



ΠΑΝΕΠΙΣΤΗΜΙΟ
ΘΕΣΣΑΛΙΑΣ

ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ

ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ

ΜΕΛΕΤΗ ΚΑΙ ΥΛΟΠΟΙΗΣΗ ΠΡΩΤΟΚΟΛΛΩΝ ΣΥΜΦΩΝΙΑΣ ΓΙΑ ΣΥΣΤΗΜΑΤΑ ΜΠΛΟΚ ΑΛΥΣΙΔΑΣ

ΤΣΑΒΔΑΡΗΣ ΚΥΡΙΑΚΟΣ 2117194

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

ΥΠΕΥΘΥΝΟΣ

ΤΖΙΡΙΤΑΣ ΝΙΚΟΛΑΟΣ
ΕΠΙΚΟΥΡΟΣ ΚΑΘΗΓΗΤΗΣ

Λαμία 2025



ΠΑΝΕΠΙΣΤΗΜΙΟ
ΘΕΣΣΑΛΙΑΣ

ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ

ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ

ΜΕΛΕΤΗ ΚΑΙ ΥΛΟΠΟΙΗΣΗ ΠΡΩΤΟΚΟΛΛΩΝ ΣΥΜΦΩΝΙΑΣ ΓΙΑ ΣΥΣΤΗΜΑΤΑ ΜΠΛΟΚ ΑΛΥΣΙΔΑΣ

ΤΣΑΒΔΑΡΗΣ ΚΥΡΙΑΚΟΣ 2117194

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

ΥΠΕΥΘΥΝΟΣ

ΤΖΙΡΙΤΑΣ ΝΙΚΟΛΑΟΣ
ΕΠΙΚΟΥΡΟΣ ΚΑΘΗΓΗΤΗΣ

Λαμία 2025



UNIVERSITY OF
THESSALY

SCHOOL OF SCIENCE

DEPARTMENT OF COMPUTER SCIENCE & TELECOMMUNICATIONS

STUDY AND IMPLEMENTATION OF AGREEMENT PROTOCOLS FOR BLOCKCHAIN SYSTEMS

TSAVDARIS KYRIAKOS 2117194

FINAL THESIS

ADVISOR

TZIRITAS NIKOLAOS
ASSISTANT PROFESSOR

Lamia 2025

«Με ατομική μου ευθύνη και γνωρίζοντας τις κυρώσεις ⁽¹⁾, που προβλέπονται από της διατάξεις της παρ. 6 του άρθρου 22 του Ν. 1599/1986, δηλώνω ότι:

1. Δεν παραθέτω κομμάτια βιβλίων ή άρθρων ή εργασιών άλλων αυτολεξεί **χωρίς να τα περικλείω σε εισαγωγικά** και χωρίς να αναφέρω το συγγραφέα, τη χρονολογία, τη σελίδα. Η αυτολεξεί παράθεση χωρίς εισαγωγικά χωρίς αναφορά στην πηγή, είναι λογοκλοπή. Πέραν της αυτολεξεί παράθεσης, λογοκλοπή θεωρείται και η παράφραση εδαφίων από έργα άλλων, συμπεριλαμβανομένων και έργων Συμφοιτητών μου, καθώς και η παράθεση στοιχείων που άλλοι συνέλεξαν ή επεξεργάστηκαν, χωρίς αναφορά στην πηγή. Αναφέρω πάντοτε με πληρότητα την πηγή κάτω από τον πίνακα ή σχέδιο, όπως στα παραθέματα.
2. Δέχομαι ότι η αυτολεξεί **παράθεση χωρίς εισαγωγικά**, ακόμα κι αν συνοδεύεται από αναφορά στην πηγή σε κάποιο άλλο σημείο του κειμένου ή στο τέλος του, είναι αντιγραφή. Η αναφορά στην πηγή στο τέλος π.χ. μιας παραγράφου ή μιας σελίδας, δεν δικαιολογεί συρραφή εδαφίων έργου άλλου συγγραφέα, έστω και παραφρασμένων, και παρουσίασή τους ως δική μου εργασία.
3. Δέχομαι ότι υπάρχει επίσης περιορισμός στο μέγεθος και στη συχνότητα των παραθεμάτων που μπορώ να εντάξω στην εργασία μου εντός εισαγωγικών. Κάθε μεγάλο παράθεμα (π.χ. σε πίνακα ή πλαίσιο, κλπ), προϋποθέτει ειδικές ρυθμίσεις, και όταν δημοσιεύεται προϋποθέτει την άδεια του συγγραφέα ή του εκδότη. Το ίδιο και οι πίνακες και τα σχέδια
4. Δέχομαι όλες τις συνέπειες σε περίπτωση λογοκλοπής ή αντιγραφής.

Ημερομηνία: 05/10/2025

Ο – Η Δηλ.

(1) «Όποιος εν γνώσει του δηλώνει ψευδή γεγονότα ή αρνείται ή αποκρύπτει τα αληθινά με έγγραφη υπεύθυνη δήλωση του άρθρου 8 παρ. 4 Ν. 1599/1986 τιμωρείται με φυλάκιση τουλάχιστον τριών μηνών. Εάν ο υπαίτιος αυτών των πράξεων σκόπευε να προσπορίσει στον εαυτόν του ή σε άλλον περιουσιακό όφελος βλάπτοντας τρίτον ή σκόπευε να βλάψει άλλον, τιμωρείται με κάθειρξη μέχρι 10 ετών.»

Τα δύο πρωτόκολλα παρουσιάζουν διαφορετικές φιλοσοφίες και συμβιβασμούς. Το Proof-of-Work (PoW) βασίζεται στην υπολογιστική ισχύ για ασφάλεια και ανοικτή συμμετοχή, προσφέρει υψηλή ανθεκτικότητα σε επιθέσεις, αλλά έχει πιθανά forks/reorgs και πιθανολογική τελικοποίηση, με μεγαλύτερες καθυστερήσεις και ενεργειακό αποτύπωμα. Το Proof-of-Stake (PoS) μειώνει δραστικά την κατανάλωση ενέργειας και τείνει να έχει πιο σταθερούς χρόνους επιβεβαίωσης, όμως η δικαιοσύνη/αποκέντρωση εξαρτάται από τη διανομή του stake και εμφανίζει απόκλιση όταν το stake είναι έντονα άνισο ή οι χρόνοι round/timeout δεν είναι σωστά ρυθμισμένοι. Σε θορυβώδη δίκτυα το PoW αυξάνει τη διακύμανση block time λόγω forks, ενώ το PoS διατηρεί χαμηλή και σταθερή καθυστέρηση αν οι παράμετροι είναι ισορροπημένοι. Συνολικά, το PoW μεγιστοποιεί την ανοιχτότητα με κόστος επιδόσεων και το PoS ισορροπεί απόδοση και ασφάλεια με εξάρτηση από την οικονομική κατανομή.

Λέξεις-κλειδιά:

Blockchain, πρωτόκολλα συναίνεσης, καθυστέρηση (latency), ρυθμαπόδοση (throughput), τελικοποίηση (finality), αποκέντρωση (decentralisation), κλιμάκωση, forks/reorgs, χρόνος block, διάδοση (gossip), κατανάλωση ενέργειας, αξιοπιστία δικτύου

Ανά πρωτόκολλο

PoW :

Proof-of-Work, υπολογιστική ισχύς, πιθανολογική τελικοποίηση, forks/reorgs, χρόνος εξόρυξης (mining interval), χρόνος block, ποσοστό ορφανών (orphan rate), καθυστέρηση διάδοσης, ρυθμαπόδοση, κατανάλωση ενέργειας, αριθμός γειτόνων (num neighbours).

PoS :

Proof-of-Stake (PoS), Stake-weighted proposer selection, κατανομή stake, δικαιοσύνη ηγετών, KL divergence (D_{KL}), Gini, Threshold-based finality ($\text{confirmation_threshold} \times \text{total_stake}$; NEW_BLOCK $\rightarrow$ CONFIRM $\rightarrow$ FINALIZED), Slot/Round timing (slot_len, block_interval), Round/slot timeout, Gossip, Fork choice (tip), Randomness/seed per slot, ρυθμαπόδοση, Block interval, καθυστέρηση τελικοποίησης, αποκέντρωση, ενεργειακή απόδοση, Run control (simTime)

ABSTRACT

The two protocols present different philosophies and trade-offs. Proof-of-Work (PoW) relies on computational power for security and open participation. It offers high resilience against attacks but comes with potential forks/reorgs and probabilistic finality, resulting in longer delays and a higher energy footprint. Proof-of-Stake (PoS) drastically reduces energy consumption and tends to provide more stable confirmation times; however, fairness/decentralisation depends on stake distribution, and deviations occur when the stake is heavily imbalanced or when round/timeout parameters are misconfigured. In noisy networks, PoW increases block time variance due to forks, whereas PoS maintains low and stable latency if parameters are well balanced. Overall, PoW maximises openness at the cost of performance, while PoS balances performance and security with dependence on economic distribution.

Keywords:

Blockchain, consensus protocols, latency, throughput, finality, decentralisation, scalability, forks/reorgs, block time, gossip, energy consumption, network reliability.

Per protocol:

PoW:

Proof-of-Work, computational power, probabilistic finality, forks/reorgs, mining interval, block time, orphan rate, propagation delay, throughput, energy consumption, number of neighbours.

PoS:

Proof-of-Stake (PoS), Stake-weighted proposer selection, stake distribution, leader fairness, KL-divergence (D_{KL}), Gini, Threshold-based finality ($\text{confirmation_threshold} \times \text{total_stake}$), Slot/Round timing (slot_len , block_interval) Round/slot timeout, Gossip, Fork choice (tip), Randomness/seed per slot, Throughput (blocks/s), Block interval, Finality latency, Decentralization, Energy efficiency, Run control (simTime)

Table of Contents

ΠΕΡΙΛΗΨΗ.....	7
ABSTRACT	8
<u>ΣΧΗΜΑΤΑ ΚΑΙ ΠΙΝΑΚΕΣ</u>	<u>12</u>
<u>ΚΕΦΑΛΑΙΟ 1 ΕΙΣΑΓΩΓΗ</u>	<u>13</u>
1.1 ΣΚΟΠΟΣ ΚΑΙ ΠΡΟΒΛΗΜΑ	13
1.2 ΚΙΝΗΤΡΑ ΚΑΙ ΕΡΕΥΝΗΤΙΚΑ ΕΡΩΤΗΜΑΤΑ	14
1.3 ΚΥΡΙΕΣ ΣΥΝΕΙΣΦΟΡΕΣ	15
1.4 ΜΕΘΟΔΟΛΟΓΙΑ ΚΑΙ ΠΛΑΙΣΙΟ ΑΞΙΟΛΟΓΗΣΗΣ.....	15
1.5 ΣΥΝΤΟΜΗ ΕΠΙΣΚΟΠΗΣΗ ΠΡΩΤΟΚΟΛΛΩΝ ΜΕ ΣΕΝΑΡΙΑΚΗ ΟΠΤΙΚΗ	16
1.6 ΠΕΡΙΟΡΙΣΜΟΙ ΚΑΙ ΠΕΔΙΟ ΕΦΑΡΜΟΓΗΣ.....	17
1.7 ΔΟΜΗ ΤΗΣ ΕΡΓΑΣΙΑΣ.....	17
1.8 ΟΔΗΓΟΣ ΑΝΑΓΝΩΣΗΣ ΑΠΟΤΕΛΕΣΜΑΤΩΝ.....	17
<u>ΚΕΦΑΛΑΙΟ 2 ΥΠΟΒΑΘΡΟ: ΘΕΩΡΙΑ ΣΥΝΑΙΝΕΣΗΣ ΚΑΙ ΜΕΤΡΙΚΕΣ.....</u>	<u>18</u>
2.1 ΒΑΣΙΚΕΣ ΈΝΝΟΙΕΣ BLOCKCHAIN.....	18
2.2 ΣΤΟΧΟΣ ΚΑΙ ΠΕΔΙΟ ΤΟΥ ΚΕΦΑΛΑΙΟΥ.....	19
2.3 PROOF OF WORK (POW): ΛΕΙΤΟΥΡΓΙΑ ΓΡΙΦΟΥ, FORKS/REORGs ΚΑΙ ΚΑΤΑΝΑΛΩΣΗ ΙΣΧΥΟΣ....	19
2.4 PROOF OF STAKE (POS): STAKE-ΣΤΑΘΜΙΣΜΕΝΗ ΕΠΙΛΟΓΗ, FORKS ΚΑΙ ΤΕΛΙΚΟΠΟΙΗΣΗ ΣΥΝΑΛΛΑΓΩΝ	20
2.5 ΘΕΩΡΙΑ ΣΥΝΑΙΝΕΣΗΣ ΣΕ ΚΑΤΑΝΕΜΗΜΕΝΑ ΣΥΣΤΗΜΑΤΑ	21
2.7 ΜΕΤΡΙΚΕΣ ΑΞΙΟΛΟΓΗΣΗΣ ΚΑΙ ΤΥΠΟΙ	22
2.8 ΣΥΝΔΕΣΗ ΘΕΩΡΙΑΣ ΜΕ ΤΟΝ ΠΡΟΣΟΜΟΙΩΤΗ	23
2.8 ΠΛΑΙΣΙΟ ΧΑΡΑΚΤΗΡΙΣΜΟΥ ΠΟΥ ΘΑ ΧΡΗΣΙΜΟΠΟΙΗΘΕΙ ΣΤΑ ΑΠΟΤΕΛΕΣΜΑΤΑ	25
<u>ΚΕΦΑΛΑΙΟ 3 ΜΕΘΟΔΟΛΟΓΙΑ: ΠΡΟΣΟΜΟΙΩΤΗΣ, ΡΥΘΜΙΣΕΙΣ, ΣΕΝΑΡΙΑ ΚΑΙ ΔΕΙΚΤΕΣ</u>	<u>27</u>
3.1 ΠΕΡΙΓΡΑΦΗ ΠΡΟΣΟΜΟΙΩΤΗ	27
3.2 ΚΟΙΝΕΣ ΡΥΘΜΙΣΕΙΣ (BASE).....	28
3.3 ΕΠΙΛΟΓΗ ΠΡΟΤΕΙΝΟΝΤΟΣ ΚΑΙ ΡΟΗ ΓΥΡΟΥ	29
3.4 ΣΕΝΑΡΙΑ ΑΞΙΟΛΟΓΗΣΗΣ (FAST, BASELINE, SLOW, SKEW)	29
3.5 ΔΙΑΔΙΚΑΣΙΑ ΕΚΤΕΛΕΣΗΣ.....	31
3.6 ΕΞΑΓΩΓΗ ΚΑΙ ΣΥΝΟΨΗ ΔΕΔΟΜΕΝΩΝ	32
3.7 ΠΕΡΙΟΡΙΣΜΟΙ & ΑΠΕΙΛΕΣ ΕΓΚΥΡΟΤΗΤΑΣ	32
3.8 ΣΥΝΔΕΣΗ ΜΕ ΤΑ ΑΠΟΤΕΛΕΣΜΑΤΑ	33
<u>ΚΕΦΑΛΑΙΟ 4 POW: ΑΠΟΤΕΛΕΣΜΑΤΑ & ΣΥΖΗΤΗΣΗ ΑΝΑ ΣΕΝΑΡΙΟ.</u>	<u>34</u>

4.1 ΜΕΘΟΔΟΛΟΓΙΑ ΚΑΙ ΔΙΑΓΡΑΜΜΑΤΑ ΡΟΗΣ POW	34
4.2 ΣΥΓΚΡΙΤΙΚΗ ΣΥΝΟΨΗ ΣΕΝΑΡΙΩΝ (FAST / BASELINE / SKEW / SLOW) & ΠΙΝΑΚΑΣ ΜΕΤΡΙΚΩΝ	36
4.3 ΣΥΜΠΕΡΑΣΜΑΤΑ ΚΕΦΑΛΑΙΟΥ	44
 <u>ΚΕΦΑΛΑΙΟ 5 POS: ΑΠΟΤΕΛΕΣΜΑΤΑ & ΣΥΖΗΤΗΣΗ ΑΝΑ ΣΕΝΑΡΙΟ.</u>	<u>45</u>
 5.1 ΜΕΘΟΔΟΛΟΓΙΑ	45
5.2 ΔΙΑΓΡΑΜΜΑΤΑ ΡΟΗΣ POS.....	45
5.3 ΑΠΟΤΕΛΕΣΜΑΤΑ & ΣΥΖΗΤΗΣΗ	49
5.4 ΠΙΝΑΚΑΣ ΜΕΤΡΙΚΩΝ POS ΑΝΑ ΣΕΝΑΡΙΟ	53
5.5 ΣΥΜΠΕΡΑΣΜΑΤΑ ΚΕΦΑΛΑΙΟΥ	53
 <u>ΚΕΦΑΛΑΙΟ 6 ΣΥΜΠΕΡΑΣΜΑΤΑ & ΜΕΛΛΟΝΤΙΚΗ ΕΡΓΑΣΙΑ</u>	<u>55</u>
 <u>ΒΙΒΛΙΟΓΡΑΦΙΑ</u>	<u>56</u>

Σχήματα και Πίνακες

Σχήμα 1 - Γενικές παράμετροι (base.yaml)	29
Σχήμα 2 - Παράμετροι PoW ανά σενάριο	30
Σχήμα 3 - Παράμετροι PoS ανά σενάριο	31
Σχήμα 4 - Proof-of-Work (επισκόπηση ροής).	34
Σχήμα 5 - Βρόχος εξόρυξης (mining loop).	35
Σχήμα 6 - Διάδοση & ανάκτηση γονέα (propagation / parent recovery).	36
Σχήμα 7 - Τελικό μήκος αλυσίδας (Final Chain Length)	37
Σχήμα 8 - Forks ανά σενάριο	38
Σχήμα 9 - Reorganizations ανά σενάριο	39
Σχήμα 10 - Μέσο διάστημα εξόρυξης ανά κόμβο (Mining Interval, λογαριθμική κλίμακα)	39
Σχήμα 11 - Orphan rate ανά σενάριο	40
Σχήμα 12 - Χρόνος Block στην τελική μακρύτερη αλυσίδα	41
Σχήμα 13 - Αποκέντρωση (Gini-based)	42
Σχήμα 14 - Θεωρητικά vs Προσομοιωμένα Blocks	42
Σχήμα 15 – Συγκριτικός Πίνακας Μετρικών PoW	43
Σχήμα 16 – Εποπτική απεικόνιση κύκλου PoS	46
Σχήμα 17 – Διάδοση block και οριστικοποίηση (finality)	47
Σχήμα 18 – Ροή Staking/Επιλογής Ηγέτη στο PoS.	48
Σχήμα 19 – Μέσο Throughput (blocks/s) ανά διαμόρφωση PoS	49
Σχήμα 20 – Κατανομή Διαστήματος Blocks	50
Σχήμα 21 – Αποκέντρωση (Gini-based)	50
Σχήμα 22 – Δικαιοσύνη επιλογής ηγετών (απόκλιση KL, D_KL)	51
Σχήμα 23 – Stake vs Πραγματική Συμμετοχή στην Ηγεσία.	52
Σχήμα 24 – Συγκριτικός Πίνακας Μετρικών για τις τέσσερις διαμορφώσεις PoS	53

1.1 Σκοπός και πρόβλημα

Τα κατανεμημένα καθολικά ή blockchain είναι δομές δεδομένων που οργανώνουν συναλλαγές σε αλυσίδα από blocks όπου κάθε block περιέχει κρυπτογραφικό hash του προηγούμενου και σύννοψη περιεχομένου διασφαλίζοντας ακεραιότητα αλληλουχία και δυνατότητα επαλήθευσης από κάθε κόμβο. Η εξέλιξη της αλυσίδας ελέγχεται από μηχανισμό συναίνεσης ώστε οι κόμβοι να συμφωνούν στο επόμενο έγκυρο block ή ledger δηλαδή την οριστικοποιημένη κατάσταση του καθολικού στο τέλος ενός γύρου συναίνεσης χωρίς κεντρικό εμπιστευμένο μεσολαβητή.

Η παρούσα πτυχιακή εργασία χαρακτηρίζει και περιγράφει δύο διαφορετικές προσεγγίσεις συναίνεσης σε κατανεμημένα καθολικά: Proof-of-Work (PoW, πρωτόκολλο όπου η πρόταση block απαιτεί απόδειξη υπολογιστικού έργου) και Proof-of-Stake (PoS, πρωτόκολλο όπου η επιλογή ηγέτη γίνεται με πιθανότητα ανάλογη του stake που έχει δεσμεύσει ο κόμβος). Στόχος είναι η ποσοτική αποτίμηση των βασικών δεικτών απόδοσης και αξιοπιστίας υπό ενιαίες, ελεγχόμενες συνθήκες. Συγκεκριμένα, μετριοούνται:

Throughput (συναλλαγές/δευτ., δηλαδή ο ρυθμός διεκπεραίωσης συναλλαγών ανά δευτερόλεπτο), καθυστέρηση επιβεβαίωσης ή τελικοποίησης (blocktime / finality latency, όπου blocktime είναι ο μέσος χρόνος μεταξύ δύο διαδοχικών blocks και finality latency ο χρόνος μέχρι μια συναλλαγή ή ένα block να θεωρηθεί μη αναστρέψιμο), σταθερότητα αλυσίδας (forks, reorgs, orphans, propagation delay, όπου fork είναι η προσωρινή διακλάδωση όταν συνυπάρχουν δύο έγκυρες κορυφές στο ίδιο ύψος, reorg η αναδιοργάνωση κατά την οποία εγκαταλείπεται ένα κλαδί και υιοθετείται άλλο με περισσότερη εργασία ή ύψος, orphans τα blocks που δεν εντάχθηκαν τελικά στη μακρύτερη αλυσίδα και propagation delay ο μέσος χρόνος διάδοσης ενός block στο δίκτυο) και δείκτες αποκέντρωσης ή δικαιοσύνης (κατανομή παραγωγής blocks ή ηγεσίας σε σχέση με υπολογιστική ισχύ ή μερίδιο stake και πόσο ταιριάζει αυτή με την αναμενόμενη κατανομή πόρων hashpower ή stake).

Η παρούσα εργασία εστιάζει αποκλειστικά στον δίκαιο και επαναλήψιμο χαρακτηρισμό των πρωτοκόλλων, τα οποία, λόγω διαφορετικής φιλοσοφίας και παραμέτρων, συχνά περιγράφονται υπό ετερογενείς συνθήκες. Για τον σκοπό αυτό, θέτει κοινό πειραματικό πλαίσιο ώστε οι ιδιότητες κάθε οικογένειας να καταγράφονται με συνέπεια και να αναπαράγονται με ακρίβεια, μέσω ρυθμίσεων και μεταβλητών του προσομοιωτή (expected_block_time, block_interval, round_timeout, confirmation_threshold, num_neighbours). Expected_block_time είναι ο στόχος μέσου χρόνου μεταξύ blocks στο PoW, block_interval είναι η διάρκεια slot παραγωγής block στο PoS, confirmation_threshold είναι το απαιτούμενο ποσοστό/βάρος stake ψήφων (CONFIRM) για να γίνει ένα block

Finalized, round_timeout είναι το μέγιστο διάστημα αναμονής πριν επαναληφθεί ο γύρος, num_neighbours είναι ο αριθμός γειτονικών συνδέσεων κάθε κόμβου.

Κίνητρο για τη χρήση blockchain είναι η ανάγκη καταγραφής και εκτέλεσης συναλλαγών σε ανοικτά ή πολυμερή περιβάλλοντα χωρίς κεντρικό σημείο εμπιστοσύνης. Η αποκέντρωση επιτρέπει σε ανεξάρτητους κόμβους να επιτυγχάνουν συναίνεση μέσω μηχανισμών όπως PoW και PoS αξιοποιώντας αντίστοιχους πόρους ή παραδοχές όπως υπολογιστική ισχύ stake. Με αυτόν τον τρόπο μειώνεται ο κίνδυνος λογοκρισίας ή αυθαίρετων μεταβολών ενώ ενισχύονται ακεραιότητα διαφάνεια και επαληθευσιμότητα.

Χρήσεις των blockchain περιλαμβάνουν συστήματα πληρωμών και ψηφιακά νομίσματα διακανονισμό περιουσιακών στοιχείων ιχνηλασιμότητα στην εφοδιαστική αλυσίδα ταυτοποίηση και έλεγχο πρόσβασης καθώς και εκτέλεση έξυπνων συμβολαίων όπου ο κώδικας εφαρμόζει κανόνες με τρόπο επαληθεύσιμο από όλους τους κόμβους. Στα σενάρια της εργασίας οι ιδιότητες που σχετίζονται με αυτές τις χρήσεις αποτυπώνονται μέσα από μετρικές όπως Throughput Blocktime και Finality latency και από παραμέτρους του προσομοιωτή όπως expected_block_time, block_interval, round_timeout και confirmation_threshold.

Οι τύποι blockchain χωρίζονται σε public, private και consortium ανάλογα με το ποιοι κόμβοι μπορούν να συμμετέχουν στην επικύρωση. Στα public η συμμετοχή είναι ανοικτή και η συναίνεση επιτυγχάνεται με χρήση πόρων όπως hashpower ή stake. Στα private ένας οργανισμός ορίζει τα δικαιώματα συμμετοχής και επικύρωσης. Στα consortium η διακυβέρνηση μοιράζεται ανάμεσα σε προκαθορισμένα μέλη σχηματίζοντας ομοσπονδιακό σχήμα. Επιπλέον υπάρχει διάκριση μεταξύ permissionless και permissioned blockchain ανάλογα με το αν απαιτείται έγκριση για την ένταξη ενός κόμβου στο δίκτυο. Στο επόμενο τμήμα διατυπώνονται τα κεντρικά ερωτήματα και οι άξονες χαρακτηρισμού που καθοδηγούν τα σενάρια Fast Baseline Skew Slow για PoW και PoS.

1.2 Κίνητρα και ερευνητικά ερωτήματα

Η εργασία εστιάζει στον συστηματικό χαρακτηρισμό των οικογενειών πρωτοκόλλων συναίνεσης με έμφαση στα PoW και PoS και στη χαρτογράφηση των βασικών τους ιδιοτήτων υπό ελεγχόμενες και αναπαραγώγιμες συνθήκες. Η ανάλυση οργανώνεται γύρω από άξονες που απορρέουν από τη λειτουργία τους όπως ο μηχανισμός πρότασης και δέσμευσης block, η φύση της τελικοποίησης (πιθανολογική ή καθοριστική), οι προϋποθέσεις εμπιστοσύνης και οι κρίσιμες παράμετροι που ρυθμίζουν τη συμπεριφορά τους όπως expected_block_time, block_interval, round_timeout, confirmation_threshold και num_neighbours.

Τα βασικά ερευνητικά ερωτήματα εστιάζουν στον χαρακτηρισμό. Ποιες λειτουργικές ιδιότητες ορίζουν την ταυτότητα κάθε πρωτοκόλλου PoW και PoS. Πώς επηρεάζουν οι παράμετροι λειτουργίας το προφίλ απόδοσης και αξιοπιστίας κάθε πρωτοκόλλου χωρίς πρόθεση ιεράρχησης. Ποια μοτίβα παραμένουν σταθερά όταν μεταβάλλονται χρονισμοί και συνδεσιμότητα όπως `block_interval`, `round_timeout`, `expected_block_time` και `num_neighbours` και πώς αποτυπώνονται αυτές οι ιδιότητες στις μετρικές `throughput`, `block_time`, `finality`, `latency`, `forks`, `reorgs`, `orphan_rate` και δείκτες αποκέντρωσης ή δικαιοσύνης. Στόχος είναι μια σαφής περιγραφή του λειτουργικού αποτυπώματος κάθε οικογένειας πρωτοκόλλων και των παραγόντων που το διαμορφώνουν.

1.3 Κύριες συνεισφορές

Η μελέτη υιοθετεί παραγοντικό σχεδιασμό 2×4 δύο πρωτόκολλα επί τέσσερα σενάρια Fast Baseline Skew Slow με εναρμονισμένες παραμέτρους όπου υπάρχει εννοιολογική αντιστοίχιση όπως ρυθμός `block` ή `round` τα `timeouts` και ο αριθμός γειτόνων ώστε ο χαρακτηρισμός να είναι δίκαιος και επαναλήψιμος. Η παρουσίαση γίνεται ενδο πρωτοκολλιακά με γραφήματα και πίνακες ανά σενάριο χωρίς δια πρωτοκολλιακή αντιπαραβολή. Καθιερώνεται συνεπής ονοματολογία ευρημάτων για παράδειγμα PoW Fast, PoS Skew η οποία ενισχύει σαφήνεια και ιχνηλασιμότητα σε όλο το κείμενο. Η συνεισφορά είναι η συστηματική περιγραφή των ιδιοτήτων κάθε οικογένειας ανά σενάριο με μετρικές όπως `Throughput` `Blocktime` `Finality` `latency` και ειδικές ιδιότητες όπως `forks` `reorgs` `orphan_rate` `propagation delay` `timeouts` και `retries`.

1.4 Μεθοδολογία και πλαίσιο αξιολόγησης

Τα σενάρια χαρτογραφούνται ως εξής. Το Fast χρησιμοποιεί επιθετικούς χρονισμούς και μικρά `timeouts` με πιο αραιή συνδεσιμότητα (ενδεικτικά τρεις γείτονες), δίνοντας έμφαση στη διαπερατότητα. Το Baseline επιλέγει ισορροπημένες τιμές και μέση συνδεσιμότητα (π.χ. έξι γείτονες), στοχεύοντας στη σταθερότητα. Το Skew εισάγει άνιση ισχύ ή μερίδιο (`power skew` στο PoW, `stake skew` στο PoS) και λειτουργεί ως τεστ δικαιοσύνης/αποκέντρωσης. Τέλος, το Slow υιοθετεί βραδύ ρυθμό και συντηρητικούς χρονισμούς για να ευνοηθεί η ντετερμινιστική πρόοδος και η ασφαλής τελικοποίηση.

Για κάθε σενάριο και πρωτόκολλο μετράμε `throughput`, `blocktime` ή `finality latency`, `forks/reorgs/orphans` και καθυστέρηση διάδοσης όπου έχει νόημα, καθώς και δείκτες αποκέντρωσης ή δικαιοσύνης. Ως κύριος δείκτης χρησιμοποιείται η απόκλιση `KL D_KL` που μετρά πόσο απέχει η πραγματική κατανομή παραγωγής `blocks` ή ηγεσιών από την αναμενόμενη κατανομή πόρων για παράδειγμα στο PoW η συσχέτιση `blocks` με ισχύ και στο PoS ηγετών με `stake`. Η διακοπή των πειραμάτων και η αναπαραγωγιμότητα ορίζονται σαφώς. Και στα PoW και PoS τα τρέχοντα πειράματα τερματίζουν με `simTime = 3600 s`.

Όπου απαιτείται χρησιμοποιούνται σταθεροί seeds ώστε τα αποτελέσματα να μπορούν να επαναληφθούν.

Οι παράμετροι που ελέγχουν τον χαρακτηρισμό είναι ρητά χαρτογραφημένες στον προσομοιωτή. Για το PoW χρησιμοποιείται `expected_block_time` και κατανομή `hashpower`. Για το PoS χρησιμοποιούνται `block_interval` και `confirmation_threshold`. Κοινές δικτυακές παράμετροι είναι `num_neighbours` `msg_val_delay` `queueing_delay` `processing_delay` και το `bandwidth`. Για τον ορίζοντα εκτέλεσης χρησιμοποιούνται `simulation.simTime`. Για τον ρυθμό και τα μεγέθη δεδομένων χρησιμοποιούνται `Tl_dur` `Tn` `base_block_size` `Bsize` και `Tsize`. Η ονομασία των σεναρίων είναι Fast Baseline Skew Slow και εφαρμόζεται σε κάθε πρωτόκολλο ξεχωριστά. Στο επόμενο κεφάλαιο παρουσιάζεται το θεωρητικό υπόβαθρο της συναίνεσης και χαρτογραφούνται ρητά οι μετρικές και οι παράμετροι του προσομοιωτή `expected_block_time` `block_interval` `round_timeout` `confirmation_threshold` και `num_neighbours` ώστε ο χαρακτηρισμός ανά σενάριο να είναι πλήρως τεκμηριωμένος.

1.5 Σύντομη επισκόπηση πρωτοκόλλων με σεναριακή οπτική

Στην ενότητα αυτή γίνεται μια γενική παρουσίαση των τριών οικογενειών πρωτοκόλλων που μελετώνται. Το Proof of Work PoW βασίζεται στον ανταγωνισμό υπολογιστικής ισχύος για την παραγωγή νέων blocks και αντιπροσωπεύει την πιο διαδεδομένη μορφή ανοικτής συναίνεσης. Το Proof of Stake PoS επιλέγει παραγωγούς blocks με πιθανότητα ανάλογη του μεριδίου stake και επιδιώκει μικρότερη κατανάλωση πόρων και σταθερό ρυθμό παραγωγής. Στο παρόν κεφάλαιο η περιγραφή περιορίζεται στην εντύπωση και στη βασική φιλοσοφία κάθε πρωτοκόλλου ενώ οι λεπτομέρειες αλγορίθμων και παραμέτρων παρουσιάζονται αναλυτικά στα κεφάλαια που ακολουθούν.

Στο Proof of Work (PoW) οι κόμβοι επιλύουν υπολογιστικό γρίφο κατακερματισμού, με συνέπεια υψηλή κατανάλωση πόρων και την εμφάνιση forks/reorgs. Στο Proof of Stake (PoS) η παραγωγή blocks βασίζεται στην κατανομή stake και στην ψήφο των validators, με στόχο μείωση της ενεργειακής κατανάλωσης. Η αναλυτική παρουσίαση των παραπάνω αλγορίθμων, μαζί με την εσωτερική τους λειτουργία και τις σχετικές παραμέτρους, παρατίθεται στο Κεφάλαιο 2.

Στη συνέχεια παρουσιάζονται τα τέσσερα σενάρια Fast Baseline Skew Slow με έμφαση στο πώς οι ρυθμίσεις παραμέτρων επηρεάζουν την πρόοδο και τη σταθερότητα του συστήματος. Το ίδιο πλαίσιο σεναρίων εφαρμόζεται σε όλα τα πρωτόκολλα και οι λεπτομέρειες κάθε περίπτωσης δίνονται στις αντίστοιχες υποενότητες και κεφάλαια.

1.6 Περιορισμοί και πεδίο εφαρμογής

Η προσομοίωση απλοποιεί ορισμένες πτυχές (π.χ. πλήρες δίκτυο, επιθέσεις εκτός πεδίου, οικονομικά κίνητρα. Τα αποτελέσματα ερμηνεύονται εντός κάθε πρωτοκόλλου ανά σενάριο. Η γενίκευση σε παραγωγικά δίκτυα απαιτεί επιπλέον παραμέτρους (διαφορετική τοπολογία, churn, ετερογενείς καθυστερήσεις).

Η αναπαραγωγικότητα εξασφαλίζεται με σταθερά seeds για όλες τις στοχαστικές επιλογές (όπως επιλογή ηγετών και προγραμματισμός γεγονότων). Οι ρυθμίσεις των πειραμάτων οργανώνονται σε τυποποιημένα σύνολα παραμέτρων με συνεπή ονοματολογία, ενώ σε όλες τις εκτελέσεις χρησιμοποιείται σταθερή τοπολογία οκτώ κόμβων. Δεν εξετάζονται προχωρημένα σχήματα δυναμικής ρύθμισης δυσκολίας στο PoW (πέρα από το απλό retargeting που υλοποιείται, ούτε οικονομικά κίνητρα ή επιθετικά σενάρια. Το εύρος του κεφαλαίου περιορίζεται στον λειτουργικό χαρακτηρισμό σεναρίων.

1.7 Δομή της εργασίας

Η εργασία οργανώνεται ως εξής: στο Κεφάλαιο 2 παρουσιάζεται το θεωρητικό υπόβαθρο της συναίνεσης, στο Κεφάλαιο 3 περιγράφονται ο προσομοιωτής, οι ρυθμίσεις, τα σενάρια και οι δείκτες, ακολουθούν δύο κεφάλαια με αποτελέσματα και συζήτηση ανά πρωτόκολλο (Κεφ. 4 PoW, Κεφ. 5 PoS). Το Κεφάλαιο 6 καταγράφει τα συμπεράσματα και προτείνει μελλοντική εργασία.

1.8 Οδηγός ανάγνωσης αποτελεσμάτων

Κάθε κεφάλαιο που αφορά συγκεκριμένο πρωτόκολλο ακολουθεί ενιαία ροή. Για κάθε σενάριο Fast Baseline Skew Slow παρουσιάζονται τα γραφήματα ανά μετρική με σύντομο σχολιασμό ώστε να φαίνεται τι δείχνει το διάγραμμα και γιατί έχει σημασία. Στη συνέχεια ακολουθούν συνοπτικά σχήματα που βάζουν δίπλα δίπλα όλα τα σενάρια του ίδιου πρωτοκόλλου και επιτρέπουν άμεση οπτική αποτύπωση του προφίλ του. Δεν περιλαμβάνονται δια πρωτοκολλιακές αντιπαραβολές. Ο κωδικός ονοματολογίας «Πρωτόκολλο–Σενάριο» (π.χ. PoS–Skew) χρησιμοποιείται σε άξονες, λεζάντες και κείμενο.

ΚΕΦΑΛΑΙΟ 2 Υπόβαθρο: θεωρία συναίνεσης και μετρικές

2.1 Βασικές Έννοιες Blockchain

Σκοπός του κεφαλαίου είναι να δώσει το ενιαίο λεξιλόγιο και τα κριτήρια αξιολόγησης για PoW και PoS πριν περάσουμε στη μεθοδολογία.

Τα blockchain είναι κατανεμημένα καθολικά όπου οι συναλλαγές ομαδοποιούνται σε blocks τα οποία συνδέονται αλυσιδωτά μέσω κρυπτογραφικών hashes. Κάθε block έχει επικεφαλίδα που περιλαμβάνει το hash του προηγούμενου block, τη χρονική σήμανση και μεταδεδομένα, και σώμα που περιέχει τις συναλλαγές. Αυτή η δομή εξασφαλίζει ακεραιότητα και ανίχνευση αλλαγών στην ιστορία. Οι κόμβοι του δικτύου διατηρούν ένα αντίγραφο του καθολικού—πλήρες ή μερικό—και συνεργάζονται για να συμφωνήσουν στο επόμενο block

Στον προσομοιωτή αυτό γίνεται με δύο τρόπους: στο PoW το επόμενο block παράγεται μέσω υπολογιστικού έργου και ο μέσος ρυθμός παραγωγής ρυθμίζεται από το `expected_block_time`.

Στο PoS οι ηγέτες επιλέγονται με stake-weighted sampling (βάσει stake). Ο ρυθμός των γύρων/slots καθορίζεται από το `block_interval`. Ένα block γίνεται τελικό όταν ξεπεραστεί το `confirmation_threshold`, δηλαδή όταν το συσσωρευμένο βάρος των CONFIRM ψήφων υπερβεί το `confirmation_threshold × total_stake`.

Με αυτά τα βασικά στοιχεία, οι κόμβοι μπορούν να ευθυγραμμίζονται σταθερά παρότι επικοινωνούν με καθυστερήσεις και μερικές φορές βλέπουν διαφορετική εικόνα του δικτύου.

Η βασική λειτουργία περιλαμβάνει δημιουργία block, επικύρωση, διάδοση και οριστικοποίηση. Όταν δύο blocks προταθούν σχεδόν ταυτόχρονα, το πρωτόκολλο επιλύει την ισοπαλία με κανόνα επιλογής αλυσίδας. Για το PoW ισχύει ο κανόνας longest-chain. Η σταθερότητα επηρεάζεται από forks, reorgs, orphan rate και propagation delay, τα οποία ο προσομοιωτής καταγράφει.

Για την αποτίμηση χρησιμοποιούνται μετρικές όπως throughput, blocktime, finality latency και δείκτες δικαιοσύνης (π.χ. απόκλιση KL και σχέση κατανομής ηγετών προς stake ή hashpower). Παράμετροι όπως `num_neighbours`, `msg_val_delay` και `block_val_delay` μοντελοποιούν το δίκτυο και επιτρέπουν μελέτη ευαισθησίας σε καθυστερήσεις.

Αυτό το θεωρητικό υπόβαθρο θέτει το πλαίσιο για την ανάλυση που ακολουθεί στα πρωτόκολλα PoW και PoS, προετοιμάζοντας την παρουσίαση των τεσσάρων σεναρίων Fast, Baseline, Skew και Slow. Στις επόμενες ενότητες παρουσιάζονται ξεχωριστά τα πρωτόκολλα PoW και PoS, με έμφαση στους μηχανισμούς συναίνεσης και στις παραμέτρους

expected_block_time, block_interval, round_timeout και confirmation_threshold που καθορίζουν τη συμπεριφορά τους.

2.2 Στόχος και πεδίο του κεφαλαίου

Η ενότητα αυτή συνοψίζει το πλαίσιο μέσα στο οποίο λειτουργούν τα πρωτόκολλα που μελετώνται και προετοιμάζει το έδαφος για την ανάλυση της συναίνεσης. Ξεκινάμε από τα βασικά: το blockchain είναι ένα κατανεμημένο, αποκεντρωμένο καθολικό όπου τα δεδομένα ομαδοποιούνται σε blocks και συνδέονται κρυπτογραφικά, προσφέροντας ανιχνευσιμότητα και ακεραιότητα χωρίς κεντρική αρχή. Στη συνέχεια εξετάζουμε συνοπτικά τους μηχανισμούς συναίνεσης που θα χρησιμοποιηθούν στα πειράματα και τις παραμέτρους που επηρεάζουν τη συμπεριφορά τους.

Το κίνητρο για τη χρήση blockchain προκύπτει σε περιβάλλοντα με ελλιπή αμοιβαία εμπιστοσύνη: η αξιοπιστία δεν στηρίζεται σε ενδιάμεσους αλλά σε κρυπτογραφία, διαφάνεια και αλγόριθμους συναίνεσης, μειώνοντας κινδύνους λογοκρισίας και μονοσήμαντων σημείων αστοχίας. Οι εφαρμογές έχουν επεκταθεί πέρα από τα κρυπτονομίσματα και περιλαμβάνουν πληρωμές και διασυνοριακές μεταφορές αξίας, ιχνηλασιμότητα σε αλυσίδες εφοδιασμού, αυτοματοποιημένες συμφωνίες με έξυπνα συμβόλαια, καθώς και σενάρια κοινής χρήσης δεδομένων σε διοίκηση, υγεία και δίκτυα IoT όπου απαιτείται ανθεκτικότητα και έλεγχος πρόσβασης.

Ανάλογα με το ποιος συμμετέχει και ποιος επικυρώνει, διακρίνουμε δημόσια δίκτυα με ανοικτή συμμετοχή, ιδιωτικά δίκτυα που ελέγχονται από έναν οργανισμό και consortium/permissioned δίκτυα που διαχειρίζονται συλλογικά από πολλούς οργανισμούς με ελεγχόμενη πρόσβαση. Η επιλογή τύπου επηρεάζει άμεσα τις απαιτήσεις συναίνεσης, την απόδοση και τον βαθμό αποκέντρωσης, άρα και τον τρόπο που παραμετροποιούνται τα πρωτόκολλα στα πειράματα.

Με αυτό το υπόβαθρο, στις επόμενες υποενότητες περνάμε στις ειδικότερες αρχές των μηχανισμών συναίνεσης που αξιολογούμε και στις ρυθμίσεις του προσομοιωτή που επηρεάζουν τη διάδοση, τον ρυθμό παραγωγής blocks και τη διαδικασία τελικοποίησης.

2.3 Proof of Work (PoW): λειτουργία γρίφου, forks/reorgs και κατανάλωση ισχύος

Στο Proof-of-Work (PoW), όπως περιγράφει ο Nakamoto (2008), η παραγωγή blocks προκύπτει από ανταγωνισμό υπολογιστικής ισχύος (mining_power) και η επιλογή αλυσίδας ακολουθεί τον κανόνα της μακρύτερης/βαρύτερης αλυσίδας. Η τυπική ανάλυση του κανόνα σύγκλισης παρουσιάζεται από τους Garay, Kiayias & Leonardos (2015), ενώ πρακτικές συνέπειες της καθυστέρησης διάδοσης που οδηγούν σε forks έχουν μελετηθεί από τους

Decker & Wattenhofer (2013). Στον προσομοιωτή, ο ρυθμός δημιουργίας blocks παραμετροποιείται από το `expected_block_time` και τα βάρη `mining_power`. Δεν εφαρμόζεται `retargeting` δυσκολίας.

Λόγω καθυστερήσεων διάδοσης (`propagation delay`) μπορεί να εμφανιστούν `forks` όταν πολλοί κόμβοι παράγουν έγκυρα blocks σχεδόν ταυτόχρονα. Η σύγκλιση επιτυγχάνεται όταν ένα κλαδί γίνει μακρύτερο/βαρύτερο και επικρατήσει, οπότε τα blocks του άλλου κλαδιού εγκαταλείπονται (`reorg`) και μετρώνται ως `orphans`. Αυτός ο μηχανισμός εξηγεί γιατί στο PoW η τελικοποίηση είναι πιθανολογική: όσο προστίθενται blocks πάνω από ένα παλαιότερο block, τόσο μειώνεται η πιθανότητα αναστροφής του.

Στο πλαίσιο της παρούσας εργασίας, δεν εφαρμόζεται κανόνας τελικοποίησης με `k` επιβεβαιώσεις για το PoW. Αντί αυτού καταγράφονται: `transaction latency` (από `tx.timestamp` έως `block.time_added`), `block time/throughput` και οι δείκτες `forks`, `reorgs`, `orphan rate`, καθώς και `chain length` και `propagation delay`. Επιπλέον, για την αξιολόγηση της αποκέντρωσης χρησιμοποιείται ο δείκτης Gini, ο οποίος μετρά την ανισότητα στην κατανομή παραγωγής blocks σε σχέση με την κατανεμημένη υπολογιστική ισχύ (`mining_power`).

Τέλος, τα αποτελέσματα στα σενάρια Fast/Baseline/Skew/Slow ερμηνεύονται κυρίως ως συνάρτηση των `expected_block_time`, `num_neighbours` και του προφίλ `latency`: μικρότερες καθυστερήσεις και πιο επιθετικές ρυθμίσεις αυξάνουν την πιθανότητα `forks` και το `orphan rate`, ενώ πιο «συντηρητικές» ρυθμίσεις μειώνουν τα παραπάνω με κόστος τον ρυθμό παραγωγής.

Επισκόπηση λειτουργίας (σύνοψη). Στο PoW κάθε κόμβος συγκεντρώνει εκκρεμείς συναλλαγές, σχηματίζει υποψήφιο block και ανταγωνίζεται λύνοντας ένα κρυπτογραφικό παζλ· ο πρώτος που βρίσκει λύση διαδίδει το block και οι λοιποί το επαληθεύουν και το προσθέτουν στην αλυσίδα τους. Λόγω καθυστερήσεων διάδοσης μπορεί να εμφανιστούν ταυτόχρονα έγκυρες κορυφές (`forks`). Η σύγκλιση επιτυγχάνεται όταν ένα κλαδί γίνει μακρύτερο (κανόνας `longest-chain`) και επικρατήσει, ενώ το άλλο εγκαταλείπεται (`reorg`). Η τελικοποίηση είναι κατ' ουσία πιθανολογική και ενισχύεται όσο αυξάνεται το βάθος πάνω από το υπό εξέταση block. Οι επιλογές σχεδιασμού (μέγεθος block, διάστημα blocks, όρια δικτύου) δημιουργούν σαφή ανταλλαγή ανάμεσα σε απόδοση και ασφάλεια, όπως αναλύουν οι Gervais κ.ά. (2016).

2.4 Proof of Stake (PoS): stake-σταθμισμένη επιλογή, forks και τελικοποίηση συναλλαγών

Στο Proof-of-Stake (PoS) η δημιουργία blocks δεν βασίζεται σε υπολογιστική ισχύ αλλά στην κατανομή του `stake`: κάθε κόμβος έχει ποσοστό συμμετοχής που κανονικοποιείται και ορίζει την πιθανότητα να επιλεγεί ως προτείνων (`proposer`). Η διαδοχή των `slots` ρυθμίζεται από το `block_interval` (και γενικότερα από `slot_len`), ενώ η τελικοποίηση (`finality`) είναι

threshold-based: ένα block χαρακτηρίζεται Finalized όταν το συσσωρευμένο βάρος ψήφων CONFIRM υπερβεί το $\text{confirmation_threshold} \times \text{total_stake}$. Η ιδέα stake-weighted επιλογής ηγετών και οι ιδιότητες σύγκλισης έχουν παρουσιαστεί εκτενώς στη βιβλιογραφία (Ouroboros: Kiayias et al., 2017; David et al., 2018), ενώ εναλλακτικές προσεγγίσεις τελικοποίησης συζητούνται σε συστήματα τύπου Casper FFG (Buterin & Griffith, 2017). Στον προσομοιωτή, η επιλογή ηγέτη γίνεται με stake-weighted randomness.

Η διαδικασία οργανώνεται σε rounds/slots. Σε κάθε round επιλέγεται leader, ο οποίος συνθέτει νέο block και το διαδίδει (gossip) στο δίκτυο ως NEW_BLOCK. Οι υπόλοιποι κόμβοι προχωρούν σε Validate και, αν είναι έγκυρο, εκδίδουν confirmation (stake-weighted vote) που διαδίδεται ως CONFIRM. Κάθε κόμβος κάνει tally by stake και όταν επιτευχθεί το $\text{confirmation_threshold}$ το block γίνεται Finalized και αναγγέλλεται μία φορά ως FINALIZED. Αν δεν συγκεντρωθεί εγκαίρως το απαιτούμενο βάρος ψήφων, ενεργοποιείται timeout και ξεκινά νέος γύρος.

Η τελικοποίηση μετρίεται ρητά με τον δείκτη finality latency (χρόνος από τη δημιουργία block έως τη στιγμή που περνά το threshold). Ο χρόνος των γύρων επηρεάζεται από το $\text{block_interval/slot_len}$ και τα timeouts που έχουν οριστεί στο configuration (round/slot timeout). Ενδεικτικό παράδειγμα PoS με ισχυρή τελικοποίηση και επιλογή επιτροπών μέσω VRF είναι το Algorand (Gilad et al., 2017).

Οι μετρικές που παράγονται για το PoS περιλαμβάνουν: block interval, throughput (blocks/s), finality latency, (ενδεχομένως) forks και leader frequency, για τη δικαιοσύνη (leader fairness) χρησιμοποιούνται οι δείκτες Kullback–Leibler απόκλιση (D_KL) και Gini (σύγκριση της ιδεώδους stake-proportional κατανομής ηγετών με την πραγματική της προσομοίωσης).

Τα σενάρια Fast, Baseline, Skew και Slow διαφοροποιούν παραμέτρους όπως block_interval , round/slot timeout και (όπου ισχύει) την ίδια την stake distribution, ώστε να μελετηθούν οι επιπτώσεις σε finality latency, throughput και fairness/decentralization.

Επισκόπηση λειτουργίας (σύνοψη). Το PoS αντικαθιστά την υπολογιστική ισχύ με οικονομικό μερίδιο συμμετοχής: σε κάθε slot γίνεται stake-weighted sampling για τον leader, ο οποίος προτείνει block. Λόγω καθυστερήσεων διάδοσης μπορεί να εμφανιστούν προσωρινά forks όταν δύο προτάσεις απέχουν χρονικά, όμως η αλυσίδα που συγκεντρώνει ταχύτερα το απαιτούμενο stake πάνω από το $\text{confirmation_threshold}$ επικρατεί. Η finality εκφράζεται ως επίτευξη threshold και μετρίεται ρητά από τον προσομοιωτή ως finality latency.

2.5 Θεωρία συναίνεσης σε κατανεμημένα συστήματα

Καθοριστικό ρόλο διαδραματίζει το μοντέλο χρονισμού. Στο ασύγχρονο μοντέλο δεν υπάρχουν άνω όρια καθυστέρησης μηνυμάτων και ισχύει το θεώρημα αδυναμίας FLP, σύμφωνα με το οποίο κανένας ντετερμινιστικός αλγόριθμος συναίνεσης δεν μπορεί να εγγυηθεί ταυτόχρονα ασφάλεια (safety) και ζωντάνια (liveness) όταν επιτρέπεται έστω και ένα σφάλμα διεργασίας. Στο μερικώς συγχρονισμένο μοντέλο, μετά από έναν άγνωστο χρόνο σταθεροποίησης (GST), οι καθυστερήσεις περιορίζονται από πρακτικά άνω όρια· πάνω σε αυτή την υπόθεση στηρίζεται το μεγαλύτερο μέρος των υλοποιήσιμων BFT πρωτοκόλλων. Τέλος, στο πλήρως συγχρονισμένο μοντέλο τα όρια καθυστερήσεων θεωρούνται γνωστά και σταθερά εκ των προτέρων.

Όσον αφορά το μοντέλο σφαλμάτων, γίνεται διάκριση μεταξύ crash faults, όπου οι κόμβοι απλώς παύουν να λειτουργούν και σταματούν να επικοινωνούν, και Byzantine faults, όπου οι κόμβοι μπορούν να ενεργούν αυθαίρετα ή κακόβουλα, στέλνοντας αντιφατικά ή λανθασμένα μηνύματα. Στο κλασικό πλαίσιο BFT με σταθερό γνωστό σύνολο επικυρωτών διασφαλίζεται ορθότητα όταν οι Βυζαντινοί κόμβοι είναι λιγότεροι από το ένα τρίτο $f < n/3$. Στα συστήματα τύπου Nakamoto η ασφάλεια παραμένει πιθανολογική και εξαρτάται από τον πόρο που ελέγχει ο αντίπαλος όπως η αναλογία hashpower στο PoW ή stake στο PoS και όσο μικρότερο είναι το μερίδιο του αντιπάλου τόσο μειώνεται η πιθανότητα επιτυχούς ανατροπής. Στις επόμενες ενότητες παρουσιάζονται οι οικογένειες PoW και PoS με τη βασική αρχιτεκτονική τους και τον τρόπο που εφαρμόζουν τις παραπάνω έννοιες στην πράξη.

2.7 Μετρικές αξιολόγησης και τύποι

Ως throughput νοείται ο ρυθμός διεκπεραίωσης του συστήματος, σε blocks ανά δευτερόλεπτο ή συναλλαγές ανά δευτερόλεπτο. Ως block time ορίζεται ο μέσος χρόνος ανάμεσα σε δύο διαδοχικές δεσμεύσεις στην κύρια αλυσίδα. Η finality latency μετριέται μόνο όταν υπάρχει ρητός κανόνας τελικοποίησης. Στο PoS του παρόντος simulator η finality είναι threshold-based: κάθε κόμβος αθροίζει το βάρος των CONFIRM ψήφων και ένα block θεωρείται Finalized όταν το συσσωρευμένο stake υπερβεί το $\text{confirmation_threshold} \times \text{total_stake}$. Η finality latency ορίζεται ως ο χρόνος από την πρόταση του block έως τη στιγμή που επιτυγχάνεται το παραπάνω threshold και εξαρτάται από ρυθμό διάδοσης (gossip), συμμετοχή validators και ρυθμίσεις χρονισμού. Στο PoW η αξιολόγηση εστιάζει σε block time, throughput και στη συμπεριφορά αποκλίσεων της αλυσίδας.

Οι καθυστερήσεις διάδοσης επηρεάζουν άμεσα την εμφάνιση forks. Όταν δύο blocks φτάσουν σε διαφορετικούς κόμβους με μικρή χρονική διαφορά, οι τοπικές αλυσίδες μπορεί να διακλαδωθούν προσωρινά. Η σχέση ανάμεσα στο propagation delay και στη συχνότητα των προσωρινών διακλαδώσεων έχει μελετηθεί από τους Decker & Wattenhofer (2013), ενώ η σύγκλιση προς τη μακρύτερη αλυσίδα τυποποιείται στο μοντέλο Bitcoin Backbone από

τους Garay, Kiayias & Leonardos (2015). Στον προσομοιωτή καταγράφουμε για κάθε κόμβο τον μέσο χρόνο διάδοσης block, τον αριθμό forks και reorgs και τον δείκτη orphan rate, ώστε να εκτιμούμε τη σταθερότητα της αλυσίδας κάτω από διαφορετικές καθυστερήσεις.

Στο PoW η παρέκκλιση της αλυσίδας αποτυπώνεται ως orphan rate, ενώ η reorg length δείχνει το μέγιστο μήκος αναδιοργάνωσης που παρατηρήθηκε. Με αυτόν τον τρόπο οι χρονισμοί των γύρων και η ανθεκτικότητα σε καθυστερήσεις προκύπτουν άμεσα από τα logs του προσομοιωτή.

Η δικαιοσύνη αποτιμάται ως η απόκλιση από τις θεωρητικές αναλογίες: στο PoW (βλ. §2.3, ο Nakamoto) τα βάρη προκύπτουν από το mining_power και τα αποτελέσματα από το μερίδιο παραγωγής blocks, ενώ στο PoS (βλ. §2.4, η οικογένεια Ouroboros) τα βάρη προκύπτουν από τα stakes και τα αποτελέσματα από το μερίδιο ηγεσιών. Η απόκλιση μετριέται με τον δείκτη D_KL. Τιμές κοντά στο μηδέν δείχνουν πιο δίκαιη, αναλογική συμπεριφορά. Συμπληρωματικά, ο συντελεστής Nakamoto εκφράζει τον ελάχιστο αριθμό συμμετεχόντων που αρκεί για να ελέγχουν πάνω από το 50% του αντίστοιχου πόρου ή αποτελέσματος. Μαζί, οι δύο μετρικές δίνουν καθαρή εικόνα τόσο της αποδοτικότητας όσο και της ανθεκτικότητας κάθε πρωτοκόλλου.

Οι παραπάνω μετρικές χρησιμοποιούνται συστηματικά στην ανάλυση των πρωτοκόλλων στα Κεφάλαια 4 και 5, με αναφορά στις τιμές που προέκυψαν από τα σενάρια Fast, Baseline, Skew και Slow. Η βασική ανταλλαγή ανάμεσα σε ρυθμαπόδοση και καθυστέρηση, καθώς και τα όρια κλιμάκωσης ανοιχτών blockchains, έχουν αναλυθεί από τους **Croman κ.ά. (2016)**, υπογραμμίζοντας τον ρόλο των παραμέτρων δικτύου και σχεδιαστικών επιλογών στα πρακτικά όρια απόδοσης.

2.8 Σύνδεση θεωρίας με τον προσομοιωτή

Η παράμετρος simulation.simTime καθορίζει τον ορίζοντα του πειράματος στον προσομοιωμένο χρόνο με ανώτατη διάρκεια προσομοίωσης (simulation.simTime). Οι δικτυακές καθυστερήσεις—όπως queueing_delay, processing_delay, use_latency και diff_city_latency_ms διαμορφώνουν το προφίλ επικοινωνίας και επηρεάζουν άμεσα την παρατηρούμενη συμπεριφορά των πρωτοκόλλων. Μεγαλύτερες καθυστερήσεις αυξάνουν την πιθανότητα forks στο PoW και οδηγούν σε περισσότερες επαναπροσπάθειες (retries) στο PoS.

Η συνδεσιμότητα περιγράφεται από το num_neighbours, δηλαδή τον αριθμό σταθερών γειτόνων ανά κόμβο. Το gossip είναι ενεργό στο μοντέλο μετάδοσης και δεν χρησιμοποιείται ως διακριτός παράγοντας μεταξύ σεναρίων. Κατά συνέπεια, η ταχύτητα διασποράς blocks και μηνυμάτων εξαρτάται κυρίως από τον βαθμό του γράφου (num_neighbours), τα μονοπάτια καθυστέρησης και το διαθέσιμο bandwidth/ουρές. Μεγαλύτερος βαθμός τείνει να μειώνει τους χρόνους διάδοσης και τον κίνδυνο αποκλίσεων ή forks, όμως παράγει

περισσότερη κίνηση (περισσότερες αποστολές ανά μήνυμα) που μπορεί να αυξήσει την ουρά/καθυστερήση στο δίκτυο όταν το bandwidth είναι περιορισμένο. Αντίθετα, αραιότερη συνδεσιμότητα επιβραδύνει τη διάδοση και μπορεί να αυξήσει τη διασπορά του block time ή να καθυστερήσει τη σύγκλιση.

Στο PoW ο ρυθμός εξόρυξης καθορίζεται από την παράμετρο `expected_block_time` σε συνδυασμό με τα βάρη `hashpower` ανά κόμβο. Στα logs, το `mining_interval_avg` συνοψίζει τον μέσο χρόνο μεταξύ επιτυχών τοπικών εξορύξεων ανά κόμβο. Κατά την εκτέλεση καταγράφονται συστηματικά γεγονότα και μετρικές της αλυσίδας, ενδεικτικά `fork_detected`, `reorganization`, `orphan_rate`, `propagation_delay_avg`, `mining_interval_avg`, καθώς και ανά κόμβο `Throughput`, `Blocktime` και `Latency`, ώστε να είναι δυνατή η πλήρης αποτίμηση σταθερότητας και απόδοσης. Σημείωση: στο κείμενο ο δείκτης Block time (s) αντιστοιχεί στο πεδίο “Blocktime” του log.

Η σύγκριση ανάμεσα στη θεωρία και την προσομοίωση ακολουθεί ξεκάθαρη λογική. Τα “THEORETICAL PoW OUTPUTS” προκύπτουν απευθείας από την αναλογία `hashpower` μεταξύ κόμβων και αποτυπώνουν το αναμενόμενο μερίδιο παραγωγής blocks. Αντιθέτως, τα “Simulated” αποτελέσματα καταγράφουν την πραγματική δυναμική του συστήματος, επιβεβαιώνοντας ότι οι αφίξεις των blocks ακολουθούν Poisson διαδικασία—δηλαδή τα χρονικά διαστήματα μεταξύ διαδοχικών blocks είναι εκθετικά (με μέση τιμή ίση με τη διακύμανση και ιδιότητα `memoryless`), γεγονός που παράγει φυσικές «συστάδες» και «κενά»—και αναδεικνύοντας την επίδραση των δικτυακών καθυστερήσεων στην εμφάνιση forks, orphans και στις αποκλίσεις του πραγματικού block time από τη θεωρητική τιμή.

Στο PoW δεν εφαρμόζεται κανόνας τελικοποίησης με k επιβεβαιώσεις· αντί γι’ αυτό καταγράφεται `latency` συναλλαγών (από `tx.timestamp` έως `block.time_added`), μαζί με `blocktime/throughput`. Στο PoS, ένα block θεωρείται τελικό όταν ικανοποιηθεί το `confirmation_threshold` (stake-weighted). Η `finality latency` που καταγράφεται είναι ο χρόνος από τη δημιουργία του block έως τη στιγμή που τελικοποιείται (`now - time_created`) και αναφέρεται ανά κόμβο ως `FinalityLatency`.

Στο PoS η επιλογή προτείνοντος (leader) υλοποιείται μέσω του μηχανισμού `proposer_selection`, όπου οι πιθανότητες είναι σταθμισμένες από τα μερίδια συμμετοχής (stake). Με άλλα λόγια, όσο μεγαλύτερο το stake ενός επικυρωτή, τόσο υψηλότερη η πιθανότητα να προτείνει το επόμενο block, διατηρώντας αναλογία κοντά στη θεωρητική κατανομή βαρών.

Αξιολόγηση δικαιοσύνης. Η δικαιοσύνη της επιλογής ηγετών εκτιμάται συγκρίνοντας την καμπύλη «Stake vs Actual Leader Share» (θεωρητική πιθανότητα έναντι πραγματικής συχνότητας ηγεσίας). Η διαφορά ποσοτικοποιείται με την απόκλιση KL (D_{KL}), ενώ ο βαθμός αποκέντρωσης αποτυπώνεται με δείκτες `decentralisation` (π.χ. κατανομή ηγετών ανά κόμβο).

Ανθεκτικότητα. Η συμπεριφορά του συστήματος είναι ευαίσθητη σε skew της κατανομής stake (που μπορεί να επηρεάσει τη δικαιοσύνη), καθώς και σε παραμέτρους δικτύου όπως timeouts, οι οποίες μπορούν να αυξήσουν τις καθυστερήσεις, τις επαναπροσπάθειες και τη διασπορά των επιδόσεων.

Ο χρονισμός του γύρου ελέγχεται από παραμέτρους όπως το round_timeout (το ανώτατο όριο διάρκειας ενός γύρου), το commit_delay (μικρή καθυστέρηση πριν το τελικό commit ώστε να συλλεγούν επιπλέον ψήφοι και το msg_val_delay (τεχνητή καθυστέρηση /μοντελοποίηση δικτυακής αδράνειας για μηνύματα επικύρωσης). Ο σωστός συνδυασμός τους σταθεροποιεί τη διαδικασία ψηφοφορίας και μειώνει τα περιθώρια παρεξηγήσεων λόγω δικτυακών διακυμάνσεων.

Η αξιοπιστία και η επιχειρησιακή υγεία του πρωτοκόλλου αποτιμώνται μέσω δεικτών όπως ο αριθμός γύρων με timeout, οι στατιστικές της τελικοποίησης (avg/median/95p latency) και το μέσο μέγεθος του pool συναλλαγών, που αποτυπώνει τον βαθμό συμφόρησης του δικτύου. Μικρές τιμές latency και περιορισμένο pool υποδηλώνουν ομαλή και προβλέψιμη λειτουργία.

Τέλος, τα trade-offs είναι σαφή: ένα μικρό round_timeout επιταχύνει τους γύρους και μπορεί να μειώσει τη διάμεσο καθυστέρηση τελικοποίησης, αλλά αυξάνει τον κίνδυνο timeouts και επιπλέον retries όταν οι καθυστερήσεις δικτύου μεγαλώνουν. Αντιθέτως, μεγαλύτερα timeouts μειώνουν τις αστοχίες quorum και εξομαλύνουν τις «ουρές» στο 95ο εκατοστημόριο της latency, με τίμημα υψηλότερους μέσους χρόνους και πιθανή διόγκωση του pool. Η ρύθμιση commit_delay βοηθά να βρεθεί η ισορροπία ανάμεσα στη γρήγορη σύγκλιση και στη σταθερή, ανθεκτική τελικοποίηση.

2.8 Πλαίσιο χαρακτηρισμού που θα χρησιμοποιηθεί στα αποτελέσματα

Για κάθε πρωτόκολλο χαρακτηρίζονται τέσσερις λειτουργικές ρυθμίσεις Fast Baseline Skew Slow πάνω σε κοινή τοπολογία οκτώ κόμβων. Το Fast υιοθετεί επιθετικά διαστήματα και μικρά timeouts επιδιώκοντας υψηλό throughput με ταυτόχρονο προσεκτικό έλεγχο για forks ή timeouts. Το Baseline λειτουργεί ως ρεαλιστικό σημείο αναφοράς. Το Skew εισάγει εσκεμμένη ανισορροπία πόρου ή μεριδίου hashpower stake ηγεσιών ώστε να αποτυπωθούν η δικαιοσύνη και ο βαθμός αποκέντρωσης. Το Slow χρησιμοποιεί μεγάλα διαστήματα και συντηρητικά thresholds δίνοντας έμφαση στη σταθερότητα και στην παρατηρησιμότητα των επιδράσεων. Οι ρυθμίσεις συνδέονται ρητά με παραμέτρους από τον προσομοιωτή όπως expected_block_time, block_interval, round_timeout, confirmation_threshold και num_neighbours.

Ο χαρακτηρισμός γίνεται ως προς τον ρυθμό διεκπεραίωσης Throughput τον μέσο χρόνο παραγωγής ή ενσωμάτωσης Blocktime και την καθυστέρηση τελικοποίησης Finality latency. Παράλληλα αποτυπώνονται δείκτες δικαιοσύνης και αποκέντρωσης όπως η απόκλιση KL

D_KL και η κατανομή ηγετών καθώς και ιδιότητες που είναι ειδικές ανά πρωτόκολλο. Για το PoW καταγράφονται forks reorgs orphan rate και propagation delay. Για το PoS αποτυπώνονται leader skew και forks σε συνάρτηση με confirmation_threshold και block_interval. Επιπλέον εξετάζεται η ευαισθησία στις δικτυακές καθυστερήσεις και στον βαθμό συνδεσιμότητας num_neighbours.

ΚΕΦΑΛΑΙΟ 3 Μεθοδολογία: προσομοιωτής, ρυθμίσεις, σενάρια και δείκτες

3.1 Περιγραφή προσομοιωτή

Χρησιμοποιούμε προσομοιωτή διακριτών γεγονότων (discrete-event) σε Python για τη μελέτη πρωτοκόλλων συναίνεσης. Ένας χρονοδρομολογητής διατηρεί ουρά γεγονότων (αποστολές/λήψεις μηνυμάτων, παραγωγές blocks, timeouts) και προχωρά τον προσομοιωμένο χρόνο με βάση τις καθυστερήσεις δικτύου και εκτέλεσης. Η υλοποίηση ακολουθεί τη λογική και το μοντέλο του SymBChainSim, όπως παρουσιάζονται και επεκτείνονται στις σχετικές εργασίες των Diamantopoulos κ.ά. (2022a και 2022b).

Το δίκτυο μοντελοποιείται ως peer-to-peer overlay με $N=8$ κόμβους κατανεμημένους γεωγραφικά σε πόλεις όπως Johannesburg, Geneva, Cromwell, Tirana, Heredia, Barcelona, Montreal και Warsaw. Η επιλογή κοινής τοπολογίας 8 κόμβων διατηρεί σταθερές τις συνθήκες σε όλα τα σενάρια, εξασφαλίζοντας ότι τα αποτελέσματα των μετρήσεων είναι άμεσα συγκρίσιμα. Για κάθε μήνυμα εφαρμόζεται ενιαίο μοντέλο καθυστέρησης και χωρητικότητας. Η καθυστέρηση μετάδοσης αντλείται από πίνακα μετρημένων τιμών (use_latency: random), ώστε κάθε ζεύγος κόμβων να έχει διαφορετική καθυστέρηση. Οι τιμές αυτές ορίζονται κατά την αρχικοποίηση και παραμένουν σταθερές ανά ζεύγος κόμβων σε όλη τη διάρκεια της εκτέλεσης (δεν ανανεώνονται ή προσαρμόζονται ανά γύρο/πρωτόκολλο). Οι τοπικές (ίδιας πόλης) συνδέσεις αρχικοποιούνται με same_city_latency_ms ≈ 1000 ms (με τυχαία απόκλιση ± 50 ms), προσομοιώνοντας βαρύ περιβάλλον, ενώ οι υπόλοιπες τιμές καθυστέρησης προέρχονται από τις καταχωρήσεις του χάρτη. Επιπλέον, προστίθενται καθυστέρηση ουράς και καθυστέρηση επεξεργασίας — από 0.1 s η καθεμιά — ώστε να αποτυπώνεται ρεαλιστικά η συμπεριφορά των κόμβων υπό φόρτο. Το εύρος ζώνης ορίζεται στα 10 MB/s με στοχαστική διακύμανση ± 2 MB/s και κατώτατο όριο το 1 MB/s, ώστε να αποφεύγονται μη ρεαλιστικά υψηλοί ρυθμοί.

Η τοπολογία αρχικοποιείται με σταθερό αριθμό γειτόνων ανά σενάριο (num_neighbours) και με ενεργοποιημένο gossip σε όλες τις διαμορφώσεις. Έτσι ελέγχεται συστηματικά τόσο η συνδεσιμότητα όσο και η ένταση διάδοσης, που με τη σειρά τους καθορίζουν την πιθανότητα forks (στο PoW) και τις επαναπροσπάθειες (σε PoS).

Η ανάλυση των μετρήσεων και η παραγωγή των γραφημάτων πραγματοποιήθηκε με χρήση Python, αξιοποιώντας τις βιβλιοθήκες NumPy και Matplotlib για τον υπολογισμό και την οπτικοποίηση των αποτελεσμάτων.

3.2 Κοινές ρυθμίσεις (base)

Στην ενότητα αυτή παρατίθενται οι παράμετροι που χρησιμοποιεί ο προσομοιωτής, ακριβώς όπως ορίζονται στα αρχεία ρυθμίσεων (YAML). Κάθε πίνακας οργανώνεται σε τέσσερις στήλες σεναρίων (Fast, Baseline, Skew, Slow).

Γενικές ρυθμίσεις (δικτύου/εφαρμογής/προσομοίωσης).

Οι παράμετροι του `base.yaml` καθορίζουν το κοινό περιβάλλον μέσα στο οποίο εκτελούνται όλα τα πρωτόκολλα. Η μεταβλητή `Nn` ορίζει τον αριθμό κόμβων του δικτύου, ενώ τα `TI_dur` και `Tn` καθορίζουν κάθε πότε και πόσες συναλλαγές δημιουργεί ο κόμβος παραγωγός. Οι παράμετροι `Tsize` και `base_transaction_size` προσθέτουν σταθερό μέγεθος σε κάθε συναλλαγή, ώστε να προσομοιώνεται το κόστος μετάδοσης. Αντίστοιχα, το `Bsize` καθορίζει το μέγιστο φορτίο συναλλαγών που μπορεί να μεταφέρει ένα block και το `base_block_size` προσθέτει ένα σταθερό overhead στο μέγεθος του block για τις επικοινωνίες.

Σε επίπεδο δικτύου το `gossip` είναι μόνιμα ενεργό και η διάδοση γίνεται προς τους σταθερούς γείτονες κάθε κόμβου. Το πλήθος των γειτόνων καθορίζεται από το `num_neighbours`: μεγαλύτερες τιμές επιταχύνουν τη διάδοση αλλά αυξάνουν και την κίνηση στο δίκτυο, με πιθανό κόστος συμφόρησης. Οι παράμετροι `use_latency` και τα σχετικά όρια καθυστέρησης (`same_city_latency_ms`, `diff_city_latency_ms` με τις διασπορές τους) καθορίζουν το στοχαστικό μοντέλο μεταφοράς, δηλαδή πόσο χρόνο κάνουν τα μηνύματα να φτάσουν. Οι τιμές `processing_delay` και `queueing_delay` προσθέτουν καθυστερήσεις στο τοπικό επίπεδο, ώστε να προσομοιώνεται η φόρτωση κάθε κόμβου. Το μοτίβο αυτό αντιστοιχεί στο κλασικό epidemic/gossip πρωτόκολλο των Demers κ.ά. (1987), όπου οι κόμβοι διαδίδουν περιοδικά νέα αντικείμενα σε τυχαίους γείτονες μέχρι να επιτευχθεί ευρεία κάλυψη του δικτύου.

Τέλος, οι παράμετροι της ενότητας `simulation` (όπως το `simTime`) ορίζουν τον ορίζοντα της προσομοίωσης και πότε αυτή θα τερματίσει, ενώ το `init_CP` δηλώνει με ποιο πρωτόκολλο ξεκινάει η εκτέλεση.

Παράμετρος (base.yaml)	Τιμή (Default)
<code>application.CP (init)</code>	ForkingPow
<code>simulation.init_CP</code>	ForkingPow
<code>application.Nn</code>	8
<code>application.TI_dur</code>	1.0
<code>application.Tn</code>	5
<code>application.Tsize</code>	0.002
<code>application.base_transaction_size</code>	512
<code>network.gossip</code>	True
<code>network.num_neighbours</code>	3

network.use_latency	random
network.bandwidth.mean/dev/min	10/2/1
network.processing_delay	0.1
network.queueing_delay	0.1
network.same_city_latency_ms/dev	1000/50
network.diff_city_latency_ms/dev	50/50
simulation.simTime	3000
simulation.stop_after_blocks	-1

Σχήμα 1 - Γενικές παράμετροι (base.yaml)

3.3 Επιλογή Προτείνοντος και Ροή Γύρου

Σε αυτή την ενότητα περιγράφεται πώς επιλέγεται ο προτείνων κόμβος ανά γύρο μέσω του μηχανισμού proposer_selection, καθώς και πώς εξελίσσεται η ροή του γύρου με βάση το block_interval και το timeout.

Στο Proof-of-Work (PoW) ακολουθείται ο κανόνας longest/heaviest-chain. Ο προτείνων block αναδεικνύεται με πιθανότητα ανάλογη του mining_power και όταν ο συγχρονισμός ή η διάδοση καθυστερούν αυξάνει η πιθανότητα forks και reorgs. Για λόγους απλοποίησης δεν εφαρμόζεται retargeting δυσκολίας. Η παραγωγή blocks προσομοιώνεται από το expected_block_time μαζί με τα βάρη mining_power.

Στο Proof-of-Stake (PoS) ο proposer επιλέγεται βάσει stake (stake-weighted sampling) και η εκτέλεση είναι slot-/round-driven. Το block_interval ορίζει τον ρυθμό των slots και το timeout τη μέγιστη διάρκειά τους. Η τελικοποίηση (finality) είναι threshold-based: με confirmation_threshold ένα block γίνεται Finalized όταν το συσσωρευμένο βάρος CONFIRM ψήφων υπερβεί το confirmation_threshold × total_stake. Στα πειράματα η καθυστέρηση τελικοποίησης αποτυπώνεται ως finality latency (χρόνος από την πρόταση έως την επίτευξη του threshold).

3.4 Σενάρια αξιολόγησης (Fast, Baseline, Slow, Skew)

Κάθε πρωτόκολλο αξιολογείται σε τέσσερις οικογένειες ρυθμίσεων — Fast, Baseline, Slow και Skew — ώστε να παρατηρηθεί η συμπεριφορά του τόσο σε «επιθετικές» όσο και σε «συντηρητικές» διαμορφώσεις, αλλά και υπό συνθήκες έντονα ανισοκατανεμημένων πόρων. Η λογική είναι να καταγραφεί πώς αντιδρά το σύστημα όταν πιέζεται στα όριά του, πώς λειτουργεί σε τυπικές ρυθμίσεις, πώς σταθεροποιείται σε ήρεμες συνθήκες και πώς επηρεάζεται από συγκέντρωση ισχύος. Στη συνέχεια παρουσιάζονται συνοπτικά οι ρυθμίσεις που ενεργοποιούνται σε κάθε πρωτόκολλο, μαζί με τις τιμές ανά σενάριο στους αντίστοιχους πίνακες.

Στο PoW το `min_block_interval` λειτουργεί ως κατώφλι και αποτρέπει μη ρεαλιστικά πολύ μικρά διαστήματα μεταξύ δύο blocks. Το `Bsize` ορίζει το μέγιστο φορτίο συναλλαγών που χωρά σε κάθε block και το `base_block_size` προσθέτει σταθερό overhead στο μέγεθος του block για σκοπούς μηνυμάτων χωρίς να επηρεάζει τον χρόνο διάδοσης. Αν σε YAML σεναρίου PoW δηλωθούν `application.TI_dur` ή `application.Tn`, ή δοθεί `num_neighbours` ως override, τότε αυτές οι τιμές υπερισχύουν των γενικών μόνο για το συγκεκριμένο τρέξιμο. Οι συγκεκριμένες τιμές για Fast, Baseline, Skew και Slow φαίνονται στο επόμενο σχήμα.

Παράμετρος	Fast	Baseline	Skew	Slow
<code>expected_block_time</code>	2.0	30.0	30.0	100.0
<code>min_block_interval</code>	0.36	3.0	3.0	60.0
<code>mining_power</code>	0=5.0, 1=3.0, 2=2.0, 3=1.5, 4=1.0, 5=0.8, 6=0.4, 7=0.2	0=5.0, 1=3.0, 2=2.0, 3=1.5, 4=1.0, 5=1.2, 6=0.5, 7=1.3	0=8.0, 1=1.0, 2=0.5, 3=0.5, 4=0.3, 5=0.3, 6=0.2, 7=0.2	0=5.0, 1=3.0, 2=2.0, 3=1.5, 4=1.0, 5=0.8, 6=0.4, 7=0.2
<code>Bsize</code>	1.0	1.0	1.0	1.0
<code>base_block_size</code>	1.0	1.0	1.0	1.0
<code>num_neighbours</code>	3	6	6	6
<code>application.TI_dur</code>	1	1	1	3
<code>application.Tn</code>	5	5	5	1
<code>simulation.simTime</code>	3600	3600	3600	3600
<code>simulation.stop_after_time</code>	3600	3600	3600	3600

Σχήμα 2 - Παράμετροι PoW ανά σενάριο

Στο PoS τα δικαιώματα πρότασης blocks κατανέμονται βάσει stakes. Το `block_interval` καθορίζει κάθε πότε ξεκινά νέος γύρος και το `timeout` είναι η προθεσμία του γύρου· αν δεν συγκεντρωθεί έγκαιρα η απαιτούμενη υποστήριξη, ο γύρος κλείνει και ξεκινά ο επόμενος. Η τελικοποίηση καθορίζεται από το `confirmation_threshold`: ένα block γίνεται Finalized όταν το συσσωρευμένο βάρος των CONFIRM ψήφων υπερβεί το `confirmation_threshold × total_stake`. Το `proposer_selection` ορίζει τη μέθοδο επιλογής προτείνοντα, συνήθως βάσει stake. Οι ρυθμίσεις της ενότητας `simulation` (όπως `simTime`) καθορίζουν τον ορίζοντα εκτέλεσης. Αν στο YAML του PoS υπάρχουν `network.gossip` ή `network.num_neighbours`, τότε ισχύουν μόνο για το συγκεκριμένο σενάριο. Ο πίνακας που ακολουθεί συγκεντρώνει τις ρυθμίσεις ανά σενάριο.

Παράμετρος	Fast	Baseline	Skew	Slow
block_interval	3	10	10	40
slot_len	3	10	10	40
timeout	6	12	12	1.5
confirmation_threshold	0.67	0.67	0.67	0.67
stakes	node_0=50, node_1=20, node_2=15, node_3=15, node_4=25, node_5=25, node_6=25, node_7=25	node_0=50, node_1=20, node_2=15, node_3=15, node_4=25, node_5=25, node_6=25, node_7=25	node_0=120, node_1=30, node_2=15, node_3=10, node_4=8, node_5=7, node_6=5, node_7=5	node_0=50, node_1=20, node_2=15, node_3=15, node_4=25, node_5=25, node_6=25, node_7=25
network.num_neighbours	7	7	7	7
simulation.simTime	3600	3600	3600	3600
execution.proposer_selection	stake	stake	stake	stake

Σχήμα 3 - Παράμετροι PoS ανά σενάριο

Πλήρη αρχεία ρυθμίσεων: src/Configs/base.yaml; PoW: src/Configs/Pow_config.yaml (fast), Pow_config_baseline.yaml, Pow_config_skew.yaml, Pow_config_slow.yaml; PoS: src/Configs/Pos_config.yaml (fast), Pos_config_baseline.yaml, Pos_config_skew.yaml, Pos_config_slow.yaml

3.5 Διαδικασία εκτέλεσης

Για κάθε σενάριο εκτελείται μία κύρια προσομοίωση με σταθερό PRNG seed, ώστε τα αποτελέσματα να είναι πλήρως αναπαραγώγιμα. Οι παράμετροι του δικτύου (latencies, bandwidth) και οι λίστες γειτόνων (num_neighbours) φορτώνονται από τα αρχεία ρυθμίσεων και παραμένουν σταθερές καθ' όλη τη διάρκεια της εκτέλεσης.

Στο PoW η εκτέλεση είναι time-driven με ορίζοντα simTime (π.χ. 3600 s). Ο ρυθμός εξόρυξης προκύπτει από το expected_block_time και τα βάρη hashpower, χωρίς μηχανισμό dynamic difficulty.

Στο PoS ακολουθείται slot driven λογική: ο τερματισμός της προσομοίωσης γίνεται μέσω simTime, ενώ ο χρονισμός καθορίζεται από block_interval/slot_len και τα timeouts.

Τα logs του προσομοιωτή καταγράφουν συστηματικά counters και μέσους όρους — π.χ. `fork_detected`, `reorganization`, `TOTAL ROUNDS SCHEDULED` — καθώς και μετρικές ανά κόμβο όπως `Throughput`, `Blocktime`, `mining_interval_avg`. Αυτό επιτρέπει συγκριτική ανάλυση της προόδου, της σταθερότητας και της απόδοσης.

3.6 Εξαγωγή και σύνοψη δεδομένων

Τα αποτελέσματα εξάγονται απευθείας από τα logs του προσομοιωτή χωρίς ενδιάμεση “διόρθωση” ή εξομάλυνση. Τα scripts (Python/NumPy/Matplotlib) διαβάζουν τα αρχεία καταγραφής, χαρτογραφούν τα πεδία και υπολογίζουν συνοπτικές στατιστικές ανά κόμβο. Όπου παρατίθενται αποσπάσματα logs, διατηρείται η ακριβής μορφή των ετικετών, `Blocktime`, `mining_interval_avg`, ενώ στο κείμενο χρησιμοποιούμε την ενοποιημένη ορολογία `Block time (s)`, `Finality latency (s)`, `Throughput (blocks/s)`.

Ο `Throughput (blocks/s)` προκύπτει από τα επιβεβαιωμένα `blocks` ανά μονάδα προσομοιωμένου χρόνου ενώ `Throughput (tx/s)` χρησιμοποιείται μόνο όταν γίνεται ρητή μετατροπή (`blocks/s` × μέσες συναλλαγές ανά `block`). Ο `Block time (s)` είναι ο μέσος χρόνος παραγωγής/ενσωμάτωσης `block` (στο log: `Blocktime`).

Για το PoW αξιοποιείται επιπλέον το `mining_interval_avg` (μέσος χρόνος μεταξύ τοπικών επιτυχών εξορύξεων ανά κόμβο), καθώς και μετρητές όπως `fork_detected`, `reorganization`, `orphan_rate`, `propagation_delay_avg`. Για το PoS, η `finality latency (s)` υπολογίζεται ως ο χρόνος μέχρι την οριστικοποίηση, σύμφωνα με τον εκάστοτε κανόνα (π.χ. `confirmation_threshold`).

Σημειώσεις συνέπειας μονάδων & συμβολισμών. Σε όλο το κείμενο και τις λεζάντες εφαρμόζονται οι μονάδες `s`, `blocks/s` και `tx/s`. Οι φηγούρες/πίνακες προκύπτουν αποκλειστικά από τα εξαγόμενα των logs με σταθερό PRNG seed, ώστε η διαδικασία να είναι αναπαραγώγιμη.

3.7 Περιορισμοί & απειλές εγκυρότητας

Η μελέτη εκτελείται σε κλίμακα οκτώ κόμβων, αρκετή για ελεγχόμενη ανάλυση αλλά όχι αντιπροσωπευτική άμεσης κλιμάκωσης σε εκατοντάδες ή χιλιάδες συμμετέχοντες. Το δίκτυο είναι στατικό: οι καθυστερήσεις (`use_latency`, `same_city_latency_ms`) και το εύρος ζώνης ορίζονται κατά την αρχικοποίηση και παραμένουν σταθερά για όλη τη διάρκεια της προσομοίωσης. Δεν περιλαμβάνονται κακόβουλοι κόμβοι ή επιθέσεις.

Τα πρωτόκολλα υλοποιούνται με σκόπιμες απλοποιήσεις: στο PoW δεν υπάρχει μηχανισμός δυναμικής προσαρμογής δυσκολίας (`difficulty retargeting`), ο ρυθμός εξόρυξης

καθορίζεται αποκλειστικά από το `expected_block_time` και τα βάρη `hashpower`. Οι βασικές τιμές `latency` των ζεύξεων παραμένουν σταθερές όπως ορίζονται στην αρχικοποίηση. Στο PoS χρησιμοποιείται απλουστευμένος μηχανισμός επιλογής ηγέτη, με πιθανότητες ανάλογες του `stake`, χωρίς εξωτερικό `randomness beacon` ή σύνθετο `committee reshuffling`. Το σύνολο επικυρωτών και τα μερίδια `stake` θεωρούνται σταθερά καθ' όλη τη διάρκεια των πειραμάτων (δεν μοντελοποιούνται `staking/unstaking`, `slashing` ή μεταβολές διανομής). Η τελικοποίηση στο παρόν PoS είναι `threshold-based`: ένα `block` γίνεται `Finalized` όταν το συσσωρευμένο βάρος `CONFIRM` ψήφων υπερβεί το `confirmation_threshold × total_stake`, ενώ οι παράμετροι `expected_block_time` (PoW), `block_interval/timeout` (PoS) διατηρούνται σταθερές, όπως και οι βασικές καθυστερήσεις ζεύξεων (`latency`).

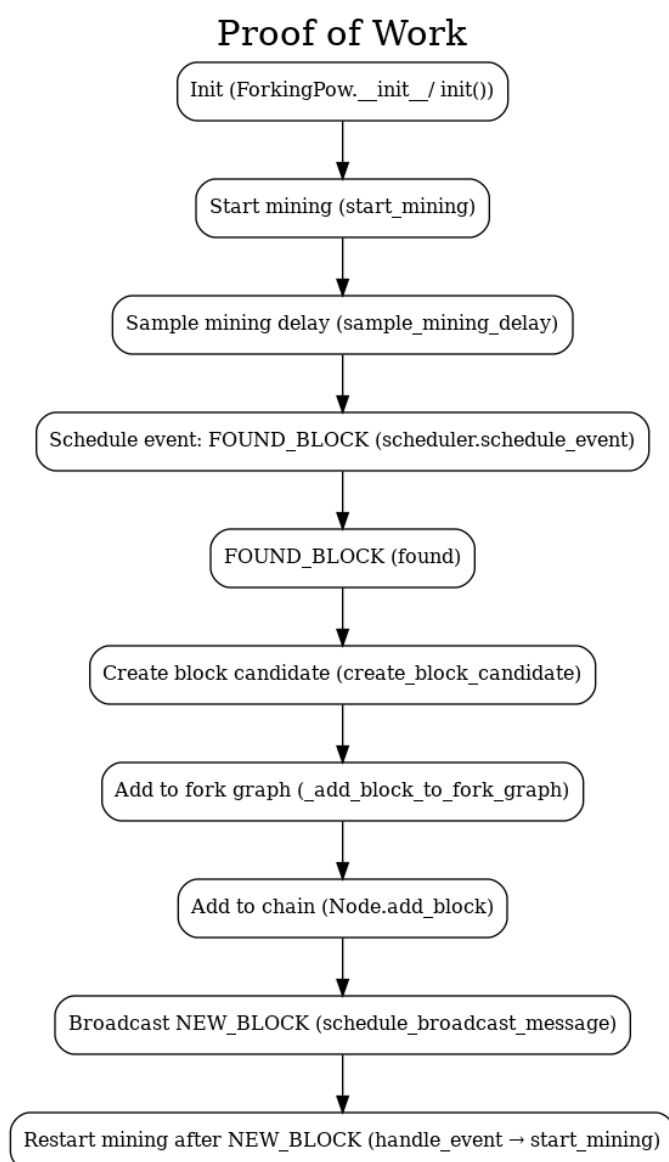
Τέλος, τα αποτελέσματα παρουσιάζουν ευαισθησία σε κρίσιμες παραμέτρους: στο PoS επηρεάζονται κυρίως από τα `timeouts` και το `confirmation_threshold`, ενώ στο PoW από τον βαθμό συνδεσιμότητας (`num_neighbours`), το `expected_block_time` και το προφίλ καθυστερήσεων (`latency`).

3.8 Σύνδεση με τα αποτελέσματα

Οι ορισμοί/ρυθμίσεις του παρόντος κεφαλαίου χρησιμοποιούνται στο Κεφ. 4 και 5 για την ανάλυση των τεσσάρων σεναρίων ανά πρωτόκολλο (`Fast`, `Baseline`, `Slow`, `Skew`), με κοινό τρόπο συγκριτικής απεικόνισης (ανά-κόμβο όπου έχει νόημα και συνολικά ανά πρωτόκολλο).

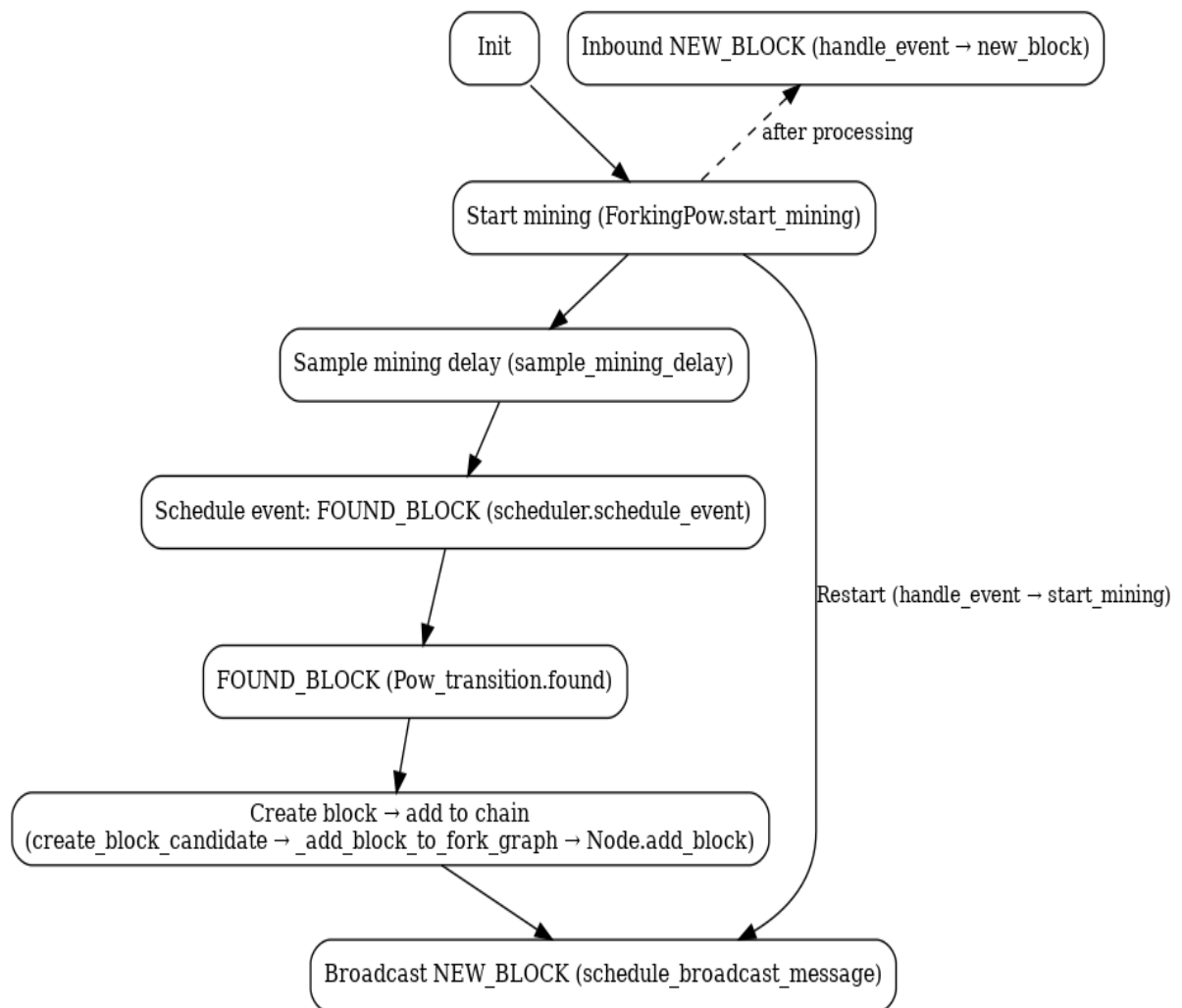
4.1 Μεθοδολογία και Διαγράμματα Ροής PoW

Στο κεφάλαιο αυτό παρουσιάζονται τα αποτελέσματα της προσομοίωσης του πρωτοκόλλου Proof-of-Work (PoW) με χρήση του SymBChainSim. Τα σενάρια που εξετάστηκαν είναι Fast, Baseline, Skew και Slow. Οι μετρικές που μετρήθηκαν περιλαμβάνουν chain length, forks, reorgs, orphan rate και mining intervals.



Σχήμα 4 - Proof-of-Work (επισκόπηση ροής).

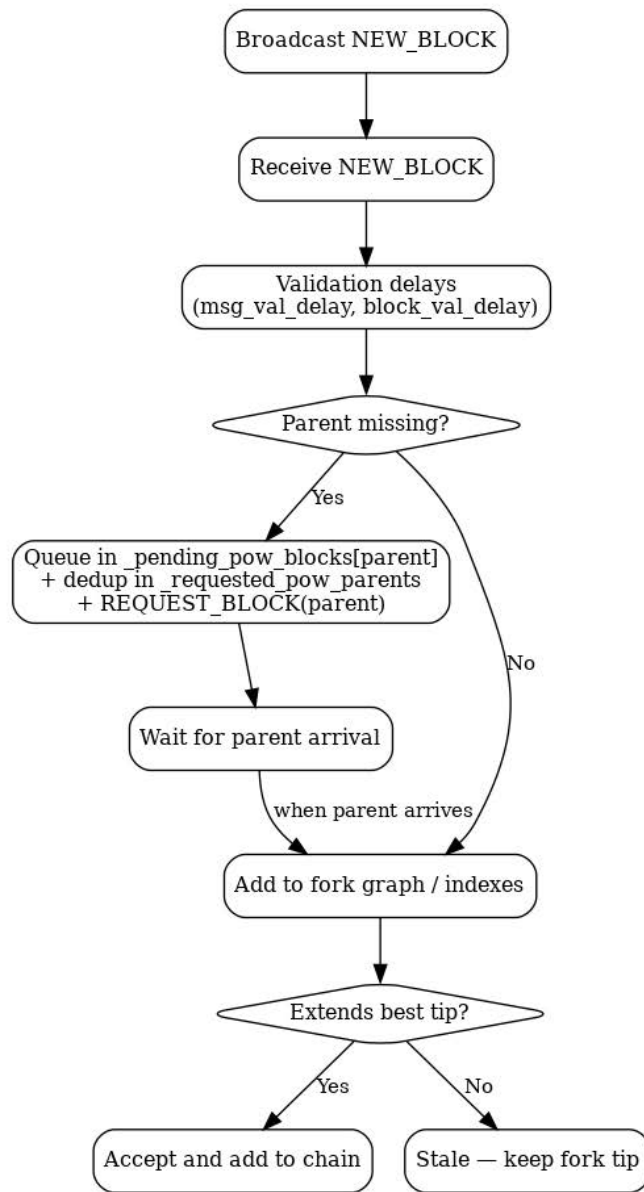
Το σχήμα 1 δείχνει τη ροή του PoW από την αρχικοποίηση έως την εύρεση/δημιουργία block, την καταχώρισή του στο fork graph και στην τοπική αλυσίδα (Node.add_block), τη μετάδοση του NEW_BLOCK και την επανεκκίνηση της εξόρυξης μετά από κάθε NEW_BLOCK. Υπογραμμίζει γιατί προκύπτουν forks: δύο ή περισσότεροι κόμβοι μπορούν να βρουν block σχεδόν ταυτόχρονα και να τα μεταδώσουν πριν διαδοθούν πλήρως τα προηγούμενα.



Σχήμα 5 - Βρόχος εξόρυξης (mining loop).

Στο PoW του Σχήματος 2, ο ρυθμός εξόρυξης δεν προκύπτει από difficulty/retargeting ούτε από ρητό ελάχιστο διάστημα προσπαθειών. Αντ' αυτού καθορίζεται παραμετρικά από το expected_block_time και κατανέμεται αναλογικά με τα βάρη hashpower ανά κόμβο (δειγματοληψία καθυστέρησης εξόρυξης → sample_mining_delay). Στα logs, το mining_interval_avg αποτυπώνει τον μέσο χρόνο μεταξύ επιτυχών τοπικών εξορύξεων ανά κόμβο. Κατά την εκτέλεση, καταγράφονται συστηματικά γεγονότα και μετρικές της αλυσίδας—ενδεικτικά: fork_detected, reorganization, orphan_rate, propagation_delay_avg, mining_interval_avg, καθώς και ανά κόμβο Throughput, Block time και Latency—

διευκολύνοντας την αποτίμηση σταθερότητας και απόδοσης. Σημ.: στα logs το “Block time” αντιστοιχεί στον δείκτη *Block time (s)* που χρησιμοποιείται στο κείμενο.

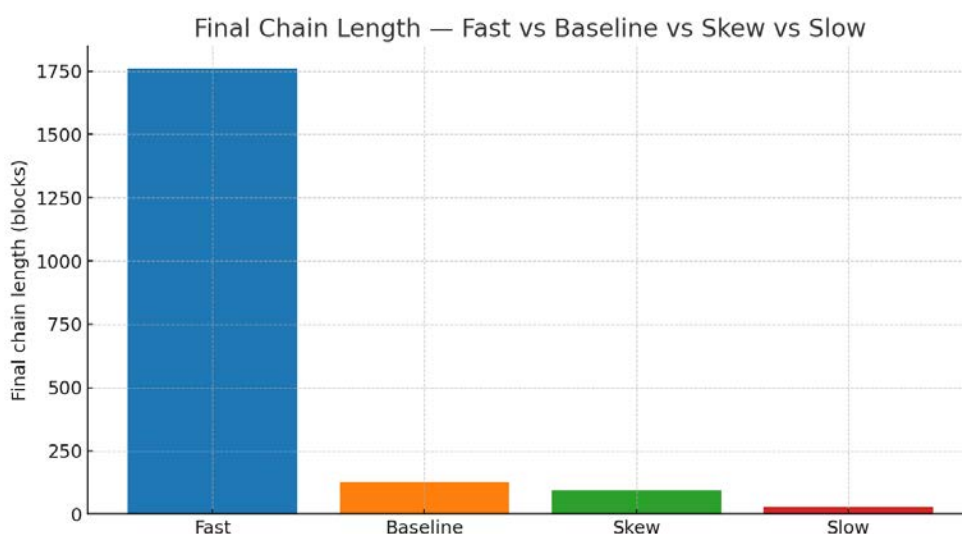


Σχήμα 6 - Διάδοση & ανάκτηση γονέα (propagation / parent recovery).

Στο σχήμα 3 δείχνει ότι με τη λήψη NEW_BLOCK ο κόμβος εφαρμόζει καθυστερήσεις επικύρωσης (msg_val_delay, block_val_delay), ελέγχει αν λείπει ο γονέας και αν ναι τοποθετεί το block στην _pending_pow_blocks[parent], κάνει dedup στο _requested_pow_parents και στέλνει REQUEST_BLOCK(parent) μέχρι να φτάσει ο γονέας, κατόπιν το block καταχωρίζεται στο fork graph και στα ευρετήρια, αν επεκτείνει το καλύτερο tip γίνεται αποδοχή και προσθήκη στην αλυσίδα, αλλιώς χαρακτηρίζεται stale και παραμένει tip του κλάδου, γεγονός που συνδέει άμεσα το propagation delay με το orphan rate και τη συχνότητα reorgs.

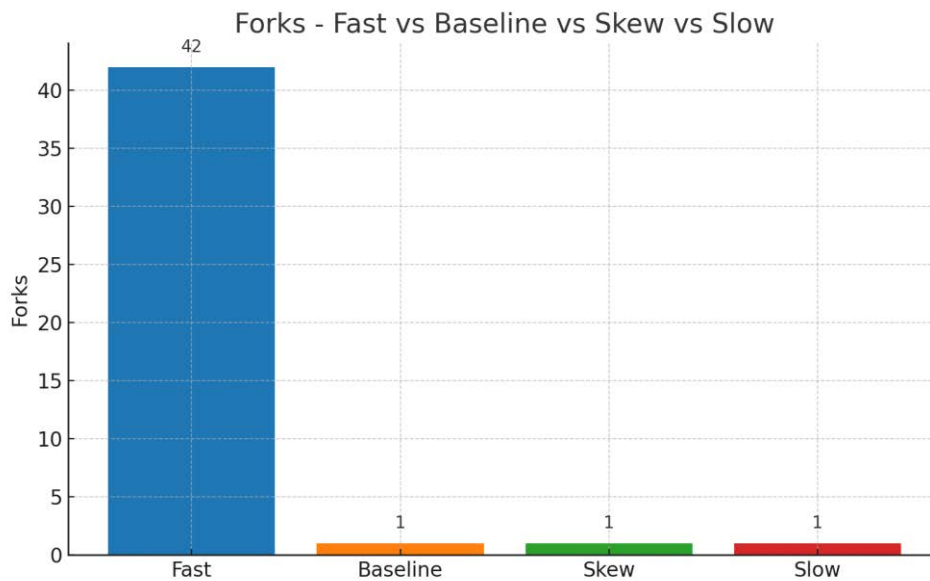
4.2 Συγκριτική σύνοψη σεναρίων (Fast / Baseline / Skew / Slow) & Πίνακας Μετρικών

Στο σύνολο των μετρικών τα τέσσερα σενάρια δείχνουν καθαρά το κλασικό trade-off ταχύτητας–σταθερότητας. Το Fast μεγιστοποιεί την παραγωγή blocks αλλά «πληρώνει» με έντονα forks/reorgs, το Slow είναι σχεδόν άψογο σε σταθερότητα αλλά ιδιαίτερα αργό, το Baseline ισορροπεί ρυθμό και αξιοπιστία και το Skew αποκαλύπτει τα όρια δικαιοσύνης όταν η ισχύς συγκεντρώνεται.



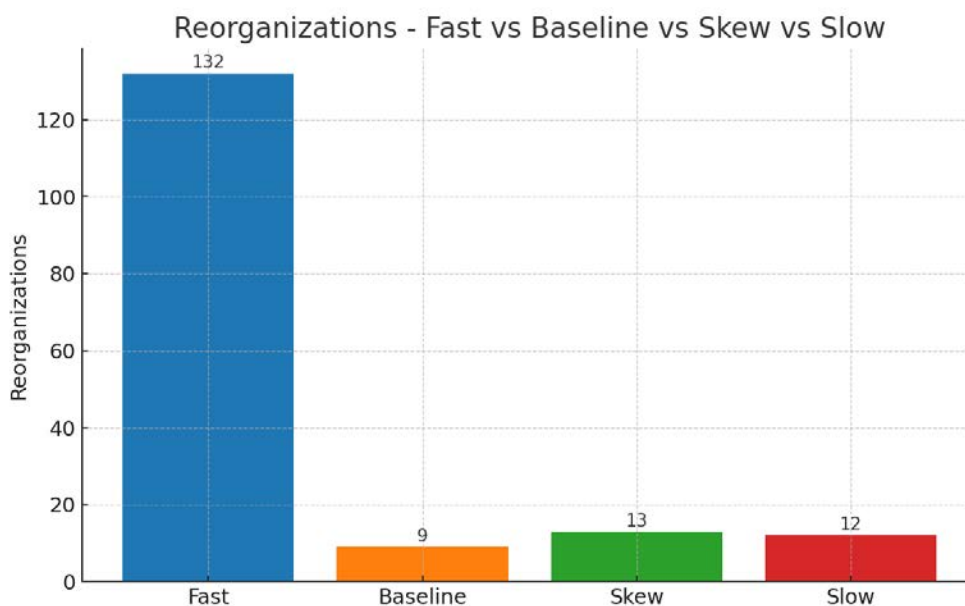
Σχήμα 7 - Τελικό μήκος αλυσίδας (Final Chain Length)

Στο σχήμα 7 δείχνει ότι το τελικό μήκος της αλυσίδας ανά σενάριο είναι Fast 1772 blocks Baseline 134 blocks Skew 139 blocks Slow 34 blocks, η μεγάλη τιμή στο Fast οφείλεται στον στόχο παραγωγής 2 δευτερόλεπτα και στο υψηλό network rate που επιτρέπουν πολύ περισσότερα blocks να φτάσουν και να παραμείνουν στη longest chain παρά τις αυξημένες συγκρούσεις, στο Baseline και στο Skew με στόχο 30 δευτερόλεπτα η παραγωγή είναι χαμηλότερη και οι τελικές αλυσίδες είναι συγκρίσιμες με μικρή απόκλιση 134 έναντι 139 επειδή η άνιση κατανομή hashpower στο Skew δίνει σε λίγους κόμβους περισσότερες επιτυχίες και περιορίζει τις απώλειες από ανταγωνιστικά blocks, στο Slow με στόχο 100 δευτερόλεπτα ο συνολικός ρυθμός παραγωγής είναι πολύ χαμηλός και η longest chain μένει σύντομη με 34 blocks, συνολικά το τελικό μήκος κλιμακώνεται κυρίως με τον στόχο block time και τη διάρκεια εκτέλεσης ενώ δευτερεύοντα ρόλο παίζουν οι ορφανήσεις και οι αναδιοργανώσεις που αφαιρούν blocks από την τελική ακολουθία.



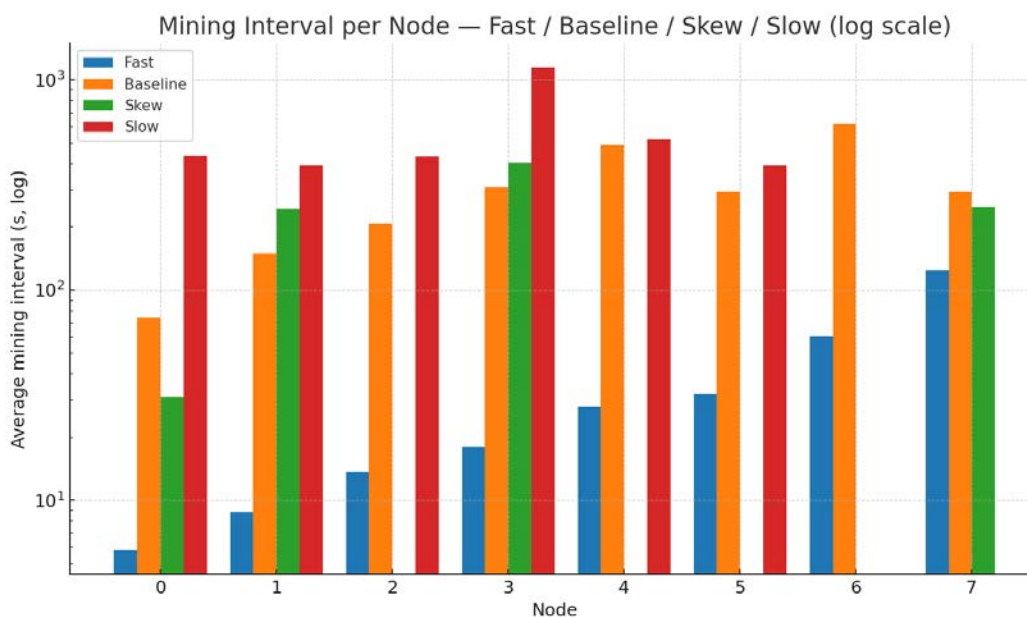
Σχήμα 8 - Forks ανά σενάριο

Όπως δείχνει το σχήμα 5, τα αποτελέσματα κάθε σεναρίου PoW ευθυγραμμίζονται με τη φυσική δυναμική του δικτύου: στο Fast το μέσο διάστημα μεταξύ blocks είναι μικρό σε σχέση με τον χρόνο διάδοσης και επεξεργασίας, με αποτέλεσμα πολλά έγκυρα blocks να κυκλοφορούν σχεδόν ταυτόχρονα και να ανταγωνίζονται μεταξύ τους, κάτι που αυξάνει αισθητά τα forks και τα ορφανά. Στα Baseline και Skew το διάστημα μεταξύ blocks είναι αρκετά μεγαλύτερο, οπότε τα blocks προλαβαίνουν να διαδοθούν πριν εμφανιστεί το επόμενο και οι συγκρούσεις είναι σπάνιες, ενώ η ανισοκατανομή ισχύος στο Skew επηρεάζει κυρίως το ποιος παράγει περισσότερα blocks και όχι τη συχνότητα των συγκρούσεων. Στο Slow, όπου το διάστημα μεταξύ blocks είναι πολύ μεγάλο, τα forks πρακτικά μηδενίζονται. Οι αναδιοργανώσεις εμφανίζονται λιγότερες από τα forks επειδή δεν οδηγεί κάθε σύγκρουση σε αλλαγή της κεφαλής της αλυσίδας. Τέλος, οι τιμές ανά κόμβο τείνουν να ταυτίζονται επειδή το δίκτυο συγκλίνει πλήρως: όλοι οι κόμβοι καταλήγουν να γνωρίζουν το ίδιο σύνολο blocks και να μοιράζονται την ίδια κύρια αλυσίδα, άρα το συνολικό αποτύπωμα σε forks και ορφανά είναι κοινό ανά σενάριο.



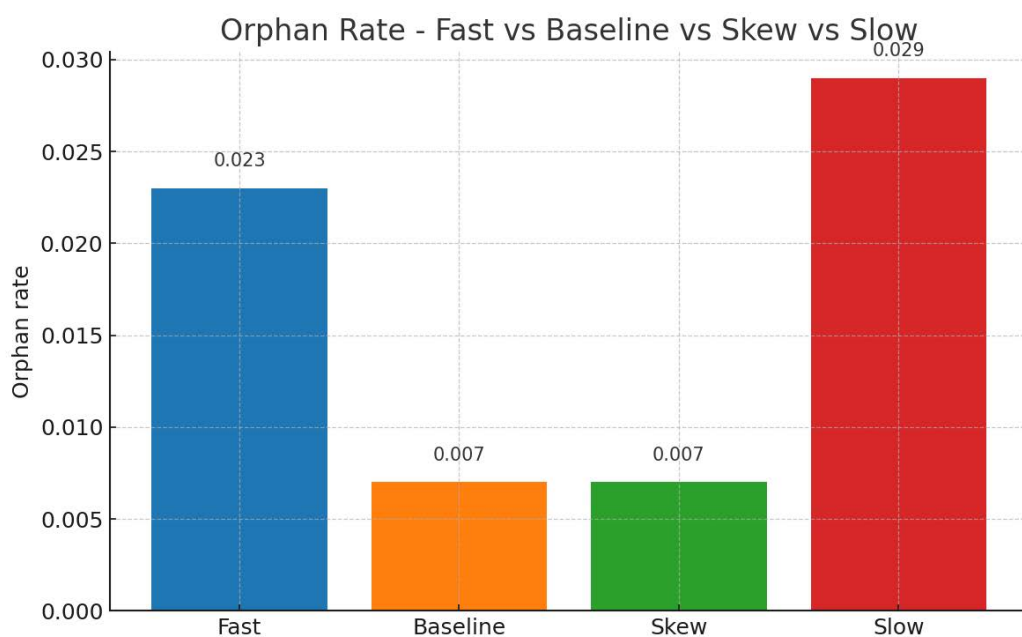
Σχήμα 9 - Reorganizations ανά σενάριο

Στο σχήμα 6 δείχνει ότι οι reorgs ανά σενάριο ακολουθούν το ίδιο μοτίβο με τα forks, κορυφώνονται στο Fast με 132 και περιορίζονται δραστικά στο Baseline 9 και στο Skew 13, ενώ παραμένουν χαμηλά στο Slow 12. Η εικόνα εξηγείται από τη σχέση block time προς χρόνο διάδοσης των blocks, στο Fast ο στόχος των 2 s είναι συγκρίσιμος με τον χρόνο διάδοσης και νέα blocks φτάνουν ταυτόχρονα σε διαφορετικούς κόμβους προκαλώντας συχνές αλλαγές στην καλύτερη αλυσίδα, όσο ο ρυθμός παραγωγής χαλαρώνει στα Baseline Skew Slow το δίκτυο προλαβαίνει να διαδώσει πλήρως κάθε block πριν εμφανιστεί ανταγωνιστικό και έτσι οι αναδιοργανώσεις μειώνονται.



Σχήμα 10 - Μέσο διάστημα εξόρυξης ανά κόμβο (Mining Interval, λογαριθμική κλίμακα)

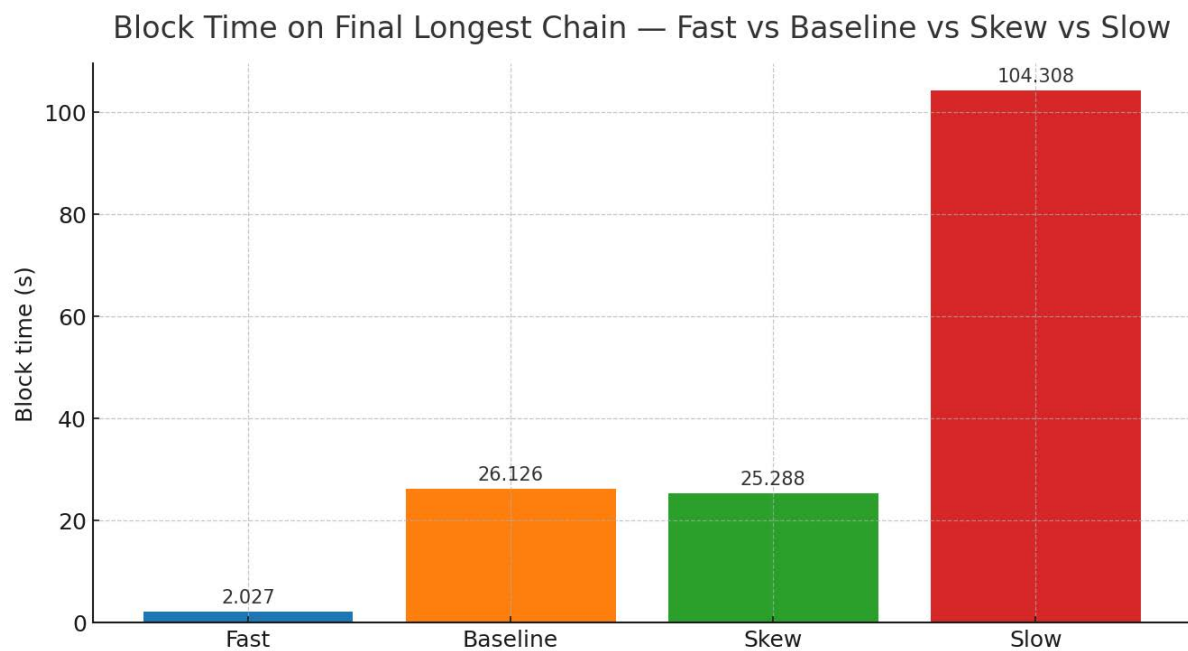
Στο σχήμα 7 δείχνει ότι ο μέσος χρόνος μεταξύ επιτυχιών εξόρυξης ανά κόμβο ακολουθεί στενά τη θεωρία για PoW, το διάστημα είναι ουσιαστικά αντίστροφο της διαθέσιμης hashpower κάθε κόμβου και κλιμακώνεται με τον στόχο παραγωγής του σεναρίου, στο Fast οι ισχυροί κόμβοι εμφανίζουν μικρά και κοντά μεταξύ τους διαστήματα ενώ οι αδύναμοι μεγαλώνουν προοδευτικά επειδή μπαίνουν σπανιότερα στο παράθυρο επιτυχίας, στο Baseline τα διαστήματα μετατοπίζονται συνολικά προς τα πάνω αλλά η σχετική ιεραρχία ισχύος παραμένει ορατή με ομοιόμορφη απόσταση ανάμεσα σε γειτονικούς κόμβους, στο Skew η ανισοκατανομή κάνει την εικόνα πιο έντονη αφού λίγοι κόμβοι με υψηλή ισχύ εμφανίζουν πολύ μικρά διαστήματα και αρκετοί χαμηλής ισχύος μένουν με μηδενική τιμή επειδή στο συγκεκριμένο τρέξιμο δεν εξόρυξαν ούτε ένα block, στο Slow τα διαστήματα εκτοξεύονται για όλους λόγω του πολύ μεγάλου expected_block_time και της μικρής παραγωγής blocks με αποτέλεσμα ο τυχαίος θόρυβος να αφήνει ορισμένους αδύναμους κόμβους χωρίς επιτυχία, συνολικά το γράφημα αποτυπώνει ότι όσο αυξάνει ο στόχος block time και όσο πιο άνιση είναι η κατανομή ισχύος τόσο μεγαλύτερα και πιο διασπαρμένα γίνονται τα διαστήματα εξόρυξης και τόσο πιο συχνά βλέπουμε κόμβους χωρίς καταγεγραμμένη επιτυχία.



Σχήμα 11 - Orphan rate ανά σενάριο

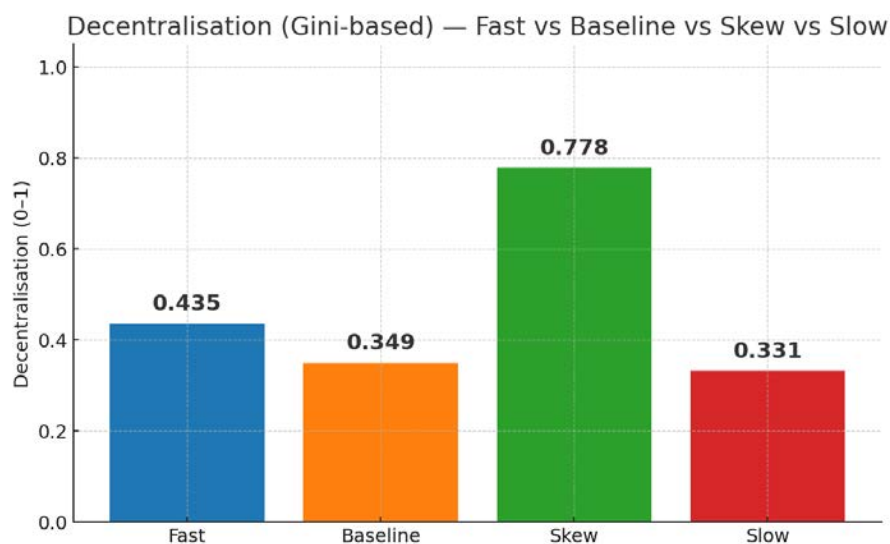
Όπως αποτυπώνεται στο σχήμα 9, ο orphan rate ανά σενάριο είναι Fast 0.023, Baseline 0.007, Skew 0.007, Slow 0.029. Ο δείκτης εκφράζει το ποσοστό των blocks που δεν εντάχθηκαν τελικά στην καλύτερη αλυσίδα και συνδέεται κυρίως με τη σχέση ανάμεσα στον μέσο χρόνο διάδοσης και στο διάστημα μεταξύ διαδοχικών blocks. Στο Fast ο ρυθμός παραγωγής είναι υψηλός και το διάστημα μεταξύ blocks είναι συγκρίσιμο με τον χρόνο διάδοσης, άρα αυξάνει η πιθανότητα δύο κόμβοι να εξορύξουν σχεδόν ταυτόχρονα και ένα από τα blocks να μείνει ορφανό. Στα Baseline και Skew ο στόχος παραγωγής είναι πιο χαλαρός και το δίκτυο προλαβαίνει να διαδώσει τα blocks πριν εμφανιστεί ανταγωνιστικό, γι' αυτό ο orphan rate πέφτει σε 0.007, ενώ το Skew δεν τον αλλάζει ουσιαστικά γιατί

επηρεάζει κυρίως το ποιος κερδίζει την εξόρυξη και όχι τη συχνότητα των επικαλύψεων. Στο Slow ο δείκτης ανεβαίνει ελαφρά σε 0.029 επειδή το πλήθος blocks είναι μικρό, άρα ακόμη και ένα μόνο ορφανό αντιστοιχεί σε μεγαλύτερο ποσοστό, πρόκειται για επίδραση του μικρού παρονομαστή και της τυχειότητας των λίγων γεγονότων και όχι για ουσιαστική υποβάθμιση της διάδοσης. Στα αποτελέσματά σου η τιμή που εκτυπώνεται ανά κόμβο είναι ίδια γιατί ο orphan rate υπολογίζεται σε επίπεδο αλυσίδας και έπειτα αναπαράγεται για κάθε κόμβο.



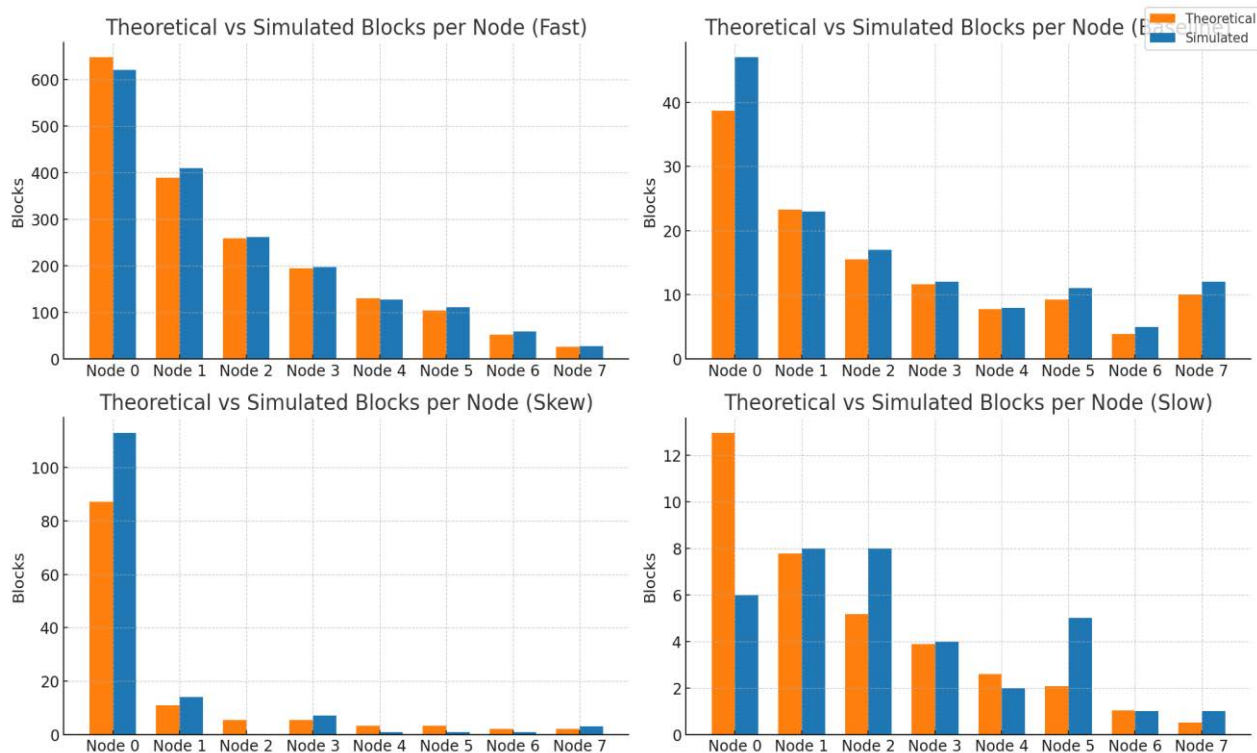
Σχήμα 12 - Χρόνος Block στην τελική μακρύτερη αλυσίδα

Όπως αποτυπώνεται για τον χρόνο block στην τελική μακρύτερη αλυσίδα ανά σενάριο στο σχήμα 12, το αποτέλεσμα ακολουθεί τον στόχο παραγωγής αλλά με διαφορετικό εύρος διακύμανσης, στο Fast ο realized block time είναι κοντά στον στόχο αλλά τείνει να εμφανίζει μικρή υπέρβαση και μεγαλύτερη διασπορά επειδή αρκετά ενδιάμεσα blocks μένουν ορφανά και δεν μετρούνται στην τελική αλυσίδα άρα τα μετρήσιμα διαστήματα είναι μεταξύ διαδοχικών blocks που τελικά επικρατούν μετά από σύντομες αναδιοργανώσεις, στο Baseline η απόκλιση μικραίνει και ο χρόνος συγκλίνει στον στόχο με πιο στενή κατανομή αφού το δίκτυο προλαβαίνει να διαδώσει τα blocks πριν εμφανιστεί ανταγωνιστικό, στο Skew ο ρυθμός παραγωγής δεν αλλάζει άρα ο realized block time παραμένει παρόμοιος με το Baseline και η κύρια διαφορά είναι ποιοι κόμβοι παράγουν τα blocks και όχι πόσο συχνά καταγράφονται στην τελική αλυσίδα, στο Slow ο μέσος χρόνος επίσης ευθυγραμμίζεται με τον μεγάλο στόχο αλλά η διακύμανση αυξάνει λόγω των λίγων συνολικά blocks όπου η τυχειότητα παίζει μεγαλύτερο ρόλο, συνολικά η μέτρηση μόνο στην τελική longest chain τείνει να μετατοπίζει ελαφρά προς τα πάνω τον χρόνο στο Fast εξαιτίας των συχνών forks ενώ στα πιο αργά σενάρια η τιμή σταθεροποιείται γύρω από τον στόχο με σαφώς μικρότερη επίδραση από ορφανά και reorgs.



Σχήμα 13 - Αποκέντρωση (Gini-based)

Στο σχήμα 13 δείχνει ότι ο δείκτης αποκέντρωσης υπολογισμένος ως Gini πάνω στην κατανομή των blocks ανά κόμβο στην τελική αλυσίδα διαφέρει έντονα ανά σενάριο, στο Skew η τιμή 0.778 είναι η μεγαλύτερη και υποδηλώνει υψηλή ανισότητα δηλαδή συγκέντρωση παραγωγής σε λίγους κόμβους, στα Fast και Baseline οι τιμές 0.435 και 0.349 αντίστοιχα δείχνουν πιο ομοιόμορφη κατανομή με μικρότερες διαφορές μεταξύ κόμβων, στο Slow η τιμή 0.331 είναι η χαμηλότερη άρα η πιο κοντά σε ισοκατανομή αν και το μικρό πλήθος blocks αυξάνει την τυχαιότητα, συνολικά όσο πιο άνιση είναι η ισχύς κατακερματισμού και όσο λιγότερες ευκαιρίες έχουν οι αδύναμοι κόμβοι τόσο αυξάνει ο Gini και μειώνεται η αποκέντρωση όπως αποτυπώνεται καθαρά στο Skew.



Σχήμα 14 - Θεωρητικά vs Προσομοιωμένα Blocks

Στο σχήμα 14 δείχνει ότι η σύγκριση θεωρητικών με προσομοιωμένων blocks ανά κόμβο ακολουθεί την κατανομή της hashpower αλλά οι διαφορές αλλάζουν με το πόσα blocks παρήχθησαν συνολικά, στο Fast όπου έχουμε πολλά blocks οι μπάρες των δύο σειρών σχεδόν ταυτίζονται γιατί οι πολλές ευκαιρίες εξομαλύνουν την τύχη και κάθε κόμβος καταλήγει κοντά στην αναλογία που του αναλογεί, στο Baseline η αντιστοίχιση παραμένει καλή αλλά οι πιο αδύναμοι κόμβοι αρχίζουν να αποκλίνουν λίγο περισσότερο, στο Skew η παραγωγή συγκεντρώνεται σε λίγους ισχυρούς κόμβους και για πολλούς αδύναμους υπάρχουν ελάχιστες ή καθόλου επιτυχίες οπότε ένα ή δύο blocks παραπάνω ή παρακάτω κάνουν μεγάλη εντύπωση σε σχέση με το αναμενόμενο, στο Slow τα blocks είναι λίγα και η τύχη παίζει μεγάλο ρόλο με αποτέλεσμα μεγαλύτερες σχετικές διαφορές ανά κόμβο ακόμη κι αν το σύνολο παραμένει συμβατό με τα βάρη, με απλά λόγια όσο λιγότερα blocks παράγονται τόσο πιο έντονα φαίνονται οι αποκλίσεις μεταξύ θεωρητικού και προσομοιωμένου για κάθε κόμβο ενώ όσο αυξάνει ο όγκος παραγωγής οι δύο καμπύλες συγκλίνουν.

Στη συγκεντρωτική αποτίμηση περιλαμβάνουμε και τον δείκτη D_KL ως μέτρο δικαιοσύνης παραγωγής blocks (αναμενόμενο από hashpower vs παρατηρούμενο μερίδιο).

Μετρική	Fast	Baseline	Skew	Slow
Chain length (blocks)	1772	134	139	34
Forks	42	1	1	1
Orphan rate (%)	0,023	0,007	0,007	0,029
Χρόνος Block στην τελική μακρύτερη αλυσίδα	2,027	26,126	25,228	104,308
Reorganization	132	9	13	12
D_KL	0.435	0349	0,778	0,331

Σχήμα 15 – Συγκριτικός Πίνακας Μετρικών PoW

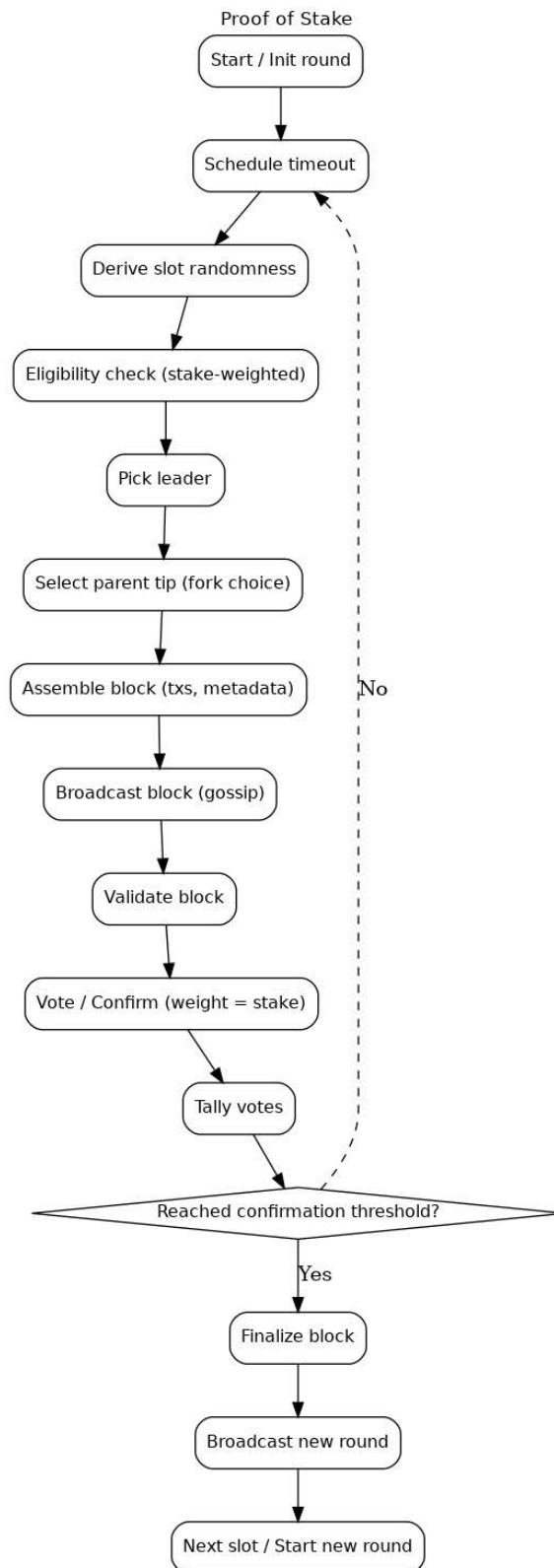
4.3 Συμπεράσματα Κεφαλαίου

Συνοψίζοντας, με βάση τα αριθμητικά αποτελέσματα του PoW: Fast καταγράφει το μεγαλύτερο μήκος αλυσίδας 1772 blocks, orphan rate 0.023 και 132 reorganizations, άρα τον υψηλότερο ρυθμό παραγωγής με αυξημένες απώλειες και αναδιοργανώσεις. Baseline φτάνει τα 134 blocks με orphan rate 0.007 και 9 reorganizations και εμφανίζει πιο σταθερή και ισορροπημένη συμπεριφορά. Skew έχει 139 blocks με orphan rate 0.007 και 13 reorganizations αλλά δείκτη αποκέντρωσης 0.778 που δείχνει ισχυρή συγκέντρωση παραγωγής σε λίγους κόμβους. Slow καταλήγει σε 34 blocks με orphan rate 0.029 και 12 reorganizations και δίνει τη χαμηλότερη παραγωγή και τους μεγαλύτερους χρόνους επιβεβαίωσης. Η εικόνα αυτή λειτουργεί ως σημείο αναφοράς για το Κεφάλαιο 5 όπου εξετάζεται το PoS στα ίδια σενάρια για άμεση σύγκριση με το PoW.

5.1 Μεθοδολογία

Στο κεφάλαιο αυτό παρουσιάζεται ο μηχανισμός συναίνεσης Proof of Stake (PoS) και τα πειραματικά αποτελέσματα σε τέσσερις διαμορφώσεις (Fast, Baseline, Skew, Slow), αντιστοίχως με το Κεφάλαιο 4 για το PoW. Ο proposer κάθε slot στο PoS επιλέγεται με πιθανότητα ανάλογη του stake (proposer_selection: "stake"). Η οριστικοποίηση (finality) βασίζεται κατώφλι στάθμισης (threshold): ένα block τελικοποιείται όταν το συσσωρευμένο stake των CONFIRM υπερβεί το $\text{confirmation_threshold} \times \text{total_stake}$ (π.χ. ~ 0.67). Η εκτέλεση είναι slot-driven (βλ. slot_len, block_interval) και ο τερματισμός της προσομοίωσης ελέγχεται από simTime. Οι μετρικές που αξιολογούνται είναι Throughput (blocks/s), Block time (s), Finality latency (s) και δείκτες αποκέντρωσης/δικαιοσύνης (π.χ. Gini και απόκλιση μεταξύ ιδεώδους κατανομής βάσει stake και πραγματικής ηγεσίας/proposer share).

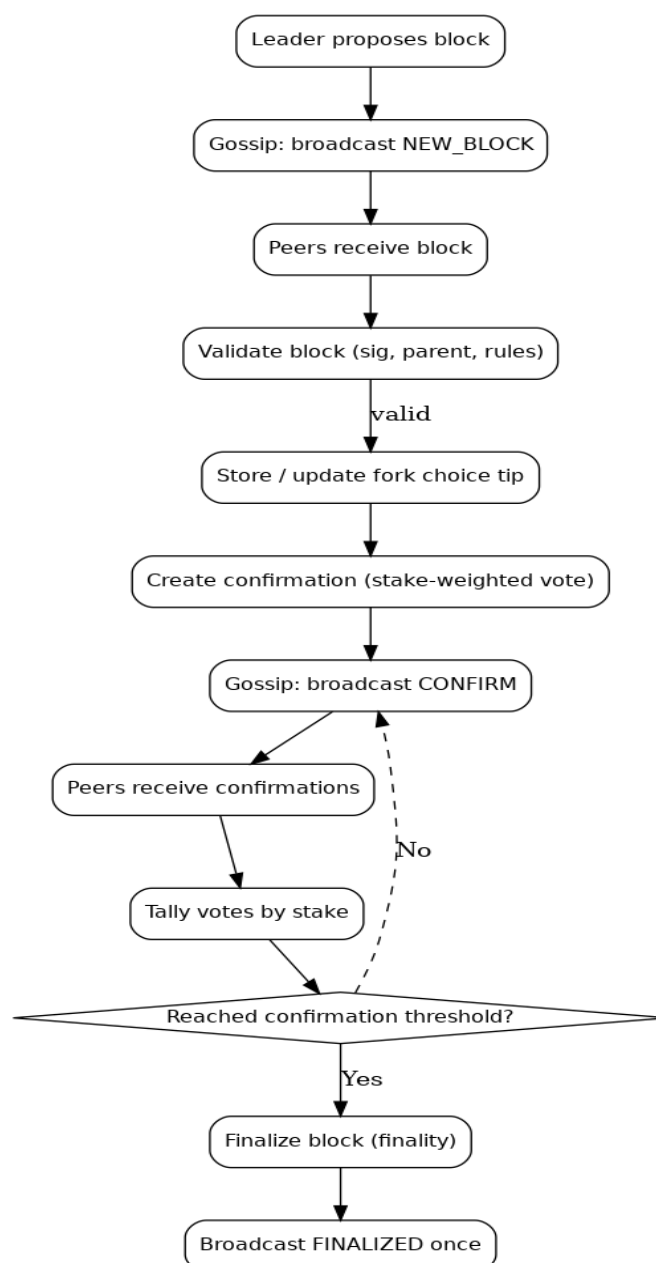
5.2 Διαγράμματα Ροής PoS



Σχήμα 16 – Εποπτική απεικόνιση κύκλου PoS

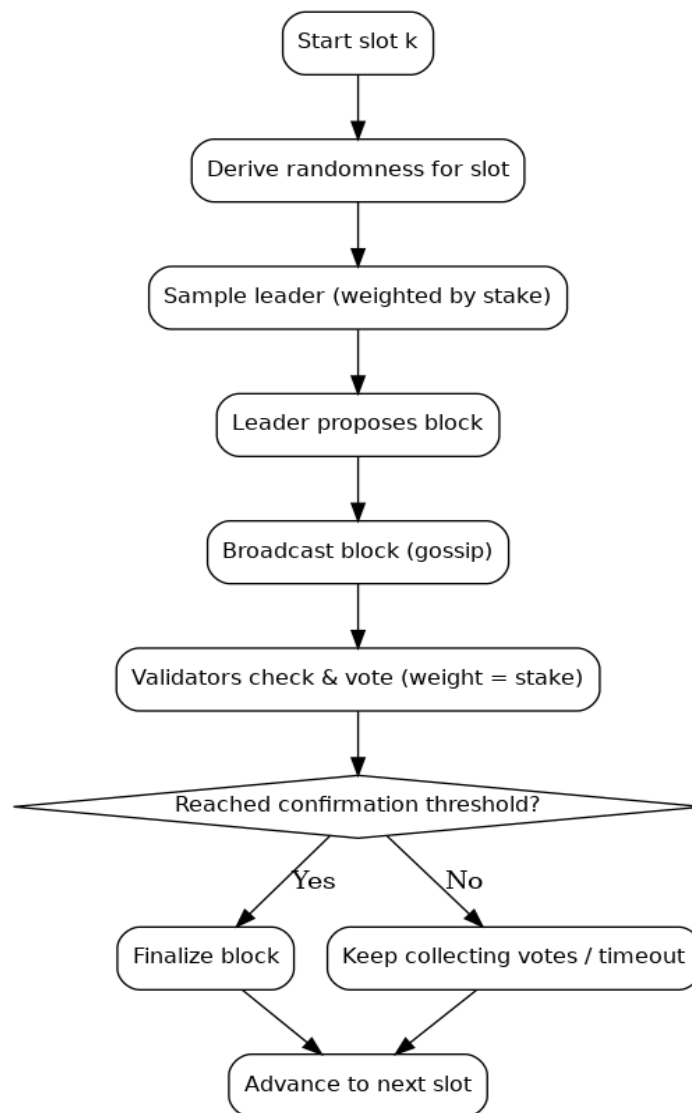
Στο σχήμα 16 ο κύκλος PoS ξεκινά με Start και Schedule timeout. Παράγεται slot randomness και γίνεται stake-weighted eligibility check για την επιλογή του leader. Ο leader

επιλέγει το parent tip σύμφωνα με το fork choice, συναρμολογεί το block και το μεταδίδει με gossip ως Broadcast block. Οι peers το Validate και στη συνέχεια Vote/Confirm με βάρος ίσο με το stake τους. Κάθε κόμβος κάνει Tally votes και ελέγχει το confirmation threshold, δηλαδή αν το συσσωρευμένο βάρος ψήφων υπερβαίνει $\text{confirmation_threshold} \times \text{total_stake}$. Αν δεν έχει επιτευχθεί συνεχίζεται η συλλογή ψήφων ή ενεργοποιείται το timeout όπως δείχνει η διακεκομμένη ροή. Αν έχει επιτευχθεί γίνεται Finalize block, έπειτα Broadcast new round και ξεκινά το Next slot με τον επόμενο γύρο.



Σχήμα 17 – Διάδοση block και οριστικοποίηση (finality)

Στο σχήμα 17 ο proposer του τρέχοντος slot/round δημιουργεί block στο τρέχον fork-choice tip και το διαδίδει με gossip ως NEW_BLOCK. Οι peers το λαμβάνουν, κάνουν Validate (υπογραφές, parent, κανόνες) και αν είναι valid ενημερώνουν το τοπικό tip και στέλνουν confirmation (stake-weighted vote) ως CONFIRM. Κάθε κόμβος κάνει tally by stake και ελέγχει το confirmation threshold, δηλαδή αν το συσσωρευμένο βάρος ψήφων υπερβαίνει $\text{confirmation_threshold} \times \text{total_stake}$. Αν δεν έχει φτάσει συνεχίζεται η συλλογή confirmations, αν έχει φτάσει το block γίνεται Finalized και ανακοινώνεται μία φορά ως FINALIZED. Η finality είναι threshold-based και συνάδει με τη ροή NEW_BLOCK → CONFIRM → FINALIZED.

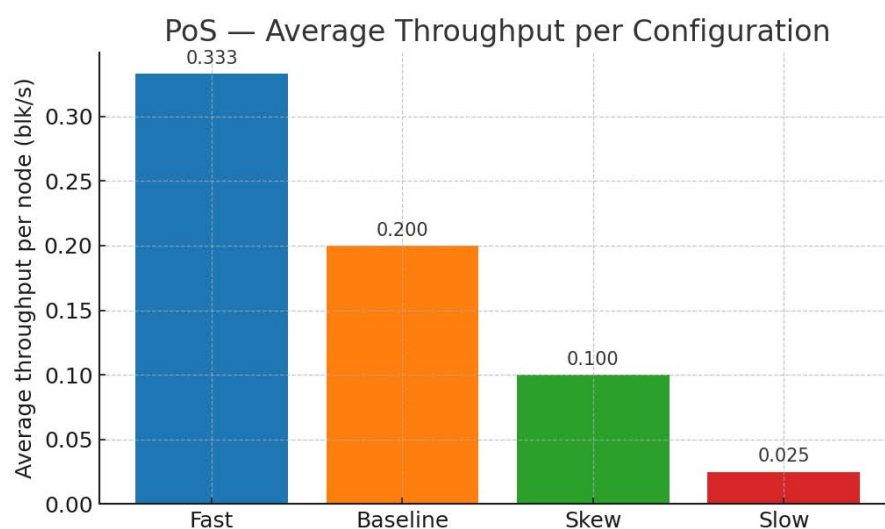


Σχήμα 18 – Ροή Staking/Επιλογής Ηγέτη στο PoS.

Όπως αποτυπώνεται στο Σχήμα 18, σε κάθε slot k παράγεται randomness για το slot, γίνεται sample leader με πιθανότητα ανάλογη του stake, ο leader προτείνει block πάνω στο τρέχον tip και το μεταδίδει με gossip ως Broadcast block. Οι validators ελέγχουν το block και

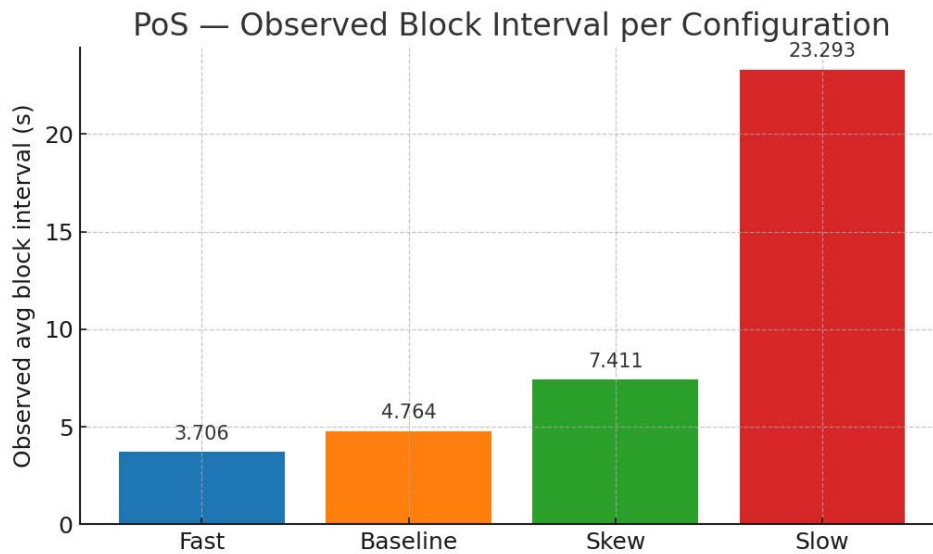
ψηφίζουν με βάρος ίσο με το stake τους (check & vote). Κάθε κόμβος αθροίζει τις ψήφους και ελέγχει αν έχει επιτευχθεί το confirmation threshold, δηλαδή αν το συσσωρευμένο stake των ψήφων υπερβαίνει $\text{confirmation_threshold} \times \text{total_stake}$. Αν ναι το block γίνεται Finalize block, αν όχι συνεχίζεται η συλλογή ψήφων ή ενεργοποιείται timeout. Τέλος το σύστημα προχωρά στο επόμενο slot.

5.3 Αποτελέσματα & Συζήτηση



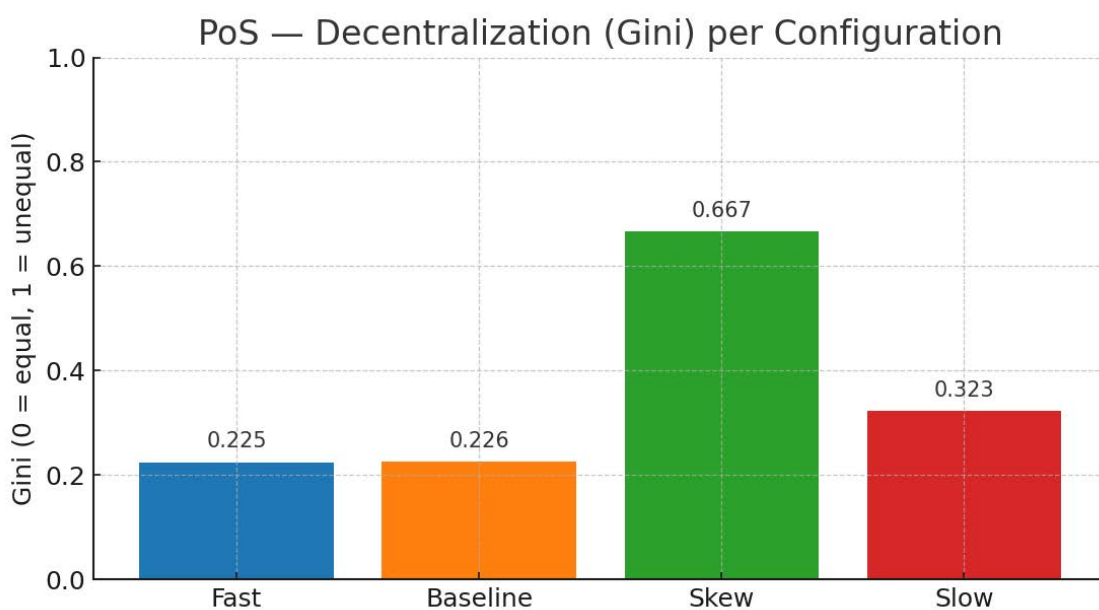
Σχήμα 19 – Μέσο Throughput (blocks/s) ανά διαμόρφωση PoS.

Το σχήμα 19 αποτυπώνει το Average throughput per node (blk/s), δηλαδή τελικοποιημένα blocks / συνολικό χρόνο προσομοίωσης, κανονικοποιημένο ανά κόμβο· μετρά μόνο blocks που έφτασαν σε finality βάσει confirmation threshold, αγνοεί ορφανά/μη έγκυρα. Η σύγκριση των τεσσάρων ρυθμίσεων δείχνει ότι όσο μικρότερο είναι το block_interval και όσο ταχύτερη η διάδοση (gossip latency, timeouts) τόσο υψηλότερος ο ρυθμός παραγωγής τελικοποιημένων blocks· αντίθετα, μεγαλύτερες καθυστερήσεις δικτύου ή/και αργός ρυθμός slots μειώνουν το throughput. Με άλλα λόγια, οι ρυθμίσεις με πιο «ελαφρύ» δίκτυο και επιθετικό χρονισμό πετυχαίνουν υψηλότερη απόδοση, ενώ οι «βαριές» ή αργές ρυθμίσεις περιορίζουν τον καθαρό ρυθμό finality.



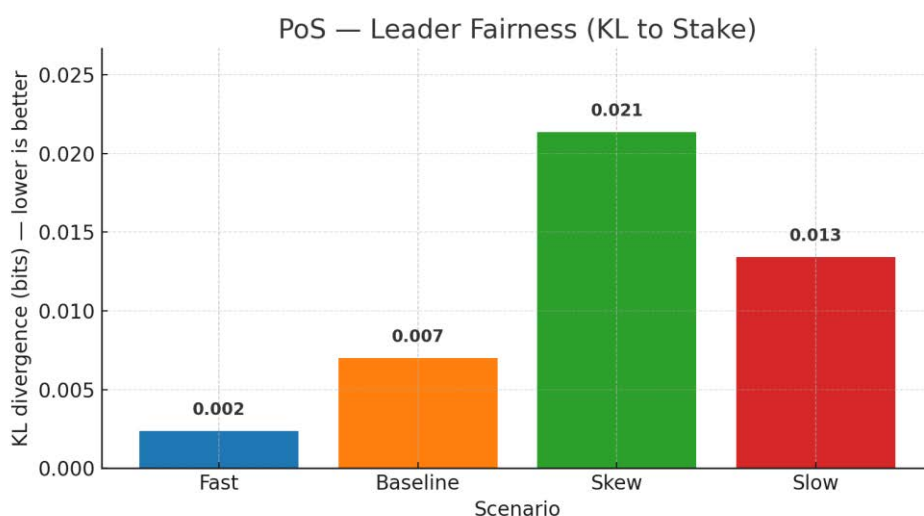
Σχήμα 20 – Κατανομή Διαστήματος Blocks

Στο Σχήμα 20 συνοψίζεται, με μία ματιά, το Observed avg block interval (s) — (ο μέσος χρόνος μεταξύ διαδοχικών blocks που έφτασαν σε finality, σε δευτερόλεπτα, όπως προκύπτει από τα logs του run ανά configuration). Όσο χαμηλότερη η στήλη, τόσο πυκνότερη η παραγωγή blocks και τόσο υψηλότερο το throughput, αφού $\text{throughput} \approx 1 / \text{block interval}$. Η διάταξη δείχνει καθαρά ότι πιο «επιθετικές» ρυθμίσεις χρονοπρογραμματισμού και ταχύτερη διάδοση (μικρό block_interval, χαμηλό gossip latency, ορθά timeouts) οδηγούν σε μικρότερους μέσους χρόνους μεταξύ blocks, ενώ πιο «βαριές» ή αργές ρυθμίσεις αυξάνουν τον χρόνο αυτό, αντανακλώντας περιορισμούς από το δίκτυο και τον ρυθμό των slots. Με άλλα λόγια, το γράφημα λειτουργεί ως καθρέφτης των ρυθμίσεων: μικρά intervals γρηγορότερη αλυσίδα, ενώ μεγάλα intervals βραδύτερη πρόοδος προς finality ανά slot.



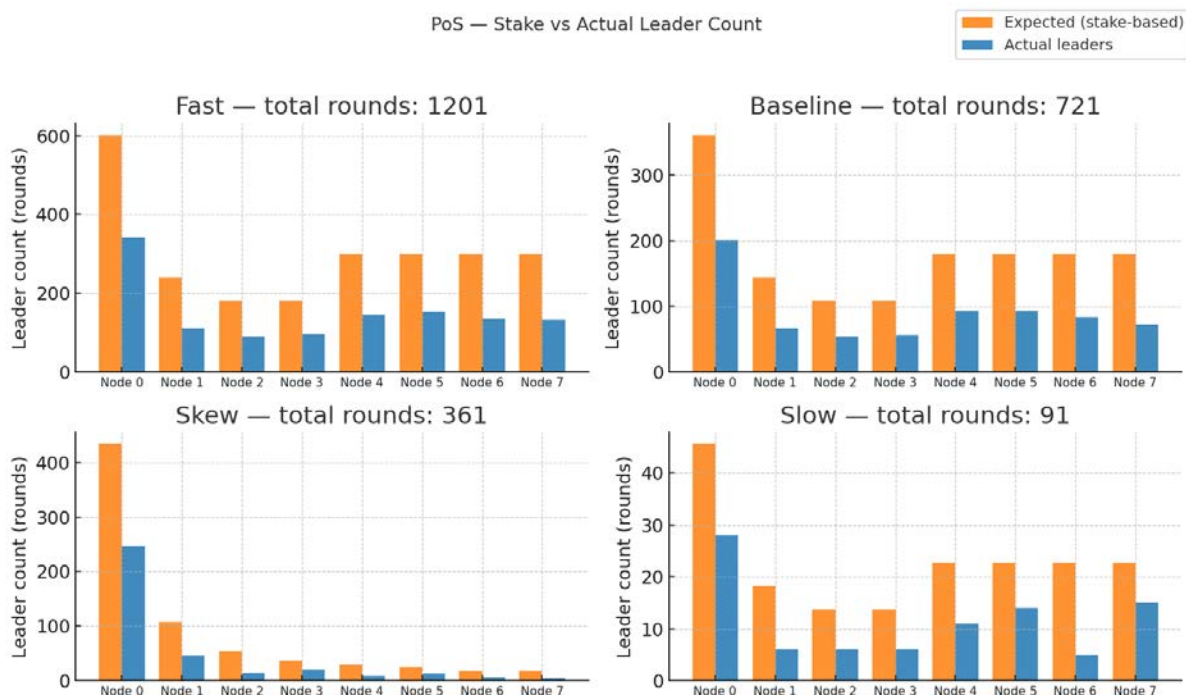
Σχήμα 21 – Αποκέντρωση (Gini-based)

Εξετάζοντας το Σχήμα 21, φαίνεται ο δείκτης Gini για την κατανομή της ηγεσίας (proposer share) ανά configuration: τιμές κοντά στο 0 σημαίνουν ισόρροπη συμμετοχή, ενώ τιμές προς το 1 δηλώνουν συγκέντρωση σε λίγους κόμβους. Με τα τρέχοντα αποτελέσματα, το Skew έχει τη μεγαλύτερη ανισοκατανομή, το Slow βρίσκεται ενδιάμεσα, ενώ Fast και Baseline εμφανίζουν παρόμοια χαμηλή τιμή. Η σύμπτωση Fast/Baseline οφείλεται στο ίδιο stake vector και στον ίδιο stake-weighted leader sampling μηχανισμό. Οι διαφορές στον χρονισμό (π.χ. μικρότερο block_interval στο Fast) επηρεάζουν τον ρυθμό παραγωγής blocks αλλά όχι τις σχετικές πιθανότητες επιλογής ηγέτη ανά κόμβο. Έτσι ο Gini, που μετρά τη σχετική ανισότητα του παρατηρούμενου proposer share, αντικατοπτρίζει κυρίως το stake και το sampling, ενώ οι παράμετροι χρονισμού και δικτύου επηρεάζουν κυρίως το throughput και τις καθυστερήσεις, όχι τη μέτρηση της αποκέντρωσης.



Σχήμα 22 – Δικαιοσύνη επιλογής ηγετών (απόκλιση KL, D_KL)

Όπως φαίνεται στο Σχήμα 22, η Leader Fairness (KL-Divergence, bits) μετρά την απόσταση ανάμεσα στην παρατηρούμενη κατανομή proposer share και στην ιδεώδη stake-proportional κατανομή. Πρακτικά, συγκρίνουμε πόσο συχνά κάθε κόμβος έγινε ηγέτης σε σχέση με το μερίδιό του σε stake, και όσο μικρότερη είναι η τιμή τόσο πιο πιστά εφαρμόζεται το stake-weighted sampling. Η τιμή του D_KL μεγαλώνει όταν κόμβοι με υψηλό stake παίρνουν σχετικά λιγότερα ή περισσότερα blocks από ό,τι αναλογεί, γιατί οι αποκλίσεις για τους μεγάλους κόμβους έχουν μεγαλύτερο βάρος. Έτσι, τα Fast και Baseline έχουν πολύ μικρή απόκλιση επειδή χρησιμοποιούν το ίδιο stake vector και τον ίδιο μηχανισμό δειγματοληψίας· οι διαφορές στον χρονισμό αλλάζουν πόσα blocks παράγονται συνολικά, όχι όμως τις σχετικές πιθανότητες επιλογής ηγέτη. Το Skew έχει τη μεγαλύτερη τιμή, επειδή με έντονα άνισο stake ακόμη και μικρές μετακινήσεις του παρατηρούμενου μεριδίου από τον στόχο μετρούν περισσότερο στο D_KL (η απόκλιση του κόμβου με το μεγαλύτερο stake κυριαρχεί). Το Slow είναι ενδιάμεσα, γιατί οι λίγοι γύροι και το μεγάλο block_interval αυξάνουν τη μεταβλητότητα και έτσι το παρατηρούμενο share εμφανίζει μεγαλύτερες τυχαίες αποκλίσεις από το stake-proportional target.



Σχήμα 23 – Stake vs Πραγματική Συμμετοχή στην Ηγεσία.

Το σχήμα 23 παρουσιάζει, για κάθε σενάριο, τη σύγκριση μεταξύ αναμενόμενων και πραγματικών μετρήσεων ηγεσίας ανά κόμβο σε απόλυτες τιμές rounds. Οι αναμενόμενες τιμές προκύπτουν απλά ως $\text{total rounds} \times \text{stake\%}$ για κάθε κόμβο, δηλαδή τι θα περιμέναμε αν το sampling του leader ήταν ιδεατά stake-proportional χωρίς επιπλέον επιδράσεις από latency ή χρονισμό. Οι πραγματικές τιμές είναι οι leader counts που κατέγραψε ο simulator από τα logs. Η οπτική σύγκριση δείχνει ότι στα Fast και Baseline τα διαστήματα μεταξύ των δύο ράβδων ανά κόμβο είναι μικρά, πράγμα που σημαίνει πως το stake-weighted sampling λειτουργεί όπως αναμένεται και οι αποκλίσεις οφείλονται κυρίως σε τυχαία διακύμανση του finite run. Στο Skew, ο κόμβος με μεγάλο stake παραμένει κυρίαρχος αλλά ορισμένοι κόμβοι με μικρό stake εμφανίζουν ακόμη μικρότερη συμμετοχή από την αναμενόμενη, ένδειξη ότι οι συνθήκες δικτύου ή ο ρυθμός διάδοσης περιορίζουν την ορατότητα των προτάσεών τους και μετατοπίζουν μέρος της πιθανότητας ηγεσίας. Στο Slow, οι απόλυτες μετρήσεις είναι μικρές λόγω λίγων rounds, με αποτέλεσμα μεγαλύτερη σχετική μεταβλητότητα και πιο εμφανείς αποκλίσεις ανά κόμβο. Συνολικά, το γράφημα καθρεφτίζει την ίδια εικόνα που δίνουν οι δείκτες fairness: μικρές αποκλίσεις σε Fast και Baseline συμβαδίζουν με χαμηλό KL-divergence και παρόμοια Gini, ενώ οι μεγαλύτερες διαφορές στο Skew και το Slow εξηγούν τις υψηλότερες τιμές απόκλισης. Ως μεθοδολογική σημείωση, οι μετρήσεις ηγεσίας βασίζονται μόνο σε blocks που ολοκλήρωσαν τη διαδικασία έως το confirmation threshold, άρα οι μετρήσεις είναι εναρμονισμένες με το μοντέλο finality του κώδικα.

5.4 Πίνακας μετρικών PoS ανά σενάριο

Μετρική	Fast	Baseline	Skew	Slow
Μέσο Throughput (blocks/s)	0,333	0,2	0,1	0,025
Κατανομή Διαστήματος Blocks	3,7	4,76	7,41	23,29
Max Finality Latency	1,668	1,968	2,607	1,868
Decentralisation (Gini)	0,225	0,226	0,667	0,323
Δικαιοσύνη ηγετών (D_KL)	0.002	0.007	0,021	0,013

Σχήμα 24 – Συγκριτικός Πίνακας Μετρικών για τις τέσσερις διαμορφώσεις PoS.

Συνολικά και με το σχήμα 24 τα αποτελέσματα δείχνουν ξεκάθαρα ότι το Μέσο Throughput κινείται αντίστροφα με την Κατανομή Διαστήματος Blocks (μικρότερο interval $\Rightarrow$ περισσότερα blocks/s), με το Fast να υπερέχει και το Slow να υστερεί. Η Max Finality Latency δεν είναι αμιγώς μονοτονική—το Skew έχει τη μεγαλύτερη καθυστέρηση τελικοποίησης, επειδή οι συνθήκες διάδοσης/χρονοπρογραμματισμού αυξάνουν την αναμονή μέχρι να φτάσει το confirmation threshold. Η Decentralisation (Gini) είναι ίδια για Fast/Baseline (ίδιο stake-based sampling) και πιο χαμηλή στο Skew—δηλαδή πιο ισόρροπη συμμετοχή στην ηγεσία. Σε όρους δικαιοσύνης ηγετών (D_KL), τα Fast και Baseline είναι πολύ κοντά στο ιδεώδες stake-proportional, ενώ το Skew και, σε μικρότερο βαθμό, το Slow παρουσιάζουν μεγαλύτερη απόκλιση.

5.5 Συμπεράσματα Κεφαλαίου

Στο κεφάλαιο αυτό αξιολογήθηκε πειραματικά το PoS μοντέλο με threshold-based finality και stake-weighted leader sampling. Τα αποτελέσματα δείχνουν ότι η απόδοση κλιμακώνεται κυρίως με τον χρονισμό: μικρότερο block_interval και ομαλή gossip latency οδηγούν σε υψηλότερο Throughput (blocks/s) και μικρότερο Block interval (s), ενώ πιο «βαριές» ρυθμίσεις (Slow) μειώνουν την παραγωγή και αυξάνουν τους χρόνους. Η Max Finality Latency επηρεάζεται από τη σταθερότητα της διάδοσης και το πόσο γρήγορα συγκεντρώνεται το απαιτούμενο βάρος ψήφων στο confirmation threshold.

Ως προς τη δικαιοσύνη (fairness), οι μετρήσεις Gini και D_KL υποδεικνύουν ότι όταν το stake vector και ο μηχανισμός stake-weighted sampling παραμένουν σταθερά (π.χ. Fast/Baseline), η σχετική κατανομή της ηγεσίας (proposer share) είναι κοντά στο ιδεώδες stake-proportional, ακόμη κι αν αλλάζει ο ρυθμός παραγωγής blocks. Αντίθετα, σενάρια με

λιγότερους γύρους και μεγαλύτερη μεταβλητότητα δικτύου (Skew/Slow) εμφανίζουν μεγαλύτερη απόκλιση (higher D_KL) και ελαφρώς πιο άνιση συμμετοχή (Gini), γεγονός που συνδέεται με τη στοχαστική φύση περιορισμένων runs και την ασύμμετρη ορατότητα blocks.

Η σύγκριση Expected vs Actual Leader Count ανά κόμβο επιβεβαιώνει εμπειρικά τα παραπάνω: όπου οι συνθήκες δικτύου είναι σταθερές, οι πραγματικοί μετρητές ηγεσίας ευθυγραμμίζονται με τις αναμενόμενες τιμές από το stake%, ενώ σε «βαριά» σενάρια παρατηρούνται αποκλίσεις που εξηγούν τις αυξημένες τιμές στους δείκτες αποκέντρωσης/δικαιοσύνης.

Πρακτικά, το μοντέλο threshold-based finality αποδείχθηκε αποτελεσματικό και προβλέψιμο: όταν το tally by stake υπερβαίνει το $\text{confirmation_threshold} \times \text{total_stake}$, η τελικοποίηση επιτυγχάνεται γρήγορα και διατηρείται η πρόοδος της αλυσίδας χωρίς να απαιτείται πολυβάθμια επιβεβαίωση. Η ρύθμιση των timeouts και του $\text{slot_len/block_interval}$ αναδεικνύεται ως βασικός μοχλός επίδοσης, χωρίς να αλλοιώνει—από μόνη της—τη σχετική πιθανότητα επιλογής ηγέτη που καθορίζεται από το stake.

Κεφάλαιο 6 Συμπεράσματα & Μελλοντική Εργασία

Η εργασία χαρακτήρισε τις οικογένειες PoW και PoS σε ενιαίο, ελεγχόμενο πλαίσιο με έμφαση στα σενάρια Fast, Baseline, Skew, Slow. Για το PoW καταγράφηκαν ιδιότητες που συνδέονται με την πιθανολογική τελικοποίηση και τη δυναμική των forks/reorgs, τον orphan rate και το propagation delay, σε συνάρτηση με τις παραμέτρους expected_block_time και num_neighbours. Για το PoS η αξιολόγηση ευθυγραμμίστηκε με τον τρέχοντα κώδικα: ο proposer επιλέγεται με stake-weighted sampling και η τελικοποίηση είναι threshold-based ένα block γίνεται Finalized όταν το συσσωρευμένο βάρος CONFIRM ψήφων υπερβεί το confirmation_threshold $\times$ total_stake. Η εκτέλεση είναι slot-driven με τερματισμό μέσω sim_time, ενώ τα μηνύματα/γεγονότα ακολουθούν τη ροή NEW_BLOCK $\rightarrow$ CONFIRM $\rightarrow$ FINALIZED. Οι βασικές μετρικές που μετρήθηκαν είναι Throughput (blocks/s), Block interval (s), Finality latency (s), καθώς και δείκτες αποκέντρωσης/δικαιοσύνης (Gini, D_KL έναντι της ιδεώδους stake-proportional κατανομής).

Σε επίπεδο σεναρίων, το Baseline λειτούργησε ως ισορροπημένο σημείο αναφοράς, το Fast αύξησε τη διαπερατότητα (μικρότερο block_interval, χαμηλότερη καθυστέρηση διάδοσης) και πέτυχε υψηλότερο throughput με μικρότερο block interval, το Skew ανέδειξε ζητήματα fairness (διαφορές μεταξύ stake και πραγματικού proposer share) και επιβάρυνση σε finality latency υπό λιγότερο ευνοϊκές συνθήκες διάδοσης, ενώ το Slow έδειξε χαμηλό ρυθμό προόδου με μεγαλύτερα διαστήματα μεταξύ blocks και αυξημένη μεταβλητότητα. Στα αποτελέσματα decentralization/fairness, τα Fast/Baseline εμφάνισαν παρόμοια Gini και πολύ μικρό D_KL ενώ τα Skew/Slow παρουσίασαν μεγαλύτερες αποκλίσεις όταν οι συνθήκες δικτύου/χρονισμού επηρεάζουν την ορατότητα και τη σταθερότητα της επιλογής ηγέτη.

Τα ευρήματα πρέπει να διαβαστούν υπό τους περιορισμούς του πλαισίου: finite runs (πεπερασμένος αριθμός γύρων/blocks), συνθετική μοντελοποίηση καθυστερήσεων δικτύου, απουσία ρητά κακόβουλων στρατηγικών και slashing, καθώς και απλοποιήσεις σε mempool/fees. Παρά ταύτα, το κοινό πειραματικό περιβάλλον επέτρεψε τη στοχευμένη απομόνωση παραγόντων επιδόσεων (χρονισμός, διάδοση), τελικοποίησης (threshold) και δικαιοσύνης (stake-proportionality) σε PoW/PoS, δίνοντας ένα καθαρό λειτουργικό αποτύπωμα χωρίς ιεραρχίες πρωτοκόλλων, αλλά με έμφαση στη διαφάνεια των trade-offs.

Μελλοντική εργασία. Προτείνεται: κλιμάκωση σε μεγαλύτερα δίκτυα με ρεαλιστικές τοπολογίες και γεωγραφική διασπορά, διερεύνηση adaptive timeouts και δυναμικού block_interval βάσει φόρτου/latency, ενσωμάτωση αντιπάλου και κανόνων slashing/ανίχνευσης διπλών ψήφων, μελέτη stake churn και epoch-level μηχανισμών randomness/committee-selection και λεπτομερή παρακολούθηση fairness με windowed D_KL και ανά-epoch Gini. Συνολικά, η επιλογή πρωτοκόλλου πρέπει να ευθυγραμμίζεται με το ζητούμενο μείγμα απόδοσης – καθυστέρησης – αποκέντρωσης, και το προτεινόμενο πλαίσιο παρέχει τις βάσεις για πιο ρεαλιστικές και γενικεύσιμες αξιολογήσεις στο επόμενο βήμα.

1. Diamantopoulos, G., Tziritas, N., Bahsoon, R., & Theodoropoulos, G. (2022, December). Digital twins for dynamic management of blockchain systems. In 2022 Winter Simulation Conference (WSC) (pp. 2876-2887). IEEE.
2. Diamantopoulos, G., Tziritas, N., Bahsoon, R., & Theodoropoulos, G. (2022, October). Dynamic data-driven digital twins for blockchain systems. In International Conference on Dynamic Data Driven Applications Systems (pp. 283-292). Cham: Springer Nature Switzerland.
3. Nakamoto, S. (2008). *Bitcoin: A peer-to-peer electronic cash system*. <https://bitcoin.org/bitcoin.pdf>
4. Garay, J., Kiayias, A., & Leonardos, N. (2015). The bitcoin backbone protocol: Analysis and applications. In J. A. Garay & R. Gennaro (Eds.), *Advances in Cryptology – EUROCRYPT 2015* (pp. 281–310). Springer.
5. Decker, C., & Wattenhofer, R. (2013). Information propagation in the Bitcoin network. *2013 IEEE Thirteenth International Conference on Peer-to-Peer Computing (P2P)*, 1–10. IEEE.
6. Gervais, A., Karame, G. O., Wüst, K., Glykantzis, V., Ritzdorf, H., & Capkun, S. (2016). On the security and performance of proof-of-work blockchains. *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security (CCS '16)*, 3–16. ACM.
7. Kiayias, A., Russell, A., David, B., & Oliynykov, R. (2017). Ouroboros: A provably secure proof-of-stake blockchain protocol. In J. Katz & H. Shacham (Eds.), *Advances in Cryptology – CRYPTO 2017* (pp. 357–388). Springer.
8. David, B., Gaži, P., Kiayias, A., & Russell, A. (2018). Ouroboros Praos: An adaptively secure, semi-synchronous proof-of-stake protocol. In J. B. Nielsen & V. Rijmen (Eds.), *Advances in Cryptology – EUROCRYPT 2018* (pp. 66–98). Springer.
9. Gilad, Y., Hemo, R., Micali, S., Vlachos, G., & Zeldovich, N. (2017). Algorand: Scaling Byzantine agreements for cryptocurrencies. *Proceedings of the 26th ACM Symposium on Operating Systems Principles (SOSP '17)*, 51–68. ACM.
10. Buterin, V., & Griffith, V. (2017). Casper the friendly finality gadget. *arXiv*. <https://arxiv.org/abs/1710.09437>
11. Demers, A., Greene, D., Hauser, C., Irish, W., Larson, J., Shenker, S., Sturgis, H., Swinehart, D., & Terry, D. (1987). Epidemic algorithms for replicated database maintenance. *Proceedings of the Sixth Annual ACM Symposium on Principles of Distributed Computing (PODC '87)*, 1–12. ACM.
12. Croman, K., et al. (2016). On scaling decentralized blockchains. In J. Clark et al. (Eds.), *Financial Cryptography and Data Security: Workshops* (pp. 106–125). Springer.

Ιδιαίτερες ευχαριστίες. Ο συντάκτης ευχαριστεί θερμά τον κ. Διαμαντόπουλο, Γεώργιο για την έμπνευση που προσέφερε και για την ευγενική διάθεση τμήματος του κώδικά του, σχετικού με PBFT και το πλαίσιο Bigfoot, που συνέβαλε ουσιαστικά στη διαμόρφωση και τον εμπλουτισμό της παρούσας εργασίας.