Κατανεμημένα Συστήματα

Ραπτόπουλος Πέτρος
Τσένος Γρηγόρης
Σαφός Κωνσταντίνος

Εισαγωγή

Το BlockChat είναι μια πλατφόρμα ανταλλαγής μηνυμάτων και καταγραφής δοσοληψιών που στηρίζεται σε ένα απλό blockchain, παρέχοντας στους χρήστες ασφαλή και αξιόπιστη αποστολή νομισμάτων και καταγραφή των μηνυμάτων που ανταλλάσσονται μεταξύ των συμμετεχόντων.

Ο κάθε χρήστης/κόμβος του BlockChat έχει ένα BlockChat wallet με coins, τα BCC (BlockChat Coins) που είναι απαραίτητα για να πραγματοποιεί transactions (αποστολές νομισμάτων ή μηνυμάτων). Το κάθε wallet αποτελείται από:
(α) ένα private key γνωστό μόνο στον χρήστη-κάτοχο του wallet, που του επιτρέπει να στέλνει νομίσματα/μηνύματα σε άλλους χρήστες και
(β) το αντίστοιχό public key, που απαιτείται για να λάβει ένας χρήστης νομίσματα ή μηνύματα από άλλον χρήστη. Το public key είναι η διεύθυνση του wallet του χρήστη αλλά ένας χρήστης αρκεί να γνωρίζει απλά το id ενός άλλου για να επικοινωνήσει μαζί του. Το σύστημα συνδέει το id με το αντίστοιχο public key αυτόματα

Ένας χρήστης κάνει συναλλαγές ξοδεύοντας BCC και υπογράφοντας τη συναλλαγή με το private key του. Κατά τη μεταφορά νομισμάτων, ο αποστολέας χρεώνεται 3% του ποσού που στέλνει. Στην αποστολή μηνυμάτων, ο αποστολέας χρεώνεται 1BCC ανά χαρακτήρα. Ο χρήστης μπορεί σε μια συναλλαγή να μεταφέρει νομίσματα και να στείλει μήνυμα, ταυτόχρονα, με τη κανονική χρέωση (3% του ποσού που στέλνει συν 1BCC ανά χαρακτήρα του μηνύματος).

Η κάθε συναλλαγή που δημιουργείται γίνεται broadcast σε όλο το δίκτυο BlockChat και οι κόμβοι που συμμετέχουν στη διαδικασία της επικύρωσης των blocks, μόλις λάβουν τη νέα συναλλαγή, την

επικυρώνουν, πρώτα ελέγχοντας από ποιόν χρήστη προήλθε η συναλλαγή χρησιμοποιώντας το public key, και πιστοποιώντας πως υπάρχουν αρκετά νομίσματα στο wallet του αποστολέα για να εκτελέσει η συναλλαγή. Αν όλοι οι έλεγχοι επιτύχουν, η συναλλαγή προστίθεται στο τρέχον block. Όταν το τρέχον block γεμίσει, τότε οι κόμβοι ξεκινούν τη διαδικασία της επικύρωσης ακολουθώντας αλγόριθμο Proof-of-Stake. Όποιος τελικά επικυρώσει το block, το κάνει broadcast σε όλο το δίκτυο, ώστε να γίνει γνωστό σε όλους τους κόμβους που συμμετέχουν. Ο επικυρωτής του block είναι και αυτός στου οποίου το wallet πιστώνονται οι χρεώσεις των συναλλαγών που έχουν συμπεριληφθεί στο block.

Εκκίνηση και Δίκτυο

Η επικοινωνία στο δίκτυο έχει υλοποιηθεί ως REST API με χρήση Flask.

Ο χρήστης ξεκινά ένα κόμβο με την εντολή:
python src/run.py -p <PORT> -n <NODES> -capacity <CAPACITY> (στον πρώτο κόμβο βάζει και argument -bootstrap), όπου
<PORT> το port το οποίο αντιστοιχεί στον συγκεκριμένο κόμβο
<NODES> το συνολικό πλήθος των κόμβων που θα υπάρχουν στο δίκτυο, και
<CAPACITY> το πλήθος των συναλλαγών που καθιστούν ένα block στην αλυσίδα

Το μέγεθος του δικτύου και το capacity των block πρέπει να είναι γνωστά από πριν και δίνονται ως arguments από τον κάθε χρήστη τη στιγμή που δημιουργείται ο κάθε κόμβος.

Ο πρώτος κόμβος που μπαίνει στο δίκτυο πρέπει να είναι ο bootstrap. Αφού δημιουργήσει το wallet του και αποκτήσει public και private key, αυτός γράφεται μόνος του στη λίστα με τους κόμβους, έχει id = 0, και δημιουργεί το πρώτο block με μια μόνο συναλλαγή κατά την οποία δίνει στον εαυτό του το πλήρες σύνολο όλων των νομισμάτων του δικτύου, 1000 * τον αριθμό των χρηστών (συμπεριλαμβανομένου του ίδιου). Έπειτα προσθέτει το genesis block, αυτό, στην αλυσίδα (χωρίς επικύρωση) και αρχίζει να ακούει στο port του, περιμένοντας τους υπόλοιπους κόμβους για να τους καταγράψει και να τους στείλει το id τους.

Όταν μπαίνει ένας άλλος κόμβος στο δίκτυο στέλνει στον bootstrap το public key του και λαμβάνει το id που του αντιστοιχεί. Αν κάποιος κόμβος προσπαθήσει να μπει στο δίκτυο πριν τον bootstrap, δε θα τα καταφέρει αφού δε θα υπάρχει ο bootstrap για να τον καταγράψει.

Όταν εισαχθούν όλοι οι κόμβοι, ο bootstrap κάνει broadcast σε όλους το blockchain όπως έχει διαμορφωθεί μέχρι εκείνη τη στιγμή, τα ζεύγη ip address/port καθώς και τα public keys των wallets όλων των κόμβων που συμμετέχουν στο σύστημα. Για κάθε έναν από τους άλλους κόμβους στο δίκτυο, ο bootstrap κόμβος εκτελεί ένα transaction όπου του μεταφέρει 1000 BCC. Έτσι, μετά την εισαγωγή όλων των κόμβων, ο καθένας τους έχει 1000 BCC στο wallet του και πλέον μπορούν να αρχίσουν τα transactions στο δίκτυο. Επιπλέον κόμβοι που θα προσπαθήσουν να εισέλθουν λαμβάνουν μήνυμα σφάλματος πως το δίκτυο είναι γεμάτο. Θεωρούμε πως δεν υπάρχουν αποχωρήσεις κόμβων από το σύστημα.

Blocks

Ένα block είναι μια λίστα από transactions που αποθηκεύεται στο blockchain, μια λίστα από blocks. Τα transactions σε κάθε block ισούνται με το capacity που ορίζεται κατά την δημιουργία των κόμβων. Όλα τα blocks έχουν το ίδιο capacity, και άρα αριθμό transactions, εκτός από το genesis block. Πέρα από τα transactions, το κάθε block έχει επιπλέον τις εξής πληροφορίες:

- index: ο αύξων αριθμός του block,

- timestamp: η χρονική στιγμή της δημιουργίας του block,
- validator: το public key του κόμβου που επικύρωσε το block,
- current_hash: το hash του block,
- previous_hash: το hash του προηγούμενου block στο blockchain.

Wallet

Στον κάθε κόμβο αντιστοιχεί ένα wallet. Ένα wallet συνδέεται με ένα ζεύγος public/private key. Το public key δρα ως διεύθυνση του wallet, την οποία μπορεί κανείς να μοιραστεί με οποιονδήποτε προκειμένου να μπορεί να δεχτεί μηνύματα. Το private key χρησιμοποιείται για να υπογράφονται τα transactions, ώστε να εξασφαλίζεται ότι μόνο ο κάτοχος του wallet μπορεί να ξοδέψει χρήματα από το wallet στέλνοντας κάποιο μήνυμα. Αυτό γίνεται με τη συνάρτηση sign_transaction(). Οποιοσδήποτε γνωρίζει το public key ενός wallet μπορεί να επαληθεύσει ότι ένα transaction που με αποστολέα το wallet αυτό έχει δημιουργηθεί από τον κάτοχό του με την verify_transaction(). Στο wallet είναι αποθηκευμένη μια λίστα με όλα τα transactions που αφορούν τον κόμβο-ιδιοκτήτη.

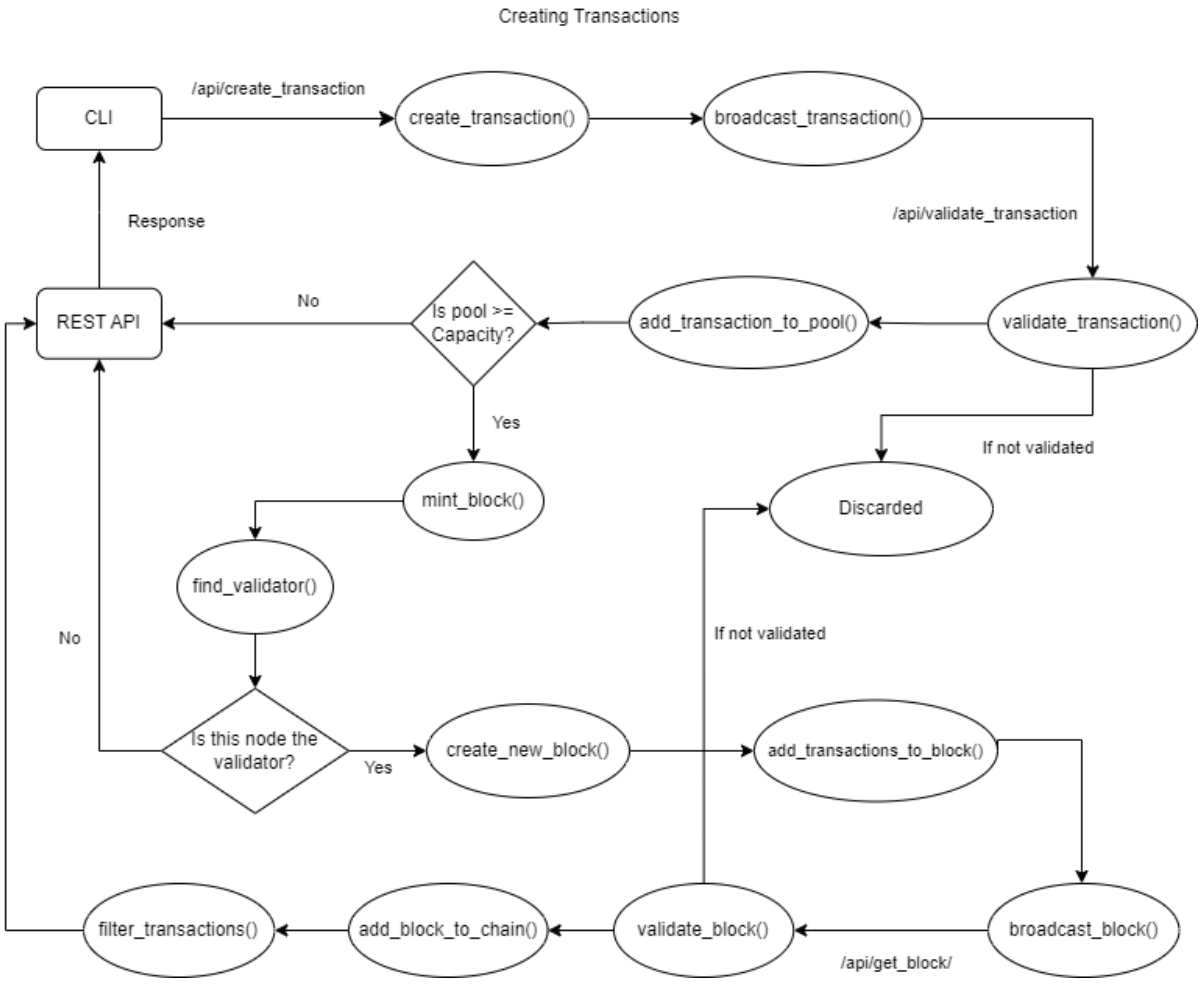
Transactions

Κάθε transaction περιέχει πληροφορίες για την αποστολή νομισμάτων και μηνύματος από ένα wallet σε ένα άλλο. Οι πληροφορίες που περιλαμβάνει είναι:

- sender_address: Το public key του wallet από το οποίο προέρχεται το μήνυμα.
- receiver_address: Το public key του wallet-παραλήπτη του μηνύματος.
- amount: το ποσό των νομισμάτων προς αποστολή (δε συμπεριλαμβάνει τις χρεώσεις). Αν ο χρήστης θέλει να στείλει μόνο μήνυμα μπορεί να το θέσει 0.
- message: το String του μηνύματος που στέλνεται. Αν ο χρήστης θέλει να στείλει μόνο νομίσματα μπορεί να το παραλείψει (defaults to an empty string "")
- nonce: μετρητής που διατηρείται ανά αποστολέα και αυξάνεται κατά 1 μετά από κάθε αποστολή
- transaction_id: το hash του transaction
- Signature: Υπογραφή που αποδεικνύει ότι ο κάτοχος του wallet δημιούργησε αυτό το transaction.
- TTL: Ο χρόνος ζωής του transaction. Ισούται με το index του πιο πρόσφατου block στην αλυσίδα του κόμβου τη στιγμή της δημιουργίας του transaction. Αυτό συγκρίνεται με το index το τωρινού πιο πρόσφατου block και αν το transaction δεν επικυρωθεί εγκαίρως, απορρίπτεται.

Ένα transaction μπορεί να δημιουργηθεί από τον κάτοχο του wallet, ο οποίος θα χρεωθεί το amount που θέλει να στείλει, το κόστος του μηνύματος και τις χρεώσεις. Κάθε transaction γίνεται broadcast σε όλα τα μέλη του blockchain και κάθε κόμβος το προσθέτει στο transaction pool του. Κατά τη λήψη ενός transaction από οποιονδήποτε node, καλείται η συνάρτηση validate_transaction που ελέγχει την ορθότητά του.

Για την πρόληψη επιθέσεων επανάληψης (replay attacks), όπου κάποιος κακόβουλος κόμβος θα μπορούσε να ξαναστείλει ένα transaction που έλαβε για να χρεώσει ξανά τον αποστολέα, κάθε συναλλαγή περιέχει ένα πεδίο nonce. Το nonce είναι ένας μετρητής που διατηρεί κάθε λογαριασμός στο BlockChat και που αυξάνεται κατά ένα με κάθε εξερχόμενη συναλλαγή. Αυτό εμποδίζει την υποβολή της ίδιας συναλλαγής περισσότερες από μία φορές στο δίκτυο.



Κόμβοι

Κάθε κόμβος χαρακτηρίζεται από το unique id τους. Ο bootstrap κόμβος έχει id = 0. Σε κάθε κόμβο αντιστοιχεί ένα wallet για το οποίο μόνο αυτός ο κόμβος έχει το private key. Στο wallet είναι αποθηκευμένη μια λίστα με όλα τα transactions που αφορούν τον συγκεκριμένο κόμβο. Ο κάθε κόμβος επίσης έχει αποθηκευμένο τον αριθμό των συναλλαγών που καθιστούν ένα block στην αλυσίδα (capacity) και ένα μετρητή των συναλλαγών που έχει πραγματοποιήσει αυτός ο κόμβος (send_counter). Τέλος, ο κάθε κόμβος κρατά και κάποιες πληροφορίες που αφορούν το δίκτυο γενικότερα. Συγκεκριμένα έχουν:

- Chain: Ένα αντίγραφο του blockchain όπως είναι τη παρούσα στιγμή.
- transaction_pool: Μια ουρά που περιέχει όλα τα transactions που έχει επικυρώσει ο κόμβος αλλά δεν έχουν μπει ακόμα σε κάποιο block της αλυσίδας. Όταν ο κόμβος λαμβάνει ένα καινούριο block, αφαιρεί από αυτή την ουρά τα transactions που έχουν καταγραφεί στο νέο block.
- chainState_ring και softState_ring: Δύο λίστες με τις πληροφορίες των υπόλοιπων κόμβων του δικτύου (id, ip, port, public_key, balance, stake (όταν ανανεώνεται, ανανεώνεται και το διαθέσιμο υπόλοιπο του λογαριασμού), nonces (Το nonces είναι μια λίστα που κρατά όλα τα nonces που έχει δει ο κόμβος)).
  Η chainState_ring αντιστοιχεί στη κατάσταση αμέσως μετά το τελευταίο block της αλυσίδας οπότε, ενώ οι πληροφορίες της είναι επικυρωμένες, δεν είναι σίγουρα επίκαιρες.

Οι πληροφορίες της softState_ring αντικατοπτρίζουν και τις αλλαγές που προκύπτουν από τα transactions του transaction pool του κόμβου, τα οποία δεν έχουν προστεθεί στην αλυσίδα ακόμα. Αυτό σημαίνει πως οι πληροφορίες της, ενώ είναι επίκαιρες, **δεν** είναι επικυρωμένες.

Ακολουθούμε το μοντέλο λογαριασμών (account model) για τη διατήρηση των δεδομένων κατάστασης (state) του blockchain. Όπως είδαμε, ο κάθε κόμβος διατηρεί δύο λίστες με τους υπάρχοντες λογαριασμούς και τα υπόλοιπά τους, μια για τη κατάσταση της αλυσίδας και μια για το soft state του δικτύου όπως το βλέπει ο κόμβος μέσα από τα εισερχόμενα, αλλά όχι ακόμα επικυρωμένα, transactions. Με κάθε συναλλαγή, οι κόμβοι που τη λαμβάνουν μειώνουν το υπόλοιπο του λογαριασμού του αποστολέα στο softState_ring σύμφωνα με το ποσό των νομισμάτων που επιθυμεί να στείλει, την χρέωση στην αποστολή, και με τον κανόνα χρέωσης των μηνυμάτων (1BCC για κάθε χαρακτήρα).

Όταν ένας κόμβος λαμβάνει ένα transaction, ξεκινά να το κάνει validate. Πρώτα κάνει verify την υπογραφή, στη συνέχεια επιβεβαιώνει πως ο αποστολέας έχει αρκετό υπόλοιπο κοιτώντας το softState_ring και τέλος σιγουρεύει πως δεν έχει ξαναδεί το nonce του transaction. Αν το transaction γίνει validated, το προσθέτει στο transaction_pool του και ανανεώνει το softState_ring. Αλλιώς, το transaction απορρίπτεται.

Μετά τη λήψη ενός νέου block, ο κάθε κόμβος το επικυρώνει. Επικυρώνει πως το hash του block είναι το αναμενόμενο και πως έχει σωστό hash για το προηγούμενο block. Έπειτα επικυρώνει όλα τα transactions του block με βάση το chainState_ring για να σιγουρευτεί πως τα transactions του block μπορούν να ολοκληρωθούν και όλες οι χρεώσεις να πληρωθούν. Τέλος ο κάθε κόμβος αυξάνει το υπόλοιπο του validator σύμφωνα με τα συνολικές χρεώσεις που χρεώθηκαν οι αποστολείς των transactions που συμπεριλαμβάνονται στο block, ανανεώνει το chainState_ring του, και αφαιρεί από το transaction_pool του τα transactions που επικυρώθηκαν με το νέο block. Επίσης, με τη λήψη του νέου block, το softState_ring ανανεώνεται και γίνεται ίσο με το chainState_ring. Αν έχουν μείνει transactions στο transaction_pool, ο κόμβος επικυρώνει ξανά αυτά τα transactions και ενημερώνει το softState_ring.

Όταν ένας κόμβος λαμβάνει ένα transaction με το οποίο το transaction_pool του έχει αρκετά transactions για να συμπληρωθεί block, ξεκινά τη διαδικασία Proof-of-Stake για να βρει τον validator του νέου block. Αν ο ίδιος είναι ο validator, κάνει mine το block και το κάνει broadcast σε όλο το δίκτυο. Αν δεν είναι, απλά το προσθέτει στο pool και δεν κάνει τίποτα άλλο.

Proof-of-Stake

Το Proof of Stake (PoS) είναι ο μηχανισμός consensus που χρησιμοποιεί το Blockchat για την εκλογή ενός validator για κάθε block της αλυσίδας. Στο PoS, το ποιος θα επικυρώσει το επόμενο μπλοκ δεν εξαρτάται από την υπολογιστική δύναμη, αλλά από το ποσό κρυπτονομίσματος (staking) που κατέχει ένας κόμβος.

Στην πράξη, κάθε κόμβος που θέλει να συμμετάσχει στη διαδικασία του validation, δεσμεύει για τον σκοπό αυτό το ποσό που θέλει από το πορτοφόλι του με τη συνάρτηση stake(amount). Αυτή η κλήση μεταφράζεται σε ένα transaction του οποίου το receiver_address είναι 0 και το ποσό αυτό που θέλει να δεσμεύσει ο κάθε κόμβος. Κατά τη λήψη τέτοιων transactions ελέγχεται, εκτός των άλλων, ότι υπάρχει το staking ποσό στο wallet του αποστολέα.

Η πιθανότητα να είναι ο επόμενος validator είναι ανάλογη του ποσού που δέσμευσε σε σχέση με το συνολικό ποσό που έχει δεσμευτεί από όλους τους πιθανούς validators. Αν επομένως ένας κόμβος έχει το 5% του συνολικού staking ποσού, τότε έχει και 5% πιθανότητα να επιλεγεί ως validator για το επόμενο block.

Η διαδικασία είναι η εξής: Μόλις συμπληρωθεί το τρέχον block με <capacity> transactions, τότε ο κόμβος καλεί μια ψευδοτυχαία γεννήτρια αριθμών που θα του υποδείξει ποιος θα είναι ο validator του block. Για

να είναι ντετερμινιστικό το αποτέλεσμα της κλήσης της από οποιονδήποτε κόμβο, πρέπει να χρησιμοποιηθεί από όλους τους κόμβους το ίδιο seed, το οποίο καθορίζεται από το hash του προηγούμενου block της αλυσίδας. Ο κόμβος που αναγνωρίζει τον εαυτό του ως validator επικυρώνει το block και το στέλνει με broadcast σε όλους. Για την εύρεση του validator από την ψευδοτυχαία γεννήτρια με βάση τις πιθανότητες του κάθε κόμβου να επιλεγεί, ακολουθούμε τη λογική της «λοτταρίας»: Ο κάθε κόμβος βάζει στον κουβά τόσους λαχνούς, όσα BCC έχει δεσμεύσει, δίνοντάς τους διαδοχικούς αριθμούς. Η γεννήτρια τραβάει έναν λαχνό, δηλαδή ένα νούμερο από το 1 μέχρι τον συνολικό αριθμό των δεσμευμένων BCC και έτσι θα καθορίσει τον επόμενο validator.

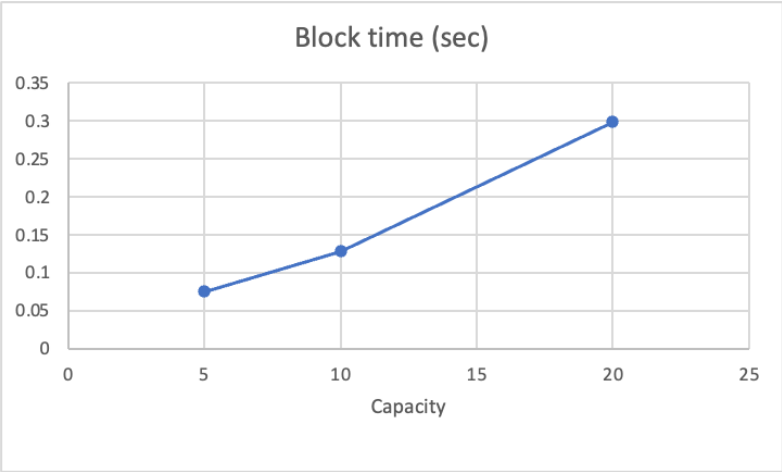
Πειράματα

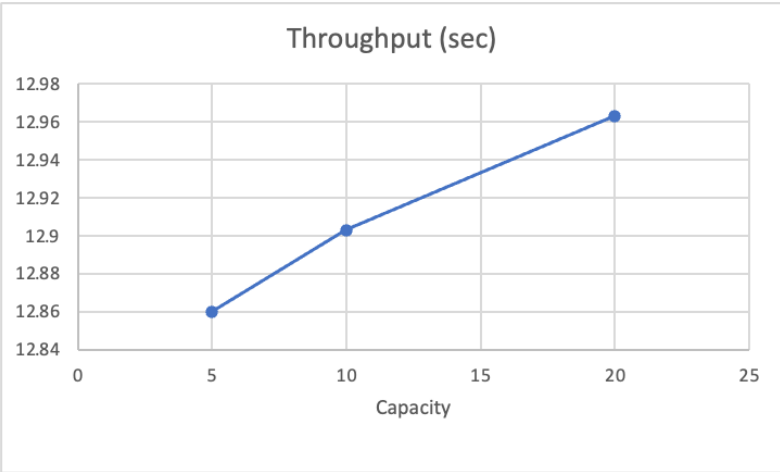
1) Απόδοση του Συστήματος

Στήνουμε ένα BlockChat με 5 clients. Αφού όλοι οι κόμβοι εισέλθουν στο σύστημα, ο καθένας διαβάζει ένα αρχείο transX.txt, όπου X το node id του κόμβου. Το κάθε αρχείο περιέχει 100 μηνύματα προς άλλους κόμβους με τη μορφή <recipient_node_id> <message> και έτσι ο κάθε κόμβος δημιουργεί και στέλνει στο δίκτυο 100 transactions. Με σταθερό staking 10 BCC για κάθε κόμβο και capacity 5, 10 και 20 καταγράφουμε το Throughput (ρυθμαπόδοση) του συστήματος, δηλαδή πόσα transactions εξυπηρετούνται στην μονάδα του χρόνου (transactions/sec), και το Block time, δηλαδή τον μέσο χρόνο που απαιτείται για να προστεθεί ένα νέο block στο blockchain (time/block):

Capacity	5	10	20
Throughput	12.860	12.903	12.963
Block time	0.075	0.1282	0.2986

Παρατηρούμε πως η ρυθμαπόδοση παραμένει σχετικά σταθερή (διαφορά 0.1 transactions/sec) ενώ το Block time είναι ανάλογο του capacity (χρειάζεται διπλάσιος χρόνος για να γεμίσει ένα διπλάσιο block)

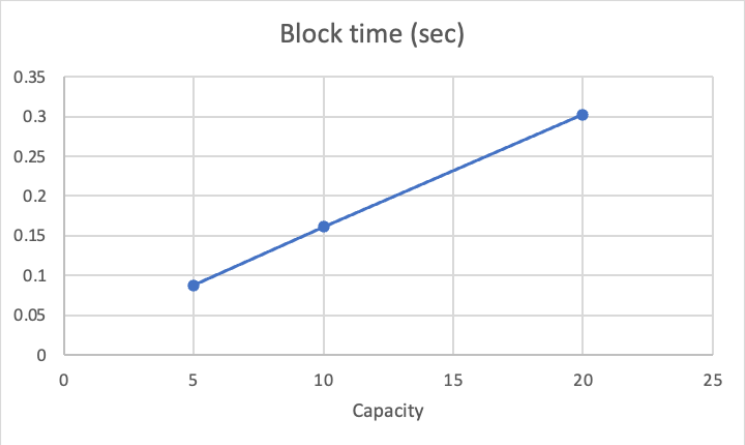


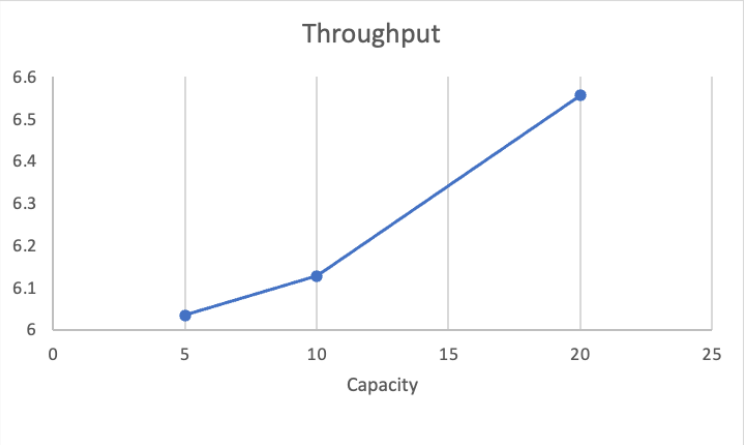


2) Κλιμακωσιμότητα του Συστήματος

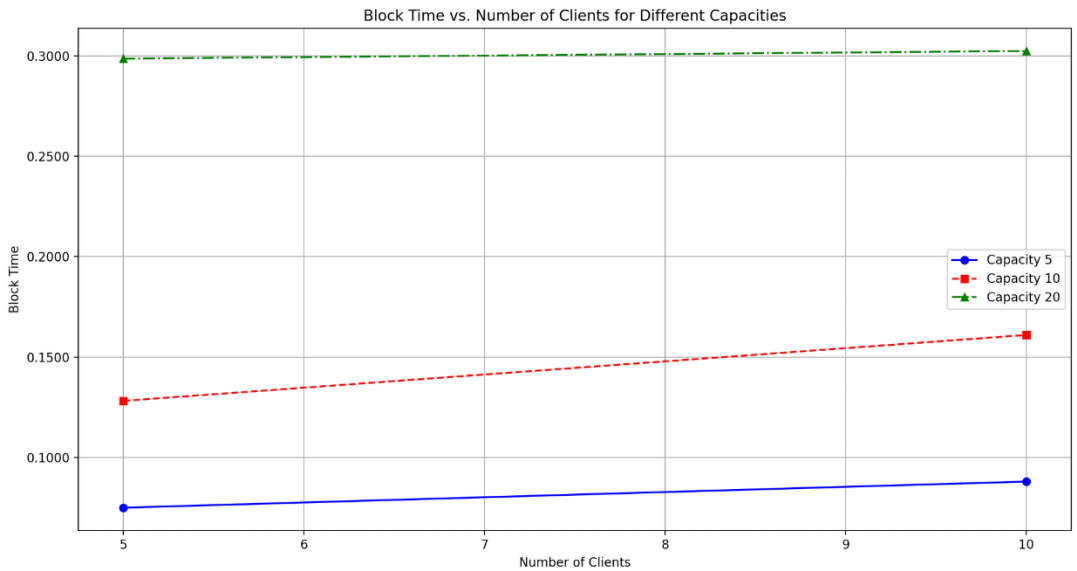
Επαναλαμβάνουμε το πείραμα, αυτή τη φορά με 10 clients και συγκρίνουμε με τα προηγούμενα αποτελέσματα:

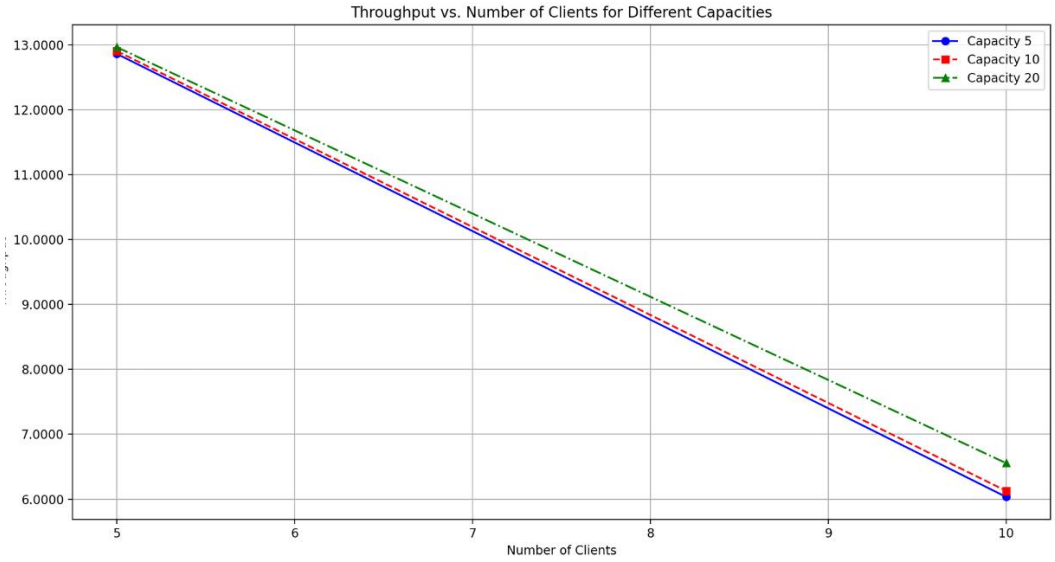
Capacity	5	10	20
Throughput	6.035	6.1274	6.5552
Block time	0.088	0.161	0.3024





Παρατηρούμε πως, πάλι το Block time είναι ανάλογο του capacity αλλά και ότι με την αύξηση του capacity τώρα, που έχουμε περισσότερους κόμβους, αυξάνεται και το throughput, καθώς η αυξημένη κίνηση στο δίκτυο σε συνδυασμό με το χαμηλό capacity οδηγούν σε συνεχείς κλήσεις των συναρτήσεων mint_block και get_block και άρα σε καθυστέρηση στη δημιουργία νέων transactions.





3) Δικαιοσύνη

Επαναλαμβάνουμε το πείραμα με τους 5 κόμβους και capacity 5, μόνο που αυτή τη φορά ένας από τους κόμβους έχει staking 100 BCC. Επειδή οι χρεώσεις πιστώνονται στον validator και ένας κόμβος έχει 10-πλάσιο stake από τους άλλους και άρα επιλέγεται πολύ πιο συχνά, παρατηρούμε πως σιγά-σιγά όλα τα νομίσματα του συστήματος τείνουν να μαζεύονται στον κόμβο αυτό, ενώ οι υπόλοιποι κόμβοι δεν έχουν, και ούτε καταφέρνουν ποτέ να αποκτήσουν, καν αρκετά νομίσματα για να πραγματοποιήσουν τα 100 transactions του πειράματος.