντίγραφα των κόμβων 3 και 249 και 45 και 3 αντίστοιχα.

4.4.3 Πρωτόκολλο δημιουργίας νέων block

Η διαδικασία δημιουργίας και πιστοποίησης block βασίζεται τόσο στην τοπολογία σύμφωνα με την οποία συνδέονται οι κόμβοι , καθώς και στη δημιουργία αντιγράφων της τοπολογίας αυτής. Σε αυτή την περίπτωση η δημιουργία
τλβλοςκ δεν βασίζεται σε κάποιο είδους Consensus, το οποίο περιλαμβάνει, δυνητικά, τη συμμετοχή όλων των κόμβων, αφού σε τέτοια ενέργεια απαιτείται όλοι οι κόμβοι να έχουν πλήρες αντίγραφο του blockchain ώστε να μπορούν να πιστοποιήσουν τη διαδικασία.

Η βάση της ιδέας είναι ότι το κάθε block που δημιουργείται ανά τακτά χρονικά διαστήματα αποθηκεύεται , σύμφωνα με συγκεκριμένο κριτήριο, σε συγκεκριμένο κόμβο που συμμετέχει στο δίκτυο. Παράλληλα τα δεδομένα του κάθε κόμβου αντιγράφονται και σε γειτονικούς κόμβους, σύμφωνα με το παράγοντα αντιγραφής (replication factor). Συνεπώς το πρωτόκολλο που "χτίστηκε", λαμβάνει υπόψη τόσο το πρωτόκολλο διασύνδεσης των κόμβων όσο και τον τρόπο αποθήκευσης των block, αφού αποτελούν περιοριστικούς παράγοντες.

Βασικό πρωτόκολλο

Ο κόμβος που δημιουργεί το εκάστοτε νέο block είναι εκείνος που επιλέχθηκε να αποθηκεύσει το αμέσως προηγούμενο block. Οι κόμβοι που συμμετέχουν στη διαδικασία επικύρωσης και τελικά αποδοχής του νέου block, είναι όσοι διατηρούν αντίγραφο των περιεχομένων του προαναφερθέντος κόμβου.

Υποθέσεις:

- Έστω ότι στο σύστημα που έχουμε, έχει τεθεί το replication factor των δεδομένων ίσο με $r = 3$.
- Έστω λοιπόν ότι βρισκόμαστε στο ύψος x του distributed blockchain και αρχίζει η διαδικασία δημιουργίας νέου block για το ύψος x+1.

- Εστω ακόμα ότι ο κόμβος n αποθήκευσε το block που δημιουργήθηκε και επικυρώθηκε στο ύψος x και οι κόμβοι k, l κρατούν αντίγραφο του κόμβου n.
- Εστω ότι κάθε συναλλαγή που δηλώνεται από κάποιον κόμβο γίνεται broadcast και ότι όλοι οι κόμβοι διατηρούν ένα shallow copy των υπολοίπων του κάθε λογαριασμού που υπάρχει στο δίκτυο, το οποίο συγχρονίζεται συνεχώς.

Σε κάθε περίπτωση δημιουργίας νέου βλοςκ ακολουθούνται τα εξής βήματα:

Βήμα 1- Συλλογή συναλλαγών και πρόταση νέου block.
Ο κόμβος n συλλέγει όσες συναλλαγές έχουν δηλωθεί και ,όσες πληρούν τις προϋποθέσεις, συμπεριλαμβάνονται σε ένα νέο block. Αυτό το block είναι το υποψήφιο προς αποθήκευση στο distributed blockchain, ωστόσο πρέπει να πληροί συγκεκριμένα κριτήρια με βασικότερο να μην διασπά τη συνέχεια του distributed blockchain. Το βασικό και μόνο συνδετικό στοιχείο μεταξύ των blocks στο distributed blockchain είναι το πεδίο previous block hash στο blockheader. Ετσι δημιουργείται μια η αλληλουχία των διάσπαρτων στο δίκτυο που συνιστά το distributed blockchain.

Βήμα 2- Ψήφιση απο τους Replicas.
Οι κόμβοι n,k,l ελέγχουν για την εγκυρότητα του block και ψηφίζουν γι αυτή, εφόσον είναι οι μόνοι κόμβοι που μπορούν να έχουν την πληροφορία για το αμέσως προηγούμενο block του distributed blockchain. Σημειώνεται ότι αν το replication factor του δικτύου είναι r τότε για να βρεθεί θετική η ψήφος και να εγκριθεί το προτεινόμενο block χρειάζεται τουλάχιστον τα 2/3 των συμμετεχόντων στην ψηφοφορία, δηλαδή ρ* 2/3 . Σε περίπτωση αποτυχίας η διαδικασία ξεκινά ξανά από το Βήμα 1.

Βήμα 3- Αποθήκευση του block.
Σε περίπτωση που το Βήμα 2 είναι επιτυχές τότε το επικυρωμένο block αποθηκεύεται σε συγκεκριμένο κόμβο σύμφωνα με τα κριτήρια που περιγράφονται στη Διαδικασία αποθήκευσης Distributed blockchain. Παράλληλα γίνονται και αντίγραφα ορίζονται σύμφωνα με τον παράγοντα αναπαραγωγής.

Επέκταση Βασικού πρωτοκόλλου

Το βασικό πρωτόκολλο δημιουργίας βλοςκ μπορεί να επεκταθεί με στόχο την ασφάλεια και σαν αντίκτυπο την ανταλλαγή περισσότερων μηνυμάτων στο δίκτυο καθώς και μεγαλύτερο blocktime. Η επέκταση, λοιπόν, προτείνει την επανειλημμένη εκτέλεση του Βήματος 2 του Βασικού πρωτοκόλλου για να φτάσουμε στο επιθυμητό βάθος ελέγχου. Αναλυτικότερα αν επαναληφθεί το Βήμα 2, 3 φορές τότε:

1. Την πρώτη φορά οι κόμβοι που αποθηκεύει το block ύψους x θα ψηφίσουν και θα γνωστοποιήσουν στο δίκτυο το πεδίο το block hash του block ύψους x.

2. Την δεύτερη φορα οι κόμβοι που έχουν αποθηκεύσει το block ύψους x-1 πιστοποιήσουν την υπαρξη του και θα γνωστοποιήσουν στο δίκτυο το πεδίο block hash του block ύψους x-1.

3. Την τρίτη φορα οι κόμβοι που έχουν αποθηκεύσει το block ύψους x-1 πιστοποιήσουν την υπαρξη του και θα γνωστοποιήσουν στο δίκτυο το πεδίο block hash του block ύψους x-1.

Η επέκταση στοχεύει στην αποφυγή αλλοίωσης της αλυσίδας από κακόβουλους δράστες και την εκμετάλλευση της διαδικασίας προς όφελος τους.

Κεφάλαιο 5

Εργαλεία και τεχνολογίες

Στο κεφάλαιο αυτο παρουσιάζονται τα κυριότερα εργαλεία και τεχνολογίες που χρησιμοποιήθηκαν για την ανάπτυξη της Proof of concept εφαρμογής της παρούσας διπλωματικής εργασίας. Πιο συγκεκριμένα παρουσιάζονται η γλώσσα προγραμματισμου και τα εργαλεία ανάπτυξης λογισμικού που χρησιμοποιήθηκαν.

5.1 Γλώσσα προγραμματισμού Java 8

Η Java [18] [19] είναι μια γενικού σκοπού, αντικειμενοστραφής και class-based γλώσσα προγραμματισμού, σχεδιασμένη με στόχο τη μικρότερη δυνατή εξάρτηση από το περιβάλλον υλοποίησης. Σκοπός είναι ο ζομπιλεδ κώδικας της Java να έχει τη δυνατότητα να "τρέξει" σε κάθε πλατφόρμα που την υποστηρίζει ανεξαρτήτως υλισμικού (WORA- write once, run anywhere) χωρίς την ανάγκη χρησιμοποίησης εκ νέου του compiler. Ο μεταγλωτισμένος Java κώδικας μετατρέπεται τυπικά σε bytecode που έχει τη δυνατότητα να "τρέξει" σε JVM (Java Virtual Machine). Το συντακτικό παραπέμπει στις γλώσσες C, C++ ωστόσο έχει αρκετά λιγότερες low-level επιλογές. Πρόκειται για μια πολύ δημοφιλή γλώσσα προγραμματισμού που χρησιμοποιούν εκατομμύρια προγραμματιστές.

5.1.1 Χαρακτηριστικά της Java

Portability:Για κάθε συνδυασμού υλισμικού ο γραμμένος κώδικας της Java πρέπει να τρέχει το ίδιο. Ο κώδικας της Java όταν γίνεται compile, δημιουργεί μια απεικόνιση του κώδικα που εκτελείται στο Java Virtual Machine, τον Java bytecode. Πρόκειται για κάτι ανάλογο με τις εντολές μηχανής που δεν μεταφράζονται ανάλογα τον συνδυασμό υλισμικού, αντίθετα υπάρχει μια καθολική μετάφραση έτσι ώστε ο κάθε χρήστης που έχει εγκατεστημένο κάποιο Java Runtime Environment να έχει τη δυνατότητα να τρέξει τον compiled κώδικα αμέσως.

Performance:Τα προγράμματα που είναι γραμμένα σε Java έχουν τη φήμη ότι "τρέχουν" πιο αργά και απαιτούν περισσότερη μνήμη από αυτά που έχουν αναπτυχθεί σε C++. Δεδομένου της χρήσης του Java Virtual Machine για την εκτέλεση προγραμμάτων υπάρχει ένα overhead. Η χρησιμοποίηση της τεχνικής Just-in-Time compilation (JiT) και η περαιτέρω

βελτίωση της γλώσσας τόσο σε efficiency όσο και features (ConcurrentMaps, Automatic Garbage Collection, Multi-core processing) έχει αναδείξει τη Java σε μία από τις κυρίαρχες γλώσσες ανάπτυξης λογισμικού τόσο στον εμπορικό τομέα (Android, Big Data Applications κλπ) όσο και στον επιστημονικό τομέα. Φυσικά έχει χρησιμοποιηθεί για την ανάπτυξη Cryptocurrency που υλοποιείται με τη βοήθεια της τεχνολογίας blockchain.

5.2 Maven

Το Maven [2] [1] θεωρείται ένα πολύ ισχυρό εργαλείο ανάπτυξης έργων και βασίζεται στο POM (project object model) και χρησιμοποιείται για Java Projects. Εξυπηρετεί 2 βασικούς σκοπούς: Περιγράφει πως αναπτύσσεται το λογισμικό καθώς και περιέχει τις εξαρτήσεις του λογισμικού. Περιέχει δηλαδή τις εξαρτήσεις του κώδικα σε εξωτερικά modules/components, τη σειρά με την οποία γίνονται build τα διάφορα επιμέρους κομμάτια, καθώς και οδηγίες για το compilation και το packaging του τελικού εκτελέσιμου αρχείου. Ολη η πληροφορία περιέχεται σε ένα XML αρχείο που βρίσκεται στο μέσα στο project το οποίο by-default είναι το pom.xml.

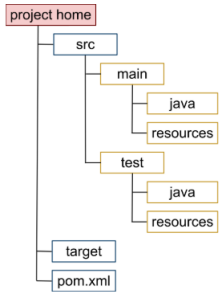
Το Maven δίνει τη δυνατότητα στο developer, έχοντας συντάξει μία φορά το pom.xml αρχείο, να μπορεί να κάνει αλλαγές στο κώδικα και τελικά να περιέχει τα ίδια dependencies , να κάνει test στον κώδικα και compile με τον ίδιο ακριβώς τρόπο.

Οποιοδήποτε external στοιχείο/βιβλιοθήκη στα πλαίσια του maven, είναι διαθέσιμο σε repositories στο διαδίκτυο

Βασικές έννοιες του Maven: Dependencies and Repositories: Dependencies είναι εξωτερικές Java βιβλιοθήκες που είναι απαραίτητες για το προτζεκτ. Repositories είναι διευθύνσεις που βρίσκονται τα dependencies. Αν οι εξαρτήσεις που περιγράφονται δεν βρεθούν τοπικά στο υπολογιστή το Maven τις κατεβάζει αυτόματα από τα online repositories και τις αποθηκεύει τοπικά για μελλοντική χρήση.

Build Life Cycles, Phases and Goals: Το build life cycle αποτελείται από μία αλληλουχία φάσεων (Phases) και κάθε φάση αποτελείται από στόχους (Goals). Αν ζητηθεί μέσω του maven η εκτέλεση ενός lifecycle τότε όλες οι φάσεις που περιέχονται σε αυτό εκτελούνται. Αντίστοιχα αν εκτελεστεί μία συγκεκριμένη φάση τότε όλοι οι στόχοι (Goals) που περιέχονται, εκτελούνται.

Build Profiles: Μέσω του Maven μπορούν να ρυθμιστούν προφίλ εκτέλεσης, που δίνει τη δυνατότητα για το build του προτζεκτ με διαφορετικές ρυθμίσεις. Για παράδειγμα μπορεί να χρειάζεται να γίνει build του project για το προσωπικό υπολογιστή του developer για έλεγχο και αργότερα σε παραγωγικό περιβάλλον ενός μεγάλου οργανισμού, και αναλόγως την περίσταση επιλέγονται άλλες ρυθμίσεις.



Σχήμα 5.1: Δομή Project Semux

Μια τυπική διάταξη ενός Maven πρότζεκτ φαίνεται στην εικόνα 5.1 .

Όνομα Διεύθυνσης	Σκοπός
project home	Περιέχει το pom.xml και τις υποδιευθύνσεις.
src/main/java	Περιέχει τον παραδοτέο Java πηγαίο κώδικα του project.
src/main/resources	Περιέχει τα παραδοτέα resources του project, όπως property files.
src/test/java	Περιέχει τον test Java πηγαίο κώδικα του project.
src/test/resources	Περιέχει τα resources του project που χρησιμοποιούνται για το δοκιμή του test κώδικα.

Σχήμα 5.2: Επεξήγηση βασικών διευθύνσεων ενός Maven project

Βασικές command-line εντολές του Maven είναι η mvn install, που πακεταρει τα dependencies, το pom.xml και το εκτελεσιμο jar σε ένα αρχείο το οποίο είναι εκτελέσιμο. Υπάρχουν και άλλες φάσεις όπως η validate, compile κλπ οι οποίες στα πλαίσια της διπλωματικής δεν ήταν απαραίτητες.

5.3 Eclipse

Το Eclipse [11] αποτελεί ένα περιβάλλον ανάπτυξης λογισμικού (Eclipse IDE) με πολλά flavors που υποστηρίζουν διάφορες γλώσσες προγραμματισμού όπως Scala,C,C++ κλπ. Το πιο ευρέως γνωστό και διαδεδομένο είναι αυτό της Java αφού είναι ένα από τα 3 καλύτερα IDE ανάπτυξης Java. Παρέχει plugins και add-ons που διευκολύνουν την συγγραφή κώδικα καθώς και την διαδικασία του debugging και testing. Παράλληλα υπάρχει integration για Maven projects. Για τις ανάγκες της εργασίας χρησιμοποιήθηκε το Eclipse IDE με Maven integration και η γλώσσα προγραμματισμού Java 8.

5.4 Semux cryptocurrency

Σχήμα 5.3

Το Semux [22] είναι μια open-source υλοποίηση [23] κρυπτονομίσματος στη γλώσσα προγραμματισμού Java που βγήκε live στο κοινό στις 20 Ιανουαρίου 2018. Πρόκειται για μία προσπάθεια δημιουργίας ενός κρυπτονομίσματος το οποίο ως στόχο να πάρει μερίδιο της αγοράς των κρυπτονομισμάτων, έχοντας να αντιμετωπίσει ισχυρούς αντιπάλους όπως το Bitcoin, Ethereum, Cardano κλπ. Το κρυπτονόμισμα βασίζεται στο Blockchain και το πρωτόκολλο συμφωνίας μεταξύ των κόμβων είναι το dPoS.

Ένα από τα κύρια χαρακτηριστικά του είναι η ανοχή σε σφάλματα.Σε fault-tolerant υπολογιστικά συστήματα και ειδικότερα στα κατανεμημένα συστήματα, το BFT (Byzantine fault tolerance) είναι το κύριο χαρακτηριστικό που προσφέρει ανοχή σε σφάλματα/προβλήματα τύπου Βυζαντινών στρατηγών (Byzantine Generals' Problem). Το Semux λοιπον χρησιμοποιεί ένα τέτοιου είδους πρωτόκολλο.

Παράλληλα, προβλέπεται η μελλοντική ενσωμάτωση και υποστήριξη έξυπνων συμβολαίων ,που μοιάζουν με αυτά του κρυπτονομίσματος Ethereum, και προς το παρόν βρίσκεται σε εξελικτικό επίπεδο.

Γενικές πληροφορίες Το νόμισμα του Semux είναι το Sem. Η ελάχιστη δυνατή υποδιαίρεση είναι 0.0000001 Sems.

5.4.1 Αρχιτεκτονική Semux

Σχήμα 5.4

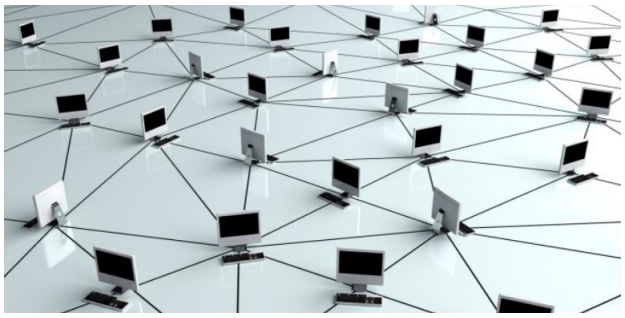
Delegated Proof Of Stake (D-PoS):
Αποτελεί τη μέθοδο συμφωνίας μεταξύ των κόμβων του δικτύου του Semux. Αυτό το είδος consensus χρησιμοποιεί τη -σε πραγματικό χρόνο- ψηφοφορία σε συνδυασμο με ένα σύστημα κοινωνικής φήμης για επιτευχθεί η συμφωνία. Σε σχέση με τα άλλα είδη consensus που χρησιμοποιούνται, θεωρείται από τα λιγότερο centralized, αφού δυνητικά συμπεριλαμβάνει στον αλγόριθμο τον κάθε κόμβο. Κάθε κάτοχος κρυπτονομίσματος μπορεί να ασκήσει σε ένα βαθμό επιρροή για το τι συμβαίνει στο δίκτυο. Παράλληλα το DPoS συμβάλει στην χαμηλή κατανάλωση ενέργειας καθώς και στην μεγάλη κλιμακωσιμότητα.
Υπάρχουν 2 είδη ρόλων στο δίκτυο: Οι κάτοχοι κρυπτονομισμάτων (token holders) και οι εκπρόσωποι (delegates). Οι εκπρόσωποι είναι ειδικοί λογαριασμοί του δικτύου οι οποίοι ψηφίζονται από τους συμμετέχοντες του δικτύου.

Το βάρος της ψήφου ενός token holder είναι ανάλογο με την περιουσία του σε κρυπτονομισμα. Είναι σημαντικό σε αυτο το πρωτόκολλο οι εκπρόσωποι να επιλέγονται με γνώμονα το καλύτερο συμφέρον του δικτύου, δεδομένου ότι αυτοί διατηρούν την ισορροπία και ρυθμίζουν τη λειτουργία του. Σε κάποιες υλοποιήσεις D-PoS, για να γίνει ένας κόμβος εκπρόσωπος οφείλει να δεσμεύσει μέρος της περιουσίας του για όσο διάστημα είναι σε αυτή τη θέση ώστε να αποθαρρυνθεί η κακόβουλη συμπεριφορά του.

Οι delegates δεν έχουν την δυνατότητα να αλλάξουν το περιεχόμενο κάποιας συναλλαγής, παρά μόνο να την αποκλείσουν από ένα block. Ωστόσο, αυτή η κίνηση είναι άνευ ουσίας αφού το επόμενο προς δημιουργία block θα συμπεριλάβει αυτές τις συναλλαγές, δίνοντας το reward για το validation τους στον επόμενο delegate. Συνεπώς θα υπάρχει μία μικρή καθυστέρηση στην πραγματοποίηση των συναλλαγών. Παράλληλα, μία τέτοια κακόβουλη ενέργεια θα οδηγούσε αναπόφευκτα στην "περιθωριοποίηση" τέτοιου είδους συμμετεχόντων.

Όπως γίνεται εμφανές λοιπόν, κάθε εκπρόσωπος έχει πιθανότητα να επικυρώσει ένα νέο block, το οποίο υπογράφεται από τον ίδιο, παίρνοντας μία ανταμοιβή για τη δουλειά του.

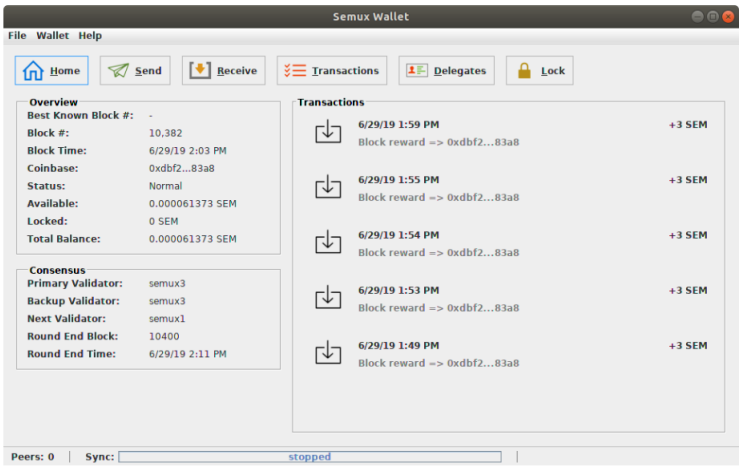
Στην περίπτωση του Semux λοιπον για να γίνει κάποιος κόμβος εκπρόσωπος είναι απαραίτητο να έχει στη πορτοφόλι του τουλάχιστον 1000 SEM συν τον φόρο που πληρώνεται για κάθε συναλλαγή. Ολοι οι εκπρόσωποι μπαίνουν σε μία λίστα, ώστε να μπορούν να ψηφιστούν για να γίνουν validators, δηλαδή για να μπορούν να επικυρώσουν νέα βλοχς. Στο Semux μπορούν να υπάρχουν ταυτόχρονα στο δίκτυο 100 validators, που μεταφράζεται στους κόμβους που έχουν ψηφιστεί περισσότερο και είναι ενεργοί. Για να είναι κάποιος validator χρειάζεται να έχει υλισμικό υψηλών επιδόσεων, περισσότερη μνήμη RAM από τον μέσο κόμβο (16GB) καθώς και γρήγορη σύνδεση στο διαδίκτυο (200Mbps).



Σχήμα 5.5

Διασύνδεση κόμβων:

Η διασύνδεση των κόμβων στο δίκτυο του Semux γίνεται μέσω ενός δικτύου ομότιμων κόμβων ή αλλιώς Peer to Peer network. Ουσιαστικά κάθε κόμβος συνδέεται με έναν αριθμό άλλων κόμβων του δικτύου και επικοινωνεί ανά τακτά χρονικα διαστήματα μαζι τους. Οι διευθύνσεις τους διατηρούνται σε μία δομή αποθήκευσης (ConcurrentHashMap) και αποθηκεύονται τόσο οι ενεργές διευθύνσεις όσο και οι διευθύνσεις των παλαιότερα συνδεδεμένων κόμβων. Ιδιαίτερη σημασία δίνεται στους κόμβους που θεωρούνται ύποπτοι οι οποίοι "μπαίνουν" σε μαύρη λίστα αν δεν πληρούν τις προϋποθέσεις του δικτύου η θεωρούνται ύποπτα τα στοιχεία τους. Το δίκτυο που δημιουργείται είναι αδόμητο, ωστόσο ο κάθε κόμβος έχει περιορισμό για τον αριθμό των ανοιχτών συνδέσεων που μπορεί να διατηρεί ταυτόχρονα, γεγονός που οδηγεί στην ομαλή κατανομή και σε λογικούς χρόνους μεταφοράς των μηνυμάτων.

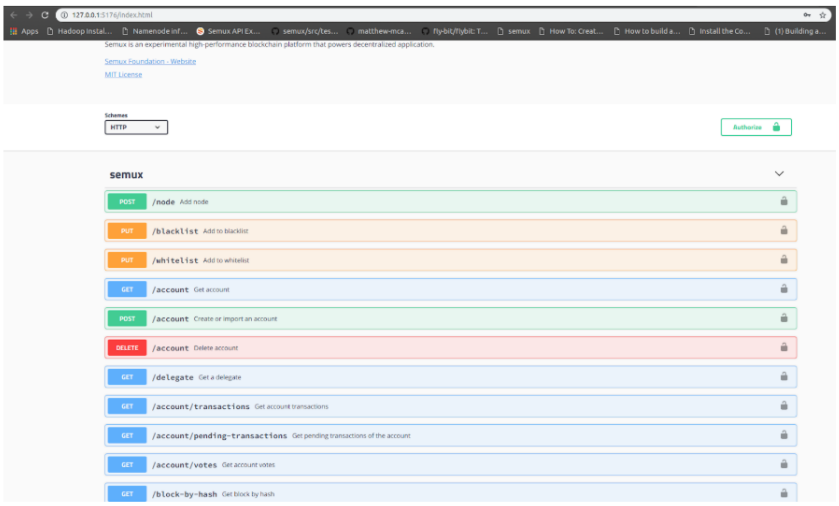


Σχήμα 5.6: Κεντρική οθόνη Semux

Γραφικό περιβάλλον:

Το Semux έχει σχεδιαστεί για να είναι φιλικό προς τον χρήστη ο οποίος δεν έχει άμεση σχέση με τον προγραμματισμό και τον διευκολύνει με το να εκθέτει γραφικό περιβάλλον, μέσω του οποίου μπορεί να εξερευνήσει ο ενδιαφερόμενος ποιοι είναι οι εκπρόσωποι του δικτύου, ποιο είναι το υπόλοιπο του, τι συναλλαγές έχουν γίνει καθώς και να κάνει και ο ίδιος συναλ-

λαγές.



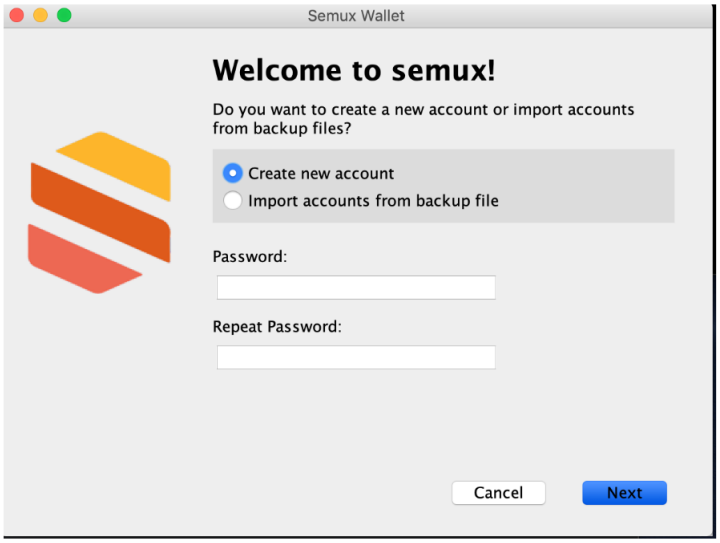
Σχήμα 5.7: Semux API Explorer

Παράλληλα εκθέτει και APIs για τον προγραμματιστή που μπορούν να χρησιμεύσουν στον έλεγχο καθώς και στην ανάπτυξη εφαρμογών οι οποίες επικοινωνούν με την συγκεκριμένη εφαρμογή.

Περιγραφή διαδικασίας σύνδεσης και συμμετοχής κόμβου στο δίκτυο.

Ας υποθέσουμε ότι ένας κόμβος λογαριασμός θέλει να συνδεθεί στο δίκτυο του Semux.

Αν είναι η πρώτη φορά που ανοίγει η εφαρμογή και ο χρήστης δεν έχει λογαριασμό (wallet) τότε δημιουργείται αυτόματα το wallet και ζητείται από τον χρήστη να βάλει τον κωδικό του. Αυτό συνεπάγεται ότι ο χρήστης πλέον κατέχει ένα μοναδικό Peer Id, το οποίο είναι το αναγνωριστικό του στοιχείο για τους άλλους χρήστες.



Σχήμα 5.8: Οθόνη σύνδεσης στο Semux

Επειτα η εφαρμογή εξερευνά το δίκτυο για άλλους Peers. Αυτό συμβαίνει μέσω ενός DNS (domain name server) που διατηρεί κάποιες διευθύνσεις γνωστών κόμβων. Με το που συνδεθεί με τουλάχιστον ένα κόμβο τότε αρχίζει να ζητάει διευθύνσεις άλλων κόμβων με τις οποίες είναι συνδεδεμένος αυτός ο κόμβος. Παράλληλα αρχίζει τον συγχρονισμό του τοπικού αντιγράφου του Blockchain μέχρις ότου φτάσει στο μέχρι τότε, πιο πρόσφατα δημιουργημένο block. Δηλαδή ζητάει από τους κόμβους με τους οποίους είναι συνδεδεμένος με τη σειρά δημιουργίας τους, τα βλοχς που έχουν οι χρήστες. Φυσικά δεν αποθηκεύει άκριτα τα blocks που λαμβάνει, αλλά κάνει ελέγχους για την εγκυρότητα τους. Ετσι λοιπον γνωστοποιείται σταδιακά η πορεία των συναλλαγών αλλά και τα υπόλοιπα όλων των λογαριασμών που έχουν συναλλαχθεί στο δίκτυο.

Τελικά όταν τελειώσει ο συγχρονισμός, ο κόμβος μπορεί, σε περίπτωση που πληρει τις προϋποθέσεις να γίνει εκπρόσωπος και δυνητικά να επικυρώσει νέα block . Για να συναλλαχθεί δεν είναι απαραίτητο να έχει ολοκληρωθεί ο συγχρονισμός, αφού η μεγάλη πλειοψηφία του δικτύου μπορεί να επιβεβαιώσει ότι είναι έγκυρες οι συναλλαγές στις οποίες συμμετέχει.

Κάθε συναλλαγή η οποία δηλώνεται από έναν κόμβο, αφού ελέγχεται η εγκυρότητα της (αν υπάρχουν οι λογαριασμοί, αν υπάρχει το υπόλοιπο κλπ), γίνεται broadcast σε όλους τους κόμβους με τους οποίους είναι συνδεδεμένος και στη συνέχεια αυτοί οι κόμβοι την κάνουν broadcast και σε άλλους. Έτσι το μήνυμα της συναλλαγής φτάνει σε κάθε κόμβο του δικτύου. Τελικά κάποιος ένας Validator επιλέγεται να δημιουργήσει το επόμενο block ο οποίος συμπεριλαμβάνει όλες τις έγκυρες συναλλαγές και τις τοποθετεί στο block, παίρνοντας την ανταμοιβή της κάθε συναλλαγής.

Consensus Protocol

Το Consensus πρωτόκολλο του Semux για τη δημιουργία νέου block διαρκεί κατά προσέγγιση 30 δευτερόλεπτα και επαναλαμβάνεται συνεχώς. Σε κάθε κύκλο οι Validators θα περάσουν 6 στάδια μέχρις ότου παραχθεί ένα νέο block. Διευκρινίζεται ότι όταν αναφερομαστε σε ύψος, εννοούμε μήκος της αλυσίδας.

Βημα 1 - *New Height* (διάρκεια 3 δευτερόλεπτα αυστηρά)

Σε αυτό το στάδιο ανακοινώνεται ότι αρχίζει ένας νέος κύκλος του πρωτοκόλλου. Το νέο ύψος της αλυσιδας αυξάνεται κατά 1 και στέλνεται ένα μήνυμα NewHeightMessage πρός όλους τους συμμετέχοντες του δικτύου.

Βημα 2 - *Propose* (διάρκεια 12 δευτερόλεπτα αυστηρά)

Αυτό το στάδιο σηματοδοτεί την πρόταση από έναν Validator για το πως θα είναι ένα νέο block. Δηλαδή συλλέγει τις διαθέσιμες συναλλαγές μέσα σε ένα βλοχκ , το οποίο το κάνει βροαδχαστ σε όλους τους validators και τίθεται υπό ψηφοφορία. Αν κάποιος από τους validators ψηφίσει αρνητικά (π χ λόγω του ότι έχει μη έγκυρες συναλλαγές ή διότι περιέχει συναλλαγές που δεν έχει άγνωστες γι αυτόν λόγω καθυστέρησης δικτύου) τότε ξαναγίνεται νέα πρόταση για block. Σημειώνεται ότι ο Validator που επιλέγεται κάθε φορά να προτείνει βλοχκ επιλέγεται τυχαία, ώστε να υπάρχει δίκαιη κατανομή των rewards των block.

Βημα 3 - Validate (διάρκεια 6 δευτερόλεπτα αυστηρά)

Σε αυτό το στάδιο γίνεται ο έλεγχος των περιεχομένων του block. Στην ουσία γίνεται η επικύρωση των συναλλαγών που περιέχονται μέσα σε ένα block, οι οποίες εκτελούνται εικονικά τοπικά σε κάθε κόμβο-validator. Αν λοιπόν ένας validator έχει λάβει την πρόταση τότε επιβεβαιώνει την εγκυρότητα των περιεχομένων της και ψηφίζει αρνητικά ή θετικά. Το μήνυμα της ψήφου γίνεται βροαδςαστ σε όλους τους ενεργούς validators του δικτύου.

Βημα 4 - Precommit (διάρκεια 6 δευτερόλεπτα αυστηρά)

Σε αυτό το στάδιο γίνεται έλεγχος από ταν κάθε validator για το αν τα 2/3 των ενεργών validator έχουν ψηφίσει θετικά στο προηγούμενο στάδιο. Αν ισχύει το παραπάνω τότε ο κάθε κόμβος κάνει βροαδςαστ μια θετική απάντηση προς όλους τους validators αλλιώς απαντά αρνητικά. Σε περίπτωση που αποτύχει αυτό το στάδιο τότε ξαναγυρνάμε στο Βήμα 2 - Propose.

Βημα 5 - Commit (διάρκεια 3 δευτερόλεπτα η λιγότερο)

Σε αυτό το βήμα οι validators κάνουν broadcast το μήνυμα ότι έχουν λάβει μήνυμα από το προηγούμενο Βήμα. Αν οι ίδιοι έχουν ήδη λάβει μήνυμα από το Βήμα 5 τότε μεταβαίνουν στο επόμενο Βήμα

Βημα 6 - Finalize (διάρκεια 3 δευτερόλεπτα η λιγότερο)

Σε αυτό το βήμα επανελέγχονται οι ψήφοι που στάλθηκαν στο Βήμα 4 για επανεπιβεβαίωση και τελικά γράφονται οι ψήφοι των validators στο βλοςκ. Στη συνέχεια το block προστίθεται στην αλυσίδα. Αν το Βήμα αυτό λήξει επιτυχώς τότε ξαναεκκινείται η διαδικασία από το Βήμα 1.

Από τη μία οι validators καθορίζουν και φροντίζουν για τη δημιουργια της αλυσίδας, από την άλλη όμως οι υπόλοιποι κόμβοι του δικτύου πρέπει να συμπεριληφθούν στη διαδικασία ενημέρωσης της αλυσίδας. Συνεπώς σε κάθε μήνυμα NewHeight (του Βήματος 1) οι “απλοί” κόμβοι προσθέτουν το νέο block στο τοπικό αντίγραφο της αλυσίδας τους.

Κεφάλαιο 6

Σχεδιασμός και υλοποίηση Συστήματος

Στο κεφάλαιο αυτό θα παρουσιαστούν λεπτομέρειες που αφορούν τον σχεδιασμό και την υλοποίηση της αποκεντρωμένης εφαρμογής που αναπτύχθηκε στα πλαίσια της παρούσας διπλωματικής εργασίας.

6.1 Εργαλεία

Χρησιμοποιήθηκαν οι εξής εκδόσεις των εργαλείων που έχουν περιγραφεί:

- Semux version 1.4.0
- JDK 1.8
- Maven 3.6.0
- Eclipse 2018-12 (4.10.0)
- Ubuntu Linux 16.04 LTS

6.2 Γιατί επιλέχθηκε το Semux·

Η Java είναι η γλώσσα προγραμματισμού που κατέχω. Το Semux είναι από τα λίγα open source κώδικα κρυπτονομίσματα των οποίων η υλοποίηση έχει γραφτεί σε Java και είναι το πιο γνωστό εξ αυτών. Παράλληλα υπάρχει μια κοινότητα που στηρίζει και συντηρεί και ανανεώνει το λογισμικό που γράφεται συνεχώς.

6.3 Δυσκολίες στην υλοποίηση

Η υλοποίηση ενός PoC (Proof of Concept) της πρότασης συνεπάγεται την τροποποίηση και μετατροπή κομμάτι του πηγαίου κώδικα της εφαρμογής του Semux.

Οι δυσκολίες που αντιμετωπίστηκαν κατά τη διάρκεια της υλοποίησης αναλύονται σε 2 άξονες: Ο πρώτος είναι η φύση του open source κώδικα. Γενικότερα σε μία ανάπτυξη open source υπάρχει συνεργασία πολλών developers, χωρίς ωστόσο να τηρούνται οι αυστηρές διαδικασίες ενός μεγάλου οργανισμού. Αυτό συνεπάγεται ότι δεν φτιάχνεται κάποιου είδους αρχείο σχεδιασμού με αυστηρές προδιαγραφές η αυστηρά πρότυπα. Δηλαδή δεν είναι δυνατόν να τηρηθεί κατά την περίοδο ανάπτυξης μοντέλα τύπου καταρράκτης αφού οι προδιαγραφές σπάνια συλλέγονται πριν απο την έναρξη του έργου. Συνεχώς παρέχονται νέες εκδόσεις οι οποίες δοκιμάζονται και ανανεώνονται με νέες λειτουργικότητες μέχρι να εκδοθεί μία σταθερή (stable) έκδοση. Στο πλαίσιο αυτό και δεδομένου ότι η εφαρμογή του Semux είναι σχετικά καινούργια στο χώρο, κατέστη απαραίτητο να επιλεχθεί η πιο σταθερή και ανανεωμένη έκδοση που παρέχεται. Ο δεύτερος είναι η μη ύπαρξη εγχειριδίων υλοποίησης και χρήσης (documentation). Η μέθοδος ανάπτυξης που ακολουθείται σε open source έργα και οι συνεχείς αλλαγές συνιστούν δύσκολη την ύπαρξη και συντήρηση εγχειριδίων που περιγράφουν το σύστημα και το πως λειτουργεί, αφού σύντομα γίνονται outdated. Για αυτό το λόγο, χρησιμοποιήθηκε reverse engineering, μια επίπονη διαδικασία, για την κατανόηση του κώδικα και των διαδικασιών με σκοπό την τροποποίηση όπου χρειαζόταν.

6.4 Παραδοχές

Η εφαρμογή ntuaSemux έχει εκπαιδευτικό χαρακτήρα και όχι εμπορικό. Συνεπώς έχουν γίνει κάποιες παραδοχές με στόχο να μειώσουν την πολυπλοκότητα της υλοποίησης, και τελικά να φανούν καλύτερα τα χαρακτηριστικά των τεχνολογιών που χρησιμοποιήθηκαν. Οι παραδοχές που έγιναν αφορούν στην σχεδιαση και στην λειτουργία του συστηματος

Το ntuaSemux σαν εφαρμογή προσφέρει ασφάλεια στις συναλλαγές. Η παραδοχή αυτή υιοθετείται από το σύνολο της κοινότητας που το χρησιμοποιεί αφού οι αλγόριθμοι κρυπτογράφησης που χρησιμοποιούνται θεωρούνται ασφαλείς. Ετσι δεν χρειάζεται μέριμνα απο μερους της εφαρμογής για επιπρόσθετη ασφάλεια.

- Δεν υπάρχουν κακόβουλοι δράστες που συνδέονται ώστε να ξεγελάσουν με κάποιο τρόπο τους υπόλοιπους συμμετέχοντες.
- Η δικτυακή απόκριση των κόμβων που συμμετέχουν δεν υπερβαίνει τα $80ms$.
- Τα δικτυακά μηνύματα τα οποία στέλνονται είναι πάντοτε επιτυχημένα και δεν υφίστανται οποιοδήποτε είδους αλλοίωση.
- Η τοπολογία του δικτύου δημιουργείται προτού αρχίσει η διαδικασία δημιουργίας block. Αυτό σημαίνει ότι αφότου συνδεθούν όλοι κόμβοι μεταξύ τους στις ¨σωστές¨ θέσεις ξεκινάει ένας προαποφασισμένος κόμβος τη δημιουργία του block με ύψος 1.
- Οι συναλλαγές που δηλώνονται από κάθε κόμβο x αφορούν τη μεταφορά μονάδων ntuaSEM από τον ίδιο τον με λογαριασμό ξ σε κάποιον υπάρχων λογαριασμό κόμβου του δικτύου.

- Οι λογαριασμοί των κόμβων του δικτύου είναι μοναδικοί.
- Κάθε συναλλαγή που δηλώνεται γίνεται broadcast σε όλους τους κόμβους του δικτύου.
- Κάθε κόμβος που υπάρχει στο δίκτυο διατηρεί ένα ρηχό αντίγραφο (shallow copy) των υπολοίπων των λογαριασμών του δικτύου.
- Δεν υπάρχει αμοιβή για την παραγωγή των blocks προς τον κόμβο που το παράγει.
- Εχουν καταχωρηθεί σε συγκεκριμένους λογαριασμούς στο genesis block μεγάλα ποσά κρυπτονομίσματος για την δυνατότητα εκτέλεσης συναλλαγών.
- Δεν εξετάζονται περιπτώσεις αποτυχιών κόμβων.
- Το replication φαστορ ισούται με το 1 δηλαδή δεν υπάρχει αντίγραφο των δεδομένων του κάθε κόμβου σε άλλο κόμβο του δικτύου.
- Υλοποιήθηκε το βασικό πρωτόκολλο δημιουργίας νέων block που περιγράφεται.

6.5 Υλοποίηση

ΓΙα το σκοπό της διπλωματικής τροποποιήθηκε ο πηγαίος κώδικας της εφαρμογής Semux. Οι αλλαγές έγιναν σε συγκεκριμένα σημεία του κώδικα ώστε να δημιουργηθεί ένα σύστημα στο οποίο:

- Οι κόμβοι θα είναι συνδεδεμένοι σε διάταξη Chord. (networking)
- Ολοι οι κόμβοι του δικτύου θα είναι ισότιμοι υπό την έννοια του ρόλου τους.
- Δεν θα αποθηκεύουν όλοι οι κόμβοι το blockchain αλλά θα αποθηκεύεται στο δίκτυο το **distributed blockchain**. (blockchain implementation + Consensus)

6.5.1 Κατακερματισμός

Για το hashing που χρησιμοποιείται στη διαδικασία της παραγωγής του block-chord-id καθώς και του node-chord-id υλοποιήθηκε μία συνάρτηση που δέχεται σαν είσοδο συμβολοσειράString το block hash και το peer-id αντίστοιχα και βγάζει σαν έξοδο έναν ακέραιο αριθμό.

Δέχεται μία συμβολοσειρά(String) παράγει έναν ακέραιο αριθμό πολλαπλασιάζοντας την εκάστοτε τιμή hash με το 31 και προσθέτοντας την τιμή του κάθε χαρακτήρα της εισόδου. Τελικά ο αριθμός που προκύπτει διαιρείται ακέραια (div) με το 10.000.000. Το εύρος του αποτελέσματος , εμπειρικά, είναι στο διάστημα (-300,300), βολικό για την χρήση που απαιτείται.

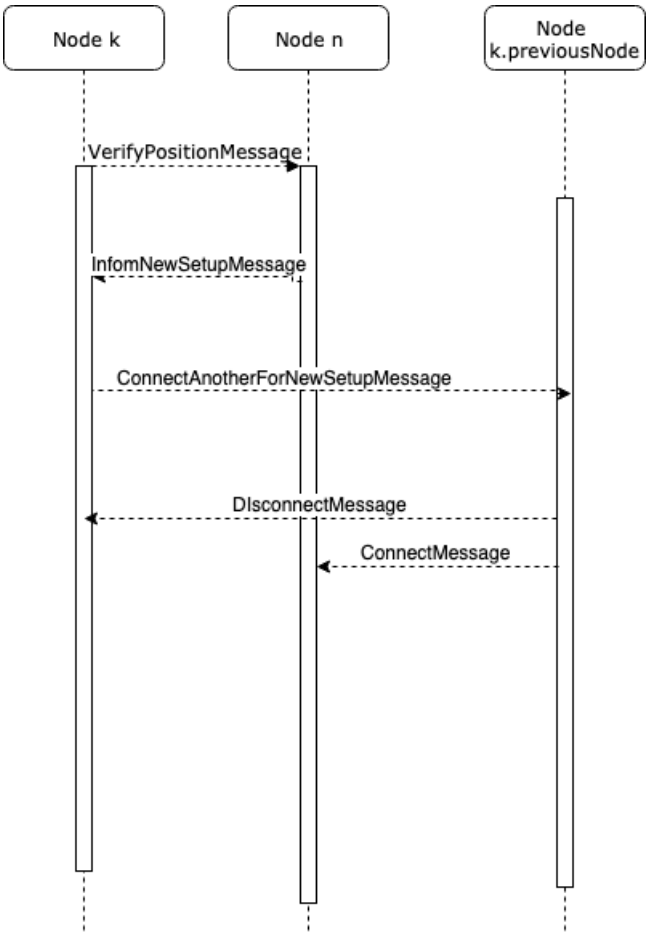
```
public static int hashFunc(String s) {
     int hash = 7;
     for (int i = 0; i < s.length(); i++) {
```

```
        hash = hash*31 + s.charAt(i);
    }
    hash=hash/10000000;
    return hash;
}
```

6.5.2 Network

Η εφαρμογή όσον αφορά το κομμάτι της διασύνδεσης κόμβων, έχει έναν handler ο οποίος διαχειρίζεται τις συνδέσεις με άλλους κόμβους. Ακολουθεί το λογικό διάγραμμα της νέας υλοποίησης.

Έστω κόμβος n ο κόμβος που θέλει να συνδεθεί στο δίκτυο, k ο κόμβος με τον οποιο επικοινωνεί ο $k.previousNode$ ο τοπολογικά προηγούμενος κόμβος του k. Παρακάτω φαίνεται το διάγραμμα που περιγράφει μια επιτυχημένη -τοπολογικά - σύνδεση του νεοεισελθεντες κόμβου, δηλαδή με άλλα λόγια το happy path.



Σχήμα 6.1: Happy path σύνδεσης ενός νέου κόμβου

sendPositionDirectionsToPeer

Η μέθοδος sendPositionDirectionsToPeer, στην οποία ο ήδη υπάρχων κόμβος στο δίκτυο, καθοδηγεί το νέο κόμβο προς την σωστή θέση στο Chord. Έστω s το αποτέλεσμα του hashFunc του κόμβου που προϋπάρχει στο δίκτυο, $s.previousNode$ το αποτέλεσμα της hashFunc του συνδεδεμένου, προηγούμενου κόμβου του s στο Chord και n το αποτέλεσμα της hashFunc του νέος κόμβος που θέλει να συνδεθεί στο δίκτυο. Τότε αν ισχύει ότι το

$(n <= sn > s.previousNode)||(s > s.previousNode(n < s.previousNoden <= s)||s > s.previousNode)$ ο κόμβος n ανήκει χωροταξικά ανάμεσα στον κόμβο s και τον κόμβο $s.previousNode$. Αλλιώς εξετάζονται ο νέος κόμβος δρομολογείται προς επικοινωνία στον επόμενο του n ,στη διάταξη του Chord, κόμβο.

Εξαίρεση αποτελεί η περίπτωση που μόνο ένας κόμβος υπάρχει. Τότε αυτόματα συνδέονται μεταξύ τους οι 2 κόμβοι, και ο ένας γίνεται successor του άλλου.

```
private void sendPositionDirectionsToPeer(Peer peer) {
       // I am alone in the network
  if (this.peer == aloneInNetwork()) {
      sendMessage(new ConnectMeMessage());
      setPreviousNode(peer);
      setNextNode(peer);
  }
  //the peer is in correct position in dht (before me)
  //i should do something to change my setup
  else if( peer.isInCorrectPosition()){
     sendMessage(new VerifyPositionMessage( true, this.previousNode));
  }
  //else redirect him to the next
  else {
     sendMessage(new VerifyPositionMessage( false, this.nextNode));
  }
}
```

onConnectMe

Η μέθοδος onConnectMe, κατά την οποία ο κόμβος που θέλει να συνδεθεί στην τοπολογία ενημερώνεται ότι δεν υπάρχει παρά μόνο ένας κόμβος ενεργός και κατά συνέπεια πρέπει να συνδεθούν μεταξύ τους.

```
private void onConnectMe(ConnectMeMessage msg) {
  setPreviousNode(msg.peer);
  setNextNode(msg.peer);
}
```

onVerifyPosition

Η μέθοδος onVerifyPosition, στην οποία ο κόμβος που θέλει να συνδεθεί στην σωστή θέση, κατά Chord τοπολογία, πληροφορείται για το αν έχει επικοινωνήσει με τον κατάλληλο κόμβο. Σε περίπτωση που δεν έχει επικοινωνήσει με τον σωστό κόμβο, τότε δέχεται μήνυμα με περιεχόμενο φαλσε και με τη διεύθυνση του επόμενο κόμβου προς επικοινωνία (δηλαδή

του successor του κόμβου που επικοινώνησε). Στη συνέχεια αποσυνδέεται με αιτιολογία $WRONGPOSITION$ και συνδέεται στον επόμενο κόμβο.

Σε περίπτωση που βρίσκεται τοπολογικά στη κατάλληλη θέση τότε εκκινείται η διαδικασία αλλαγής συνδέσεων των υπαρχόντων κόμβων. Δηλαδή στέλνει μήνυμα επιβεβαίωσης $InformNewSetup(1)$ στον κόμβο με τον οποίο έχει συνδεθεί και αποδεικνύεται ότι είναι ο successor του.

```
 private void onVerifyPosition(VerifyPositionMessage msg) {
     //in  a wrong position , then proceed to the next candidate and disconnect
         from current node
     if (msg.getCurrentPositionState()==false) {
        doConnectToNextNode(msg.nextNodeAddress);
        disconnectFromPeer(ReasonCode.WRONG_POSITION);
     }
     //in right position
     //then send InformNewSetup to my nextNode in dht
     else {
        //set channelMgr.nextNode
     setNextNode(msg.peer);
sendMessage(new InformNewSetupMessage(1));
     setNextNode(msg.peer);
     }
   }
```

onInformNewSetup

```
private void onInformNewSetup(InformNewSetupMessage msg) {
     //inform my last previous that the setup has changed and he should connect
         to his new next
     //if in channels previousNode is contained then you can send him a message
         that my    previous node has changed
     if(channelMgr.previousNode!=null) {
        previousChannel.sendMessage(new
            ConnectAnotherForNewSetupMessage(this.peer) ;
       }
       //now the correct node is set to the previousNode variable
setPreviousNode(msg.peer);
   }
```

onConnectAnotherForNewSetup

Η μέθοδος onConnectAnotherForNewSetup, κατά την οποία ο πρώην predecessor συνδέεται με το νεοσυνδεθέντα κόμβο του δικτύου και τον κάνει συγγεσσορ του. Παράλληλα στέλνει μήνυμα αποσύνδεσης στον παλαιό successor του με αιτιολογία.

```
    private void onConnectAnotherForNewSetup(ConnectAnotherForNewSetupMessage
        msg) {
    Node node=msg.getNode();
    doConnectToOneNode(msg.getNode);
    //if my next node isn't the same as my prev node (means it's not only the
        two of us in the network) then disconnect from him.
      msgQueue.disconnect(ReasonCode.CONNECT_ANOTHER_FOR_NEW_SETUP);
}
```

6.5.3 Consensus

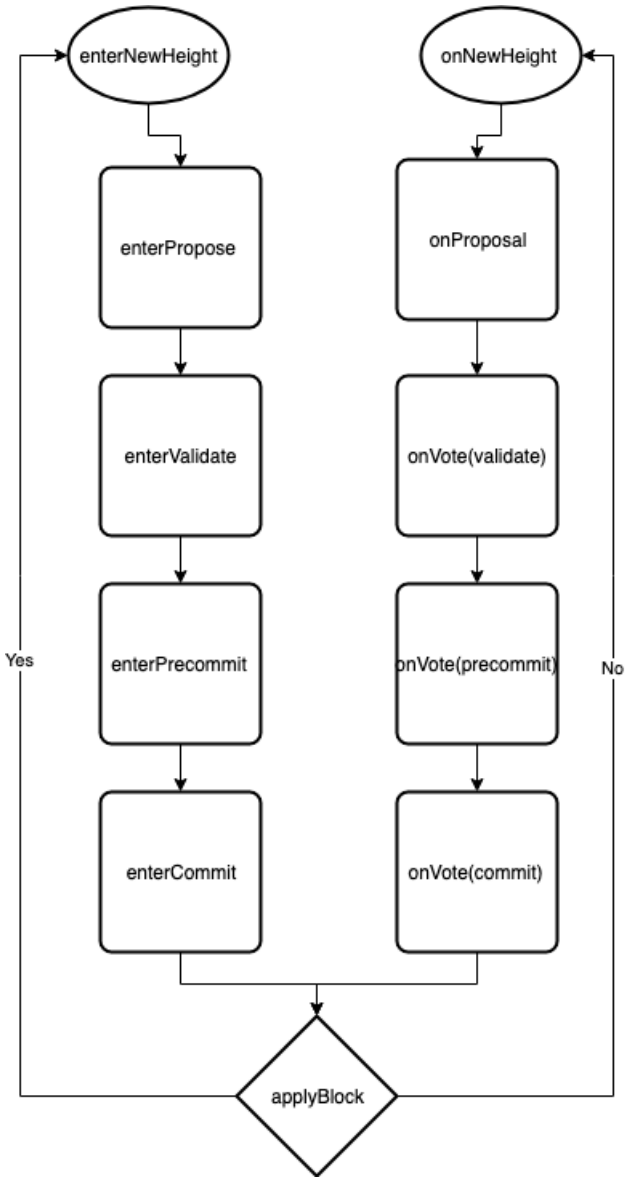
Παρακάτω φαίνεται η ροή του πρωτοκόλλου που ακολουθείται για τη διαδικασία παραγωγής και αποθήκευσης των νέων block.

Σε κάθε γύρο παραγωγής block υπάρχουν 2 είδη κόμβων:

- Ο κόμβος ο οποίος έχει αποθηκεύσει το τελευταίο block και κατά συνέπεια είναι αυτός που "αναλαμβάνει" να δημιουργήσει το επόμενο block στο distributed blockchain.

- Οι κόμβοι οι οποίοι διαδίδουν στην τοπολογία του Chord τα μηνύματα που προέρχονται από τον προαναφερθέντα κόμβο.

Σε κάθε γύρο λοιπόν το πρώτο είδος κόμβου ακολουθεί την αριστερή ροή του σχήματος ενώ το δεύτερο είδος τη δεξιά. Οπως φαίνεται και οι 2 ροές καταλήγουν στο ίδιο σημείο (apply block) στο οποίο ένας από τους κόμβους αποθηκεύει το παραγόμενο block. Αυτός ο κόμβος στον επόμενο γύρο είναι αυτός που θα παράξει και το block.



Σχήμα 6.2

enterNewHeight

```
protected void enterewHeight() {
      // update previous block
      Block prevBlock = chain.getLatestBlock();
      height = prevBlock.getNumber() + 1;
      proposal = null;
      // reset votes and events
      clearVotes();
      clearTimerAndEvents();
      if( iPropose ) {
         resetTimeout();
```

```
//Broadcast NEW_HEIGHT messages to ALL peers.
   NewHeightMessage msg = new NewHeightMessage(height);
        broadcast(msg);
       }
    }
```

onNewHeight

```
   protected void onNewHeight(long newHeight) {
   this.height=newHeight;
   resetProposalCounters();
    NewHeightMessage msg = new NewHeightMessage(height);
    broadcast(msg);
    }
```

enterPropose

```
    protected void enterPropose() {
        state = State.PROPOSE;
        resetTimeout(config.bftProposeTimeout());
        if (proposal == null) {
            Block block = proposeBlock();
            proof = new Proof(height, view, precommitVotes.getRejections());
            proposal = new Proposal(proof, block.getHeader(),
                block.getTransactions());
            proposal.sign(coinbase);
        }
        broadcast(new ProposalMessage(proposal));
    }
```

onProposal

```
      protected void onProposal(Proposal p) {
       state = State.PROPOSE;
      //store and forward proposal
      ProposalMessage msg = new ProposalMessage(p);
     broadcast(msg);
      proposal = p;
```

enterValidate

```
     protected void enterValidate() {
       state = State.VALIDATE;
```

```
  resetTimeout(config.bftValidateTimeout());
    // validate block proposal
    boolean valid = (proposal != null) &&
        validateBlock(proposal.getBlockHeader(), proposal.getTransactions());
    // construct vote
    Vote vote = valid ? Vote.newApprove(VoteType.VALIDATE, height, view,
        proposal.getBlockHeader().getHash())
            : Vote.newReject(VoteType.VALIDATE, height, view);
    // always broadcast vote directly.
    broadcast(new VoteMessage(vote));
}
```

enterPreCommit

```
protected void enterPreCommit() {
    state = State.PRE_COMMIT;
  resetTimeout(config.bftPreCommitTimeout());
    precommitVotes, commitVotes);
    Optional<byte[]> blockHash = validateVotes.anyApproved();
    Vote vote = blockHash.map(bytes -> Vote.newApprove(VoteType.PRECOMMIT,
        height, view, bytes))
            .orElseGet(() -> Vote.newReject(VoteType.PRECOMMIT, height,
                view));
    vote.sign(coinbase);
    // always broadcast vote directly.
    precommitVotes.addVote(vote);
    broadcast(new VoteMessage(vote));
}
```

enterCommit

```
    protected void enterCommit() {
    state = State.COMMIT;
  resetTimeout(config.bftCommitTimeout());
    Optional<byte[]> blockHash = precommitVotes.anyApproved();
    if (!blockHash.isPresent()) {
        throw new ntuaSemuxBftException("Entered INVALID COMMIT STAGE ");
    } else {
        // create a COMMIT vote
        Vote vote = Vote.newApprove(VoteType.COMMIT, height, view,
            blockHash.get());
        vote.sign(coinbase);
        // always broadcast vote directly.
        commitVotes.addVote(vote);
        broadcast(new VoteMessage(vote));
```

```
    }
}
```

onVote

```
    protected void onVote(Vote v) {
    if (v.getHeight() == height
            && v.getView() == view
            && v.validate()) {
        switch (v.getType()) {
        case VALIDATE:
           boolean valid = (proposal != null) &&
               validateBlock(proposal.getBlockHeader(),
               proposal.getTransactions());
           validateVotes.addVote(v);
           state = State.VALIDATE;
           if (countVal==1) {
              VoteMessage msg = new VoteMessage(v);
               broadcast(msg);
           }
            break;
        case PRECOMMIT:
            precommitVotes.addVote(v);
            state = State.PRE_COMMIT;
           if (countPre==1) {
                    VoteMessage msg = new VoteMessage(v);
                    broadcast(msg);
           }
            break;
        case COMMIT:
            commitVotes.addVote(v);
            state = State.COMMIT;
           if (countCom==1) {
                VoteMessage msg = new VoteMessage(v);
                broadcast(msg);
           }
            break;
        }
    }
}
```

Κεφάλαιο 7

Αποτελέσματα

Σε αυτό το κεφάλαιο θα γίνει παρουσίαση των συμπερασμάτων που προκύπτουν της πρότασης. Λόγω του ότι η υλοποίηση αποτελεί Proof of concept δεν κατέστει δυνατόν να πραγματοποιηθούν όλες οι λειτουργικότητες που παρουσιάζονται. Στους τομείς που δεν μπορούν να εξαχθούν, παρουσιάζονται θεωρητικοί υπολογισμοί με στόχο να δείξουν τις διαφορές σε σχέση τους ¨ανταγωνιστές¨.

7.1 Μετρήσεις

Για τις ανάγκες των μετρήσεων, επιλέχθηκαν κάποιοι κόμβοι οι οποίοι στο Genesis Block τους ανατέθηκε ένας μεγάλος αριθμός Sem ώστε να μπορούν να κάνουν συναλλαγές. Για τις μετρήσεις γίνονταν 7 συναλλαγές το δευτερόλεπτο στο δίκτυο στο δίκτυο κόμβων. Μετρήθηκαν τα εξής:

- Μέσος καταναλισκόμενος αποθηκευτικός χώρος ανά κόμβο.
- Μέσο δικτυακό εύρος ζώνης (network bandwidth) ανά κόμβο.
- Αριθμός εισερχόμενων μηνυμάτων ανά κόμβο.

Μεσος καταναλισκόμενος χώρος

Για τη μέτρηση του δεσμευμένου χώρου για τις ανάγκες του Distributed Blockchain αφέθηκε δίκτυο 15 κόμβων για κατά μέσο όρο 15 ώρες. Σαν μεταβλητή ήταν οι συναλλαγές ανά δευτερόλεπτο που γίνονταν στο δίκτυο. Τα αποτελέσματα κατάγράφονται παρακάτω:

- ρατε 6.7 txs 60.76 kB
- ρατε 21.7 txs 224.27 kB
- ρατε 23.3 txs 245.43 kB

Μέσο δικτυακό εύρος ζώνης

Για τη μέτρηση του μέσου έυρους ζώνης δημιουργήθηκε ένα δίκτυο 15 κόμβων στο οποίο πραγματοποιούνταν 6.7 συναλλαγές το δευτερόλεπτο. Στη συνέχεια μέσω του εργαλείου nettop που έχει τη δυνατότητα να δώσει μετρήσεις bandwidth ανά διεργασία, καταγράφηκαν οι τιμές για τα εισερχόμενα και εξερχόμενα δεδομένα της κάθε εφαρμογής από τη στιγμή της. Βρέθηκε η μέση τιμή και τελικά διαρέθηκε με το συνολικό χρόμο λειτουργιας του δικτύου. Τα αποτελέσματα που προέκυψαν ήταν τα εξής:

- Για 6.7 συναλλαγές το δευτερόλεπτο : $bytesin = 1.92kB/second$, $bytesout = 2.08kB/second$.
- Για 23.3 συναλλαγές το δευτερόλεπτο : $bytesin = 8.83kB/second$, $bytesout = 8.75kB/second$.

Οι μετρήσεις αυτές φαίνονται λογικές αφού τα μεγέθη που καταγράφηκαν είναι ανάλογα των συναλλαγών ανά δευτερόλπτο.

Αριθμός μηνυμάτων ανά κόμβο

Ο αριθμός μηνυμάτων ανά κόμβο μπορεί να υπολογιστεί ακριβώς σε σχέση με τις συναλλαγές που συμβαίνουν ανά δευτερόλεπτο. Τα εισερχόμενα μηνύματα ισούνται με τα εξερχόμενα λόγω της δομής του πρωτοκόλλου, δηλαδή του γεγονότος ότι ο κάθε κόμβο επαναπροωθεί το μήνυμα που λαμβάνει. Τα μηνύματα είναι που στέλνονται από έναν κόμβο. οφείλονται :

- **στο πρωτοκόλλου συμφωνίας :** Το πρωτόκολλο συμφωνίας έχει 5 στάδια που συνεπάγεται ότι κάθε κόμβος δέχεται και λαμβάνει 5 μηνύματα ανα 30 δευτερόλεπτα. (0,17 messages/second
- **στην τακτική επικοινωνία των κόμβων (ping -pong) :** Αυτού του είδους η επικοινωνία εκκινείται κάθε 30 δευτερόλεπτα από κάθε κόμβο προς 2 άλλους. Για κάθε μήνυμα ping που λαμβάνει ένας κόμβος στέλνει ένα pong. (0,13 messages/second)
- **στις συναλλαγές που συμβαίνουν :** Κάθε συναλλαγή ισούται με ένα εξερχόμενα μήνυμα ανά κόμβο (numberOfTransactions messages/second

Συνεπώς τα μηνύματα που στέλνει κάθε κόμβος ανά δευτερόλεπτο σε συνάρτηση με τις συναλλαγές που συμβαίνουν ανά δευτερόλεπτο υπολογίζονται :

$$sentMessagesPerSecond = receivedMessagesPerSecond = 0.3 + txs \tag{7.1}$$

Συνεπώς σε ένα σύστημα που γίνονται 200 συναλλαγές ανα δευτερόλεπτο ο αριθμός των μηνυμάτων που στέλνονται ισούται με 200.3 μηνύματα/ δευτερόλεπτο.

7.1.1 Παρατηρήσεις

Στα πλαίσια της δοκιμής της εφαρμογής ntuaSemux έγιναν stress tests μετρήσεις για να διαπιστωθούν οι δυνατότητες της. Κατά τη διάρκεια των μετρήσεων έγινε σαφές ότι:

- Ο μέγιστος αριθμός συναλλαγών που μπορούν να υπάρξουν σε ένα μπλοκ δεν μπορεί να υπερβαίνει τις 5607.
- Ο χρόνος που χρειάζεται κάθε κόμβος για την επικύρωση ενός μπλοκ είναι ανάλογος του πλήθους των συναλλαγών που περιέχει.
- Όσο αυξάνεται το πλήθος των συναλλαγών που πραγματοποιούνται, τόσο αυξάνεται και το bandwidth που δεσμεύεται απο τον κάθε κόμβο με μέγιστο τα 25μς και μέσο όρο 12 για ένα φόρτο συναλλαγών της τάξης των 10 txs per second.

Τα συμπεράσματα της εργασίας μπορούν να αναλυθούν σε κάποιους βασικούς άξονες.

1. Καταναλισκόμενος χώρος ανά κόμβο
2. Αριθμός διαδιδόμενων μηνυμάτων
3. Ασφάλεια
4. Blocktime

7.2 Αποτελέσματα πρότασης

Σε αυτή την ενότητα θα γίνει μία σύγκριση της πρότασης με μία γενική υλοποίηση ίδιου τύπου με αδόμητο δίκτυο P2P. Αναλυτικότερα θα θεωρηθεί ότι η υλοποίηση P2P ακολουθεί κάποιες απο τις παραδοχές του Bitcoin ώστε να μπορέσουμε να λάβουμε στατιστικά από την ιστοσελίδα https://statoshi.info/[26] η οποία περιέχει αναλυτικά στοιχεία για το συγκεκριμένο κρυπτονόμισμα. Σε αυτό το σημείο πρέπει να σημειωθεί ότι θα γίνουν προσεγγίσεις σε όποιο βαθμό απαιτείται ώστε τα αποτελέσματα να ανταποκρίνονται όσο το δυνατόν γίνεται στη πραγματικότητα. Εχει μεγάλη σημασία να γίνει κατανοητό ότι αυτές οι προσεγγίσεις είναι απαραίτητες ώστε να συγκρίνονται παρόμοια συστήματα.

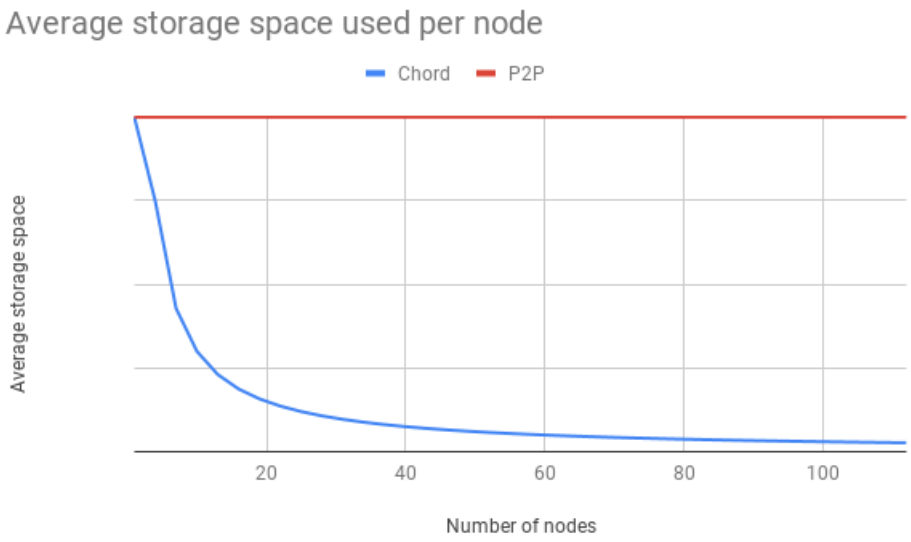
7.2.1 Μείωση καταναλισκόμενου χώρου

Ενας απο τους κύριους στόχους της πρότασης είναι η μείωση του χώρου που καταλαμβάνει το distributed blockchain σε κάθε κόμβο. Τυπικά, σύμφωνα με το πρωτόκολλο που αναλύθηκε παραπάνω κάθε κόμβος του δικτύου είναι αρμόδιος να αποθηκεύει block σύμφωνα με το block-id.

Πρώτα θα υπολογιστεί θεωρητικά ο δεσμευμένος χώρος με βάση την πρόταση και εν συνεχεία θα παρατεθούν με βάση τις μετρήσεις, αποτελέσματα.

Θεωρητικοί υπολογισμοί

Αν υποθέσουμε ότι το προτεινόμενο σύστημα έχει το replication factor ίσο με r και ότι το μέσο μέγεθος που καταλαμβάνει το διστριβυτεδ blockchain σε κάθε κόμβο ισούται με x. Δηλαδή δεδομένου του συστήματος κάθε κόμβος αποθηκεύει τα $r-1$ αντίγραφα που οφείλει



Σχήμα 7.1: Μέσος αποθηκευτικός χώορος για δεδομένο ύψος Blockchain

και το δικό του τοπικό αντίγραφο. Τότε κάθε κόμβος θα δεσμεύει χώρο για τις ανάγκες της εφαρμογής :

$$storageUsedPerNode = x * r \tag{7.2}$$

Το x ωστόσο εξαρτάται από παράγοντες τους εξής παράγοντες:

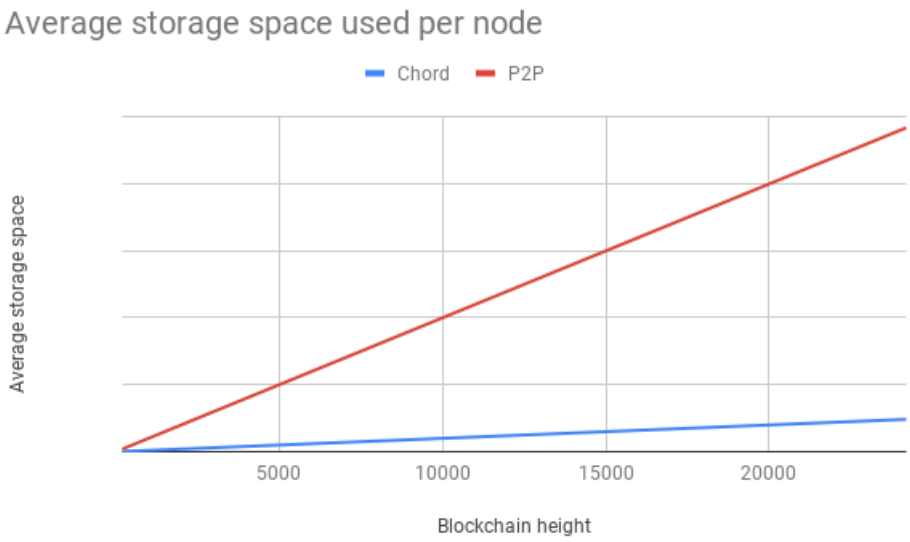
1. Τους συμμετέχοντες k στο δίκτυο του blockchain.
2. Το ύψος h στο οποίο βρίσκεται το blockchain.
3. Το replication factor r του συστήματος.

Υποθέτοντας ότι η συνάρτηση κατακερματισμού που χρησιμοποιείται οδηγεί στην, όσο τον δυνατόν, ομοιόμορφη κατανομή blocks στους συμμετέχοντες κόμβους είναι εύκολο να εξαχθεί ο τύπος υπολογισμού του x συναρτήσει του αριθμού κόμβων k και του ύψους του distributed blockchain h.

$$x = (h * avg(blocksize))/k \tag{7.3}$$

Από την παραπάνω σχέση παρατηρείται ότι όσο αυξάνεται ο αριθμός των κόμβων, μειώνεται εκθετικά η μέση χωρητικότητα x. Συγχρόνως όμως το μέγεθος x εξαρτάται και από το ύψος h ,ένα μέγεθος που συνεχώς αυξάνεται με το χρόνο. Αυτό που μας ενδιαφέρει ωστόσο είναι ένα δεδομένο ύψος (θεωρώντας ότι ο αριθμός είναι αρκετά μεγάλος) πως επηρεάζεται απο τη συμμετοχή στο δίκτυο. Φαίνεται στο διάγραμμα 7.1 ότι για δεδομένο ύψος h (μεγάλο ύψος) η μέση καταναλισκόμενη χωρητικότητα ανά κόμβο μειώνεται όσο αυξάνεται ο αριθμός των κόμβων.

Στο διάγραμμα 7.2 φαίνεται ότι ο μέσος αποθηκευτικός χώρος σε ένα δίκτυο με δεδομένο συμμετεχόντων κόμβων k αυξάνεται γραμμικά στην υλοποίηση Chord όσο μεγαλώνει το ύψος



Σχήμα 7.2: Μέσος αποθηκευτικός χώρος για δεδομένο αριθμό κόμβων

h του **distributed blockchain**. Η κλίση ωστόσο είναι πολύ μικρότερη σε σχέση με την υλοποίηση P2P και ορίζεται απο το λόγο h/k , δηλαδή ύψους προς αριθμού κόμβων.

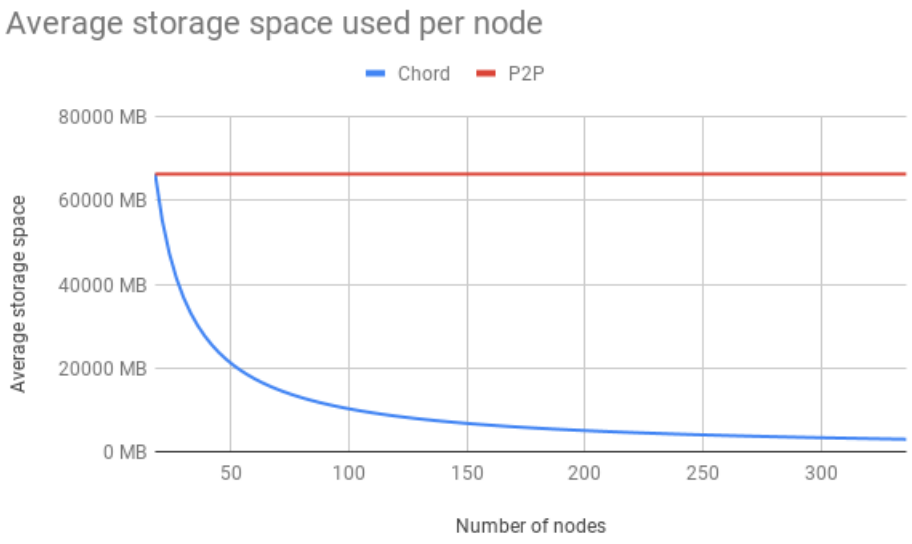
Συγχρόνως για δεδομένο αριθμό συνδεδεμένων κόμβων η μέση καταναλισκόμενη χωρητικότητα ανά κόμβο αυξάνεται με μικρή κλίση όσο αυξάνεται το ύψος.

Η αξία των παραπάνω γραφημάτων δεν μπορεί να εκτιμηθεί μέχρις ότου δοθεί ένα διάγραμμα που αφορά την μέση υλοποίηση ενός p2p-blockchain. Σε μία τέτοια περίπτωση οι κάθε κόμβος διατηρεί το πλήρες αντίγραφο της αλυσίδας. Όπως είναι επακόλουθο στα αντίστοιχα διαγράμματα θα υπάρχουν διαφορές στη περίπτωση του διαγράμματος με δεδομένο το ύψος του blockchain συναρτήσει του αριθμού των κόμβων.

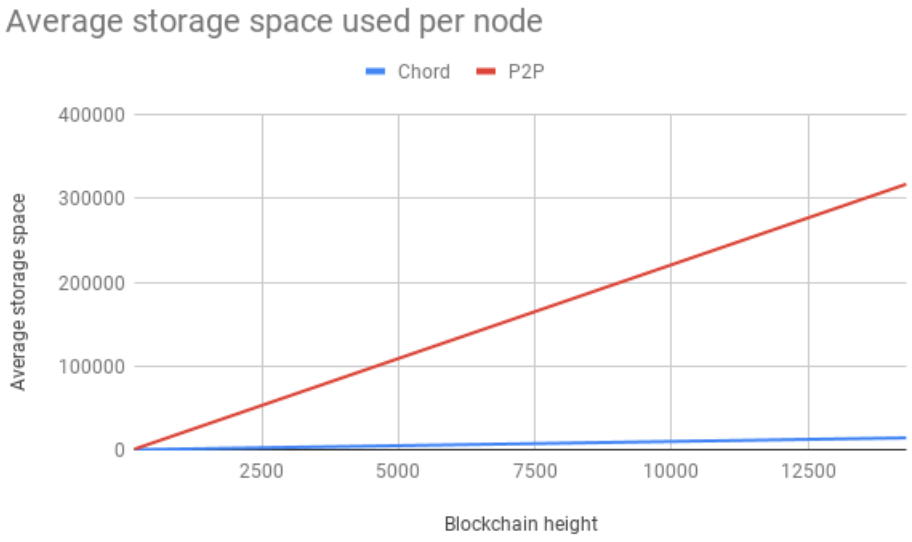
Πειραματικοί υπολογισμοί

Σύμφωνα με το στατιστικά για το Bitcoin το μέσο μέγεθος ενός μπλοκ είναι 664 $Kilobytes$. Σε αυτό το σημείο θα θεωρήσουμε μία υλοποίηση της πρότασης της οποίας το μέσο μπλοκ καταλαβάνει χώρο 664 $Kilobytes$ και έχει παράγοντα αναπαραγωγής ίσο με 15. Τότε παράγονται τα αντίστοιχα διαγράμματα. Στο διάγραμμα 7.3 μπορεί να γίνει αντιληπτή η διαφορά τάξης μεγέθους του χρησιμοποιούμενου χώρου ανά κόμβο για δεδομένο ύψος σε σχέση με τους συμμετέχοντες κόμβους, ενώ στο διάγραμμα 7.4 για δεδομένο αριθμό κόμβων (336 κόμβοι) σε σχέση με το ύψος του blockchain.

Από τα διαγράμματα 7.3 7.4 φαίνεται σε πραγματικά νούμερα ότι η μείωση του χώρου που καταλαμβάνει το blockchain ανά κόμβο είναι τεράστια,. Σε ένα δίκτυο που ακολουθεί τα πρωτόκολλα της πρότασης με 50 κόμβους, το οποίο βρίσκεται σε ύψος 100.000 κάθε κόμβος αποθηκεύει κατά μέσο όρο 19.500 Mb σε αντιδιαστολή με την υλοποίηση P2P που αποθηκεύει 66.400 Mb. Η διαφορά μεγαλώνει περισσότερο αν στο δίκτυο της πρότασης συμμετέχουν 300 κόμβοι, με συνέπεια ο καθένας να αποθηκεύει 3320 Mb.



Σχήμα 7.3: Μέσος αποθηκευτικός χώρος για δεδομένο ύψος Blockchain



Σχήμα 7.4: Μέσος αποθηκευτικός χώρος για δεδομένο αριθμό κόμβων

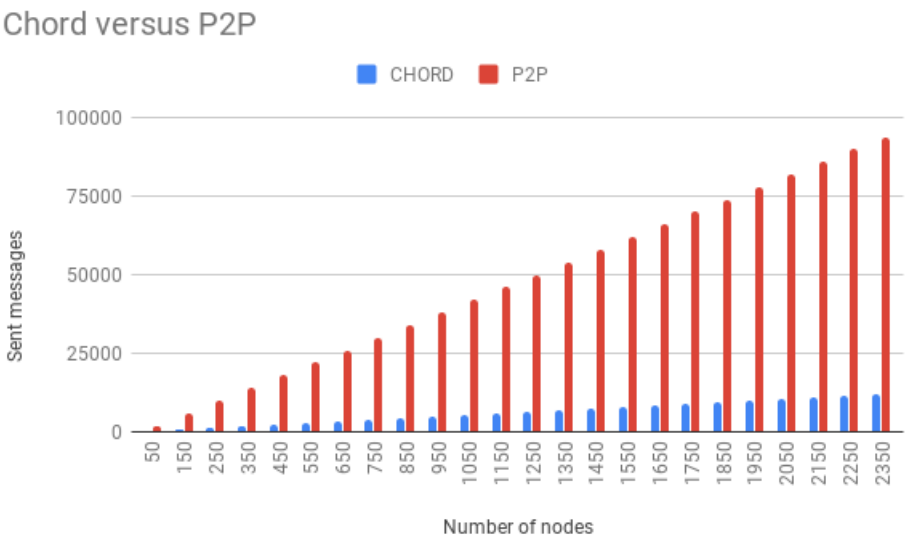
7.2.2 Διαδιδόμενα μηνύματα

Θεωρητικοί υπολογισμοί

Η υλοποίηση με βάση το Chord έχει το θετικό ότι καταφέρνει να μειώσει το πλήθος των μηνυμάτων που εκπέμπονται από τον κάθε κόμβο. Η συνηθισμένη πρακτική στη μέση υλοποίηση είναι το broadcast του κάθε μηνύματος σε έναν αριθμό κόμβων του δικτύου.

Για παράδειγμα το δίκτυο του Bitcoin επιτρέπει σε κάθε κόμβο τη σύνδεση με μέχρι και άλλους 125 κόμβους. Για κάθε μήνυμα που πρέπει να γίνει broadcast επιλέγονται τυχαία

Σχήμα 7.5: Συνολικά απεσταλμένα μηνύματα ανά γύρο Consensus



8 ενεργοί κόμβοι από κάθε κόμβο, γεγονός που μπορεί να σημαίνει ότι ορισμένοι κόμβοι λαμβάνουν μηνύματα παραπάνω από μια φορές.

Στη περίπτωση ενός blockchain κρυπτονομίσματος που έχει ως βάση το Chord κάθε κόμβος λαμβάνει 1 φορά το κάθε μήνυμα δεδομένης της τοπολογίας. Αν θεωρήσουμε ότι 2 υλοποιήσεις διαφέρουν μόνο στον τρόπο διασύνδεσης και όχι στις επιμέρους διαδικασίες που αφορούν τα πρωτόκολλο Συμφωνίας (Consensus) μεταξύ των κόμβων. Παράλληλα έστω ότι το πρωτόκολλο συμφωνίας περιλαμβάνει 5 βήματα εκ των οποίων κάθε βήμα απαιτεί βροαδςαστ από κάθε κόμβο. Στην περίπτωση της P2P υλοποίησης broadcast συνεπάγεται την αποστολή του μηνύματος σε 8 τυχαίους κόμβους ενώ στην περίπτωση του CHORD την αποστολή στον επόμενο -στην τοπολογία- κόμβο.

Το διάγραμμα 7.5 περιγράφει -θεωρητικά- συναρτήσει των κόμβων τον αριθμό των μηνυμάτων που διαδίδονται στο δίκτυο της εκάστοτε υλοποίησης. Στη θεώρηση ότι δεν υπάρχει κάποιου είδους αποτυχία κατά την διάρκεια της εκτέλεσης ενός κύκλου του πρωτοκόλλου Consensus τότε το διάγραμμα μπορεί να δώσει μία ενδεικτική εικόνα της διαφοράς των P2P-CHORD.

Πρακτικοί υπολογισμοί

Οπως υπολογίστηκε παραπάνω στην υλοποίηση τα εισερχόμενα μηνύματα ανά κόμβο που αφορούν το κομμάτι του Consensus και το κομμάτι της διάδοσης των συναλλαγών ισούνται με $0.3 + txs$ ανά δευτερόλεπτο. Αντλώντας στοιχεία από το https://statoshi.info/ φαίνεται ότι κάθε κόμβος στέλνει 22.02 $messages/second$ κατά μέσο όρο στην υλοποίηση του Bitcoin με κατά μέσο όρο 4 συναλλαγές το δευτερόλεπτο. Στην περίπτωση της υλοποίησης ntuaSemux τα εισερχόμενα μηνύματα ανέρχονται σε 4.3 $messages/second$.

Σε μία πλήρη υλοποίηση της πρότασης σύμφωνα με τα στατιστικά που έχουν εξαχθεί για

υλοποίηση Chord με αριθμό κόμβων 1.000 και 10.000 και συχνότητα συντήρησης των finger table κάθε 10 και 120 δευτερόλεπτα [38]αντίστοιχα εξάγονται τα εξής στοιχεία.

Number of sent messages per second	Frequency of finger table maintenance	Churn	Nodes
9.3 messages / sec	120 s	Low	1.000
19.3 messages / sec	10 s	Low	1.000

Σχήμα 7.6: Πειραματικές τιμές εξερχομένων μηνυμάτων σε διάταξη Chord

Τα στοιχεία αυτά συμπεριλαμβάνουν τα σταλθέντα μηνύματα ενός δικτύου για τη συντήρηση των finger tables , τη δημιουργία αντιγράφων και τις αναζητήσεις που υλοποιεί το Chord. Μάλιστα όπως αναφέρεται, η κινητικότητα στο δίκτυο δεν έδειξε να επηρεάζει ουσιαστικά τον αριθμό μηνυμάτων. Με βάση τις μετρήσεις αρκεί στην εξίσωση να συμπεριλάβουμε και τα μηνύματα που στέλνονται λόγω του Consensus, δηλαδή $5messages/30seconds = 0.17messages/sec$ και λόγω των συναλλαγών, δηλαδή $4messages/second$.

Άρα τα σταλθέντα μηνύματα σε μία υλοποίηση της πρότασης φτάνουν τα $23.47messages/second$.

Σύμφωνα με το γεγονός ότι κάθε κόμβος στην P2P υλοποίηση, όταν κάνει broadcast ένα μήνυμα στέλνει σε αλλους 8 κόμβους και θεωρώντας ότι έχει Consensus 5 βημάτων τότε τα σταλθέντα μηνύματα ανά δευτερόλεπτο υπολογίζονται :

- ping-pong messages : $1.62pingMessages/second + 2.4pongMessages/second$
- transaction messages : $4 * 7.5 = 30cmessages/second$
- Consensus messages : $5 * 7.5/30 = 1.33messages/second$.

Αρα συνόλικά σε μία υλοποίηση P2P τα συνολικά σταλθέντα μηνύματα ανά κόμβο υπολογίζονται $34.7messages/second$

Φαίνεται λοιπόν ότι η πρόταση-Chord καταφέρνει να μειώσει τον αριθμό των μηνυμάτων κατά περίπου $11messages/second$.

7.2.3 Ασφάλεια

Η πρόταση πρέπει να πληρεί και προϋποθέσεις ασφάλειας. Πέρα από το αυτονόητο, δηλαδή ότι κάθε κόμβος πρέπει να διαθέτει την πλέον ενημερωμένη έκδοση του λογισμικού, να συνδέεται απο μοναδική θέση δικτύου, να συνδέεται μοναδικός λογαριασμός κλπ.

Ουσιώδης είναι η αποφυγή κακόβουλων ενεργειών με στόχο των επηρεασμό της πορείας του distributed blockchain και την εκμετάλλευση του πρωτοκόλλου που ακολουθεί για ιδίον όφελος.

Με στόχο την προαγωγή της "δημοκρατίας" υπό την έννοια ότι όλοι οι συμμετέχοντες κόμβοι έχουν την ευκαιρία της συμμετοχής στη διαδικασία παραγωγής block γίνονται εκπτώσεις στη σε θέματα ασφάλειας. Σε υλοποιήσεις P2P όλο το δίκτυο πιστοποιεί την εγκυρότητα

P(majority)	Network Nodes	Replication Factor	Agent's Controlled nodes
4.07%	260	15	88
5.75%	260	15	91
7.99%	260	15	94
10.94%	260	15	97
14.74%	260	15	100
19.54%	260	15	103
25.44%	260	15	106

Σχήμα 7.7: Υπολογισμός πιθανότητας επίτευξης πλειοψηφίας από έναν κακόβουλο δράστη

του προς πρόταση νέου βλοςκ και κατά συνέπεια για να ελεγχθεί από κακόβουλους παράγοντες το δίκτυο θα πρέπει ο δράστης να κατέχει μεγαλύτερο από το 50% του δικτύου, είτε σε υπολογιστική ισχύ, είτε σε αξία.

Στη συγκεκριμένη πρόταση, η διαδικασία της παραγωγής νέου block εξαρτάται άμεσα από την τοπολογία που ακολουθείται καθώς και από το replication factor ο οποίος έχει οριστεί. Πιο συγκεκριμένα η πιθανότητα κάποιος κακόβουλος χρήστης να ελέγχει τα 2/3 των κόμβων, οι οποίοι σε ένα συγκεκριμένο γύρο παραγωγής νέου block καλούνται να ψηφίσουν υπολογίζεται ως εξής:

Ας θεωρήσουμε ότι το δίκτυο της πλατφόρμας αποτελείται απο n κόμβους, τηρείται παράγοντας αναπαραγωγής r καθώς και ότι ο πιθανός κοκόβουλος δράστης έχει υπό τον έλεγχο του k κόμβους.

Δεδομένου ότι σε κάθε γύρο οφείλουν οι r κόμβοι, οι οποίοι είναι συνεχόμενοι τοπολογικά, να επικυρώσουν το νέο μπλοκ, και ότι αρκεί η έγκριση των 2/3 αυτών , υπολογίζεται η πιθανότητα να ελέγχεται η ελάχιστη απαιτούμενη πλειοψηφία για έναν δεδομένο κύκλο.

$$P(a) = \frac{\binom{r}{(2/3)r}}{\binom{n}{(2/3)r}}$$

Στον αριθμητή περιγράφονται οι επιτυχημένοι συνδυασμοί σε σχέση με το συνόλο των πιθανών συνδυασμών. Η αντίθετη πιθανότητα $P'(a)$ ορίζεται ως εξής:

$$P'(a) = 1 - P(a)$$

Εώς τώρα έχει υπολογιστεί η πιθανότητα του να πετύχει κάποιος την απαραίτητη πλειοψηφία ανά κύκλο παραγωγής μπλοκ. Αυτή η πιθανότητα, συνδέεται με τον αριθμό των κόμβων που ελέγχει ο κακόβουλος δράστης, με σκοπό τον υπολογισμό της πιθανότητας , ο δράστης ελέγχοντας k κόμβους να τύχει να έχει την πλειοψηφία.

$$P(majority) = 1 - (P'(a)^{\binom{k}{(2/3)r}})$$

Η παραπάνω σχέση εκφράζει τη πιθανότητα ο δράστης να ελέγχει τουλάχιστον την απαιτούμενη πλειοψηφία. Στο σχήμα 7.7 υπολογίζεται η πιθανότητα να επιτύχει την απαραίτητη πλειοψηφία ένας κακόβουλος δράστης, για δεδομένο αριθμό κόμβων του δικτύου (260) , με πρωτόκολλο

που διατηρεί παράγοντα αναπαραγωγή ίσο με 15. Δυστυχώς, λόγω της φύσης των πράξεων για την εύρεση του αποτελέσματος, δεν είναι εύκολος ο υπολογισμός της πιθανότητας για μεγαλύτερα μεγέθη δικτύων καθώς αποτελεί περιοριστικό παράγοντα η στρογγυλοποίηση των αριθμών[20]. Τελικά τα μεγέθη των απαιτούμενων πράξεων τείνουν να γίνονται πολύ μικρά, και κατα συνέπεια θεωρούνται 0 από κάποιο σημείο και μετά.

7.2.4 Block availability

Οι υλοποιήσεις των κρυπτονομισμάτων που βασίζονται στη τεχνολογία του Blockchain έχει συνήθως σαν χαρακτηριστικό το σταθερό ή σχεδόν σταθερό blocktime - δηλαδή τη συχνότητα δημιουργίας ενός νέου block στην αλυσίδα. Το κρυπτονόμισμα Bitcoin το καταφέρνει αυτό ρυθμίζοντας τον βαθμό δυσκολίας του προβλήματος προς επίλυση για τους κόμβους έτσι ώστε το blocktime να είναι περίπου 10 λεπτά. Αλλες υλοποιήσεις χρησιμοποιούν blocking εκδοχές πρωτοκόλλων ώστε να διατηρούν το blocktime σταθερό. Τέτοιου είδους υλοποιήσεις δεν χρησιμοποιούν σαν πρωτόκολλο Consensus Proof-of-Work.

Σε αυτό το σημείο πρέπει να επισημανθεί μια ουσιαστική διαφορά στο τρόπο προσέγγισης των πρωτοκόλλων: Σε μία P2P υλοποίηση κάθε μήνυμα - όπως αναφέρθηκε - γίνεται broadcast σε κάποιους γνωστούς κόμβους, ενώ στην πρόταση κάθε μήνυμα μεταβιβάζεται στον επόμενο τοπολογικά κόμβο καθώς και γίνεται χρήση των finger tables. Στη βάση της λογικής αυτής και σύμφωνα με στατιστικά 7.8 που έχουν αντληθεί από το δίκτυο του Bitcoin η διάδοση ενός block στο 50% των κόμβων του δικτύου καθώς και στο 90%των κόμβων. Μάλιστα ισχύει ότι :

> For blocks, whose size is larger than 20kB, each kilobyte costs an additional 80ms delay until a majority knows about the block. [36]

Παράλληλα φαίνεται και ο χρόνος διάδοσης μιας συναλλαγής.

Date	Block 50th percentile	Block 90th percentile	Transaction 50th percentile	Transaction 90th percentile
2013/11/22	5.9 seconds	122.0 seconds	1.3 seconds	13.0 seconds
2014/01/17	4.7 seconds	22.6 seconds	1.5 seconds	3.7 seconds
2015/04/22	4.128 seconds	78.7441 seconds	0.66 seconds	2.731 seconds
2017/04/01	1.771 seconds	9.803 seconds	3.57 seconds	13.616 seconds

Σχήμα 7.8: Χρόνος διάδοσης block στο κρυπτονόμισμα Bitcoin

Σε μία υλοποίηση CHORD η διάδοση του μηνύματος εξαρτάται αυστηρά απο την απόκριση των συμμετεχόντων κόμβων και τελικά από το μέγεθος του δικτύου.

Στην αυθεντική υλοποίηση του Semux χρησιμοποιείται χρονόμετρο, το οποίο χρησιμεύει για σηματοδοτήσει τη λήξη ενός σταδίου του ὂνσενσυς και να ορίσει το blocktime στα 30 δευτερόλεπτα. Αυτή η λογική διατηρήθηκε και στην μετατροπή του Semux σε NtuaSemux.

Σε αυτό το σημείο θα υπολογιστει θεωρητικά το πόσο γρήγορα ένα block είναι διαθέσιμο σε όλους τους κόμβους σε μία ολοκληρωμένη υλοποίηση Chord.

Απο μετρήσεις στην υλοποίηση φάνηκε ότι ο μέσος χρόνος επεξεργασίας(ανάγνωσης-πιστοποίησης-αποθήκευσης) του περιεχομένου ενός block το οποίο περιέχει 400 συνλλαγές ανέρχεται στα $3ms$. Επίσης βάσει των στατιστικών του [26] το μέσο latency ανάμεσα σε 2 κόμβους του Bitcoin είναι 86ms.

Τέλος απαιτούνται 4.77 hops κατά μέσο όρο σε ένα δίκτυο Chord 1.000 κόμβων για την επιτυχημένη αναζήτηση .

Τότε σε ένα δίκτυο Chord 1000 κόμβων, η πληροφορία ενός block είναι βρίσκεται από όλους τους κόμβους το πολύ σε $4.77hops * 2 * 86milliseconds = 0.820second$.

Το αποτέλεσμα προκύπτει ως εξής: Κάθε φορά ο ενδιαφερόμενος κόμβος επικοινωνεί με έναν άλλον κόμβο και περιμένει την απάντηση για πιθανή αναδρομολόγηση.

Κεφάλαιο 8

Επίλογος

8.1 Σύνοψη και συμπεράσματα

Σκοπός της διπλωματικής εργασίας ήταν η εξέταση των νέων τεχνολογιών αποκέντρωσης και η σύνταξη μίας πρότασης αρχιτεκτονικής ενός κρυπτονομίσματος και τελικά η δημιουργία μίας εφαρμογής PoC που να τηρεί τις βασικές αρχές της πρότασης. Ετσι δημιουργήθηκε μία αποκεντρωμένη εφαρμογή που αποτελεί πλατφόρμα πραγματοποιήσης συναλλαγών και έτσι είναι δυνατή η επίδειξη των λειτουργικοτήτων.

Το Blockchain είναι μία νέα τεχνολογία, η οποία βρίσκεται σε ταχεία ανάπτυξη και ¨ψάχνει' να βρει τη θέση της στο ψηφιακό σκηνικό. Συνέχεια δημιουργούνται νέες υλοποιήσεις εφαρμογών με βάση το Blockchain και φυσικά γεννιούνται συνεχώς νέα κρυπτονομίσματα. Αυτό έχει ως συνέπεια να υπάρχει διαρκής εξέλιξη, αλλαγή και συχνά κατάργηση εργαλείων των εφαρμογών. Παράλληλα δεν καθίσταται εύκολο να υπάρχει υποστήριξη από μία κοινότητα που ασχολείται με τον κλάδο της τεχνολογίας των κρυπτονομισμάτων.

Εξετάστηκαν πολλές διαφορετικές τεχνολογίες και υλοποιήσεις κρυπτονομισμάτων. Αναλυτικότερα μελετήθηκε το κρυπτονόμισμα Bitcoin τόσο σαν δομή όσο και σαν πηγαίος κώδικας, το κρυπτονόμισμα Ethereum που παρέχει το εργαλείο geth και αρνητικό ότι είναι γραμμένο στη γλώσσα Go, το κρυπτονόμισμα Cardano και φυσικά το Semux. Τελικά επιλέχθηκε το τελευταίο καθώς είναι γραμμένο στη γλώσσα Java, μία γλώσσα που κατέχω, καθώς και εκδίδει APIs με αποτέλεσμα να είναι εύκολη η υλοποίηση scripts που επικοινωνούν με την εφαρμογή.

8.2 Μελλοντικές επεκτάσεις

Στην ενότητα αυτή θα παρουσιαστούν μερικές μελλοντικές επεκτάσεις της παρούσας διπλωματικής. Οι αποκεντρωμένες εφαρμογές που βασίζονται στην τεχνολογία του Blockchain δείχνουν να έχουν μέλλον και αξίζει κανείς να επενδύσει σε αυτές. Ο λόγος είναι οτι τέτοιου είδους εφαρμογές έχουν τη δυναμική να αλλάξουν σε μεγάλο βαθμό αρκετούς τομείς της ζωής και συνεπώς έχουν τραβήξει το ενδιαφέρον πολλών ερευνητών, εταιρειών και επενδυτών.

Αρχικά, δεδομένου ότι η εφαρμογή Semux στοχεύει στην υποστήριξη σε μελλοντικές

εκδόσεις σε ¨έξυπνα' συμβόλαια. Αυτή η λειτουργικότητα θα μπορούσε να ενταχθεί και στην εφαρμογή ntuaSemux στη βάση ότι η νόρμα των κρυπτονομισμάτων θέλει την ένταξη της, εξυπηρετώντας τις ανάγκες των χρηστών τους.

Ακόμα, θα μπορούσε να γίνει μία διεύρυνση του πεδίου της εφαρμογής και να μην περιορίζεται μόνο στις χρηματικές συναλλαγές μεταξύ χρηστών.

Δεδομένου ότι η αποκέντρωση είναι trending θα μπορούσε να υποστηρίζει μία λειτουργία τύπου Dropbox /GoogleDrive , δηλαδή ο κάθενας να μπορεί να αποθηκεύει στο 'cloud' της εφαρμογής τα δεδομένα του. Αυτό σημαίνει ότι δεν υπάρχει κεντρικός πάροχος της πληροφορίας ο οποίος πιθανώς την εκμεταλλεύεται. Παράλληλα όμως μία τέτοια επέκταση πρέπει να πληροί συγκεκριμένες προδιαγραφές, ώστε τα προσωπικά δεδομένα να μην είναι ευάλωτα και να είναι πάντα διαθέσιμα. Ένας νέος ορίζοντας εφαρμογών προδιαγράφεται.

Τέλος, θα μπορούσε να επεκταθεί σαν μία εφαρμογή συναλλαγών τρίτων κρυπτονομισμάτων cryptocurrency-exchange, και να αποτελεί έγκυρη και έμπιστη αρχή για τις συναλλαγές. Οι συναλλαγές αυτές θα είναι πλήρως αποκεντρωμένες και διαθέσιμες στο κοινό. Παράλληλα θα καλύψει το κενό της αγοράς για τη διαφάνεια τέτοιου είδους συναλλαγών και ανάλυσης του καθορισμού της τιμής.

8.3 Προσωπικό σχόλιο

Η πρόταση της αρχιτεκτονικής που αναλύεται σε προηγούμενα κεφάλαια δείχνει να είναι πολλά υποσχόμενη και να αντιμετωπίζει κάποια από τα προβλήματα των υπάρχοντων υλοποιήσεων. Για να γίνει αντιληπτή η αξία της, θα πρέπει να γίνει μία πλήρης υλοποίηση του πρωτοκόλλου που προτείνεται ώστε να δοκιμαστεί από το κοινό και να τελικά να φανούν πιθανά πλεονεκτήματα η μειονεκτήματα που επαληθεύουν η απορρίπτουν τα όποια πειράματα η υπολογισμούς γίνονται σε ¨αποστειρωμένο' -εργαστηριακό περιβάλλον. Αυτή η πρόταση, γίνεται και ως κινητοποίηση της κοινότητας που ασχολείται με τη τεχνολογία του Blockchain με σκοπό την αλλαγή της πορείας πλεύσης προς μία πιο αποδοτική, λιγότερο απαιτητική σε πόρους λύση. Δεν είναι απαραίτητα αυτή η πρόταση η βέλτιστη λύση, ωστόσο είναι μια προσπάθεια για την αναζήτηση της.

Βιβλιογραφία

[1] Apache maven. https://en.wikipedia.org/wiki/Apache_Maven. Accessed: 2019-08-30.

[2] Apache maven project. https://maven.apache.org/. Accessed: 2019-08-30.

[3] Bernstein - Blockchain for intellectual property. https://www.bernstein.io/. Accessed: 2019-08-30.

[4] Bitcoin. https://en.wikipedia.org/wiki/Bitcoin. Accessed: 2019-08-30.

[5] Bitcoin cryptocurrency. https://bitcoin.org/el/. Accessed: 2019-08-30.

[6] Blockchain 2019 Legislation. http://www.ncsl.org/research/financial-services-and-commerce/blockchain-2019-legislation.aspx. Accessed: 2019-08-30.

[7] Cardano cryptocurrency. https://www.cardano.org/en/home/. Accessed: 2019-08-30.

[8] Cryptography. https://en.wikipedia.org/wiki/Cryptography. Accessed: 2019-08-30.

[9] Cryptokitties. https://www.cryptokitties.co/. Accessed: 2019-08-30.

[10] Distributed computing. https://en.wikipedia.org/wiki/Distributed_computing. Accessed: 2019-08-30.

[11] Eclipse foundation. https://www.eclipse.org/. Accessed: 2019-08-30.

[12] Ethereum cryptocurrency. https://www.ethereum.org/. Accessed: 2019-08-30.

[13] Fork (blockchain). https://en.wikipedia.org/wiki/Fork_(blockchain). Accessed: 2019-08-30.

[14] Game theory. https://en.wikipedia.org/wiki/Game_theory. Accessed: 2019-08-30.

[15] History of the World Wide Web. https://en.wikipedia.org/wiki/History_of_the_World_Wide_Web. Accessed: 2019-08-30.

[16] Holochain. `https://holochain.org`. Accessed: 2019-08-30.

[17] Introduction to Security and Privacy on the Blockchain, αυτηορ=halpin, harry and piekarska, marta, βοοκτιτλε=2017 ieee european symposium on security and privacy workshops (euros&pw), παγεσ=1–3, ψεαρ=2017, οργανιζατιον=ieee.

[18] Java 8 Oracle. `https://www.oracle.com/technetwork/java/javase/overview/java8-2100321.html`. Accessed: 2019-08-30.

[19] Java programming language, ηοωπυβλισηεδ = `https://en.wikipedia.org/wiki/java_(programming_language)`, νοτε = accessed: 2019-08-30.

[20] Round off error. `https://en.wikipedia.org/wiki/Round-off_error`. Accessed: 2019-08-30.

[21] Satoshi Nakamoto. `https://en.wikipedia.org/wiki/Satoshi_Nakamoto`. Accessed: 2019-08-30.

[22] Semux cryptocurrency. `https://www.semux.org/`. Accessed: 2019-08-30.

[23] Semux source code. `https://github.com/semuxproject/semux-core`. Accessed: 2019-08-30.

[24] Size of the Bitcoin blockchain from 2010 to 2019, by quarter (in megabytes) . `https://www.statista.com/statistics/647523/worldwide-bitcoin-blockchain-size`. Accessed: 2019-08-30.

[25] Software engineering . `https://en.wikipedia.org/wiki/Software_engineering`. Accessed: 2019-08-30.

[26] Statoshi- Real time Bitcoin Nodes stats. `https://statoshi.info`. Accessed: 2019-08-30.

[27] Visa. `https://usa.visa.com/run-your-business/small-business-tools/retail.html`. Accessed: 2019-08-30.

[28] Baliga, Arati. Understanding blockchain consensus models. Ιν *Persistent*. 2017.

[29] Bayer, Dave and Haber, Stuart and Stornetta, W Scott. Improving the efficiency and reliability of digital time-stamping. Ιν *Sequences Ii*, παγες 329–334. Springer, 1993.

[30] Buterin, Vitalik and others. A next-generation smart contract and decentralized application platform. *white paper*, 3:37, 2014.

[31] Cachin, Christian. Architecture of the hyperledger blockchain fabric. Ιν *Workshop on distributed cryptocurrencies and consensus ledgers*, ολυμε 310, παγε 4, 2016.

[32] Chohan, Usman W. The Double Spending Problem and Cryptocurrencies. *Available at SSRN 3090174*, 2017.

[33] Crespo, Arturo and Garcia-Molina, Hector. Semantic overlay networks for p2p systems. Iν *International Workshop on Agents and P2P Computing*, παγες 1–13. Springer, 2004.

[34] Dabek, Frank and Brunskill, Emma and Kaashoek, M Frans and Karger, David and Morris, Robert and Stoica, Ion and Balakrishnan, Hari. Building peer-to-peer systems with Chord, a distributed lookup service. Iν *Proceedings Eighth Workshop on Hot Topics in Operating Systems*, παγες 81–86. IEEE, 2001.

[35] De Angelis, Stefano and Aniello, Leonardo and Baldoni, Roberto and Lombardi, Federico and Margheri, Andrea and Sassone, Vladimiro. Pbft vs proof-of-authority: applying the cap theorem to permissioned blockchain. 2018.

[36] Decker, Christian and Wattenhofer, Roger. Information propagation in the bitcoin network. Iν *IEEE P2P 2013 Proceedings*, παγες 1–10. IEEE, 2013.

[37] Fulmer, Nathan. Exploring the Legal Issues of Blockchain Applications. *Akron L. Rev.*, 52:161–186, 2018.

[38] Furness, Jamie and Kolberg, Mario. The Effects of Churn on Complex Search Techniques.

[39] Haber, Stuart A and Stornetta Jr, Wakefield S. Method for secure time-stamping of digital documents, Αυγ. 4 1992. US Patent 5,136,647.

[40] Kiayias, Aggelos and Russell, Alexander and David, Bernardo and Oliynykov, Roman. Ouroboros: A provably secure proof-of-stake blockchain protocol. Iν *Annual International Cryptology Conference*, παγες 357–388. Springer, 2017.

[41] Korpela, Kari and Hallikas, Jukka and Dahlberg, Tomi. Digital supply chain transformation toward blockchain integration. Iν *proceedings of the 50th Hawaii international conference on system sciences*, 2017.

[42] Lamport, Leslie and Shostak, Robert and Pease, Marshall. The Byzantine generals problem. *ACM Transactions on Programming Languages and Systems (TOPLAS)*, 4(3):382–401, 1982.

[43] Li, Jingming and Li, Nianping and Peng, Jinqing and Cui, Haijiao and Wu, Zhibin. Energy consumption of cryptocurrency mining: A study of electricity consumption in mining cryptocurrencies. *Energy*, 168:160–168, 2019.

[44] Mahajan, Prerna and Sachdeva, Abhishek. A study of encryption algorithms AES, DES and RSA for security. *Global Journal of Computer Science and Technology*, 2013.

[45] Nakamoto, Satoshi and others. Bitcoin: A peer-to-peer electronic cash system. 2008.

[46] Savelyev, Alexander. Contract law 2.0:'Smart'contracts as the beginning of the end of classic contract law. *Information & Communications Technology Law*, 26(2):116–134, 2017.

[47] Stoica, Ion and Morris, Robert and Karger, David and Kaashoek, M Frans and Balakrishnan, Hari. Chord: A scalable peer-to-peer lookup service for internet applications. *ACM SIGCOMM Computer Communication Review*, 31(4):149–160, 2001.

[48] Truby, Jon. Decarbonizing Bitcoin: Law and policy choices for reducing the energy consumption of Blockchain technologies and digital currencies. *Energy research & social science*, 44:399–410, 2018.

Γλωσσάριο

Ελληνικός όρος	Αγγλικός όρος
κομμάτι	block
Συνεπής συνάρτηση κατακερματισμού	Consistent Hashing function
Πρωτόκολλο συμφωνίας	Consensus protocol
Αυτο-κλιμακωσιμότητα	Self-scalability
Αλυσίδα συστοιχιών	Blockchain
Συστοιχία της Γένεσης	Genesis block
Το πρόβλημα των βυζαντινών στρατηγών	Byzantine Generals problem
ιδιωτικότητα	privacy
κάνναβος	grid
παράθυρο	window
συνάθροιση	aggregation
χρονόσημο	timestamp