

## Εξαμηνιαία Εργασία στα Κατανεμημένα Συστήματα - BlockChat

Κακούρης Δημήτριος (03119019)  
Μπάρλου Όλγα (03119217)  
Κοσμάς Θωμάς (03119845)

### Υλοποίηση/Βιβλιοθήκες

Η υλοποίηση του backend του συστήματος έγινε σε Python. Η επικοινωνία μεταξύ των κόμβων γίνεται με sockets, και συγκεκριμένα χρησιμοποιώντας την βιβλιοθήκη **websockets** της Python.

- **Websocket Implementation:**

Συγκεκριμένα, κατά την δημιουργία σύνδεσης ανάμεσα σε δύο κόμβους ορίζουμε δύο socket connections ανα ζευγάρι για τη διαχείριση κλήσεων εκτός της λήψης transactions και άλλες δύο για transaction handling. Ο λόγος αυτής της απόφασης είναι σχεδιαστικός και προέρχεται από πειραματισμό κατά την ανάπτυξη του δικτύου. Λόγω του format *request/receive* των websockets και για λόγους παραλληλοποίησης και ταυτόχρονων κλήσεων ανάμεσα καταλήγουμε σε ξεχωριστά sockets για handling λήψης transactions και διαφορετικά για handling των υπόλοιπων κλήσεων. Η υλοποίηση με websockets έναντι http πλεονεκτεί στο latency της επικοινωνίας, η χρήση του είναι ιδανική για P2P networks. Άπαξ και δημιουργηθεί ένα ws connection μένει ανοιχτό για επικοινωνία και έτσι γλιτώνουμε το initialization latency σε κάθε κλήση που αντιμετωπίζει ένα http implementation. Το πλεονέκτημα γίνεται εμφανές στο transaction throughput.

Για την αλληλεπίδραση του χρήστη με το σύστημα BlockChat, χρησιμοποιήθηκε η βιβλιοθήκη inquirer της Python, για δημιουργία διαδραστικής εφαρμογής CLI.

Η διεξαγωγή των πειραμάτων έγινε σε 5 και 10 virtual machines που δημιουργήθηκαν στην πλατφόρμα DigitalOcean.

- Asyncio Library

Τα locks, broadcasts, events και γενικότερα τα κομμάτια του κώδικα που αφορούν ασύγχρονες διαδικασίες έγιναν μέσω της βιβλιοθήκης της Python **Asyncio**.

- Pycryptodome

Τα cryptographic functions που χρησιμοποιούνται στο κώδικα πάρθηκαν από τη βιβλιοθήκη pycryptodome.

- Inquirer

Το CLI μενού/επιλογέας και question/answer format υλοποιήθηκε με τη βοήθεια της βιβλιοθήκης Inquirer.

## Δομή συστήματος

### Backend

Κάθε **κόμβος** του συστήματος είναι και ένα instance της κλάσης `Node()`. Οι δομές από τις οποίες αποτελείται ο κάθε node είναι ένα id, πεδία με την IP και το port του στο δίκτυο, η δική του εκδοχή του blockchain (`chain`, instance της κλάσης `Blockchain()`), το wallet του (instance της κλάσης `Wallet()`), πεδίο με το επικυρωμένο, από το blockchain, stake του, ένα dictionary με το state, όπου είναι αποθηκευμένα τα balances και τα stakes των κόμβων του δικτύου, ένα double-ended queue όπου διατηρούνται τα pending transactions, ένα dictionary όπου διατηρούνται τα buffered blocks, και δομές συγχρονισμού, ένα event για το validation του block και την διαχείριση των pending transactions, και ένα lock για την προσθήκη συναλλαγών σε block.

Κάθε κόμβος έχει και το δικό του **wallet**. Το wallet αυτό αποτελείται από τα πεδία public και private key, nonce και balance. Το nonce και το balance είναι 0 όταν αρχικοποιείται το wallet. Το nonce αυξάνεται κατά 1 κάθε φορά που επικυρώνεται μια συναλλαγή που έχει κάνει ο κάτοχος του wallet, και αποτρέπει από το να συμβούν replay attacks. Συγκεκριμένα, το nonce λαμβάνεται υπόψη στο hashing του transaction, όπου δημιουργείται το id της συναλλαγής (`transaction_id`). Έτσι, στην διαδικασία του `verify_signature()`, αφού λαμβάνεται υπόψη το `transaction_id`, δεν θα γίνει verify κάποια συναλλαγή που πάει να ξανασταλεί στο δίκτυο.

Κάθε κόμβος έχει και το δικό του **blockchain**, το οποίο αποτελείται από μια λίστα όπου καταγράφονται τα blocks που έχουν επαληθευτεί, και ένα lock το οποίο χρησιμοποιείται στην συνάρτηση `mint_block()`, και στην συνάρτηση `new_block()`, η οποία βάζει το μόλις επικυρωμένο block στο blockchain.

Κάθε **block** που δημιουργείται είναι instance της κλάσης `Block()`, η οποία έχει τα πεδία `index` (αύξων αριθμός του block), `previous_hash` (το hash του προηγούμενου block στο blockchain), `timestamp` (το timestamp δημιουργίας του block), `current_hash` (το hash του block), `validator` (το public key του validator του block), `capacity` (γίνεται ίσο με την σταθερά capacity την οποία καθορίζει ο χρήστης), και `transactions` (λίστα με τα transactions που έχουν μπει στο block).

Για κάθε αποστολή νομισμάτων, μηνύματος, ή δέσμευση ποσού για staking, δημιουργείται ένα **transaction**. Η κλάση `Transaction()` αποτελείται από τα πεδία `sender_address` (το public key του αποστολέα), `receiver_address` (το public key του παραλήπτη), `type_of_transaction` (καθορίζει το είδος του transaction και μπορεί να πάρει μια από τις τιμές 'coin' και 'message'), `nonce` (το καθορίζει ο αποστολέας και το κάνει ίσο με την τιμή του nonce που έχει αποθηκευμένη στο wallet του), `amount` (το ποσό των νομισμάτων προς αποστολή, εάν πρόκειται για μήνυμα, στην αρχικοποίηση της συνάρτησης το πεδίο παίρνει τιμή ίση με το μήκος του μηνύματος), και `message` (η default τιμή είναι None, όταν πρόκειται για μήνυμα αποθηκεύει το μήνυμα προς αποστολή).

### BlockChat client

Η διεπαφή με τον χρήστη είναι μια διαδραστική εφαρμογή CLI.

Ο χρήστης μπορεί να επιλέξει μεταξύ των παρακάτω επιλογών:

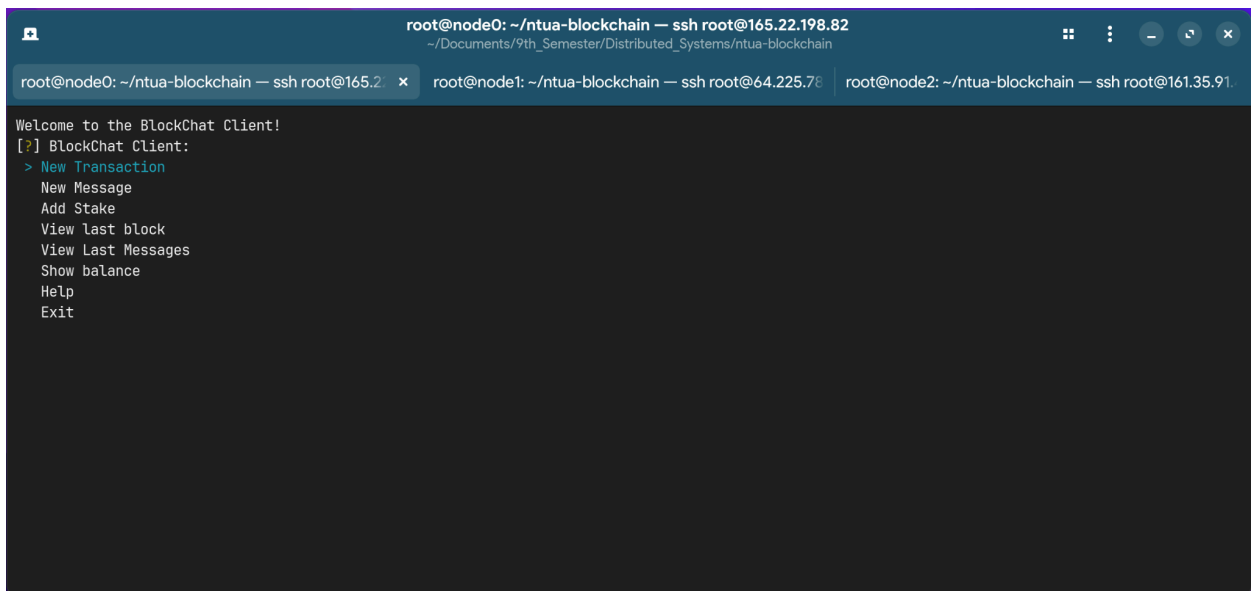
- **New Transaction:** Ο χρήστης αποστέλλει νομίσματα σε κάποιον άλλο χρήστη, καθορίζοντας το receiver ID και το ποσό που θέλει να στείλει. Η επιλογή αυτή καλεί την

`create_transaction()` με παραμέτρους `type_of_transaction='coin'`, `receiver_address` το public key του wallet που αντιστοιχίζεται στο receiver ID που πέρασε ο χρήστης, και `amount` ίσο με το ποσό που καθόρισε ο χρήστης.

- **New Message:** Ο χρήστης αποστέλλει ένα μήνυμα σε κάποιο άλλο χρήστη, καθορίζοντας το receiver ID και το μήνυμα που θέλει να στείλει. Η επιλογή αυτή καλεί την `create_transaction()` με παραμέτρους `type_of_transaction='message'` και `receiver_address` το public key του wallet που αντιστοιχίζεται στο receiver ID που πέρασε ο χρήστης. Η παράμετρος `amount` καθορίζεται με την αρχικοποίηση του instance της κλάσης `Transaction()`, και παίρνει τιμή ίση με το μήκος του μηνύματος (σε χαρακτήρες).
- **Add stake:** Ο χρήστης καθορίζει το ποσό που θέλει να δεσμεύσει για staking. Καλείται η συνάρτηση `stake(amount)`.
- **View last block:** Ο χρήστης βλέπει λεπτομέρειες του τελευταίου επικυρωμένου block. Τυπώνεται το public key του validator και οι συναλλαγές που μπήκαν στο block αυτό. Για κάθε συναλλαγή φαίνονται τα πεδία `sender_address`, `recipient_address`, `type_of_transaction`, `amount`, `message`, `nonce`, `transaction_id`, και `signature`.
- **View Last Messages:** Ο χρήστης βλέπει τα μηνύματα που έχει λάβει, και τα έλαβε με το τελευταίο επικυρωμένο block.
- **Show balance:** Ο χρήστης βλέπει το balance του και το ποσό που έχει δεσμεύσει για staking, τόσο το hard/επικυρωμένο state όσο και το soft, πριν επικυρωθεί το block.
- **Help:** Τυπώνεται κείμενο που εξηγεί τις επιλογές του client.
- **Exit:** Με αυτή την επιλογή κλείνει ο client.

Παρακάτω παρατίθενται κάποια στιγμιότυπα από την εφαρμογή CLI:

To client menu:



```
root@node0: ~/ntua-blockchain — ssh root@165.22.198.82
~/Documents/9th_Semester/Distributed_Systems/ntua-blockchain

root@node0: ~/ntua-blockchain — ssh root@165.22.198.82 x root@node1: ~/ntua-blockchain — ssh root@64.225.78 root@node2: ~/ntua-blockchain — ssh root@161.35.91

Welcome to the BlockChat Client!
[?] BlockChat Client:
> New Transaction
New Message
Add Stake
View last block
View Last Messages
Show balance
Help
Exit
```

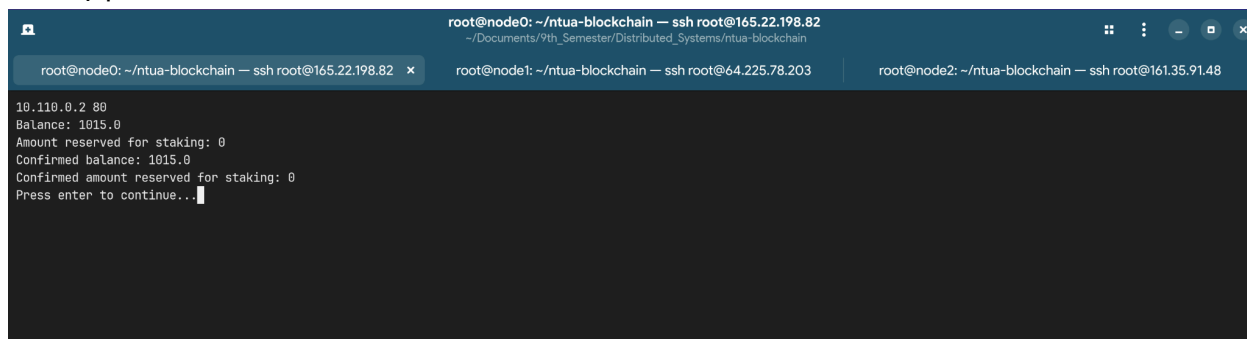
## Επιλογή ‘View last block’:

```
root@node1: ~/ntua-blockchain — ssh root@64.225.78.203
root@node0: ~/ntua-blockchain — ssh root@165.22.198.82
root@node1: ~/ntua-blockchain — ssh root@64.225.78.203
root@node2: ~/ntua-blockchain — ssh root@161.35.91.48

{
  "validator": "-----BEGIN PUBLIC KEY-----\nMIIBIjANBgkqhkiG9w0BAQEFAAOCAQIAMIIBcQKAAQEA2C5LUP/7EhaELWmN03x\n'seEQ4QDny3G68FynQQf1G3A6xTNYVcqTNXpFGwF8yunApd0XBfEHsZeS2ChZopq4C\nnm4ApoDR45lncXae1gvnC2gm6z2s8u1r6036+HCAgoGR1pwRDn2p1txH3f+Smvey6\nlnL8vSmPkSDKa6Eg9tsDpt6T/n8oQeK156zjeTLRVK01aSH71Ym14uvby1G6fnXJ\n\nlV0kbg5VMncL/4sVmx2WvoaAKTbdxh6nxt+gmnS182Fr\nbJFS16CHUXPKnocsVj\n\nlnpj6aajmm7umt7h3Z119aR7/21IhJymkgmw7b0S1R0Tyg7xtc4eye09sJ/2nx4fyb\nlnRwIDAQAB\n\n-----END PUBLIC KEY-----",
  "transactions": [
    {
      "sender_address": "-----BEGIN PUBLIC KEY-----\nMIIBIjANBgkqhkiG9w0BAQEFAAOCAQIAMIIBcQKAAQEA2C5LUP/7EhaELWmN03x\n'seEQ4QDny3G68FynQQf1G3A6xTNYVcqTNXpFGwF8yunApd0XBfEHsZeS2ChZopq4C\nnm4ApoDR45lncXae1gvnC2gm6z2s8u1r6036+HCAgoGR1pwRDn2p1txH3f+Smvey6\nlnL8vSmPkSDKa6Eg9tsDpt6T/n8oQeK156zjeTLRVK01aSH71Ym14uvby1G6fnXJ\n\nlV0kbg5VMncL/4sVmx2WvoaAKTbdxh6nxt+gmnS182Fr\nbJFS16CHUXPKnocsVj\n\nlnpj6aajmm7umt7h3Z119aR7/21IhJymkgmw7b0S1R0Tyg7xtc4eye09sJ/2nx4fyb\nlnRwIDAQAB\n\n-----END PUBLIC KEY-----",
      "recipient_address": "-----BEGIN PUBLIC KEY-----\nMIIBIjANBgkqhkiG9w0BAQEFAAOCAQIAMIIBcQKAAQEAxjZgB0j15jH/W3KmPdZf\n\nlN2pdLfqLHFEXHQHMLM+dcLMFs9uGKpshmyF1YlaqPa2326x+oY+Too0h79q7ysPy\n\nlnU80z1tFa1A9W0Uw09c9fjTYLmm4+MkatiCe1JOh+j74H25jiv3QvYRWUXInDop9tA\n\nlnrJhwtJ1ka/uFLHwe1lTc2sm481JYnhAahyL7bJKT0w5E86dkgPA1wx+fvFk12W\n\nlnzuyqCPbD1gY5gFZQ46ukuDEVXwu8yyR+37l1spNbMkTCyN87Bntg+eha8SRRn4\n\nlnTua82uetP3kj\n\nlnUtsnie0mqndWAX/F6FL0w08C3M2419DzhgyPmfM664Ah6WAI16j\n\nlnuQIDAQAB\n\n\n-----END PUBLIC KEY-----",
      "type_of_transaction": "coin",
      "amount": 1000,
      "message": null,
      "nonce": 0,
      "transaction_id": "59f15e893b174580a8ee9ba66d6d9ec721d640251fa8da95ea2a1df65c8b77fe",
      "signature": "qA+trAsLrFuxRceXoPmJpJWHUihjeE12oY5QdrFpkHYECw0dDU++nTth3JR16hcPeY2PvtooVBeq+3VJpTL0aYFezwBHK7cbWa7rbKqHGfNZWS+9ZheGd9z4yKy1LW\n\nPnrq5/2UmiaXALf6TL5NnfTJAwqCPmufMTU7PePi7eYsx29AYuQATRA/XvIPPRCCTCCENqwdSsL0257G0U2s\n\nlnPfkK4y5MyJo6VnQdPnhF1s6rSe6Qhd6uRY112Z+0vEny02eKJnC8nc9r/SCZE4Pu0WB0KA2FJ21jPucUeMC8j\n\nNYndp0tKCuXbnp6Z4JhZDLU7HS10rArc/A==",
    },
    {
      "sender_address": "-----BEGIN PUBLIC KEY-----\nMIIBIjANBgkqhkiG9w0BAQEFAAOCAQIAMIIBcQKAAQEA2C5LUP/7EhaELWmN03x\n'seEQ4QDny3G68FynQQf1G3A6xTNYVcqTNXpFGwF8yunApd0XBfEHsZeS2ChZopq4C\n\nlnm4ApoDR45lncXae1gvnC2gm6z2s8u1r6036+HCAgoGR1pwRDn2p1txH3f+Smvey6\nlnL8vSmPkSDKa6Eg9tsDpt6T/n8oQeK156zjeTLRVK01aSH71Ym14uvby1G6fnXJ\n\nlV0kbg5VMncL/4sVmx2WvoaAKTbdxh6nxt+gmnS182Fr\nbJFS16CHUXPKnocsVj\n\nlnpj6aajmm7umt7h3Z119aR7/21IhJymkgmw7b0S1R0Tyg7xtc4eye09sJ/2nx4fyb\nlnRwIDAQAB\n\n\n-----END PUBLIC KEY-----",
      "recipient_address": "-----BEGIN PUBLIC KEY-----\nMIIBIjANBgkqhkiG9w0BAQEFAAOCAQIAMIIBcQKAAQEAqEmVYONKT4a1dsAJ+2S\n\nlneCL4tumUD/v87Y2b085MYdfaRjZl100QgtufITkMyesVFFN9reiHuzKxmYKowm\n\nlnRoXu1UNG1HQYCUJdA0S1316/LSd1aSaW06oCzAwLq1jxTq2MWBanc6BQbEwexB\n\n\nlnUE2CftVZQ42L6CHbgkpbXj+Pd12PWEknBLZvnmUBTCELjgt1G21fBCTC0DE3g\n\nlnNnr8T3AgWx7Fay7YV1kfn+RFL16zUfutnxfgTKYe1lel9q+VJBFI0nj\n\nlnp6A63sv8U\n\nlni+L9MEwSar0uR0mfF0bpC2qeZ3T6sMbcz3r7PcLTM6IT6Cs8B69+8qxYU6mQuw\n\nlnJQIDAQAB\n\n\n-----END PUBLIC KEY-----",
      "type_of_transaction": "coin",
      "amount": 1000,
      "message": null,
      "nonce": 1,
      "transaction_id": "429ae30134e0645918f20610423ce1ce8298202fdb1df43206682d876f48fe2b1b",
      "signature": "eayZf5dKE709ghyBwx64Xecq/aPfpJTnBox4ApxWALrhJ53c1LTpTnT+40hAR1SFHwjF53c1RmiHqViu\n\nlnx8tdc0nbQXhhjQL16j3cB8NrpYW1KX51f6q6/13VQoHC31ldaY4kna6qJ0hJbmG4/mZmC7Pub3WJ4u9dx9Yzenvcae5F5fJrY/WUKx9247S122dx1eE7VFLs48Y\n\nlnXw0q3wB1T01sk0e+2eaw1j7uKfICfeX00bhYHV3wi/EUQ50JDZf29uqb7Z6S\n\nlnThruzg+fv9qyZ1eR71+L6sU5IjhL6T9UNpT76saXxT1tq7E\n\nlnrF97nuHMQ==",
    },
    {
      "sender_address": "-----BEGIN PUBLIC KEY-----\nMIIBIjANBgkqhkiG9w0BAQEFAAOCAQIAMIIBcQKAAQEA2C5LUP/7EhaELWmN03x\n'seEQ4QDny3G68FynQQf1G3A6xTNYVcqTNXpFGwF8yunApd0XBfEHsZeS2ChZopq4C\n\nlnm4ApoDR45lncXae1gvnC2gm6z2s8u1r6036+HCAgoGR1pwRDn2p1txH3f+Smvey6\nlnL8vSmPkSDKa6Eg9tsDpt6T/n8oQeK156zjeTLRVK01aSH71Ym14uvby1G6fnXJ\n\nlV0kbg5VMncL/4sVmx2WvoaAKTbdxh6nxt+gmnS182Fr\nbJFS16CHUXPKnocsVj\n\nlnpj6aajmm7umt7h3Z119aR7/21IhJymkgmw7b0S1R0Tyg7xtc4eye09sJ/2nx4fyb\nlnRwIDAQAB\n\n\n-----END PUBLIC KEY-----",
      "recipient_address": "-----BEGIN PUBLIC KEY-----\nMIIBIjANBgkqhkiG9w0BAQEFAAOCAQIAMIIBcQKAAQEAxjZgB0j15jH/W3KmPdZf\n\nlnN2pdLfqLHFEXHQHMLM+dcLMFs9uGKpshmyF1YlaqPa2326x+oY+Too0h79q7ysPy\n\nlnU80z1tFa1A9W0Uw09c9fjTYLmm4+MkatiCe1JOh+j74H25jiv3QvYRWUXInDop9tA\n\nlnrJhwtJ1ka/uFLHwe1lTc2sm481JYnhAahyL7bJKT0w5E86dkgPA1wx+fvFk12W\n\nlnzuyqCPbD1gY5gFZQ46ukuDEVXwu8yyR+37l1spNbMkTCyN87Bntg+eha8SRRn4\n\nlnTua82uetP3kj\n\nlnUtsnie0mqndWAX/F6FL0w08C3M2419DzhgyPmfM664Ah6WAI16j\n\nlnuQIDAQAB\n\n\n-----END PUBLIC KEY-----",
      "type_of_transaction": "message",
      "amount": 12,
      "message": "Hey 1 from 0",
      "nonce": 2,
      "transaction_id": "6ca2a904cfe98adae3ef97449e5ae5b904dd00132bc86dc740111f56dfb2d4d",
      "signature": "qyI\n\nlnwhQ5Xkn6I-CyoxRTWMSB1f0SKCNVnm6So0FDgV6cInfrms4/ASMDYCZKTYEx+3sqMaj\n\nlnjqQKmw9usxovucdXLo01044N/jwbkfw0AEwhZ4TR1GNf9LEBuv0Tc7jyvhGqCYnQ\n\nlnWWzq214j0ySh/Pw+xnkLVZPxyf5SUIHE4R3z1c8Xth1MeJgX2TzSzbkEHj+f2F7jbcA6Jue7q\n\nlnT17p/qPEWQpINN+waik8xqrPAVazPUSDr1t6kVJHkFWLKVfwtx8+4af5eA8ySE5yHm\n\nlnMcoCu46ffrRTqZxhndx/MKcQBE5X3k1iU\n\nlnTadHxtc48UEIj\n\nln1p1AvyUo==",
    },
    {
      "sender_address": "-----BEGIN PUBLIC KEY-----\nMIIBIjANBgkqhkiG9w0BAQEFAAOCAQIAMIIBcQKAAQEAqEmVYONKT4a1dsAJ+2S\n\nlneCL4tumUD/v87Y2b085MYdfaRjZl100QgtufITkMyesVFFN9reiHuzKxmYKowm\n\nlnRoXu1UNG1HQYCUJdA0S1316/LSd1aSaW06oCzAwLq1jxTq2MWBanc6BQbEwexB\n\n\nlnUE2CftVZQ42L6CHbgkpbXj+Pd12PWEknBLZvnmUBTCELjgt1G21fBCTC0DE3g\n\nlnNnr8T3AgWx7Fay7YV1kfn+RFL16zUfutnxfgTKYe1lel9q+VJBFI0nj\n\nlnp6A63sv8U\n\nlni+L9MEwSar0uR0mfF0bpC2qeZ3T6sMbcz3r7PcLTM6IT6Cs8B69+8qxYU6mQuw\n\nlnJQIDAQAB\n\n\n-----END PUBLIC KEY-----",
      "recipient_address": "-----BEGIN PUBLIC KEY-----\nMIIBIjANBgkqhkiG9w0BAQEFAAOCAQIAMIIBcQKAAQEAxjZgB0j15jH/W3KmPdZf\n\nlnN2pdLfqLHFEXHQHMLM+dcLMFs9uGKpshmyF1YlaqPa2326x+oY+Too0h79q7ysPy\n\nlnU80z1tFa1A9W0Uw09c9fjTYLmm4+MkatiCe1JOh+j74H25jiv3QvYRWUXInDop9tA\n\nlnrJhwtJ1ka/uFLHwe1lTc2sm481JYnhAahyL7bJKT0w5E86dkgPA1wx+fvFk12W\n\nlnzuyqCPbD1gY5gFZQ46ukuDEVXwu8yyR+37l1spNbMkTCyN87Bntg+eha8SRRn4\n\nlnTua82uetP3kj\n\nlnUtsnie0mqndWAX/F6FL0w08C3M2419DzhgyPmfM664Ah6WAI16j\n\nlnuQIDAQAB\n\n\n-----END PUBLIC KEY-----",
      "type_of_transaction": "message",
      "amount": 12,
      "message": "Hey 1 from 2",
      "nonce": 0,
      "transaction_id": "426f985b624b22308b46557f73ebc826c54f8d270bf8b4d4c6944d9ebc39c5e",
      "signature": "Tq6v\n\nlnuPskAlYqKJJrRWIIF2ABX3VnK06RSVkgbtCRdV4Y489p0/ayF0T3ILK26tGjUhe1e9eTzabD\n\nln8wffmX+JUgaapJourFAktFALmb5KfTUVQHXy0DVHVALDv/1A2R11\n\nlnHnPl0y3n7HMcG6v2qgW5RBby27rb7j3SSMxswDL/gmpJ7c1jY93mbZ4RRep\n\nlnrLfqKXnsRkvEu7Xw0Ea4L15eu+UppNzjKWlnz+cJseL560BT/jPMPo1NX/1RM6h9H\n\nlnKRnccTMaVbJ355FRH+ZfMzSday7626950/JB0wvohyCZV0wL/SkTnd6c+fWuS\n\nlnw6v779xth0n9eaveikisp9==",
    },
    {
      "sender_address": "-----BEGIN PUBLIC KEY-----\nMIIBIjANBgkqhkiG9w0BAQEFAAOCAQIAMIIBcQKAAQEAxjZgB0j15jH/W3KmPdZf\n\nlnN2pdLfqLHFEXHQHMLM+dcLMFs9uGKpshmyF1YlaqPa2326x+oY+Too0h79q7ysPy\n\nlnU80z1tFa1A9W0Uw09c9fjTYLmm4+MkatiCe1JOh+j74H25jiv3QvYRWUXInDop9tA\n\nlnrJhwtJ1ka/uFLHwe1lTc2sm481JYnhAahyL7bJKT0w5E86dkgPA1wx+fvFk12W\n\nlnzuyqCPbD1gY5gFZQ46ukuDEVXwu8yyR+37l1spNbMkTCyN87Bntg+eha8SRRn4\n\nlnTua82uetP3kj\n\nlnUtsnie0mqndWAX/F6FL0w08C3M2419DzhgyPmfM664Ah6WAI16j\n\nlnuQIDAQAB\n\n\n-----END PUBLIC KEY-----",
      "type_of_transaction": "coin",
      "amount": "100",
      "message": null,
      "nonce": 0,
      "transaction_id": "49add1317cc57d74766bc2d579909d3e432715b12fe42de338a80a58debalec58",
      "signature": "HND0a1Qh1Q0SAH/FkUXWe8oPmQVE9s0kZzCP21LYNm1dYTuaH8Kx1r8J\n\nlnKnuL/5abVpgNhs66ngprwBNLdEZfmxAnurcFfmI/jmUYLubb65oYdeux6v87Vr1o\n\nln68QIZewLcootrK4kH5nj3adnJCNLDJasTffEZ5f3LYk0YjBRFqwx2401HZHQF/mWLU6JGB13gfu8FWB14P+kQX2cL6V\n\nlnFLHnLYZTh1Xyv4j6peqr2N6tcvqpW02n06RaVpztqK2HnH1d9FDAAyLqFhmBbApQ100\n\nlnDndqRLhf5548w2JCvJyot74a5sIvKbo+Xh68qr7Av2vM5N8amnmvDq==",
    }
  ]
}

Press enter to continue...
```

### Επιλογή 'View balance':

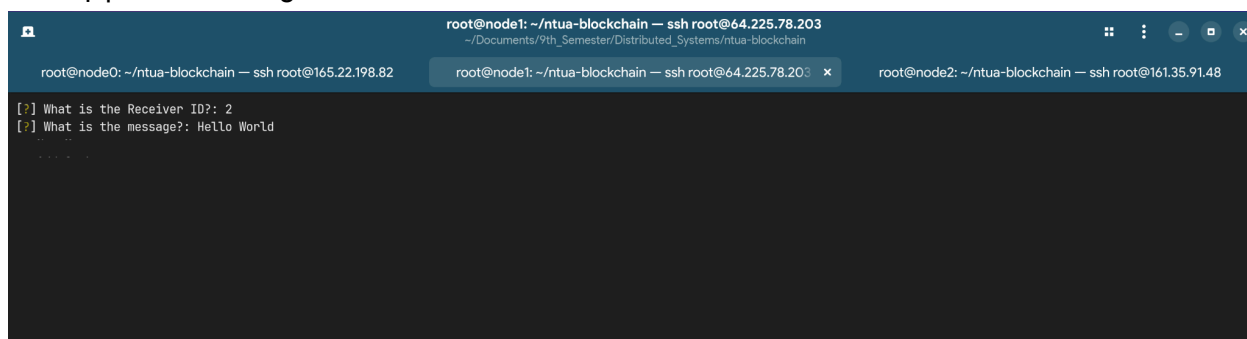


```
root@node0: ~/ntua-blockchain — ssh root@165.22.198.82
~/Documents/9th_Semester/Distributed_Systems/ntua-blockchain

root@node0: ~/ntua-blockchain — ssh root@165.22.198.82 x root@node1: ~/ntua-blockchain — ssh root@64.225.78.203 root@node2: ~/ntua-blockchain — ssh root@161.35.91.48

10.110.0.2 80
Balance: 1015.0
Amount reserved for staking: 0
Confirmed balance: 1015.0
Confirmed amount reserved for staking: 0
Press enter to continue...
```

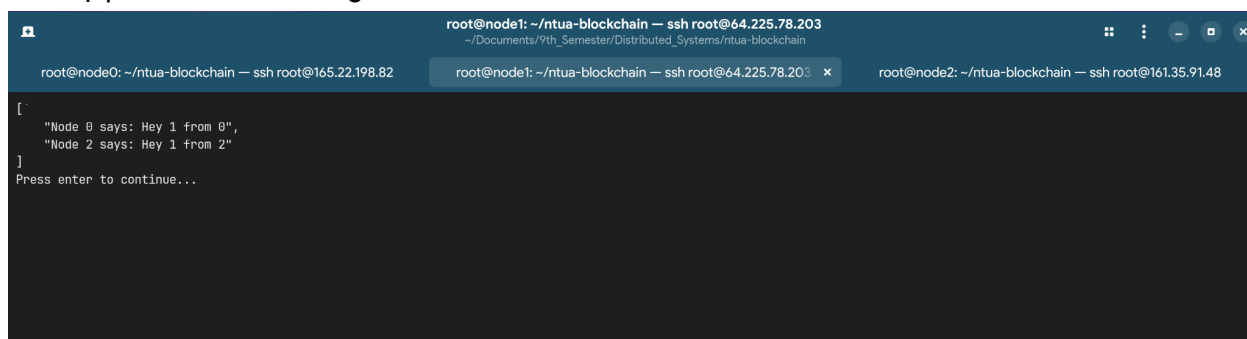
### Επιλογή 'New Message':



```
root@node0: ~/ntua-blockchain — ssh root@165.22.198.82 root@node1: ~/ntua-blockchain — ssh root@64.225.78.203 x root@node2: ~/ntua-blockchain — ssh root@161.35.91.48

[?] What is the Receiver ID?: 2
[?] What is the message?: Hello World
.....
```

### Επιλογή 'View Last Messages':



```
root@node0: ~/ntua-blockchain — ssh root@165.22.198.82 root@node1: ~/ntua-blockchain — ssh root@64.225.78.203 x root@node2: ~/ntua-blockchain — ssh root@161.35.91.48

[
  "Node 0 says: Hey 1 from 0",
  "Node 2 says: Hey 1 from 2"
]
Press enter to continue...
```

### Λειτουργία συστήματος

Αρχικά, εισέρχεται στο δίκτυο ο bootstrap κόμβος, όπου καθορίζεται το πλήθος των κόμβων που θα μπουν στο δίκτυο, και το capacity των block που θα έχει το blockchain. Όταν γίνει αυτό, γίνεται το genesis transaction, όπου ο bootstrap κόμβος λαμβάνει 1000\*n BCC, όπου η το πλήθος των αναμενόμενων κόμβων. Ύστερα, μπαίνουν οι υπόλοιποι κόμβοι στο δίκτυο διαδοχικά. Μέχρι να μπει το καθορισμένο πλήθος κόμβων στο δίκτυο, δεν επιτρέπεται σε κανέναν κόμβο να κάνει συναλλαγές και να επιλέξει οποιαδήποτε από τις επιλογές του client. Αφού μπουν όλοι οι κόμβοι, ο bootstrap κόμβος δίνει 1000 BCC στον καθένα. Ύστερα, κάθε κόμβος μπορεί να στείλει BCC ή κάποιο μήνυμα σε άλλο κόμβο, να δεσμεύσει BCC για stake,

να δει τις συναλλαγές του τελευταίου επικυρωμένου block, και να δει το balance του, τόσο το soft όσο και το hard state.

Όταν ο χρήστης επιλέξει να στείλει BCC ή μήνυμα, καθορίζει το id του κόμβου παραλήπτη, και το ποσό ή το μήνυμα. Τότε καλείται η συνάρτηση `create_transaction()`, η οποία δημιουργεί μια συναλλαγή (στην περίπτωση του μηνύματος, το ποσό της συναλλαγής είναι ίσο με το μήκος του μηνύματος που πρέπει να σταλεί). Για την συγκεκριμένη συναλλαγή, καλείται η συνάρτηση `broadcast_transaction()`, η οποία στέλνει σε όλους τους κόμβους την συναλλαγή προς επικύρωση. Κάθε κόμβος ελέγχει εάν επιτρέπεται να γίνει αυτή η συναλλαγή. Σε αυτό τον έλεγχο, βλέπει εάν μπορεί να κάνει `verify` την υπογραφή της συναλλαγής, και επιπλέον, βάσει του state από balances που κρατάει ο συγκεκριμένος κόμβος, εάν ο κόμβος που θέλει να κάνει την συναλλαγή έχει το επαρκές υπόλοιπο, δηλαδή εάν έχει περισσότερα BCC από το ποσό που θέλει να αποστείλει συν fees εάν πρόκειται για αποστολή BCC, ή εάν το μήκος του μηνύματος είναι μικρότερο από το πλήθος των BCC που έχει ως balance. Εάν δεν πληρούται κάποια από αυτές τις συνθήκες, ο κόμβος δεν βάζει την συγκεκριμένη συναλλαγή στο block του.

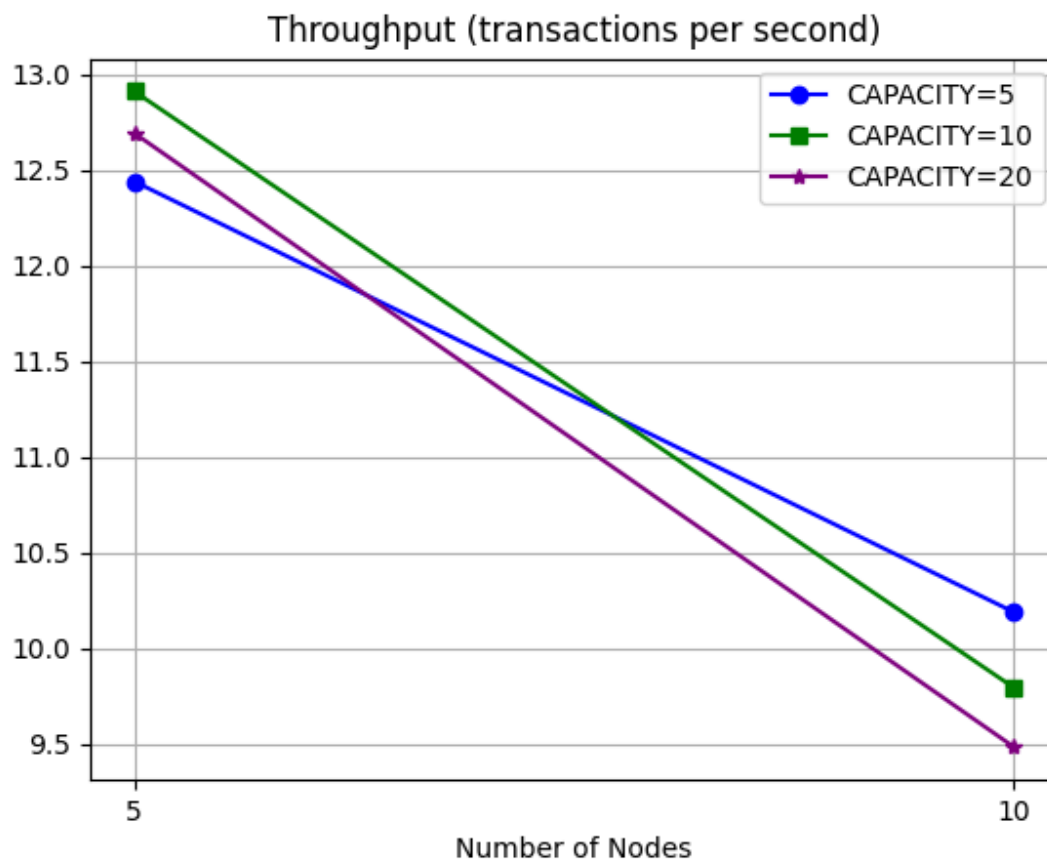
Η παραπάνω διαδικασία ακολουθείται και όταν ο κόμβος επιλέγει να δεσμεύσει κάποιο ποσό ως stake. Αυτή η ενέργεια μεταφράζεται σε transaction με παραλήπτη με `receiver_address` ίσο με 0, και στον έλεγχο οι υπόλοιποι κόμβοι ελέγχουν εάν ο κόμβος έχει τα απαραίτητα BCC στο balance του για να κάνει staking, και συγκεκριμένα εάν το ποσό που θέλει να δεσμεύσει είναι λιγότερο από το balance του συν το ήδη δεσμευμένο ποσό staking. Η συνάρτηση `stake()` είναι υλοποιημένη ώστε ο χρήστης να καθορίζει πόσο θέλει να είναι το καινούργιο ποσό staking, και όχι πόσο επιπλέον ποσό θέλει να δεσμεύσει για staking. Με άλλα λόγια, εάν αρχικά είχε δεσμεύσει 20 BCC για staking, και ύστερα επιλέξει να δεσμεύσει 40 BCC, το ποσό staking μετά την επικύρωση του block θα είναι 40 BCC (όχι  $20+40=60$  BCC).

Αφού γεμίσει το block, δηλαδή όταν το πλήθος των συναλλαγών μέσα στο block γίνει ίσο με το capacity, αρχίζει η διαδικασία validation του block. Αρχικά, επιλέγεται ο validator μέσω μιας ψευδοτυχαίας συνάρτησης. Αυτή η συνάρτηση χρησιμοποιεί ένα seed το οποίο έχει υπολογιστεί βάσει του hash του τελευταίου επικυρωμένου block, το οποίο κάνει ντετερμινιστικό το αποτέλεσμα επιλογής του validator, για όλους τους κόμβους που εκτελούν την συνάρτηση. Η πιθανότητα να επιλεγεί κάποιος κόμβος ως validator αυξάνεται βάσει του stake που έχει δεσμεύσει, και συγκεκριμένα βάσει του τελευταίου επικυρωμένου stake. Ο validator που θα επιλεγεί από αυτή την διαδικασία είναι αυτός που θα επικυρώσει το block του. Μετά, όλοι οι κόμβοι λαμβάνουν το block αυτό, και βάσει των συναλλαγών που περιέχονται σε αυτό, αλλάζουν τα hard balances τους, αλλά και το state που διατηρούν, βάσει αυτού του επικυρωμένου block. Αυτό είναι εφικτό, επειδή στο state υπάρχουν ξεχωριστά πεδία όπου διατηρούνται τα balances που έχουν προκύψει από την τελευταία επικύρωση block (`valid_balance`, `valid_stake`), και τα soft balances (`balance`, `stake`), οπότε στο update αυτό γίνεται revert στο τελευταίο επικυρωμένο state και κάθε κόμβος αλλάζει το soft και hard state του, αλλά και το soft και hard state που διατηρεί για τους υπόλοιπους κόμβους. Σε αυτή την διαδικασία, ο validator συλλέγει τα fees των συναλλαγών και τα BCC που χρεώθηκαν όσοι κόμβοι έστειλαν κάποιο μήνυμα. Αυτές οι αλλαγές απεικονίζονται και στο state όλων των κόμβων. Μετά από την ενημέρωση των balances, για κάθε κόμβο ορίζεται ως `current_block` το επικυρωμένο, ακυρώνοντας, έτσι, το block που είχε ο καθένας και δεν κατάφερε να επικυρώσει. Όσο ένας κόμβος περιμένει με γεμάτη τη δικιά του εκδοχή του block προσθέτει τα νεοεισερχόμενα transactions στο πεδίο του node, με όνομα `pending_transactions`. Όταν λάβει νέο block το προσθέτει στο blockchain του και

δημιουργεί καινούργιο με previous hash το hash του νεοεισερχόμενου block, μετά τη δημιουργία αυτού προσπαθεί να κάνει poleft τις συναλλαγές που έχουν συσσωρευτεί στο pending\_transactions. Έτσι συνεχίζεται και επαναλαμβάνεται η λήψη block και η επιμήκυνση του blockchain του κάθε κόμβου. Αξίζει να σημειωθεί ότι επειδή μπορεί να υπάρχει *out of order* λήψη block απο ένα κόμβο υπάρχει δομή buffer όπου αποθηκεύονται με αυξουσα σειρά index τα block που έχουν ληφθεί ώστε να προστεθούν και να ενημερώσουν τα hard και soft balances όταν παραληφθούν τα ενδιάμεσα blocks.

### Πειράματα/Μετρήσεις

Για μέτρηση της απόδοσης και της κλιμακωσιμότητας του συστήματος, διεξήχθησαν tests με τα δεδομένα αρχεία με συναλλαγές για 5 και 10 κόμβους. Σε αυτά τα test καταγράφηκε το throughput και το block time του συστήματος, για τιμές capacity 5, 10 και 20, και σταθερό staking για κάθε κόμβο, ίσο με 10 BCC. Τα αποτελέσματα φαίνονται στα παρακάτω γραφήματα:



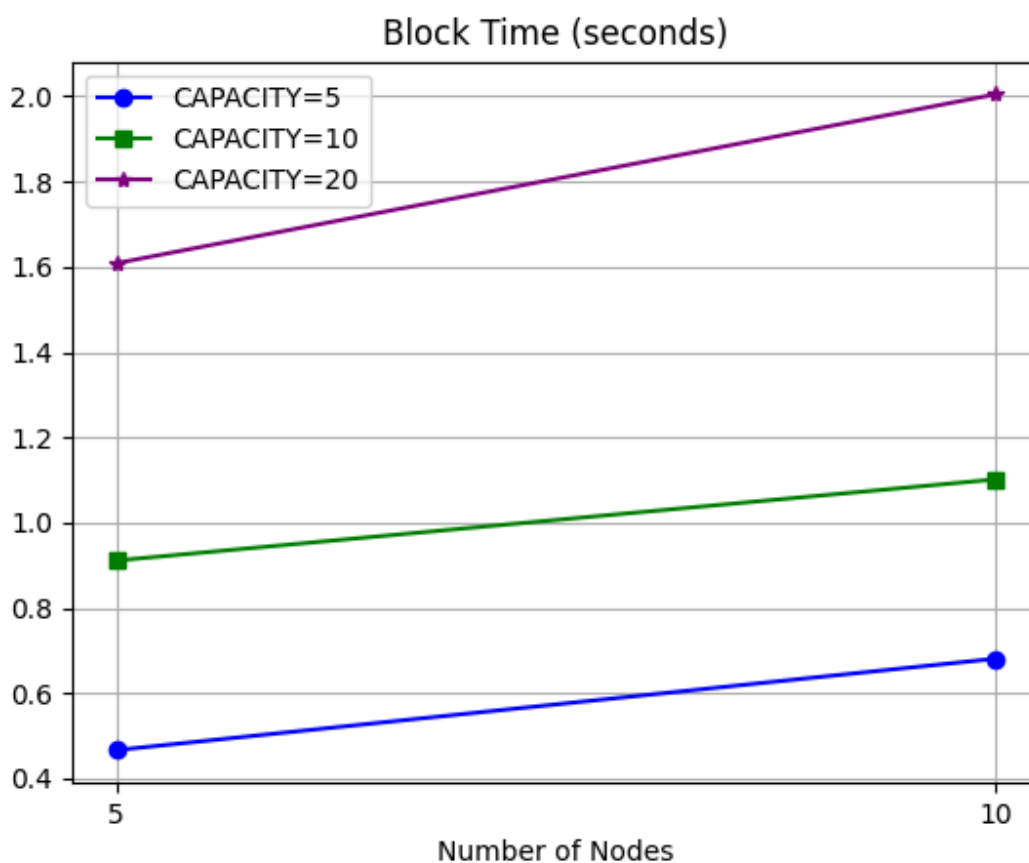
Παρατηρείται, αρχικά, ότι το throughput μειώνεται καθώς αυξάνεται ο αριθμός των κόμβων. Αυτό είναι αναμενόμενο, καθώς, όσο αυξάνεται ο αριθμός των κόμβων, αυξάνεται επίσης το contention για τη πρόσβαση σε συναρτήσεις που εξυπηρετούν την προσθήκη ενός transaction/message σε block (οι οποίες προστατεύονται με locks) λόγω των περισσότερων



αιτημάτων transactions/messages από τους άλλους κόμβους. Έτσι οδηγούμαστε σε αύξηση του χρόνου εξυπηρέτησης.

Επιπλέον, φαίνεται ότι τα συστήματα με υψηλότερη χωρητικότητα έχουν μικρότερο throughput στους 5 κόμβους, γεγονός που μπορεί να υποδηλώνει ότι οι πρόσθετες συναλλαγές ανά μπλοκ απαιτούν περισσότερο χρόνο για την τελική ανανέωση των balances του state του δικτύου που έχει ο κάθε κόμβος και επίτευξη consensus, αντισταθμίζοντας τα οφέλη της υψηλότερης χωρητικότητας συναλλαγών σε κάθε block.

Για πλήθος κόμβων ίσο με 10, οι τιμές του throughput είναι πιο κοντινές, σε σχέση με το δίκτυο 5 κόμβων. Η σύγκλιση του throughput στους 10 κόμβους, ανεξάρτητα από τη χωρητικότητα, μπορεί να υποδηλώνει ένα όριο πέρα από το οποίο η χωρητικότητα ενός μπλοκ δεν είναι πλέον ο περιοριστικός παράγοντας, αλλά άλλοι παράγοντες επηρεάζουν περισσότερο την ταχύτητα επεξεργασίας συναλλαγών.



Όσον αφορά το block time, παρατηρείται ότι έχει πολύ μικρές διακυμάνσεις (αμελητέες σχεδόν για ίδιο capacity και πρακτικά σταθερό block time) με την αύξηση του αριθμού των κόμβων. Αυτό εξηγείται από το γεγονός ότι στο Proof of Stake το block μπαίνει στο blockchain όταν απλά ο επιλεγμένος validator φτάσει στο σημείο να κλείσει το block έχοντας φτάσει στο capacity του. Σε αντίθεση με το Proof of Work (PoW) που περιλαμβάνει mining και συνεπώς αυξάνοντας τους κόμβους έχουμε περισσότερες πιθανότητες να γίνει mine ένα block μέχρι μια συγκεκριμένη χρονική στιγμή, το minting δεν επηρεάζεται από το μέγεθος του δικτύου.



Επιπλέον, για οποιοδήποτε δεδομένο αριθμό κόμβων, τα block υψηλότερης χωρητικότητας έχουν μεγαλύτερο block time. Αυτό μπορεί να οφείλεται στο μεγαλύτερο μέγεθος δεδομένων κάθε block λόγω περισσότερων συναλλαγών, το οποίο χρειάζεται περισσότερο χρόνο για την επικύρωση και τη διάδοση. Το φαινόμενο αυτό φαίνεται να είναι πιο έντονο με την αύξηση του αριθμού των κόμβων, γεγονός που υποδηλώνει ότι το μεγαλύτερο μέγεθος block επιδεινώνει τις προκλήσεις κλιμάκωσης ενός αναπτυσσόμενου δικτύου.

Φαίνεται, επίσης, ότι η αύξηση του block time από χωρητικότητα 5 σε 10 είναι λιγότερο απότομη από ότι από 10 σε 20. Αυτό υποδηλώνει ότι ενώ η αύξηση της χωρητικότητας προσθέτει γενικά overhead, ο ρυθμός αύξησης του οποίου μπορεί να μην είναι γραμμικός. Ο αντίκτυπος των πρόσθετων συναλλαγών στο block time γίνεται πιο σημαντικός καθώς αυξάνεται το capacity του block, υποδεικνύοντας ότι υπάρχουν μειωμένη απόδοση όσον αφορά την αποδοτικότητα με την προσθήκη περισσότερων συναλλαγών σε ένα block.

Επιπλέον, για μέτρηση της δικαιοσύνης του συστήματος, εκτελέστηκαν ξανά τα παραπάνω test για 5 κόμβους και capacity ίσο με 5, αλλά αυτή την φορά αντί για σταθερό staking ίσο με 10 BCC για κάθε κόμβο, ένας κόμβος θα δεσμεύσει ποσό για staking ίσο με 100 BCC. Διαλέξαμε ο κόμβος που θα κάνει stake 100 BCC να είναι ο bootstrap. Παρακάτω φαίνονται τα timestamps και τα ids των validators των επικυρωμένων block (για το πρώτο block το οποίο είναι το genesis και δεν έχει validator, βάλαμε το id ίσο με -1).

Περίπτωση ίσου stake για όλους τους κόμβους:

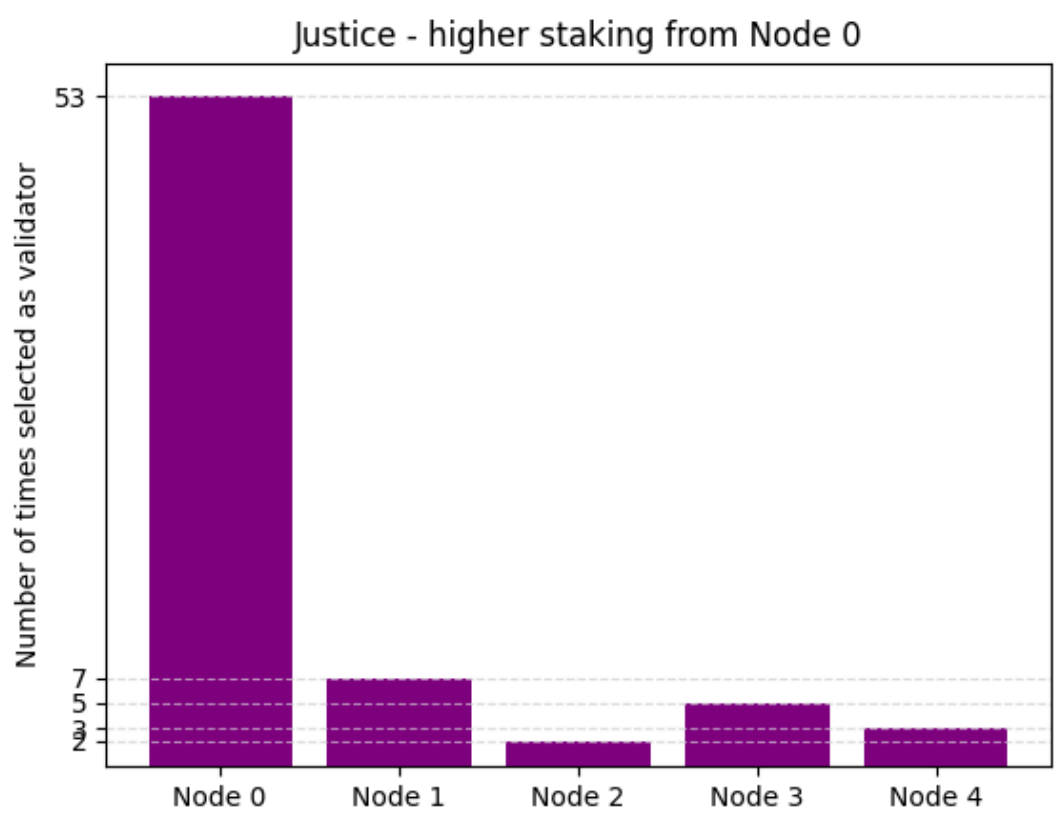
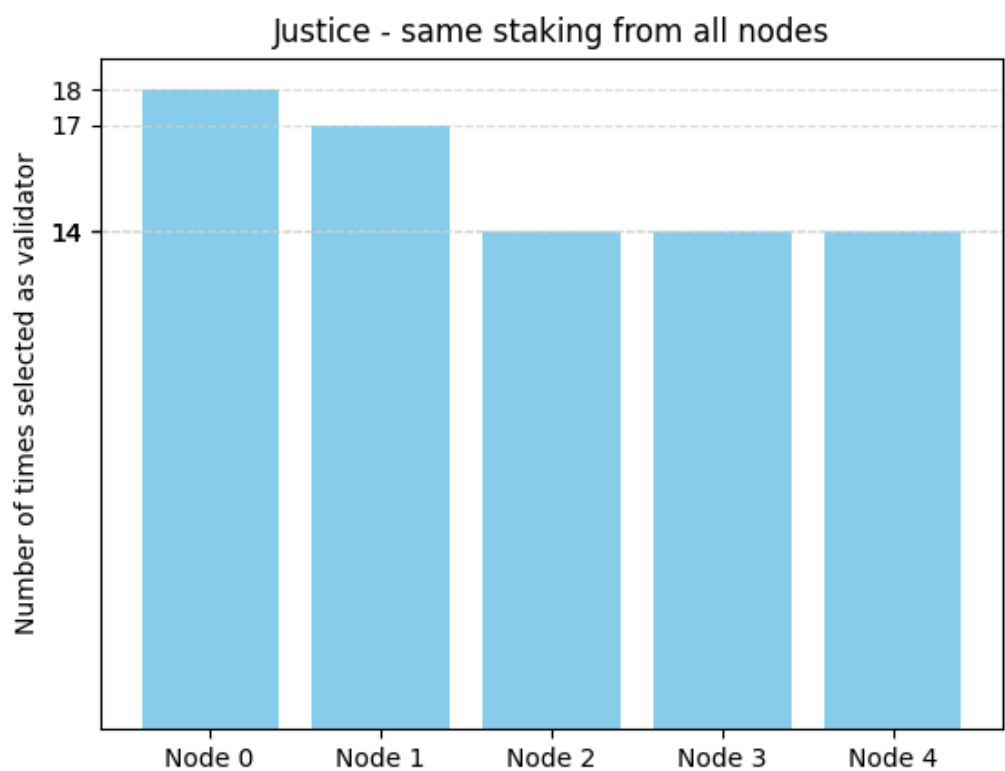
```
Blockchain timestamps and validators: [(1712752493.6152387, -1),
(1712752514.3782172, 0), (1712752517.8637052, 0), (1712752518.2550457, 3),
(1712752518.4253776, 0), (1712752518.6252542, 1), (1712752518.8856487, 4),
(1712752519.0654168, 1), (1712752519.0824697, 1), (1712752519.2756567, 0),
(1712752519.4633288, 2), (1712752519.7075336, 0), (1712752519.8582141, 3),
(1712752520.0383956, 2), (1712752520.2333448, 1), (1712752520.4439478, 4),
(1712752520.6505542, 2), (1712752520.7926743, 3), (1712752520.9694784, 0),
(1712752521.1015654, 3), (1712752521.2813382, 4), (1712752521.292995, 4),
(1712752521.3025262, 4), (1712752521.472602, 1), (1712752521.6209805, 3),
(1712752521.7975378, 1), (1712752522.0276172, 2), (1712752522.304087, 4),
(1712752522.313057, 4), (1712752522.507465, 2), (1712752522.524682, 2),
(1712752522.7054894, 1), (1712752522.9954891, 4), (1712752523.1803515, 1),
(1712752523.3275423, 0), (1712752523.5147061, 2), (1712752523.7663443, 1),
(1712752523.785391, 1), (1712752523.9644258, 3), (1712752523.9721048, 3),
(1712752524.1673625, 1), (1712752524.3152893, 3), (1712752524.3240566, 3),
(1712752524.5077305, 0), (1712752524.5206568, 0), (1712752524.658497, 3),
(1712752524.6796608, 4), (1712752524.8422556, 0), (1712752525.0918813, 2),
(1712752525.111095, 2), (1712752525.3588698, 0), (1712752525.6242871, 2),
(1712752525.7609553, 4), (1712752525.7715998, 4), (1712752525.781943, 4),
(1712752525.9986463, 2), (1712752526.167638, 0), (1712752526.1823018, 0),
```

```
(1712752526.328893, 1), (1712752526.499775, 2), (1712752526.6998947, 0),  
(1712752526.921002, 4), (1712752526.941717, 3), (1712752527.1021369, 1),  
(1712752527.1160874, 1), (1712752527.1292384, 1), (1712752527.2928963, 3),  
(1712752527.3012545, 3), (1712752527.3090286, 3), (1712752527.6333406, 1),  
(1712752528.0133286, 2), (1712752528.2640302, 4), (1712752528.2850468, 0),  
(1712752528.7209158, 2), (1712752528.8699594, 0), (1712752529.1475606, 0),  
(1712752529.4282084, 0), (1712752529.546999, 1)]
```

Περίπτωση δέσμευσης μεγαλύτερου stake από τον bootstrap node:

```
Blockchain timestamps and validators: [(1712752987.353717, -1),  
(1712753010.1861866, 0), (1712753013.6485915, 0), (1712753013.8593166, 0),  
(1712753013.9723659, 0), (1712753014.3220162, 4), (1712753014.4810946, 0),  
(1712753014.4946518, 0), (1712753014.6808875, 1), (1712753014.9383142, 3),  
(1712753014.9474258, 3), (1712753015.088599, 0), (1712753015.1038704, 0),  
(1712753015.116116, 0), (1712753015.1291766, 0), (1712753015.2549722, 0),  
(1712753015.3953354, 0), (1712753015.5136545, 0), (1712753015.6454191, 0),  
(1712753015.7525606, 0), (1712753015.854796, 0), (1712753015.975181, 0),  
(1712753016.0883067, 0), (1712753016.473158, 3), (1712753016.6385975, 0),  
(1712753016.8655608, 4), (1712753017.0266726, 0), (1712753017.0395405, 0),  
(1712753017.0529401, 0), (1712753017.0662627, 0), (1712753017.0784829, 0),  
(1712753017.201752, 0), (1712753017.3262591, 0), (1712753017.4335992, 0),  
(1712753017.544802, 0), (1712753017.7763317, 3), (1712753018.0063698, 1),  
(1712753018.1650064, 0), (1712753018.1784167, 0), (1712753018.3463876, 1),  
(1712753018.4646058, 1), (1712753018.8156495, 2), (1712753019.144352, 1),  
(1712753019.341184, 0), (1712753019.3572216, 0), (1712753019.5213926, 3),  
(1712753019.6655335, 0), (1712753019.681185, 0), (1712753019.69444, 0),  
(1712753019.711822, 0), (1712753019.9398234, 1), (1712753020.1504183, 0),  
(1712753020.1695507, 0), (1712753020.18463, 0), (1712753020.2964902, 0),  
(1712753020.4259536, 0), (1712753020.5579233, 0), (1712753020.7763343, 0),  
(1712753021.0953317, 0), (1712753021.316006, 0), (1712753021.6323814, 0),  
(1712753021.9440596, 0), (1712753022.2647028, 0), (1712753022.501797, 0),  
(1712753022.8071344, 0), (1712753023.0900958, 0), (1712753023.495731, 4),  
(1712753023.8286178, 2), (1712753024.3095894, 1), (1712753024.6743598, 0),  
(1712753024.910106, 0)]
```

Παρακάτω φαίνονται δύο bar plot για τις δύο αυτές περιπτώσεις:



Στην πρώτη περίπτωση έχουμε για όλους τους κόμβους σχεδόν ίδια πλήθη φορών που έχουν επιλεγεί ως validators. Άρα, με ίδιο stake, φαίνεται το σύστημα να διαλέγει validator με σχετικά ομοιόμορφο τρόπο.

Στην δεύτερη περίπτωση, ο Node 0 επιλέγεται σημαντικά περισσότερες φορές από τους υπόλοιπους, και συγκεκριμένα 53/70 φορές. Αυτό είναι αναμενόμενο, λόγω του ότι έχει δεκαπλάσιο stake σε σχέση με τους υπόλοιπους κόμβους.