

2026-02

pÿ œ μ » - Ä . ° ± 1 ' ½ ¬ Ä Ä Å ¾ . ' » ³ ¿ Á ¯ , 1
 pÿ ' Á Ç ¹ Ä μ ° Ä ¿ ½ ¹ ° - Â • À μ ¾ μ Á ³ ± Ã Ä Î
 pÿ ¿ » » ± Ä » Î ½ Å Á ® ½ É ½ ³ ¹ ± Ä . ½ •
 pÿ Ä . Â ' Í ¾ . Ã . Â Ä . Â • À ¯ ' ¿ Ã . Â ° ± 1
 pÿ " ¹ μ ° À μ Á ± ¹ É Ä ¹ ° ì Ä . Ä ± Â Ä É ½
 pÿ ÿ » ¿ ° » . Á É ¼ - ½ É ½ • ½ ´ ¿ - Ã Å ½ ´ μ ì 1

pÿ ' Á ¹ Ã Ä ¿ ´ ® ¼ ¿ Å , ! ¯ » ¹ Ä Ä ¿ Â

pÿ Á ì ³ Á ± ¼ ¼ ± " . ¼ ì Ã ¹ ± Â " ¹ ¿ ¯ ° . Ã . Â , £ Ç ¿ » ® ÿ ¹ ° ¿ ½ ¿ ¼ ¹ ° Î ½ • Ä ¹ Ã Ä . ¼ Î ½ ° ± ¹ " ¹ ¿ ¯ ° .
 pÿ ± ½ μ Ä ¹ Ã Ä ® ¼ ¹ ¿ • μ ¬ Ä ¿ » ¹ Â ¬ Æ ¿ Å

<http://hdl.handle.net/11728/13396>

Downloaded from HEPHAESTUS Repository, Neapolis University institutional repository

Πανεπιστήμιο Νεάπολις Πάφος
Σχολή Οικονομικών, Διοίκησης και Πληροφορικής
Τμήμα Πληροφορικής

Μελέτη και Ανάπτυξη Αλγορίθμων σε Αρχιτεκτονικές
Επεξεργαστών Πολλαπλών Πυρήνων για την Επίτευξη
της Αύξησης της Επίδοσης και της Διεκπεραιωτικότητας
των Ολοκληρωμένων Ενδο-συνδεόμενων Δικτύων

Διπλωματική Εργασία
του
Φίλιππος Αριστοδήμου

Επιβλέπουσα: Δρ. Έλενα Κακουλλή

Φεβρουάριος 2026

Περίληψη

Τα Ολοκληρωμένα Ενδο-συνδεδεμένα Δίκτυα (ΟΕΔ), γνωστά διεθνώς ως Networks-on-Chip (NoCs), αποτελούν την κυρίαρχη υποδομή επικοινωνίας στους σύγχρονους πολυπύρηνους επεξεργαστές. Η απόδοση του δικτύου επηρεάζει άμεσα τη συνολική απόδοση του συστήματος, καθιστώντας τον σχεδιασμό αποδοτικών αλγορίθμων δρομολόγησης κρίσιμο ζήτημα.

Η παρούσα εργασία παρουσιάζει τον αλγόριθμο APAR (Adaptive Phase-Aware Routing), έναν υβριδικό αλγόριθμο δρομολόγησης που συνδυάζει τα πλεονεκτήματα της ντετερμινιστικής και της προσαρμοστικής δρομολόγησης. Η βασική καινοτομία του APAR έγκειται στην ικανότητά του να ανιχνεύει δυναμικά τη φάση λειτουργίας του δικτύου βάσει των επιπέδων συμφόρησης και να προσαρμόζει τη στρατηγική δρομολόγησης ανάλογα.

Ο APAR ταξινομεί την κατάσταση του δικτύου σε τρεις φάσεις: χαμηλή, μέτρια και υψηλή συμφόρηση. Σε χαμηλή συμφόρηση χρησιμοποιεί ντετερμινιστική δρομολόγηση XY για ελάχιστη καθυστέρηση. Σε μέτρια και υψηλή συμφόρηση μεταβαίνει σε προσαρμοστική δρομολόγηση βασισμένη στο μοντέλο Odd-Even για εξισορρόπηση φόρτου. Ένας μηχανισμός υστέρησης εξασφαλίζει ομαλές μεταβάσεις μεταξύ φάσεων.

Η πειραματική αξιολόγηση βασίστηκε σε πλήρη μήτρα συνθηκών (*mesh, traffic, IR, run*) και εκτελέστηκε με προσομοίωση ακριβείας κύκλου ρολογιού. Συγκεκριμένα, εξετάστηκαν 3 μεγέθη πλέγματος (4×4 , 8×8 , 16×16), 7 συνθετικά πρότυπα κυκλοφορίας (Uniform, Transpose, Hotspot, Bit-Complement, Bit-Reverse, Neighbor, Tornado) και 18 ρυθμοί έγχυσης (0.01–0.50 flits/node/cycle), με 3 επαναλήψεις ανά διαμόρφωση. Αυτό αντιστοιχεί σε $3 \times 7 \times 18 \times 3 = 1134$ διαμορφώσεις πειράματος. Καθώς συγκρίθηκαν 5 αλγόριθμοι (XY, West-First, Odd-Even, CARL, APAR), το συνολικό πλήθος εκτελέσεων προσομοίωσης ήταν $1134 \times 5 = 5670$.

Σε κάθε εκτέλεση μετρήθηκαν βασικές μετρικές (throughput, latency, hops) μέσα σε σαφώς ορισμένο παράθυρο μέτρησης μετά από warmup, και εφαρμόστηκε φάση εκκένωσης (drain) για ολοκλήρωση των πακέτων. Επιπλέον, πραγματοποιήθηκαν έλεγχοι εγκυρότητας (no-drop μετά το drain και μηδενικά drain timeouts), ώστε τα αποτελέσματα να είναι συγκρίσιμα και αναπαραγώγιμα.

Τα αποτελέσματα δείχνουν ότι η απόδοση είναι έντονα εξαρτώμενη από το πρότυπο κυκλοφορίας (traffic-dependent): ο ντετερμινιστικός XY παραμένει ισχυρός σε ομοιόμορφη κυκλοφορία, ενώ ο APAR εμφανίζει κέρδη σε πρότυπα με έντονη δομή,

όπως το bit-reverse (π.χ. +33.5% μέγιστη διεκπεραίωση (peak throughput) έναντι XY σε 8×8). Συνολικά, η εργασία υποστηρίζει ότι η επίγνωση φάσης μπορεί να προσφέρει στοχευμένα οφέλη σε συγκεκριμένα σενάρια συμφόρησης, χωρίς όμως να τεκμηριώνει ότι μία και μοναδική πολιτική δρομολόγησης υπερέχει σταθερά σε κάθε περίπτωση.

Λέξεις-κλειδιά: Ολοκληρωμένα Ενδο-συνδεόμενα Δίκτυα, Αλγόριθμοι Δρομολόγησης, Προσαρμοστική Δρομολόγηση, Ανίχνευση Φάσης, Πολυπύρηντοι Επεξεργαστές

Abstract

Networks-on-Chip (NoC) constitute the dominant communication infrastructure in modern multicore processors. Network performance directly affects overall system performance, making the design of efficient routing algorithms a critical concern.

This thesis presents the APAR (Adaptive Phase-Aware Routing) algorithm, a hybrid routing algorithm that combines the advantages of deterministic and adaptive routing. The key innovation of APAR lies in its ability to dynamically detect the network's operating phase based on congestion levels and adapt the routing strategy accordingly.

APAR classifies the network state into three phases: low, medium, and high congestion. In low congestion, it uses deterministic XY routing for minimal latency. In medium and high congestion, it transitions to adaptive routing based on the Odd-Even turn model for load balancing. A hysteresis mechanism ensures smooth transitions between phases.

Evaluation was performed through **5670** cycle-accurate simulation experiments on three mesh sizes (4×4 , 8×8 , 16×16), seven traffic patterns and 18 injection rates (0.01–0.50 flits/node/cycle), with three runs per configuration. Five algorithms were compared: XY, West-First, Odd-Even, CARL, and APAR, with strict integrity checks (no-drop after drain and zero drain timeouts).

Results indicate strongly **traffic-dependent** behavior: deterministic XY remains strong under uniform traffic, while APAR shows benefits on structured patterns such as bit-reverse (e.g., +33.5% peak throughput vs. XY on 8×8).

Keywords: Networks-on-Chip, Routing Algorithms, Adaptive Routing, Phase Detection, Multicore Processors

Ευχαριστίες

Θα ήθελα να εκφράσω τις θερμές μου ευχαριστίες στην επιβλέπουσα καθηγήτρια της διπλωματικής μου εργασίας, Δρ. Έλενα Κακουλλή, για την πολύτιμη καθοδήγηση, τη συνεχή υποστήριξη και την εμπιστοσύνη που μου έδειξε καθ' όλη τη διάρκεια εκπόνησης της παρούσας διπλωματικής εργασίας. Ευχαριστώ εξίσου θερμά και την Δρ. Σαλώμη Ευριπίδου, που αποδέχτηκε να είναι μέλος στην αξιολόγηση της πτυχιακής μου εργασίας. Τέλος, θα ήταν παράλειψή μου να μην ευχαριστήσω τη σύζυγο μου Ειρήνη, που μου προσέφερε ψυχολογική στήριξη και κίνητρο ώστε να φέρω εις πέρας τη διπλωματική μου εργασία.

Περιεχόμενα

1	Εισαγωγή.....	10
1.1	Αντικείμενο και Κίνητρα της Έρευνας	10
1.2	Διατύπωση του Προβλήματος	11
1.3	Στόχοι της Έρευνας	11
1.4	Συνεισφορές της Εργασίας	12
1.5	Σημασία της Έρευνας.....	12
1.6	Δομή της Εργασίας	13
2	Θεωρητικό Υπόβαθρο και Σχετικές Εργασίες.....	14
2.1	Θεμελιώδεις Έννοιες Ολοκληρωμένων Ενδο-συνδεόμενων Δικτύων	14
2.1.1	Εξέλιξη των Ενδοεπικοινωνιακών Υποδομών	14
2.1.2	Αρχιτεκτονικές Συνιστώσες NoC.....	14
2.1.3	Τοπολογίες Δικτύου	15
2.1.4	Δομή Πακέτων και Έλεγχος Ροής.....	18
2.2	Αλγόριθμοι Δρομολόγησης.....	19
2.2.1	Ταξινόμηση Αλγορίθμων Δρομολόγησης.....	19
2.2.2	Αδιέξοδα και το Μοντέλο Στροφών	19
2.2.3	Δρομολόγηση ΧΥ.....	20
2.2.4	Δρομολόγηση West-First	20
2.2.5	Μοντέλο Στροφών Odd-Even	20
2.3	Δρομολόγηση με Επίγνωση Συμφόρησης.....	21
2.4	Σχετικές Εργασίες στην Προσαρμοστική Δρομολόγηση	21
2.4.1	Προσεγγίσεις Βασισμένες σε Ενισχυτική Μάθηση	21
2.4.2	Σύγκριση με State-of-the-Art Αλγορίθμους	22
2.4.3	Κενά στην Υπάρχουσα Βιβλιογραφία	23
2.5	Μετρικές Αξιολόγησης Απόδοσης	24

2.6	Πρότυπα Κυκλοφορίας.....	25
3	Μεθοδολογία	26
3.1	Επισκόπηση Πειραματικής Προσέγγισης.....	26
3.2	Πλαίσιο Προσομοίωσης.....	26
3.2.1	Αρχιτεκτονική Προσομοιωτή	26
3.2.2	Μοντέλο Δρομολογητή.....	27
3.2.3	Γεννήτριες Κυκλοφορίας	28
3.3	Παράμετροι Πειραμάτων	28
3.3.1	Διαμόρφωση Δικτύου	29
3.3.2	Ρυθμοί Έγχυσης	29
3.3.3	Περίοδος Προσομοίωσης	29
3.4	Μετρικές Αξιολόγησης	30
3.5	Αλγόριθμοι Αναφοράς	30
3.6	Περιβάλλον Εκτέλεσης	31
3.7	Έλεγχοι Εγκυρότητας Πειραματικού Πλαισίου	31
4	Ο Προτεινόμενος Αλγόριθμος APAR.....	32
4.1	Επισκόπηση Αρχιτεκτονικής	32
4.2	Μηχανισμός Ανίχνευσης Φάσης	34
4.2.1	Μετρικές Συμφόρησης	34
4.2.2	Κατώφλια Φάσεων	34
4.2.3	Μηχανισμός Υστέρησης	34
4.3	Πολιτικές Δρομολόγησης ανά Φάση	36
4.3.1	Φάση Χαμηλής Συμφόρησης.....	36
4.3.2	Φάση Μέτριας Συμφόρησης	36
4.3.3	Φάση Υψηλής Συμφόρησης	36
4.3.4	Ενδεικτικά Σενάρια Δρομολόγησης.....	37
4.4	Υλοποίηση	38
4.4.1	Δομές Δεδομένων	38
4.4.2	Αλγόριθμος Υπολογισμού Διαδρομής.....	38

4.4.3	Ενημέρωση Πληροφοριών Συμφόρησης.....	39
4.5	Ανάλυση Ορθότητας	39
4.5.1	Απαλλαγή από Αδιέξοδα	39
4.5.2	Ζωντάνια	40
4.6	Ανάλυση Πολυπλοκότητας	40
4.6.1	Χωρική Πολυπλοκότητα	40
4.6.2	Χρονική Πολυπλοκότητα	40
4.7	Σύγκριση με Υπάρχουσες Προσεγγίσεις	41
4.7.1	Σύγκριση με DyAD.....	41
4.7.2	Σύγκριση με CARL	41
4.7.3	Σύγκριση με Odd-Even.....	41
4.8	Παράμετροι Σχεδιασμού	42
4.8.1	Επιλογή Κατωφλίων.....	42
4.8.2	Εύρος Υστέρησης	42
4.8.3	Βάθος Πληροφοριών Γειτόνων.....	42
5	Πειραματικά Αποτελέσματα	43
5.1	Σύνοψη πειραμάτων και εγκυρότητα δεδομένων	43
5.2	Μέγιστη διεκπεραιωτική ικανότητα (peak throughput) στο 8×8	43
5.3	Καμπύλες throughput/καθυστέρησης ανά πρότυπο κυκλοφορίας (8×8)	45
5.4	Σύγκριση σε σταθερό προσφερόμενο φόρτο (IR=0.30, 8×8)	48
5.5	Συμπεριφορά φάσεων APAR (8×8)	51
5.6	Σημείο κορεσμού και εστίαση στην προκορεσμική περιοχή	52
5.7	Κλιμακωσιμότητα (4×4, 8×8, 16×16).....	52
5.8	Συνολική αποτίμηση των αποτελεσμάτων	54
6	Συμπεράσματα και Μελλοντική Εργασία	55
6.1	Σύνοψη Εργασίας.....	55
6.2	Κύρια Ευρήματα.....	55
6.2.1	Βελτίωση Απόδοσης	55

6.2.2	Αποτελεσματικότητα Μηχανισμού Ανίχνευσης Φάσης	56
6.2.3	Σταθερότητα Μέσω Υστέρησης.....	56
6.3	Συνεισφορές	56
6.4	Περιορισμοί και Απειλές στην Εγκυρότητα (Threats to Validity)	57
6.4.1	Εσωτερική εγκυρότητα : ορθότητα/πιστότητα προσομοιωτή.....	57
6.4.2	Κατασκευαστική εγκυρότητα: μετρικές, ορισμοί και παράθυρο μέτρησης...	57
6.4.3	Εξωτερική εγκυρότητα: γενίκευση σε πραγματικές εφαρμογές/συστήματα	58
6.4.4	Συμπερασματική εγκυρότητα (Conclusion validity): στατιστική ισχύς.....	58
6.4.5	Ευαισθησία και ρύθμιση παραμέτρων (APAR/CARL)	58
6.4.6	Ενεργειακό μοντέλο (όπου χρησιμοποιείται)	59
6.5	Μελλοντική Εργασία	59
6.6	Τελικές Παρατηρήσεις.....	60
7	Επίλογος.....	61

1 Εισαγωγή

1.1 Αντικείμενο και Κίνητρα της Έρευνας

Η αδιάκοπη αναζήτηση υψηλότερης υπολογιστικής απόδοσης οδήγησε τη βιομηχανία ημιαγωγών προς όλο και πιο σύνθετες αρχιτεκτονικές επεξεργαστών πολλαπλών πυρήνων. Οι σύγχρονοι επεξεργαστές υψηλών επιδόσεων ενσωματώνουν δεκάδες έως και εκατοντάδες επεξεργαστικούς πυρήνες σε ένα μόνο ολοκληρωμένο κύκλωμα, δημιουργώντας πρωτοφανείς προκλήσεις για την υποδομή ενδοεπικοινωνίας του επεξεργαστή. Οι παραδοσιακές προσεγγίσεις διασύνδεσης, όπως οι αφοσιωμένες συνδέσεις σημείου προς σημείο και τα κοινόχρηστα διαζεύγματα (shared buses), αποδείχθηκαν ανεπαρκείς για τέτοια μαζικά παράλληλα συστήματα λόγω θεμελιωδών περιορισμών κλιμάκωσης.

Τα Ολοκληρωμένα Ενδο-συνδεόμενα Δίκτυα (ΟΕΔ), γνωστά διεθνώς ως Networks-on-Chip (NoCs), αναδύθηκαν ως επαναστατική λύση για την αντιμετώπιση αυτών των προκλήσεων διασύνδεσης. Η έννοια παρουσιάστηκε επίσημα για πρώτη φορά από τους Dally και Towles [1], όπου προτάθηκε η αντικατάσταση της παραδοσιακής καλωδίωσης με ένα δίκτυο μεταγωγής πακέτων ενσωματωμένο μέσα στο ίδιο το τσιπ. Οι Benini και De Micheli [2] χαρακτήρισαν τα NoC ως ένα νέο παράδειγμα σχεδιασμού System-on-Chip. Η προσέγγιση αυτή προσφέρει πολλά πλεονεκτήματα, μεταξύ των οποίων η εγγενής κλιμακωσιμότητα για υποστήριξη αυξανόμενου αριθμού πυρήνων, ο αρθρωτός σχεδιασμός που επιτρέπει συστηματική επαλήθευση, η βελτιωμένη ενεργειακή απόδοση μέσω δομημένης επικοινωνίας, καθώς και η ενισχυμένη ανοχή σε σφάλματα μέσω εναλλακτικών διαδρομών [5].

Ωστόσο, η υιοθέτηση αρχιτεκτονικών NoC εισάγει νέες προκλήσεις, ιδιαίτερα στον τομέα των αλγορίθμων δρομολόγησης [6]. Ο αλγόριθμος δρομολόγησης καθορίζει τον τρόπο με τον οποίο τα πακέτα διασχίζουν το δίκτυο από την πηγή στον προορισμό, επηρεάζοντας άμεσα κρίσιμες μετρικές απόδοσης όπως η καθυστέρηση, η διεκπεραιωτική ικανότητα και η κατανάλωση ενέργειας. Ενώ οι απλοί ντετερμινιστικοί αλγόριθμοι, όπως η δρομολόγηση XY [5], παρέχουν εγγυημένη απαλλαγή από αδιέξοδα με ελάχιστο υπολογιστικό κόστος, δεν μπορούν να προσαρμοστούν σε δυναμικές συνθήκες κυκλοφορίας. Οι προσαρμοστικοί αλγόριθμοι, όπως ο Odd-Even [4] και ο West-First [3], προσπαθούν να ξεπεράσουν αυτόν τον περιορισμό, αλλά εισάγουν πρόσθετη πολυπλοκότητα.

1.2 Διατύπωση του Προβλήματος

Το κεντρικό πρόβλημα που αντιμετωπίζει η παρούσα Διπλωματική Εργασία του είναι ο σχεδιασμός και η υλοποίηση ενός προσαρμοστικού αλγορίθμου δρομολόγησης που ξεπερνά τους περιορισμούς των υφιστάμενων προσεγγίσεων, διατηρώντας παράλληλα πρακτική δυνατότητα υλοποίησης. Συγκεκριμένα, η εργασία επιδιώκει να απαντήσει στα ακόλουθα ερευνητικά ερωτήματα.

Τα ερευνητικά ερωτήματα της εργασίας διατυπώνονται ως εξής:

1. Πώς μπορεί η πληροφορία της τρέχουσας **φάσης συμφόρησης** (χαμηλή/μέτρια/υψηλή) να ενσωματωθεί στις αποφάσεις δρομολόγησης με τρόπο που να παραμένει πρακτικά υλοποιήσιμος (δηλ. με μικρό υπολογιστικό/υλοποιητικό κόστος ανά δρομολογητή);
2. Μπορεί να σχεδιαστεί ένα υβριδικό σχήμα που να εναλλάσσεται ομαλά μεταξύ πολιτικών δρομολόγησης ανά φάση, χωρίς ταλαντώσεις, μέσω μηχανισμού υστέρησης;
3. Πώς επηρεάζει η παραπάνω προσέγγιση τη διεκπεραιωτική ικανότητα (throughput) και το peak throughput σε σύγκριση με καθιερωμένους αλγορίθμους αναφοράς, υπό διαφορετικά πρότυπα κυκλοφορίας;
4. Πώς επηρεάζει η παραπάνω προσέγγιση την καθυστέρηση (latency) και τη συμπεριφορά κοντά στον κορεσμό, σε σύγκριση με τους αλγορίθμους αναφοράς;
5. Πώς μεταβάλλεται η απόδοση (throughput/latency) όταν αλλάζει το μέγεθος δικτύου (4×4 , 8×8 , 16×16) και ποια είναι η κλιμακωσιμότητα της προσέγγισης;

1.3 Στόχοι της Έρευνας

Ο πρωταρχικός στόχος της παρούσας διπλωματικής εργασίας είναι η ανάπτυξη του APAR (Adaptive Phase-Aware Routing), ενός νέου υβριδικού προσαρμοστικού αλγορίθμου δρομολόγησης για αρχιτεκτονικές NoC δισδιάστατου πλέγματος. Οι επιμέρους στόχοι περιλαμβάνουν τον σχεδιασμό ενός μηχανισμού ανίχνευσης φάσης συμφόρησης που ταξινομεί την κατάσταση λειτουργίας του δικτύου σε χαμηλή, μέτρια και υψηλή συμφόρηση με βάση την πληρότητα ενταμιευτών. Επιπλέον, στοχεύεται η ανάπτυξη πολλαπλών πολιτικών δρομολόγησης βελτιστοποιημένων για κάθε φάση, καθώς και η υλοποίηση ενός μηχανισμού ομαλής εναλλαγής μεταξύ πολιτικών μέσω υστέρησης.

Παράλληλα, διεξάγεται πειραματική αξιολόγηση συγκρίνοντας τον APAR με αλγόριθμους αναφοράς (βλ. Κεφάλαιο 3) σε πολλαπλά πρότυπα κυκλοφορίας και μεγέθη δικτύου. Τέλος, τεκμηριώνεται ότι ο APAR δεν εισάγει νέες κυκλικές εξαρτήσεις (και άρα διατηρεί την απαλλαγή από αδιέξοδα) όταν οι μη-ντετερμινιστικές επιλογές του περιορίζονται σε κατευθύνσεις επιτρεπτές από το μοντέλο στροφών Odd-Even [4].

1.4 Συνεισφορές της Εργασίας

Η παρούσα Διπλωματική Εργασία του συνεισφέρει στον τομέα του σχεδιασμού ΟΕΔ με τους ακόλουθους τρόπους. Πρώτον, προτείνεται ο αλγόριθμος APAR (Adaptive Phase-Aware Routing), ο οποίος συνδυάζει ανίχνευση φάσης συμφόρησης με βάση την πληρότητα ενταμιευτών (buffer-based phase detection) και δυναμική επιλογή πολιτικής δρομολόγησης ανά φάση. Η σχετική βιβλιογραφία και οι συναφείς προσεγγίσεις ανασκοπούνται στο Κεφάλαιο 2, όπου αποσαφηνίζεται η θέση του APAR σε σχέση με καθιερωμένες και πιο πρόσφατες μεθόδους.

Δεύτερον, υλοποιείται ένα πειραματικό πλαίσιο προσομοίωσης ακριβείας κύκλου (cycle-accurate simulation framework) για δισδιάστατα πλέγματα, στο οποίο ενσωματώνονται τόσο ο APAR όσο και οι αλγόριθμοι αναφοράς, ώστε να εξασφαλίζεται δίκαιη σύγκριση κάτω από κοινές παραμέτρους. Τρίτον, παρουσιάζονται πειραματικά αποτελέσματα για 7 πρότυπα κυκλοφορίας, 18 ρυθμούς έγχυσης και 3 μεγέθη δικτύου, τα οποία επιτρέπουν συζήτηση τόσο των περιπτώσεων όπου ο APAR υπερέχει όσο και των περιπτώσεων όπου οι κλασικές μέθοδοι (π.χ. XY) παραμένουν ισχυρές.

1.5 Σημασία της Έρευνας

Η σημασία της παρούσας έρευνας εκτείνεται σε πολλαπλά επίπεδα. Σε θεωρητικό επίπεδο, η εργασία εισάγει την έννοια της **επίγνωσης φάσης συμφόρησης** στη δρομολόγηση NoC, αναγνωρίζοντας ότι τα δίκτυα εμφανίζουν διακριτές καταστάσεις λειτουργίας (χαμηλή/μέτρια/υψηλή συμφόρηση) με διαφορετικές απαιτήσεις σε καθυστέρηση και διεκπεραιωτική ικανότητα. Η αξιοποίηση αυτής της πληροφορίας επιτρέπει έναν ελεγχόμενο συμβιβασμό μεταξύ απλότητας (ντετερμινιστική δρομολόγηση) και προσαρμοστικότητας (Odd-Even constrained επιλογές) ανάλογα με τις επικρατούσες συνθήκες.

Σε πρακτικό επίπεδο, ο APAR προσφέρει μια υλοποιήσιμη λύση που μπορεί να ενσωματωθεί σε υπάρχουσες αρχιτεκτονικές NoC με ελάχιστες τροποποιήσεις. Η πρόσθετη πολυπλοκότητα υλικού περιορίζεται σε λίγους καταχωρητές και συγκριτές ανά δρομολογητή, καθιστώντας την προσέγγιση οικονομικά βιώσιμη.

Σε επίπεδο βιομηχανίας, καθώς οι πολυπύρρηνοι επεξεργαστές συνεχίζουν να αυξάνουν τον αριθμό των πυρήνων τους, η ανάγκη για αποδοτικές υποδομές διασύνδεσης γίνεται όλο και πιο κρίσιμη. Επεξεργαστές όπως ο Intel Xeon Phi με 72 πυρήνες και ερευνητικά πρωτότυπα με εκατοντάδες πυρήνες καθιστούν επιτακτική την ανάπτυξη βελτιωμένων αλγορίθμων δρομολόγησης.

1.6 Δομή της Εργασίας

Το υπόλοιπο της διπλωματικής εργασίας οργανώνεται ως εξής. Το Κεφάλαιο 2 παρέχει το θεωρητικό υπόβαθρο των αρχιτεκτονικών NoC, των αλγορίθμων δρομολόγησης και των σχετικών εργασιών, μαζί με ολοκληρωμένη ανασκόπηση της υπάρχουσας βιβλιογραφίας. Παρουσιάζεται επίσης σύγκριση με σχετικές προσεγγίσεις όπως ο DyAD και congestion-aware μέθοδοι. Το Κεφάλαιο 3 περιγράφει τη μεθοδολογία, το πλαίσιο προσομοίωσης, την πειραματική διάταξη με **5670** πειράματα και τις μετρικές αξιολόγησης. Το Κεφάλαιο 4 παρουσιάζει τον λεπτομερή σχεδιασμό του αλγορίθμου APAR, συμπεριλαμβανομένων του μηχανισμού ανίχνευσης φάσης, των πολιτικών δρομολόγησης ανά φάση και του μηχανισμού υστέρησης για ομαλές μεταβάσεις. Το Κεφάλαιο 5 παρουσιάζει τα πειραματικά αποτελέσματα από τη σύγκριση πέντε αλγορίθμων (XY, West-First, Odd-Even, CARL, APAR) σε τρία μεγέθη πλέγματος. Τέλος, το Κεφάλαιο 6 συνοψίζει τις συνεισφορές, συζητά τους περιορισμούς και προσδιορίζει κατευθύνσεις για μελλοντική έρευνα.

2 Θεωρητικό Υπόβαθρο και Σχετικές Εργασίες

2.1 Θεμελιώδεις Έννοιες Ολοκληρωμένων Ενδο-συνδεδεμένων Δικτύων

2.1.1 Εξέλιξη των Ενδοεπικοινωνιακών Υποδομών

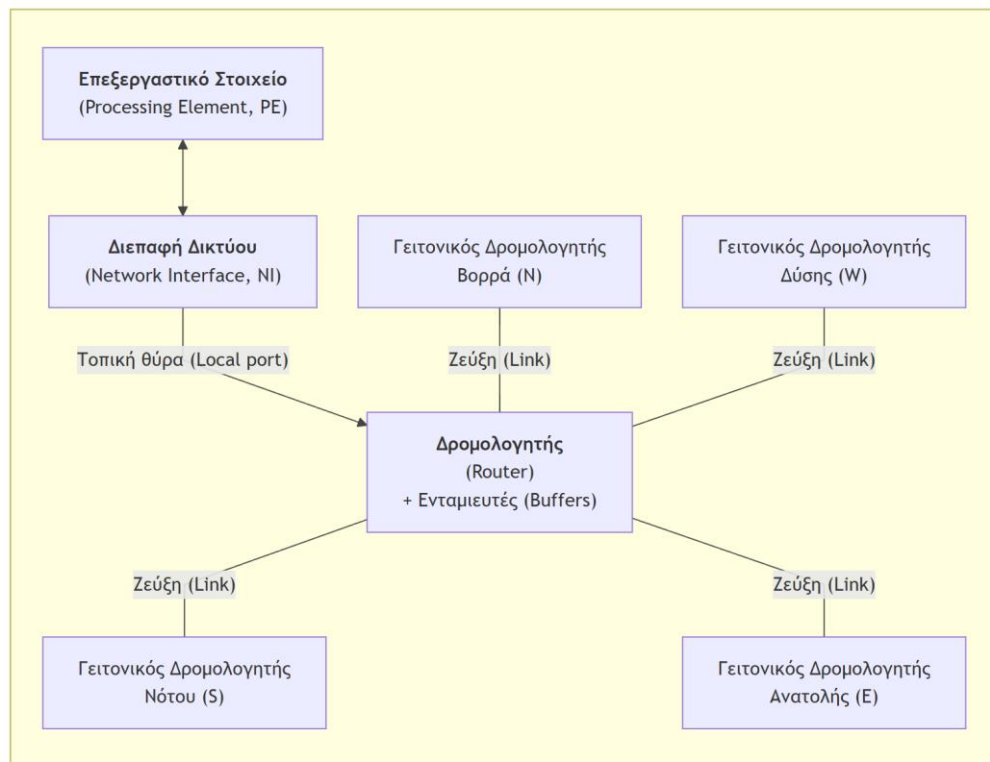
Η εξέλιξη της επικοινωνίας εντός των ολοκληρωμένων κυκλωμάτων διήλθε από διακριτές φάσεις. Οι πρώτοι μικροεπεξεργαστές βασίζονταν σε απλές συνδέσεις σημείου προς σημείο μεταξύ λειτουργικών μονάδων. Καθώς η πολυπλοκότητα των τσιπ αυξανόταν, εμφανίστηκαν οι αρχιτεκτονικές κοινόχρηστων διαζευγμάτων ως πιο κλιμακώσιμη λύση, επιτρέποντας σε πολλαπλές συνιστώσες να επικοινωνούν μέσω ενός κοινού μέσου. Ωστόσο, τα κοινόχρηστα διαζεύγματα παρουσιάζουν θεμελιώδεις περιορισμούς, καθώς το εύρος ζώνης δεν κλιμακώνεται με τον αριθμό των συνδεδεμένων συσκευών και το υπολογιστικό κόστος διαιτησίας αυξάνεται με το μέγεθος του συστήματος.

Το παράδειγμα των Ολοκληρωμένων Ενδο-συνδεδεμένων Δικτύων αντιπροσωπεύει μια θεμελιώδη μετατόπιση από τις παραδοσιακές προσεγγίσεις [2]. Αντί να αντιμετωπίζεται η διασύνδεση ως δευτερεύουσα μελέτη, τα NoC αναβαθμίζουν την υποδομή επικοινωνίας σε πρωτεύοντα σχεδιαστικό παράγοντα. Η βασική διαπίστωση, όπως διατυπώθηκε από τους Dally και Towles [1], είναι ότι η δρομολόγηση πακέτων μέσω ενός δομημένου δικτύου προσφέρει ανώτερη κλιμακωσιμότητα σε σύγκριση με τη δρομολόγηση καλωδίων μέσω μιας ad-hoc υποδομής διασύνδεσης.

2.1.2 Αρχιτεκτονικές Συνιστώσες NoC

Ένα τυπικό NoC περιλαμβάνει αρκετές βασικές συνιστώσες [5], [6]. Οι δρομολογητές λειτουργούν ως τα στοιχεία μεταγωγής που προωθούν τα πακέτα μέσω του δικτύου. Κάθε δρομολογητής συνδέεται με γειτονικούς δρομολογητές μέσω θυρών εισόδου και εξόδου, και με το τοπικό επεξεργαστικό στοιχείο μέσω μιας τοπικής θύρας. Ένας τυπικός δρομολογητής δισδιάστατου πλέγματος διαθέτει πέντε θύρες, που αντιστοιχούν στις κατευθύνσεις Βορράς, Νότος, Ανατολή, Δύση και Τοπική.

Στο Διάγραμμα 1 παρουσιάζεται συνοπτικά η τυπική αρχιτεκτονική ενός OED και η σχέση μεταξύ επεξεργαστικών στοιχείων, διεπαφών δικτύου, δρομολογητών και ζεύξεων.



Διάγραμμα 1: Αρχιτεκτονική ΟΕΔ (NoC) με επεξεργαστικό στοιχείο (Processing Element, PE), διεπαφή δικτύου (Network Interface, NI), δρομολογητές (routers), ζεύξεις (links) και ενταμιευτές (buffers). Πηγή: Ιδία επεξεργασία.

Οι ζεύξεις αποτελούν τα φυσικά κανάλια που συνδέουν γειτονικούς δρομολογητές [5]. Οι παράμετροι της ζεύξης, συμπεριλαμβανομένου του πλάτους, της συχνότητας και του βάθους προσωρινής αποθήκευσης, επηρεάζουν σημαντικά την απόδοση και την κατανάλωση ενέργειας του δικτύου [20]. Οι μονάδες διεπαφής δικτύου (Network Interface, NI) γεφυρώνουν την επικοινωνία των επεξεργαστικών στοιχείων με το NoC: αναλαμβάνουν την πακετοποίηση/αποπακετοποίηση (packetization/depacketization), την προσωρινή αποθήκευση και την προσαρμογή μεταξύ του πρωτοκόλλου επιπέδου συναλλαγών των υποσυστημάτων και του πρωτοκόλλου πακέτων/flits του δικτύου [6]. Οι ενταμιευτές αποθηκεύουν πακέτα που αναμένουν μετάδοση σε κάθε θύρα δρομολογητή, οργανωμένοι τυπικά ως ουρές First-In-First-Out [23].

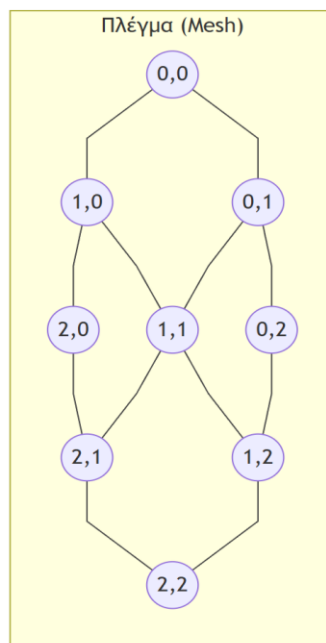
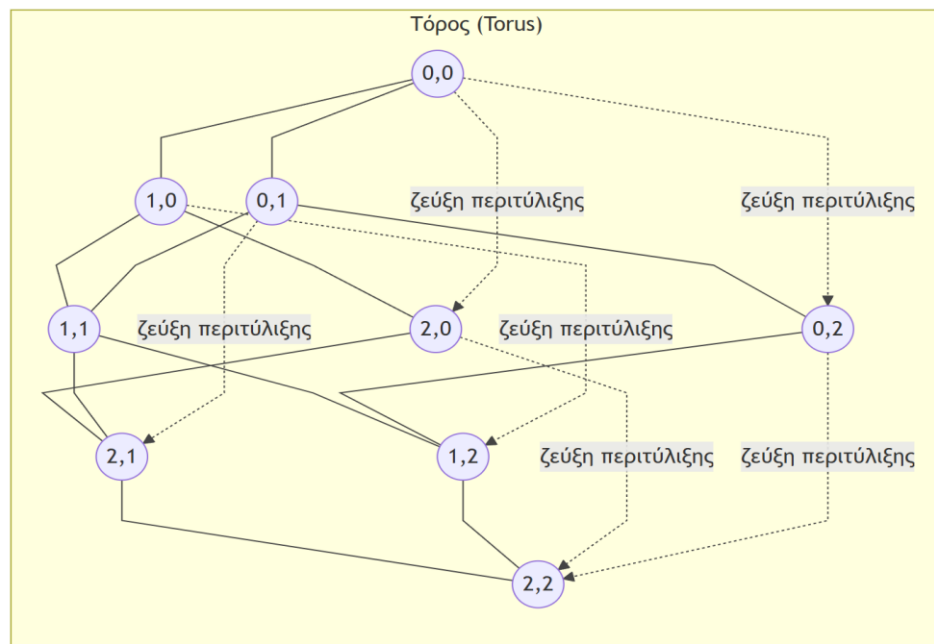
2.1.3 Τοπολογίες Δικτύου

Η τοπολογία του δικτύου καθορίζει τον τρόπο διασύνδεσης των δρομολογητών [5], [17]. Οι συνηθέστερες τοπολογίες περιλαμβάνουν το δισδιάστατο πλέγμα, όπου οι δρομολογητές

διατάσσονται σε κανονικό πλέγμα με συνδέσεις προς τους άμεσους γείτονες [16]. Η τοπολογία πλέγματος προσφέρει απλότητα, κανονική δομή και καλή κλιμακωσιμότητα. Για ένα πλέγμα $N \times N$, η διάμετρος του δικτύου είναι $2(N-1)$ βήματα.

Η τοπολογία τόρου (torus) είναι παρόμοια με το πλέγμα αλλά διαθέτει συνδέσεις περιτύλιξης στα όρια, μειώνοντας τη διάμετρο και παρέχοντας ομοιόμορφη ποικιλία διαδρομών. Η υλοποίηση είναι πιο σύνθετη λόγω των μεγάλων ζεύξεων περιτύλιξης. Ο δακτύλιος αποτελεί απλή τοπολογία κατάλληλη για μικρά συστήματα αλλά με περιορισμένη κλιμακωσιμότητα λόγω αυξανόμενης διαμέτρου. Πρόσφατες εργασίες έχουν προτείνει βελτιωμένες τοπολογίες όπως τα Express Virtual Channels [18] για μείωση της καθυστέρησης.

Στο Διάγραμμα 2 απεικονίζεται ενδεικτικά η δομική διαφορά μεταξύ πλέγματος (mesh) και τόρου (torus), όπου στον τόρο οι ζεύξεις περιτύλιξης συνδέουν τα αντίθετα άκρα.



Σημείωση: οι
διακεκομμένες ζεύξεις
δηλώνουν
ζεύξεις περιτύλιξης
(wrap-around links).

Διάγραμμα 2: Ενδεικτική απεικόνιση δισδιάστατου πλέγματος (mesh) και τόρου (torus). Οι διακεκομμένες ζεύξεις δηλώνουν συνδέσεις περιτύλιξης (wrap-around). Πηγή: Ιδία επεξεργασία.

Η παρούσα εργασία εστιάζει στην τοπολογία δισδιάστατου πλέγματος λόγω της ευρείας υιοθέτησής της σε εμπορικούς πολυπύρηνους επεξεργαστές και της καταλληλότητάς της για κανονική χωροθέτηση.

Στον Πίνακα 1 συνοψίζονται βασικά χαρακτηριστικά τοπολογιών που επηρεάζουν άμεσα την επιλογή τους (διάμετρος, βαθμός κόμβου και πολυπλοκότητα καλωδίωσης).

Πίνακας 1: Σύγκριση βασικών τοπολογιών NoC (ενδεικτικά χαρακτηριστικά)

Τοπολογία	Βαθμός κόμβου	Διάμετρος (ενδεικτικά)	Πλεονεκτήματα	Μειονεκτήματα
2D Πλέγμα (Mesh)	2–4 (εσωτ. 4)	$2(N - 1)$	Απλή διάταξη, κανονική γεωμετρία, σχετικά εύκολη χωροθέτηση και καλωδίωση	Μεγαλύτερη διάμετρος από τόρο, ανομοιόμορφη συμφόρηση στο κέντρο
Τόρος (Torus)	4	N	Μικρότερη διάμετρος, περισσότερες εναλλακτικές διαδρομές	Ζεύξεις περιτύλιξης (wrap-around) με αυξημένο μήκος/κόστος
Δακτύλιος (Ring)	2	$\lfloor N^2/2 \rfloor$	Απλή υλοποίηση για μικρά συστήματα	Μεγάλη διάμετρος, χαμηλή κλιμακωσιμότητα

2.1.4 Δομή Πακέτων και Έλεγχος Ροής

Η μετάδοση δεδομένων σε NoC ακολουθεί ένα στρωματοποιημένο πρωτόκολλο. Στο υψηλότερο επίπεδο, τα μηνύματα διαιρούνται σε πακέτα, κάθε ένα από τα οποία περιέχει πληροφορίες δρομολόγησης στην κεφαλίδα και ωφέλιμο φορτίο. Τα πακέτα μπορεί να υποδιαιρεθούν περαιτέρω σε flits, που αποτελούν τη μικρότερη μονάδα ελέγχου ροής.

Οι συνήθεις τύποι flit περιλαμβάνουν το head flit που περιέχει πληροφορίες δρομολόγησης με τις διευθύνσεις πηγής και προορισμού, τα body flits που μεταφέρουν δεδομένα ωφέλιμου

φορτίου, και το tail flit που σηματοδοτεί το τέλος του πακέτου και απελευθερώνει τους δεσμευμένους πόρους.

Οι μηχανισμοί ελέγχου ροής διέπουν την κατανομή πόρων και τη μεταφορά δεδομένων. Ο έλεγχος ροής βασισμένος σε πιστώσεις κατανέμει πιστώσεις ενταμιευτή στους ανοδικούς δρομολογητές, επιτρέποντας την αποστολή flit μόνο εφόσον υπάρχουν επαρκείς πιστώσεις και αποτρέποντας την υπερχείλιση ενταμιευτή. Τα εικονικά κανάλια επιτρέπουν σε πολλαπλά λογικά κανάλια να μοιράζονται μια φυσική ζεύξη, βελτιώνοντας τη διεκπεραιωτική ικανότητα επιτρέποντας στα πακέτα να παρακάμπτουν μπλοκαρισμένα πακέτα.

2.2 Αλγόριθμοι Δρομολόγησης

2.2.1 Ταξινόμηση Αλγορίθμων Δρομολόγησης

Οι αλγόριθμοι δρομολόγησης καθορίζουν τη διαδρομή που ακολουθούν τα πακέτα μέσω του δικτύου. Ταξινομούνται κατά μήκος πολλών διαστάσεων. Η πρώτη διάκριση αφορά τους ντετερμινιστικούς έναντι των προσαρμοστικών αλγορίθμων. Οι ντετερμινιστικοί αλγόριθμοι επιλέγουν πάντα την ίδια διαδρομή για ένα δεδομένο ζεύγος πηγής-προορισμού, ενώ οι προσαρμοστικοί αλγόριθμοι μπορούν να επιλέξουν μεταξύ πολλαπλών διαδρομών βάσει των συνθηκών του δικτύου.

Η δεύτερη διάκριση αφορά τους ελάχιστους έναντι των μη ελάχιστων αλγορίθμων. Οι ελάχιστοι αλγόριθμοι χρησιμοποιούν μόνο συντομότερες διαδρομές, ενώ οι μη ελάχιστοι αλγόριθμοι μπορεί να χρησιμοποιήσουν μακρύτερες διαδρομές για αποφυγή συμφόρησης ή σφαλμάτων. Η τρίτη διάκριση αφορά τη δρομολόγηση πηγής έναντι της κατανεμημένης. Η δρομολόγηση πηγής υπολογίζει την πλήρη διαδρομή κατά τον χρόνο έγχυσης, ενώ η κατανεμημένη δρομολόγηση λαμβάνει αποφάσεις βήμα προς βήμα σε κάθε δρομολογητή.

2.2.2 Αδιέξοδα και το Μοντέλο Στροφών

Μια κρίσιμη μελέτη στον σχεδιασμό αλγορίθμων δρομολόγησης είναι η αποφυγή αδιεξόδων. Ένα αδιέξοδο συμβαίνει όταν ένα σύνολο πακέτων σχηματίζει μια κυκλική εξάρτηση, με κάθε πακέτο να αναμένει πόρους που κατέχει κάποιο άλλο. Οι αναγκαίες συνθήκες για αδιέξοδο είναι ο αμοιβαίος αποκλεισμός (οι πόροι δεν μπορούν να μοιραστούν), η κατοχή και αναμονή (τα πακέτα κατέχουν πόρους ενώ αναμένουν άλλους),

η απουσία προεκτόπισης (οι πόροι δεν μπορούν να απελευθερωθούν βίαια), και η κυκλική αναμονή (υπάρχει κύκλος στο γράφημα αναμονής).

Το Μοντέλο Στροφών, που εισήχθη από τους Glass και Ni [3], παρέχει ένα κομψό πλαίσιο για τον σχεδιασμό προσαρμοστικών αλγορίθμων δρομολόγησης χωρίς αδιέξοδα. Η βασική διαπίστωση είναι ότι τα αδιέξοδα απαιτούν κύκλους στο γράφημα εξάρτησης καναλιών, και οι κύκλοι μπορούν να σπάσουν απαγορεύοντας συγκεκριμένες στροφές (αλλαγές κατεύθυνσης 90°).

Για ένα δισδιάστατο πλέγμα, υπάρχουν οκτώ πιθανές στροφές. Το μοντέλο στροφών προσδιορίζει ότι η απαγόρευση τουλάχιστον δύο στροφών σε κάθε μία από τις τέσσερις πιθανές κατευθύνσεις κύκλου είναι επαρκής για να εγγυηθεί την απαλλαγή από αδιέξοδα.

2.2.3 Δρομολόγηση XY

Η δρομολόγηση XY αποτελεί τον απλούστερο ντετερμινιστικό αλγόριθμο για δισδιάστατα πλέγματα [5]. Ο αλγόριθμος δρομολογεί πρώτα κατά μήκος του άξονα X μέχρι να φτάσει στη στήλη του προορισμού, και στη συνέχεια κατά μήκος του άξονα Y μέχρι τη γραμμή του προορισμού. Αυτή η διαδικασία εγγυάται συντομότερη διαδρομή και απαλλαγή από αδιέξοδα, καθώς δεν επιτρέπονται στροφές από Y σε X. Ωστόσο, η δρομολόγηση XY δεν μπορεί να προσαρμοστεί σε συμφόρηση, καθώς η διαδρομή είναι πλήρως καθορισμένη από την πηγή και τον προορισμό.

2.2.4 Δρομολόγηση West-First

Η δρομολόγηση West-First [3] απαγορεύει στροφές που θα δρομολογούσαν δυτικά μετά τη δρομολόγηση σε άλλη κατεύθυνση. Τα πακέτα πρέπει να ολοκληρώσουν όλα τα δυτικά βήματα πριν στρίψουν σε οποιαδήποτε άλλη κατεύθυνση. Αυτός ο αλγόριθμος παρέχει μερική προσαρμοστικότητα διατηρώντας την εγγύηση απαλλαγής από αδιέξοδα.

2.2.5 Μοντέλο Στροφών Odd-Even

Το μοντέλο στροφών Odd-Even, που προτάθηκε από τον Chiu [4], επιδιώκει μεγαλύτερη προσαρμοστικότητα χωρίς να θυσιάζει την απαλλαγή από αδιέξοδα. Η λογική του είναι πρακτική: δεν απαγορεύει όλες τις εναλλακτικές διαδρομές, αλλά μόνο ένα μικρό σύνολο στροφών που θα μπορούσε να δημιουργήσει κυκλικές εξαρτήσεις καναλιών. Έτσι διατηρούνται διαθέσιμες περισσότερες ελάχιστες διαδρομές από ό,τι σε πιο αυστηρά turn-model σχήματα. Στις άρτιες στήλες απαγορεύονται οι στροφές ΒΔ (Βορράς→Δύση) και ΝΔ

(Νότος→Δύση), ενώ στις περιττές στήλες απαγορεύονται οι στροφές AB (Ανατολή→Βορράς) και AN (Ανατολή→Νότος).

2.3 Δρομολόγηση με Επίγνωση Συμφόρησης

Η καθαρά προσαρμοστική δρομολόγηση βασισμένη αποκλειστικά στη διαθεσιμότητα διαδρομών συχνά αποτυγχάνει να εξισορροπήσει αποτελεσματικά το φορτίο [9]. Η δρομολόγηση με επίγνωση συμφόρησης ενσωματώνει πληροφορίες κυκλοφορίας στις αποφάσεις δρομολόγησης. Η προσέγγιση DyXY, που προτάθηκε από τους Li και συνεργάτες [7], παρακολουθεί την πληρότητα ενταμιευτή στους γειτονικούς δρομολογητές και εναλλάσσεται μεταξύ δρομολόγησης XY και YX για αποφυγή συμφορημένων περιοχών. Οι Hu και Marculescu [8] πρότειναν τον DyAD που συνδυάζει δυναμική διαίτησία με προσαρμοστική δρομολόγηση.

Οι συναρτήσεις επιλογής καθορίζουν ποια από τις έγκυρες θύρες εξόδου να επιλεγεί. Η τυχαία επιλογή παρέχει ίση πιθανότητα για όλες τις έγκυρες εξόδους. Η επιλογή βάσει επιπέδου ενταμιευτή επιλέγει την έξοδο με τον περισσότερο διαθέσιμο χώρο ενταμιευτή. Η προσέγγιση Neighbors-on-Path εξετάζει τη συμφόρηση πολλά βήματα μπροστά [9].

2.4 Σχετικές Εργασίες στην Προσαρμοστική Δρομολόγηση

2.4.1 Προσεγγίσεις Βασισμένες σε Ενισχυτική Μάθηση

Οι Feng και συνεργάτες [10] πρότειναν τον πρώτο αλγόριθμο δρομολόγησης NoC βασισμένο σε ενισχυτική μάθηση [21], συνδυάζοντας Q-learning με δρομολόγηση εκτροπής για ανοχή σφαλμάτων. Οι Farahnakian και συνεργάτες [11] εισήγαγαν τη δρομολόγηση με επίγνωση συμφόρησης βασισμένη σε Q-learning (Q-LCAR), κωδικοποιώντας το επίπεδο συμφόρησης στο χώρο καταστάσεων. Οι Zheng και Louri [12] χρησιμοποίησαν Βαθιά Q-Networks (DQN) [22] για ενεργειακά αποδοτική δρομολόγηση. Οι Wang και Louri [13] πρότειναν το CURE, ένα ενοποιημένο πλαίσιο ενισχυτικής μάθησης που βελτιστοποιεί απόδοση, ενέργεια και αξιοπιστία.

2.4.2 Σύγκριση με State-of-the-Art Αλγορίθμους

Για την πλήρη κατανόηση της θέσης του APAR στο πεδίο, είναι χρήσιμο να αναφερθούν τόσο πρόσφατες όσο και κλασικές προσεγγίσεις προσαρμοστικής δρομολόγησης.

Ενδεικτικά, πρόσφατες προσεγγίσεις αξιοποιούν βαθιά ενισχυτική μάθηση (deep reinforcement learning) για online προσαρμοστική δρομολόγηση. Το DRLAR [26] αντιμετωπίζει την επιλογή εξόδου ως πρόβλημα διαδοχικής λήψης απόφασης, όπου ο δρομολογητής παρατηρεί την τοπική κατάσταση συμφόρησης και προσαρμόζει τη συμπεριφορά του με βάση την εμπειρία που αποκτά κατά την εκτέλεση. Στόχος του είναι να μειώσει τη συμφόρηση σε επιβαρυσμένες περιοχές χωρίς να απαιτείται χειροκίνητη διαμόρφωση ξεχωριστών κανόνων για κάθε πρότυπο κυκλοφορίας.

Το DeepNR [27], από την άλλη, χρησιμοποιεί νευρωνικό μοντέλο για να αποτιμά καλύτερα τις διαθέσιμες επιλογές διαδρομής σε πιο σύνθετες και δυναμικές καταστάσεις φόρτου. Η βασική ιδέα είναι ότι η βαθύτερη αναπαράσταση της κατάστασης του δικτύου μπορεί να οδηγήσει σε καλύτερο συμβιβασμό μεταξύ throughput και latency από ό,τι απλούστερες ευριστικές μέθοδοι.

Παράλληλα, έχουν προταθεί και νεότερες στρατηγικές επιλογής εξόδου/διαδρομής με στόχο τη μείωση της καθυστέρησης [28]. Για μια συνολική εικόνα των σύγχρονων τεχνικών, βλ. τις πρόσφατες ανασκοπήσεις για adaptive routing σε 2D NoCs [29] και για δρομολόγηση/μικροαρχιτεκτονική δρομολογητών σε 3D NoCs [30].

Ο αλγόριθμος DyAD [8] των Hu και Marculescu εισήγαγε τη δυναμική διαιτησία με προσαρμοστική δρομολόγηση, εναλλασσόμενος μεταξύ ντετερμινιστικής και προσαρμοστικής λειτουργίας βάσει του επιπέδου συμφόρησης. Ο DyAD επιτυγχάνει βελτιώσεις 15-20% στη διεκπεραιωτική ικανότητα έναντι της στατικής δρομολόγησης XY. Ωστόσο, η εναλλαγή λειτουργιών είναι απότομη χωρίς μηχανισμό υστέρησης, οδηγώντας σε πιθανές ταλαντώσεις.

Ο DBAR [14] των Ma και συνεργατών επικεντρώνεται στην υποστήριξη πολλαπλών ταυτόχρονων εφαρμογών, χρησιμοποιώντας περιφερειακή επίγνωση συμφόρησης. Αν και αποτελεσματικός για πολυπρογραμματισμό, δεν εξετάζει τις φάσεις εκτέλεσης μεμονωμένων εφαρμογών.

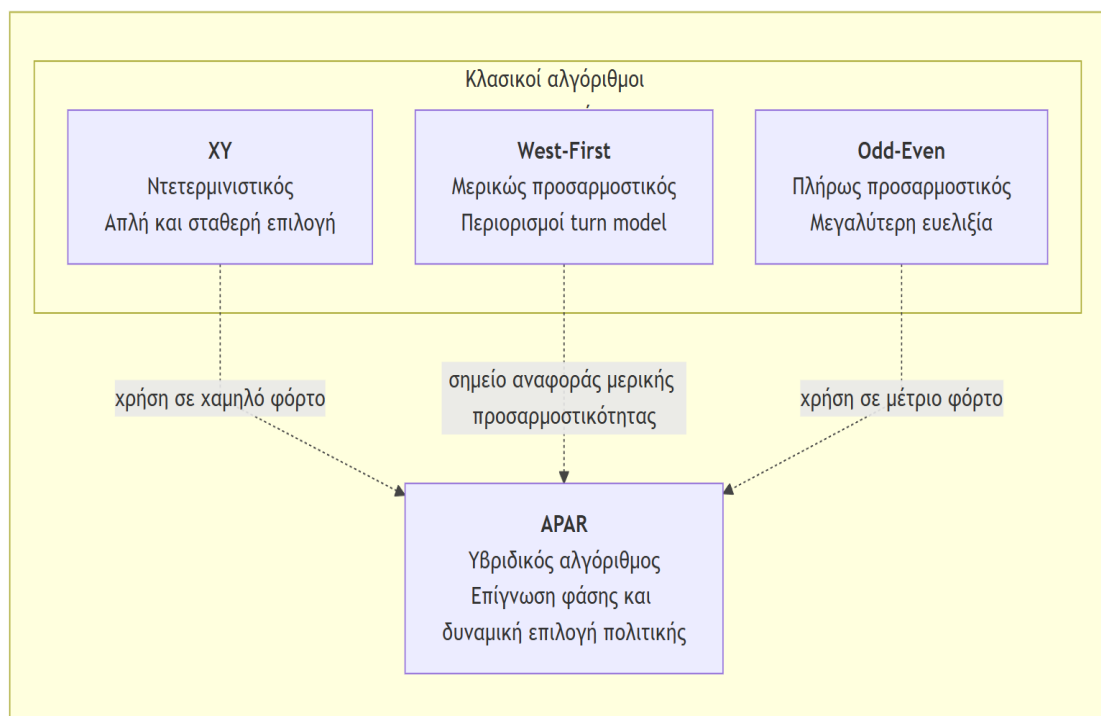
Το CURE [13] των Wang και Lourι αποτελεί ένα ενοποιημένο πλαίσιο ενισχυτικής μάθησης που βελτιστοποιεί ταυτόχρονα απόδοση, ενέργεια και αξιοπιστία. Οι αναφερόμενες

βελτιώσεις φτάνουν το 25% σε διεκπεραιωτική ικανότητα. Ωστόσο, το CURE απαιτεί σημαντικούς υπολογιστικούς πόρους για την εκπαίδευση του μοντέλου ενισχυτικής μάθησης.

Σε σύγκριση με αυτούς τους αλγορίθμους, ο APAR προσφέρει: (α) **τριφασική ανίχνευση συμφόρησης** (low/medium/high) μέσω buffer utilization, (β) μηχανισμό υστέρησης για ομαλές και σταθερές μεταβάσεις, και (γ) χαμηλότερο υπολογιστικό κόστος καθώς δεν απαιτεί εκτεταμένη εκπαίδευση όπως οι προσεγγίσεις ενισχυτικής μάθησης.

2.4.3 Κενά στην Υπάρχουσα Βιβλιογραφία

Το Διάγραμμα 3 συνοψίζει ενδεικτικά την ταξινόμηση αλγορίθμων δρομολόγησης και τη θέση του APAR ως υβριδικής προσέγγισης.



Διάγραμμα 3: Ταξινόμηση αλγορίθμων δρομολόγησης και η θέση του APAR ως υβριδικής προσέγγισης. Πηγή: Ιδία επεξεργασία.

Η ανάλυση της υπάρχουσας βιβλιογραφίας υποδεικνύει ορισμένα σημεία που αποτελούν ευκαιρίες βελτίωσης. Πρώτον, στις εργασίες που εξετάστηκαν δεν εντοπίστηκε συνδυασμός **τριφασικής ταξινόμησης συμφόρησης** με επιλογή πολιτικής και ρητό μηχανισμό

υστέρησης. Δεύτερον, πολλοί αλγόριθμοι περιορίζονται σε τοπική πληροφορία (1-hop), ενώ η περιφερειακή επίγνωση (regional awareness) εμφανίζεται πιο σπάνια. Τρίτον, αρκετά υβριδικά σχήματα εναλλάσσονται απότομα μεταξύ λειτουργιών, αντί να στοχεύουν σε ομαλές μεταβάσεις.

Αυτά τα κενά αποτέλεσαν το κίνητρο για τον σχεδιασμό του APAR, ο οποίος αντιμετωπίζει κάθε περιορισμό μέσω ανίχνευσης φάσης συμφόρησης, πολιτικών προσαρμοσμένων σε κάθε φάση και μηχανισμού υστέρησης για ομαλές μεταβάσεις.

2.5 Μετρικές Αξιολόγησης Απόδοσης

Στην παρούσα εργασία αξιολογούνται οι αλγόριθμοι δρομολόγησης με βάση ένα σύνολο καθιερωμένων μετρικών. Η καθυστέρηση (latency) εκφράζεται σε κύκλους ρολογιού προσομοίωσης (cycles), ενώ η διεκπεραιωτική ικανότητα (throughput) σε flits/node/cycle.

Πίνακας 2: Σύνοψη μετρικών αξιολόγησης

Μετρική	Μονάδα	Περιγραφή
Διεκπεραιωτική ικανότητα (Throughput)	flits/node/cycle	Flits που παραδίδονται επιτυχώς ανά κόμβο και κύκλο [5]
Καθυστέρηση (Latency)	cycles	Μέσος χρόνος από έγχυση έως παράδοση πακέτου [6]
Σημείο κορεσμού (Saturation point)	injection rate	Ρυθμός έγχυσης όπου η καθυστέρηση αυξάνεται απότομα και το throughput σταθεροποιείται [9]
Κλιμακωσιμότητα (Scalability)	—	Διατήρηση απόδοσης καθώς αυξάνει το μέγεθος πλέγματος [17]

Οι παραπάνω μετρικές ερμηνεύονται ως εξής: - **Διεκπεραιωτική ικανότητα (throughput):** βασική μετρική για ΟΕΔ· επηρεάζεται από το bottleneck των ζευξέων και από τις επιλογές δρομολόγησης. - **Καθυστέρηση (latency):** περιλαμβάνει χρόνο μετάδοσης/επεξεργασίας

και χρόνο αναμονής λόγω συμφόρησης. - **Σημείο κορεσμού (saturation point):** πρακτικός δείκτης «αντοχής» σε υψηλό φόρτο· μετά από αυτό σημειώνεται απότομη αύξηση καθυστέρησης. - **Κλιμακωσιμότητα (scalability):** εξετάζει κατά πόσο τα παραπάνω διατηρούνται όταν το πλέγμα μεγαλώνει (π.χ. $4 \times 4 \rightarrow 8 \times 8 \rightarrow 16 \times 16$).

2.6 Πρότυπα Κυκλοφορίας

Τα πρότυπα κυκλοφορίας (traffic patterns) είναι συνθετικά και ορίζουν, για κάθε πηγή, έναν μηχανισμό επιλογής προορισμού. Στον Πίνακα 3 συνοψίζονται τα πρότυπα που χρησιμοποιούνται στα πειράματα της εργασίας (βλ. Κεφάλαιο 3).

Πίνακας 3: Πρότυπα κυκλοφορίας που χρησιμοποιούνται στην αξιολόγηση

Πρότυπο	Περιγραφή / παραγωγή προορισμού	Σκοπός/χαρακτηριστικά
Uniform (ομοιόμορφο)	Τυχαίος προορισμός με ομοιόμορφη κατανομή [15]	«Γενικό» φορτίο, χωρίς έντονη δομή συμφόρησης
Transpose	$(x, y) \rightarrow (y, x)$	Διαγώνιες ροές, συμμετρική δομή
Hotspot	Με πιθανότητα p προς hotspot, αλλιώς uniform	Many-to-one συμφόρηση κοντά στο hotspot
Bit-Complement	$\text{dest} = \sim \text{src} \& (N - 1)$	Μεγάλες αποστάσεις, αυξημένη πίεση στο δίκτυο
Bit-Reverse	Αναστροφή bits της διεύθυνσης κόμβου	Δομημένες/συγχρονισμένες ροές, έντονη χωρική συσχέτιση
Neighbor	Τυχαίος άμεσος γείτονας	Τοπικές επικοινωνίες, υψηλό throughput
Tornado	$(x, y) \rightarrow ((x + N/2 - 1) \bmod N, y)$	«Λωρίδες» ροών, ανάδειξη bottlenecks

3 Μεθοδολογία

3.1 Επισκόπηση Πειραματικής Προσέγγισης

Η αξιολόγηση του προτεινόμενου αλγορίθμου APAR πραγματοποιήθηκε μέσω ενός συστηματικού πλαισίου πειραματισμού που περιλαμβάνει προσομοίωση ακριβείας κύκλου ρολογιού, πολλαπλά πρότυπα κυκλοφορίας και σύγκριση με καθιερωμένους αλγορίθμους αναφοράς. Η μεθοδολογία σχεδιάστηκε ώστε να παρέχει αξιόπιστα και αναπαραγώγιμα αποτελέσματα που επιτρέπουν δίκαιη σύγκριση μεταξύ των εξεταζόμενων αλγορίθμων.

Η πειραματική διαδικασία ακολούθησε τέσσερα διακριτά στάδια: - (1) Υλοποιήθηκε και παραμετροποιήθηκε το πειραματικό πλαίσιο προσομοίωσης. - (2) Υλοποιήθηκαν οι αλγόριθμοι δρομολόγησης που εξετάζονται στην εργασία (APAR και αλγόριθμοι αναφοράς). - (3) Σχεδιάστηκε η πειραματική μήτρα (μεγέθη πλέγματος, πρότυπα κυκλοφορίας, ρυθμοί έγχυσης) και εκτελέστηκαν οι προσομοιώσεις. - (4) Συλλέχθηκαν τα αποτελέσματα και πραγματοποιήθηκε στατιστική ανάλυση.

3.2 Πλαίσιο Προσομοίωσης

3.2.1 Αρχιτεκτονική Προσομοιωτή

Ο προσομοιωτής αναπτύχθηκε σε γλώσσα C++ με χρήση του προτύπου C++17, ακολουθώντας παρόμοια προσέγγιση με τον BookSim [19]. Η επιλογή της C++ υπαγορεύθηκε από την ανάγκη για υψηλή απόδοση εκτέλεσης, καθώς η προσομοίωση ακριβείας κύκλου ρολογιού απαιτεί σημαντικό υπολογιστικό φόρτο, ιδιαίτερα για μεγάλα δίκτυα και μακρές περιόδους προσομοίωσης.

Σημειώνεται ότι ο στόχος της εργασίας δεν είναι η ανάπτυξη ενός γενικού/βιομηχανικού προσομοιωτή, αλλά η δημιουργία ενός συνεπούς και ελεγχόμενου πειραματικού πλαισίου για συγκριτική αξιολόγηση αλγορίθμων δρομολόγησης υπό κοινές παραδοχές. Ο πηγαίος κώδικας και τα δεδομένα/artefacts της αξιολόγησης παρατίθενται στον συνοδευτικό φάκελο της εργασίας, όπως περιγράφεται στο Παράρτημα Α.

Ο προσομοιωτής οργανώνεται σε τρία βασικά υποσυστήματα. Το υποσύστημα πυρήνα περιλαμβάνει τις κλάσεις Router, Buffer, VirtualChannel, Flit και Packet που υλοποιούν τη βασική λειτουργικότητα του δικτύου. Το υποσύστημα τοπολογίας παρέχει την κλάση MeshTopology που υποστηρίζει διαμορφώσιμα πλέγματα διαστάσεων 4×4, 8×8 και 16×16.

Το υποσύστημα δρομολόγησης περιλαμβάνει τις υλοποιήσεις των αλγορίθμων XY, Odd-Even και APAR.

3.2.2 Μοντέλο Δρομολογητή

Κάθε δρομολογητής στο δίκτυο υλοποιεί ένα μοντέλο πέντε θυρών που αντιστοιχούν στις κατευθύνσεις Βορράς, Νότος, Ανατολή, Δύση και Τοπική. Κάθε θύρα διαθέτει δύο εικονικά κανάλια με βάθος ενταμιευτή τεσσάρων flits. Ο έλεγχος ροής πραγματοποιείται μέσω μηχανισμού βασισμένου σε πιστώσεις, όπου κάθε εικονικό κανάλι διατηρεί μετρητή πιστώσεων που ενημερώνεται κατά την αποστολή και λήψη flits.

Η διοχέτευση του δρομολογητή περιλαμβάνει τέσσερα στάδια [24], [25]: υπολογισμός διαδρομής (route computation), κατανομή εικονικού καναλιού (VC allocation), κατανομή διακόπτη (switch allocation) και διέλευση διακόπτη (switch traversal). Η διαιτησία στο στάδιο κατανομής διακόπτη πραγματοποιείται με αλγόριθμο round-robin για εξασφάλιση δικαιοσύνης [23].

Στον Πίνακα 4 συνοψίζονται οι βασικές παράμετροι του δικτύου και του δρομολογητή, οι οποίες διατηρούνται σταθερές σε όλα τα πειράματα.

Πίνακας 4: Βασικές παράμετροι προσομοίωσης δικτύου

Παράμετρος	Τιμή
Τοπολογία	2D Mesh (δισδιάστατο πλέγμα)
Θύρες δρομολογητή	5 (B, N, A, Δ, Τοπική)
Εικονικά κανάλια	2 VC ανά θύρα
Βάθος ενταμιευτή	4 flits ανά VC
Μέγεθος πακέτου	4 flits
Μέγεθος flit	64 bits
Έλεγχος ροής	Πιστώσεις (credit-based)
Διαιτησία	Round-robin
Στάδια διοχέτευσης	4 (route computation, VC allocation, switch allocation, switch traversal)

3.2.3 Γεννήτριες Κυκλοφορίας

Ο προσομοιωτής υποστηρίζει επτά πρότυπα κυκλοφορίας, τα οποία είναι συνθετικά (synthetic) και ορίζουν μηχανισμό επιλογής προορισμού για κάθε πηγή. Οι ορισμοί/χαρτογραφήσεις των προτύπων συνοψίζονται στο Κεφάλαιο 2 (Πίνακας 3).

- **Uniform:** ομοιόμορφη τυχαία κυκλοφορία με προορισμούς από ομοιόμορφη κατανομή.
- **Transpose:** αντιστροφή συντεταγμένων πηγής για παραγωγή διαγώνιων ροών.
- **Hotspot:** αυξημένη πιθανότητα προορισμού προς έναν hotspot κόμβο (συμφόρηση τύπου “many-to-one”).
- **Bit-Complement:** προορισμός από bitwise συμπλήρωμα της διεύθυνσης πηγής.
- **Bit-Reverse:** προορισμός από αναστροφή των bits της διεύθυνσης πηγής.
- **Neighbor:** προορισμός σε άμεσο γείτονα (τοπική κυκλοφορία).
- **Tornado:** κυκλοφορία με σταθερή μετατόπιση (pattern με έντονη χωρική δομή).

3.3 Παράμετροι Πειραμάτων

Ο Πίνακας 5 συνοψίζει την πειραματική μήτρα (factors/levels) που χρησιμοποιήθηκε στην αξιολόγηση και εξηγεί πώς προκύπτει το πλήθος των **5670** προσομοιώσεων.

Πίνακας 5: Πειραματική μήτρα και πλήθος πειραμάτων

Παράγοντας	Επίπεδα
Αλγόριθμοι	5 (XY, West-First, Odd-Even, CARL, APAR)
Μεγέθη πλέγματος	3 (4×4, 8×8, 16×16)
Πρότυπα κυκλοφορίας	7 (Uniform, Transpose, Hotspot, Bit-Complement, Bit-Reverse, Neighbor, Tornado)
Ρυθμοί έγχυσης (injection rates)	18 επίπεδα (0.01–0.50 flits/node/cycle)
Επαναλήψεις	n=3 (διαφορετικοί σπόροι)
Warmup / Measurement	200 / 2000 κύκλοι
Σύνολο	$5 \times 3 \times 7 \times 18 \times 3 = 5670$ πειράματα

Για να μειωθεί η τυχαιότητα στις συγκρίσεις, εφαρμόζεται πολιτική «κοινής τυχαιότητας» (common random numbers): για το ίδιο (*mesh, traffic, IR, run*) χρησιμοποιείται ο ίδιος seed σε όλους τους αλγορίθμους, ώστε οι συγκρίσεις να είναι ζευγαρωμένες (paired) και να μειώνεται η διακύμανση που οφείλεται στη στοχαστικότητα.

3.3.1 Διαμόρφωση Δικτύου

Τα πειράματα διεξήχθησαν σε τρία μεγέθη πλέγματος για αξιολόγηση της κλιμακωσιμότητας. Το πλέγμα 4×4 περιλαμβάνει 16 κόμβους και χρησιμοποιήθηκε για αρχική επικύρωση και ανάλυση συμπεριφοράς. Το πλέγμα 8×8 περιλαμβάνει 64 κόμβους και αποτελεί το κύριο σενάριο αξιολόγησης. Το πλέγμα 16×16 περιλαμβάνει 256 κόμβους και χρησιμοποιήθηκε για αξιολόγηση συμπεριφοράς σε μεγάλη κλίμακα.

Κάθε δρομολογητής διαθέτει δύο εικονικά κανάλια ανά θύρα με βάθος ενταμιευτή τεσσάρων flits. Το μέγεθος πακέτου ορίστηκε σε τέσσερα flits, με μέγεθος flit 64 bits. Αυτές οι παράμετροι είναι αντιπροσωπευτικές τυπικών σχεδιασμών NoC και επιτρέπουν σύγκριση με αποτελέσματα της βιβλιογραφίας.

3.3.2 Ρυθμοί Έγχυσης

Ο ρυθμός έγχυσης ορίζεται ως **flits/κόμβο/κύκλο** (offered load) και κυμάνθηκε από 0.01 έως 0.50 σε 18 επίπεδα: 0.01, 0.02, 0.03, 0.04, 0.05, 0.06, 0.08, 0.10, 0.12, 0.15, 0.18, 0.20, 0.25, 0.30, 0.35, 0.40, 0.45, 0.50. Το εύρος αυτό επιτρέπει την παρατήρηση τόσο της γραμμικής περιοχής (χαμηλό φορτίο) όσο και της περιοχής κορεσμού (υψηλό φορτίο).

3.3.3 Περίοδος Προσομοίωσης

Κάθε προσομοίωση χωρίζεται σε δύο φάσεις: θέρμανση (warmup) και μέτρηση (measurement). Τα στατιστικά συλλέγονται αποκλειστικά στο παράθυρο μέτρησης $[t_{warmup}, t_{warmup} + t_{meas})$, ώστε να μην επηρεάζονται από μεταβατικά φαινόμενα (transients) της αρχικοποίησης.

Μετά τη μέτρηση ακολουθεί φάση εκκένωσης/αποστράγγισης (drain), κατά την οποία δεν εγχέονται νέα πακέτα και η προσομοίωση συνεχίζει μέχρι να αδειάσει το δίκτυο ή μέχρι προκαθορισμένο χρονικό όριο (timeout). Η φάση drain επιτρέπει έλεγχο ακεραιότητας «μηδενικής απώλειας» (no-drop): τα εγχυθέντα flits πρέπει να ισούνται με τα flits που παραδόθηκαν μετά την εκκένωση.

3.4 Μετρικές Αξιολόγησης

Οι βασικοί ορισμοί των μετρικών δίνονται στο Κεφάλαιο 2 (Πίνακας 2). Στα πειράματα της παρούσας εργασίας, οι μετρικές υπολογίζονται στο παράθυρο μέτρησης και είναι οι ακόλουθες: - **Καθυστέρηση (latency, cycles)**: μέση καθυστέρηση πακέτου από την έγχυση έως την παράδοση. - **Διεκπεραιωτική ικανότητα (throughput, flits/node/cycle)**: delivered flits ανά κόμβο και κύκλο. - **Μέσος αριθμός αλμάτων (hop count, hops)**: μέσο πλήθος βημάτων/δρομολογητών που διασχίζει ένα πακέτο.

3.5 Αλγόριθμοι Αναφοράς

Για τη σύγκριση του APAR επιλέχθηκε ένα σύνολο αλγορίθμων που καλύπτει αντιπροσωπευτικά ντετερμινιστική, (μερικώς/πλήρως) προσαρμοστική και ενισχυτικής μάθησης (reinforcement learning) προσέγγιση.

Πίνακας 6: Αλγόριθμοι σύγκρισης και κριτήρια επιλογής

Αλγόριθμος	Κατηγορία	Απαλλαγή από αδιέξοδα	Λόγος επιλογής
XY	Ντετερμινιστικός	Dimension-order, χωρίς κύκλους εξάρτησης	Baseline χαμηλού κόστους/πολυπλοκότητας
West-First [3]	Μερικώς προσαρμοστικός	Turn model	Κλασικό μερικώς προσαρμοστικό σημείο αναφοράς
Odd-Even [4]	Πλήρως προσαρμοστικός	Turn model (εξαρτώμενο από θέση)	Κλασικό deadlock-free adaptive routing
CARL	RL / επίγνωση συμφόρησης	Επιλογές περιορισμένες σε επιτρεπτές στροφές (Odd-Even)	Αντιπροσωπευτικό baseline τύπου Q-learning, εμπνευσμένο από σχετικές RL προσεγγίσεις [11], [13]

Το πλήθος πειραμάτων και οι συνδυασμοί παραμέτρων προκύπτουν από τον Πίνακα 5.

Δεν συμπεριλήφθηκαν επιπλέον state-of-the-art αλγόριθμοι που απαιτούν εκτεταμένη παραμετροποίηση/εκπαίδευση ή εξειδικευμένο υποστηρικτικό υλικό (π.χ. προσεγγίσεις deep reinforcement learning όπως οι DRLAR/DeepNR [26], [27]), καθώς ο στόχος είναι μια δίκαιη και αναπαραγώγιμη σύγκριση υπό κοινό πειραματικό πλαίσιο.

3.6 Περιβάλλον Εκτέλεσης

Τα πειράματα εκτελέστηκαν σε σύστημα με **Windows 11 (build 26200)** και μεταγλωττιστή **g++ (GCC) 13.3.0 (MinGW-w64)** με υποστήριξη του προτύπου **C++17**, σε release build **-O2**. Τα scripts ανάλυσης/γραφημάτων εκτελέστηκαν με **Python 3.12.10** και τα πακέτα του `scripts/requirements.txt` (pandas 2.1.4, numpy 1.26.4, matplotlib 3.8.2). Ο χρόνος εκτέλεσης της προσομοίωσης (wall-clock) ανά πείραμα υπολογίζεται ότι κυμαίνεται από 50 έως 150 ms, ανάλογα με το μέγεθος του δικτύου και τη διάρκεια προσομοίωσης (η **καθυστέρηση δικτύου** μετράται ξεχωριστά σε κύκλους, όπως ορίζεται στην Ενότητα 3.4, «Μετρικές Αξιολόγησης»).

3.7 Έλεγχοι Εγκυρότητας Πειραματικού Πλαισίου

Για τις ανάγκες της εργασίας, ο προσομοιωτής λειτουργεί ως κοινό πειραματικό πλαίσιο. Για τον λόγο αυτό εφαρμόστηκαν έλεγχοι ακεραιότητας (integrity checks) και δοκιμές ορθής λειτουργίας που διασφαλίζουν ότι τα αποτελέσματα είναι συνεπή και συγκρίσιμα:

- Έλεγχος ακεραιότητας εκτέλεσης (no-drop / drain): μετά τη φάση εκκένωσης (drain) δεν παραμένουν flits στο δίκτυο και επαληθεύεται μηδενική απώλεια (Injected flits = Delivered flits).
- Έλεγχος προόδου (liveness): σε όλες τις διαμορφώσεις παρατηρείται συνεχής πρόοδος παράδοσης πακέτων και δεν εμφανίζονται «παγώματα» προσομοίωσης.
- Sanity check χαμηλού φόρτου: σε πολύ χαμηλούς ρυθμούς έγχυσης, η μέση καθυστέρηση προσεγγίζει την αναμενόμενη ελάχιστη τιμή με βάση την απόσταση Manhattan.

Οι παραπάνω έλεγχοι δεν αποτελούν πλήρη «πιστοποίηση» ενός γενικού simulator, αλλά είναι επαρκείς για το σκοπό της εργασίας: δίκαιη και αναπαραγώγιμη σύγκριση αλγορίθμων δρομολόγησης υπό κοινό μοντέλο.

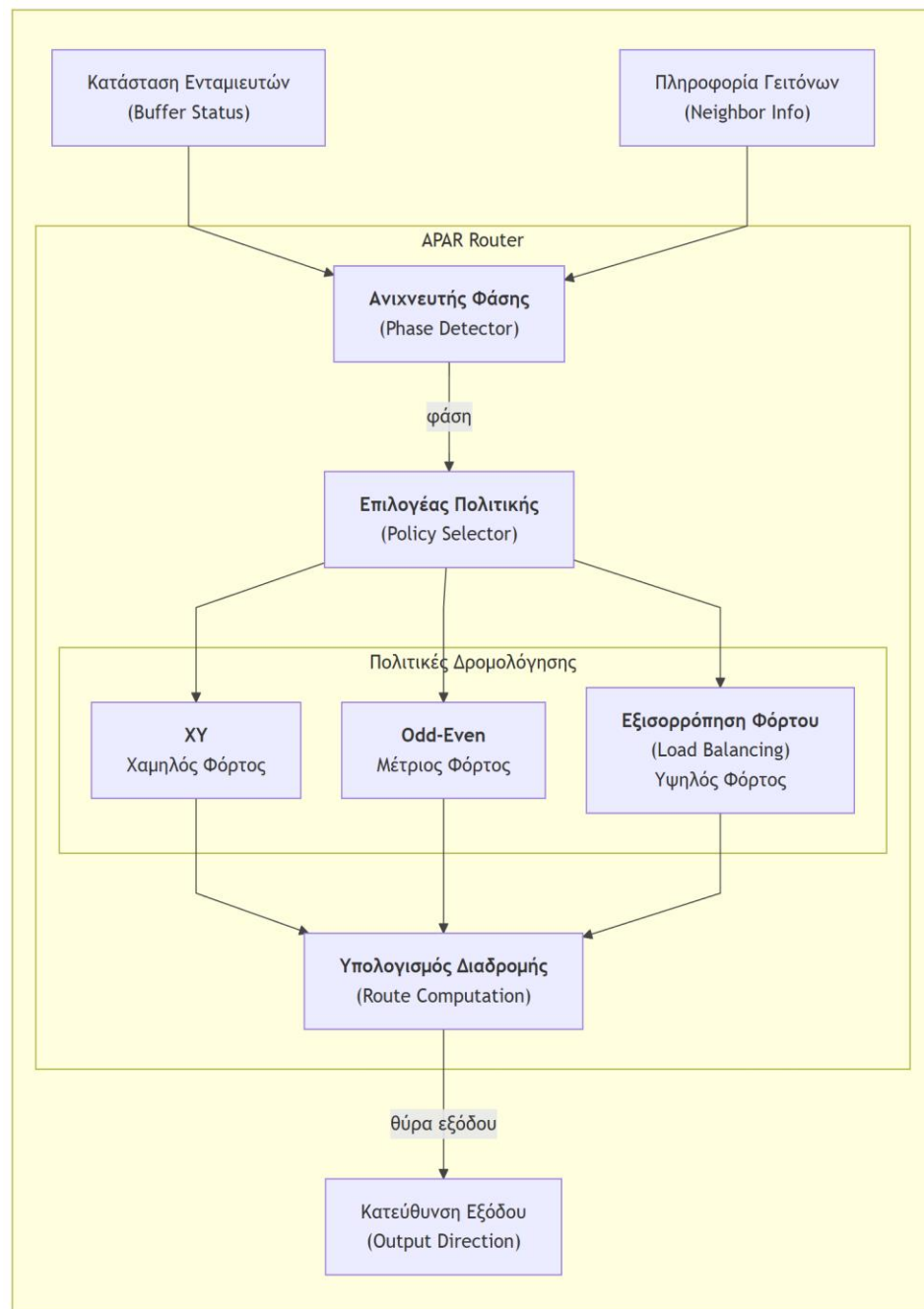
4 Ο Προτεινόμενος Αλγόριθμος APAR

4.1 Επισκόπηση Αρχιτεκτονικής

Ο αλγόριθμος APAR (Adaptive Phase-Aware Routing) αποτελεί έναν υβριδικό αλγόριθμο δρομολόγησης που συνδυάζει τα πλεονεκτήματα της ντετερμινιστικής και της προσαρμοστικής δρομολόγησης. Η βασική καινοτομία του APAR έγκειται στην ικανότητά του να ανιχνεύει τη φάση λειτουργίας του δικτύου και να προσαρμόζει δυναμικά τη στρατηγική δρομολόγησης ανάλογα με τις επικρατούσες συνθήκες.

Η αρχιτεκτονική του APAR βασίζεται σε τρία βασικά συστατικά. Το πρώτο συστατικό είναι ο μηχανισμός ανίχνευσης φάσης που παρακολουθεί τα επίπεδα συμφόρησης του δικτύου και ταξινομεί την τρέχουσα κατάσταση σε μία από τρεις φάσεις: χαμηλή συμφόρηση, μέτρια συμφόρηση και υψηλή συμφόρηση. Το δεύτερο συστατικό είναι η βιβλιοθήκη πολιτικών δρομολόγησης που περιλαμβάνει εξειδικευμένες στρατηγικές για κάθε φάση. Το τρίτο συστατικό είναι ο μηχανισμός υστέρησης που εξασφαλίζει ομαλές μεταβάσεις μεταξύ φάσεων.

Το Διάγραμμα 4 παρουσιάζει την αρχιτεκτονική του APAR σε επίπεδο δρομολογητή και τα βασικά υποσυστήματα λήψης απόφασης δρομολόγησης.



Διάγραμμα 4: Αρχιτεκτονική APAR Router — η αρχιτεκτονική του δρομολογητή APAR με τα τρία βασικά συστατικά: ανιχνευτής φάσης, επιλογέας πολιτικής και υπολογισμός διαδρομής. Πηγή: Ιδία επεξεργασία.

4.2 Μηχανισμός Ανίχνευσης Φάσης

4.2.1 Μετρικές Συμφόρησης

Η ανίχνευση φάσης βασίζεται στην παρακολούθηση της πληρότητας των ενταμιευτών σε κάθε δρομολογητή. Κάθε δρομολογητής υπολογίζει τοπικά τη μέση πληρότητα των ενταμιευτών του ως ποσοστό της συνολικής χωρητικότητας. Η τιμή αυτή ενημερώνεται σε κάθε κύκλο ρολογιού και χρησιμοποιείται για τον καθορισμό της τρέχουσας φάσης.

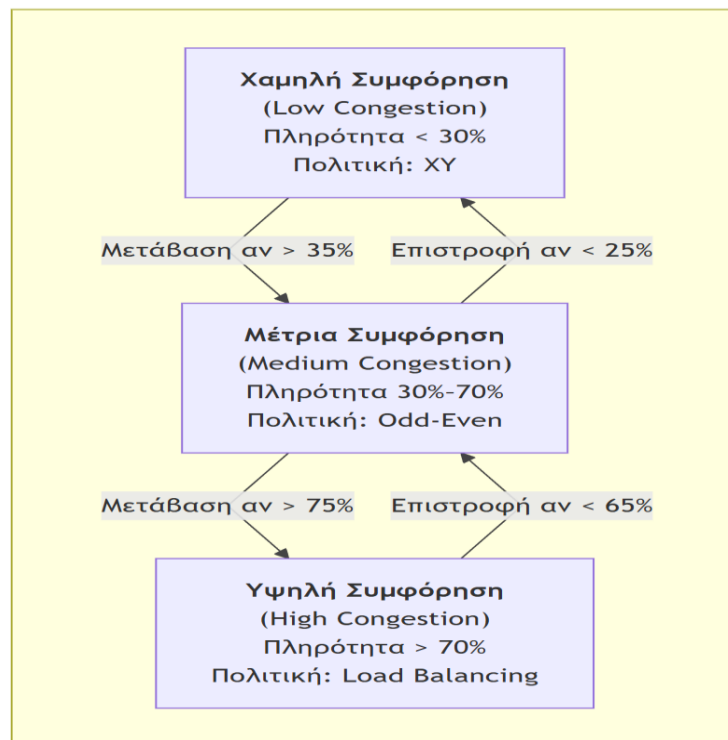
Η μετρική συμφόρησης υπολογίζεται ως ο λόγος του συνολικού αριθμού flits που βρίσκονται στους ενταμιευτές προς τη συνολική χωρητικότητα ενταμιευτών. Για δρομολογητή με πέντε θύρες, δύο εικονικά κανάλια ανά θύρα και βάθος ενταμιευτή τεσσάρων flits, η συνολική χωρητικότητα είναι 40 flits.

4.2.2 Κατώφλια Φάσεων

Ο APAR χρησιμοποιεί δύο κατώφλια για τον διαχωρισμό των τριών φάσεων. Το χαμηλό κατώφλι ορίζεται έως το 30% της χωρητικότητας ενταμιευτή (ποσοστό πληρότητας), ενώ το υψηλό κατώφλι ορίζεται στο 70%. Όταν η πληρότητα είναι κάτω από το χαμηλό κατώφλι, το δίκτυο θεωρείται σε φάση χαμηλής συμφόρησης. Όταν η πληρότητα βρίσκεται μεταξύ των δύο κατωφλίων, το δίκτυο θεωρείται σε φάση μέτριας συμφόρησης. Όταν η πληρότητα υπερβαίνει το υψηλό κατώφλι, το δίκτυο θεωρείται σε φάση υψηλής συμφόρησης.

4.2.3 Μηχανισμός Υστέρησης

Το Διάγραμμα 5 απεικονίζει τις τρεις φάσεις λειτουργίας και τη λογική μετάβασης με υστέρηση (hysteresis).



Διάγραμμα 5: Φάσεις Συμφόρησης APAR — οι τρεις φάσεις λειτουργίας του APAR και οι μεταβάσεις μεταξύ τους με μηχανισμό υστέρησης. Πηγή: Ιδία επεξεργασία.

Για την αποφυγή ταλαντώσεων μεταξύ φάσεων, ο APAR εφαρμόζει μηχανισμό υστέρησης (hysteresis) γύρω από τα κατώφλια. Πρακτικά, η μετάβαση προς *υψηλότερη* φάση απαιτεί να ξεπεραστεί το αντίστοιχο κατώφλι κατά +5%, ενώ η μετάβαση προς *χαμηλότερη* φάση απαιτεί να πέσει η πληρότητα κάτω από το κατώφλι κατά -5%.

Παράδειγμα: για κατώφλι low=30% και band=5%, η μετάβαση Low→Medium συμβαίνει όταν η πληρότητα γίνει >35%, ενώ η επιστροφή Medium→Low όταν γίνει <25%. Αντίστοιχα, για high=70% η μετάβαση Medium→High απαιτεί >75% και η επιστροφή High→Medium <65%.

Επιπλέον, εφαρμόζεται ελάχιστη διάρκεια παραμονής σε φάση (min residence time) πριν επιτραπεί νέα μετάβαση, ώστε να αποφεύγονται γρήγορες εναλλαγές όταν η συμφόρηση κυμαίνεται γύρω από τα κατώφλια.

4.3 Πολιτικές Δρομολόγησης ανά Φάση

4.3.1 Φάση Χαμηλής Συμφόρησης

Σε συνθήκες χαμηλής συμφόρησης, ο APAR εφαρμόζει **ντετερμινιστική επιλογή τύπου XY** (X-first, έπειτα Y) ώστε να ελαχιστοποιεί την επιβάρυνση (overhead) της προσαρμοστικότητας όταν το δίκτυο είναι «καθαρό».

Ωστόσο, για να διατηρείται **ενιαία εγγύηση απαλλαγής από αδιέξοδα** σε όλες τις φάσεις, η λειτουργία χαμηλής συμφόρησης υλοποιείται ως πολιτική επιλογής (selection policy) πάνω σε ένα **σταθερό σύνολο επιτρεπτών κατευθύνσεων (admissible directions)** που προκύπτει από το μοντέλο στροφών Odd-Even. Έτσι ο APAR δεν «αναμειγνύει» ανεξάρτητες συναρτήσεις δρομολόγησης (π.χ. καθαρό XY έναντι Odd-Even), αλλά διατηρεί κοινή βάση επιτρεπτότητας (admissibility) και αλλάζει μόνο το πώς επιλέγει μεταξύ ισοδύναμων ελάχιστων (minimal) επιλογών.

4.3.2 Φάση Μέτριας Συμφόρησης

Σε συνθήκες μέτριας συμφόρησης, ο APAR μεταβαίνει σε προσαρμοστική δρομολόγηση βασισμένη στο μοντέλο Odd-Even [4]. Η πολιτική αυτή επιτρέπει την επιλογή μεταξύ πολλαπλών έγκυρων διαδρομών βάσει της διαθεσιμότητας πόρων, διατηρώντας παράλληλα την εγγύηση απαλλαγής από αδιέξοδα [3].

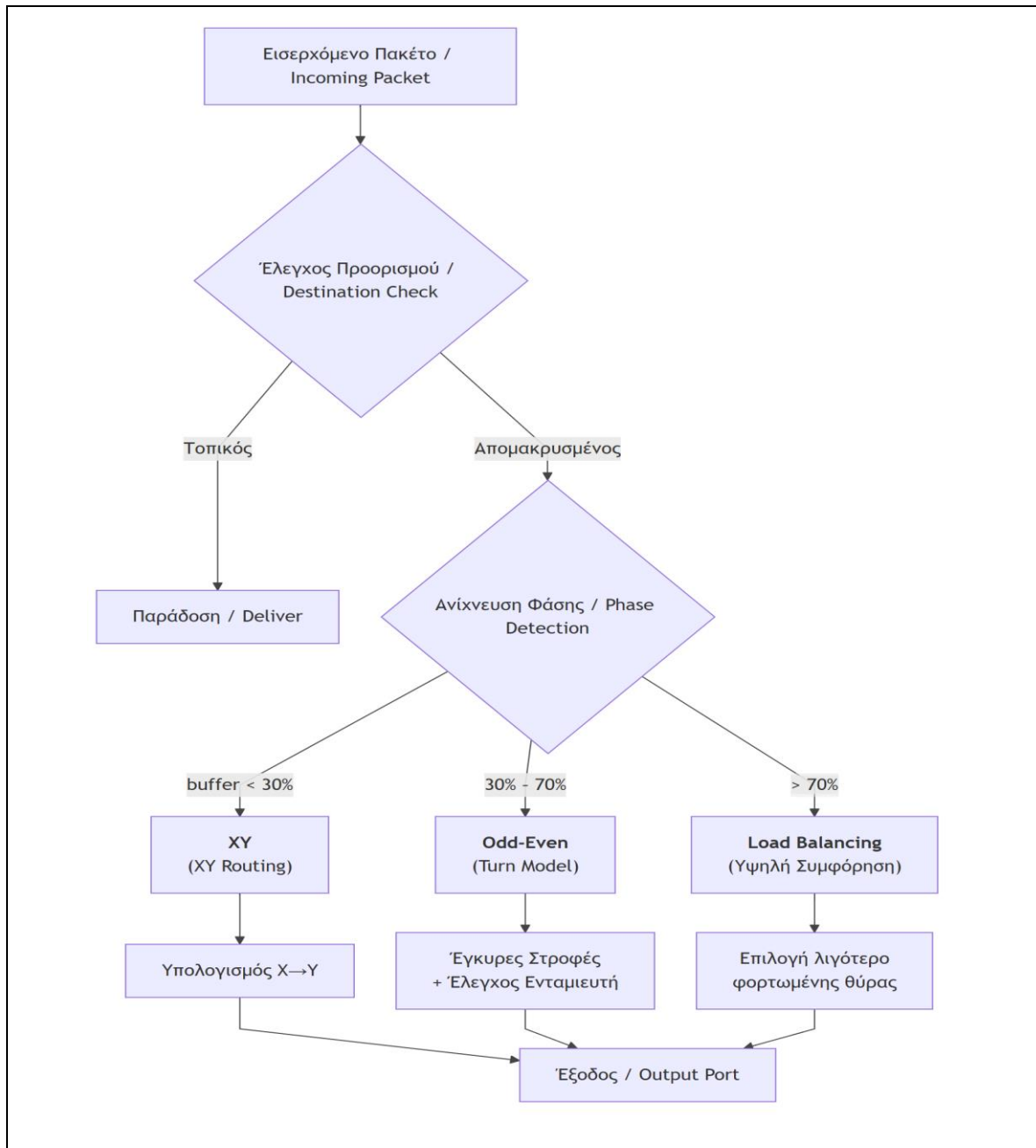
Η επιλογή μεταξύ έγκυρων εξόδων πραγματοποιείται με βάση τη διαθεσιμότητα χώρου ενταμιευτή στους γειτονικούς δρομολογητές [9]. Ο APAR επιλέγει την έξοδο που οδηγεί στον δρομολογητή με τον περισσότερο διαθέσιμο χώρο ενταμιευτή, επιτυγχάνοντας έτσι εξισορρόπηση φόρτου, παρόμοια με την προσέγγιση DBAR [14].

4.3.3 Φάση Υψηλής Συμφόρησης

Σε συνθήκες υψηλής συμφόρησης, ο APAR εφαρμόζει επιθετική εξισορρόπηση φόρτου. Η πολιτική αυτή επεκτείνει την προσαρμοστική δρομολόγηση με πρόσθετους μηχανισμούς για την αποφυγή συμφορημένων περιοχών.

Ο αλγόριθμος εξετάζει όχι μόνο τους άμεσους γείτονες αλλά και μια τοπική περιφερειακή εκτίμηση δεύτερου βαθμού (2-hop lookahead), επιλέγοντας **μεταξύ ελάχιστων επιτρεπτών κατευθύνσεων (minimal admissible directions)** εκείνη που οδηγεί σε λιγότερο συμφορημένη περιοχή. Η προσέγγιση αυτή παραμένει ελάχιστη (minimal, ίδιο hop-count), αλλά αποφεύγει να «σπρώχνει» την κίνηση προς ήδη κορεσμένους διαδρόμους.

Το Διάγραμμα 6 συνοψίζει τη ροή απόφασης του APAR από την ανίχνευση φάσης έως την τελική επιλογή θύρας εξόδου.



Διάγραμμα 6: Ροή Απόφασης Δρομολόγησης — διάγραμμα ροής της διαδικασίας λήψης απόφασης δρομολόγησης στον APAR. Πηγή: Ιδία επεξεργασία.

4.3.4 Ενδεικτικά Σενάρια Δρομολόγησης

Για να γίνει πιο συγκεκριμένος ο τρόπος λειτουργίας του APAR, παρουσιάζονται ενδεικτικά σενάρια σε δισδιάστατο πλέγμα (mesh). Σε όλα τα σενάρια, οι επιτρεπτές κατευθύνσεις

περιορίζονται από τους κανόνες Odd-Even και στη συνέχεια εφαρμόζεται η πολιτική επιλογής που αντιστοιχεί στην τρέχουσα φάση.

- **Σενάριο 1 (Φάση χαμηλής συμφόρησης – Low phase):** σε πλέγμα 4×4 , από πηγή $(0,0)$ προς προορισμό $(3,1)$ η πολιτική επιλέγει κατά προτεραιότητα Ανατολή μέχρι να επιτευχθεί ο στόχος στον άξονα X και στη συνέχεια Βορρά στον άξονα Y. Η συμπεριφορά είναι ελάχιστη (minimal) και σταθερή όταν το δίκτυο δεν παρουσιάζει συμφόρηση.
- **Σενάριο 2 (Φάση μέτριας συμφόρησης – Medium phase):** σε πλέγμα 8×8 , από $(1,1)$ προς $(3,3)$ υπάρχουν πολλαπλές ελάχιστες επιλογές (π.χ. Ανατολή ή Βορρά). Η πολιτική Medium επιλέγει την έξοδο που οδηγεί σε γειτονικό δρομολογητή με μεγαλύτερη διαθέσιμη χωρητικότητα (χαμηλότερη πληρότητα/περισσότερες πιστώσεις (credits)), ώστε να μειωθεί η πιθανότητα συμφόρησης.
- **Σενάριο 3 (Φάση υψηλής συμφόρησης – High phase):** όταν υπάρχουν δύο ελάχιστες/επιτρεπτές επιλογές, ο APAR δεν εξετάζει μόνο την άμεση κατάσταση του επόμενου δρομολογητή (1-hop), αλλά και μια περιφερειακή εκτίμηση (2-hop) της περιοχής στην οποία «οδηγεί» κάθε επιλογή. Έτσι μπορεί να αποφύγει μια επιλογή που φαίνεται καλή άμεσα αλλά κατευθύνει την κίνηση προς ήδη κορεσμένη περιοχή.

4.4 Υλοποίηση

4.4.1 Δομές Δεδομένων

Η υλοποίηση του APAR απαιτεί ελάχιστες πρόσθετες δομές δεδομένων σε σχέση με έναν τυπικό δρομολογητή. Κάθε δρομολογητής διατηρεί έναν μετρητή πληρότητας ενταμιευτή που ενημερώνεται σε κάθε κύκλο, μια μεταβλητή φάσης που υποδεικνύει την τρέχουσα φάση λειτουργίας, και έναν πίνακα με τα επίπεδα συμφόρησης των γειτονικών δρομολογητών.

4.4.2 Αλγόριθμος Υπολογισμού Διαδρομής

Ο αλγόριθμος υπολογισμού διαδρομής εκτελείται σε κάθε δρομολογητή για κάθε εισερχόμενο πακέτο. Πρώτα ελέγχεται η τρέχουσα φάση του δρομολογητή. Ανάλογα με τη φάση, επιλέγεται η κατάλληλη πολιτική δρομολόγησης. Η πολιτική επιστρέφει ένα σύνολο

έγκυρων θυρών εξόδου, από τις οποίες επιλέγεται η βέλτιστη βάση της διαθεσιμότητας πόρων.

4.4.3 Ενημέρωση Πληροφοριών Συμφόρησης

Η ανταλλαγή πληροφοριών συμφόρησης μεταξύ γειτονικών δρομολογητών πραγματοποιείται μέσω παράπλευρων σημάτων (sideband) που μεταδίδονται παράλληλα με τα δεδομένα. Κάθε δρομολογητής μεταδίδει την τρέχουσα πληρότητα ενταμιευτή του στους γείτονές του σε κάθε κύκλο ρολογιού. Αυτή η προσέγγιση εισάγει ελάχιστο υπολογιστικό κόστος ενώ παρέχει επίκαιρη πληροφόρηση για τις αποφάσεις δρομολόγησης.

Στην υλοποίηση του προσομοιωτή, η «πληρότητα» υπολογίζεται ως κλάσμα πληρότητας (occupancy fraction, 0..1) των ενταμιευτών εισόδου (input buffers) των εικονικών καναλιών (Virtual Channels, VCs) και είναι διαθέσιμη σε κάθε κύκλο για τους άμεσους γείτονες. Επιπλέον, για τις τοπικές αποφάσεις δρομολόγησης χρησιμοποιείται σήμα ανάσχεσης (backpressure) μέσω ελέγχου ροής με πιστώσεις (credit-based flow control), δηλαδή μέσω της διαθεσιμότητας downstream credits ανά output VC, ώστε η επιλογή εξόδου να λαμβάνει υπόψη την πραγματική δυνατότητα αποδοχής flits.

Η εκτίμηση 2-hop (δεύτερου βαθμού) μπορεί να υλοποιηθεί χωρίς «παγκόσμια» επικοινωνία: κάθε δρομολογητής μεταδίδει την τοπική του πληρότητα στους άμεσους γείτονες, και η περιφερειακή συμφόρηση προσεγγίζεται χρησιμοποιώντας τις τιμές που «βλέπει» ο επόμενος δρομολογητής για τη δική του γειτονιά. Στο πλαίσιο προσομοίωσης αυτό υλοποιείται ως πρόσβαση στις διαθέσιμες μετρήσεις του next-hop, ενώ σε υλικό θα μπορούσε να προκύψει με ένα επιπλέον στάδιο διάδοσης σημάτων συμφόρησης.

4.5 Ανάλυση Ορθότητας

4.5.1 Απαλλαγή από Αδιέξοδα

Ο APAR διατηρεί την απαλλαγή από αδιέξοδα διότι όλες οι επιλογές δρομολόγησης περιορίζονται σε επιτρεπτές κατευθύνσεις (admissible directions) του μοντέλου στροφών Odd-Even. Η λειτουργία χαμηλής συμφόρησης διαφοροποιεί μόνο την πολιτική επιλογής (ντετερμινιστική, XY-like), όχι το σύνολο επιτρεπτών στροφών. Κατά συνέπεια δεν εισάγονται νέες κυκλικές εξαρτήσεις καναλιών πέρα από αυτές που ήδη αποκλείει το Odd-Even μοντέλο.

Εμπειρικά, στο σύνολο των πειραμάτων της εργασίας δεν εμφανίστηκαν χρονικά όρια εκκένωσης (drain timeouts), γεγονός που αποτελεί ισχυρή ένδειξη απουσίας αδιεξόδων υπό το συγκεκριμένο μοντέλο προσομοίωσης.

4.5.2 Ζωντάνια

Η παρούσα εργασία δεν συνοδεύεται από πλήρη μαθηματική απόδειξη ζωντάνιας για τον APAR. Υπό το υιοθετημένο μοντέλο προσομοίωσης, η συμπεριφορά του αλγορίθμου είναι συμβατή με ιδιότητα προόδου: χρησιμοποιεί ελάχιστες επιτρεπτές διαδρομές στις φάσεις χαμηλής και μέτριας συμφόρησης και περιορίζει τη φάση υψηλής συμφόρησης σε ελάχιστες admissible επιλογές χωρίς να εισάγει επιπλέον κυκλικές εξαρτήσεις πέρα από το πλαίσιο Odd-Even. Επιπλέον, στο σύνολο των πειραμάτων δεν εμφανίστηκαν φαινόμενα «παγώματος» ή drain timeouts. Συνεπώς, τα πειραματικά αποτελέσματα αποτελούν ισχυρή εμπειρική ένδειξη ζωντάνιας υπό τις συγκεκριμένες παραδοχές, ενώ μια τυπική απόδειξη παραμένει αντικείμενο μελλοντικής εργασίας.

4.6 Ανάλυση Πολυπλοκότητας

4.6.1 Χωρική Πολυπλοκότητα

Η πρόσθετη χωρική πολυπλοκότητα του APAR είναι $O(1)$ ανά δρομολογητή, καθώς απαιτούνται μόνο σταθερού μεγέθους μετρητές και καταχωρητές ανεξάρτητα από το μέγεθος του δικτύου.

4.6.2 Χρονική Πολυπλοκότητα

Ο υπολογισμός διαδρομής στον APAR έχει χρονική πολυπλοκότητα $O(1)$, καθώς η απόφαση δρομολόγησης βασίζεται σε τοπικές και περιορισμένου ορίζοντα (1–2 hop) μετρήσεις συμφόρησης και δεν απαιτεί παγκόσμια γνώση ή επικοινωνία με απομακρυσμένους δρομολογητές.

4.7 Σύγκριση με Υπάρχουσες Προσεγγίσεις

4.7.1 Σύγκριση με DyAD

Ο αλγόριθμος DyAD [8] εναλλάσσεται μεταξύ ντετερμινιστικής και προσαρμοστικής δρομολόγησης βάσει του επιπέδου συμφόρησης. Ωστόσο, η εναλλαγή είναι απότομη χωρίς μηχανισμό υστέρησης, οδηγώντας σε πιθανές ταλαντώσεις. Ο APAR βελτιώνει αυτή την προσέγγιση με τρεις τρόπους: πρώτον, χρησιμοποιεί τρεις φάσεις αντί για δύο, επιτρέποντας πιο λεπτομερή προσαρμογή. Δεύτερον, εφαρμόζει μηχανισμό υστέρησης για ομαλές μεταβάσεις. Τρίτον, χρησιμοποιεί πληροφορίες γειτόνων δεύτερου βαθμού στη φάση υψηλής συμφόρησης.

4.7.2 Σύγκριση με CARL

Ο αλγόριθμος CARL (Congestion-Aware Reinforcement Learning) αποτελεί αλγόριθμο σύγκρισης βασισμένο σε ενισχυτική μάθηση (reinforcement learning) και, ειδικότερα, Q-learning. Η λήψη απόφασης γίνεται με επιλογή θύρας εξόδου που μεγιστοποιεί μια τιμή Q, υπό περιορισμούς (π.χ. Odd-Even) ώστε να διατηρείται η απαλλαγή από αδιέξοδα· η φιλοσοφία αυτή είναι συναφής με προσεγγίσεις τύπου Q-LCAR και RL-based NoC routing [11], [13].

Σε αντίθεση, ο APAR υλοποιεί ρητούς κανόνες ανά φάση με κατώφλια και υστέρηση, χωρίς διαδικασία εκπαίδευσης. Συνεπώς, η σύγκριση APAR–CARL στην παρούσα εργασία αναδεικνύει κυρίως τη διαφορά μεταξύ «κανόνων/κατωφλίων» και «μάθησης», με τα εμπειρικά αποτελέσματα να παρουσιάζονται και να συζητούνται στο Κεφάλαιο 5.

4.7.3 Σύγκριση με Odd-Even

Ο αλγόριθμος Odd-Even [4] παρέχει πλήρη προσαρμοστικότητα αλλά χωρίς επίγνωση συμφόρησης. Ο APAR χρησιμοποιεί τον Odd-Even ως βάση για τις φάσεις μέτριας και υψηλής συμφόρησης, αλλά προσθέτει επιλογή εξόδου βάσει διαθεσιμότητας ενταμιευτή. Αυτό επιτρέπει καλύτερη εξισορρόπηση φόρτου διατηρώντας την εγγύηση απαλλαγής από αδιέξοδα.

4.8 Παράμετροι Σχεδιασμού

4.8.1 Επιλογή Κατώφλιων

Τα κατώφλια φάσης (30% και 70% ως ποσοστά πληρότητας ενταμειυτή) επιλέχθηκαν βάσει πειραματικής ανάλυσης. Το χαμηλό κατώφλι στο 30% εξασφαλίζει ότι η ντετερμινιστική δρομολόγηση χρησιμοποιείται μόνο όταν η συμφόρηση είναι πραγματικά χαμηλή. Το υψηλό κατώφλι στο 70% ενεργοποιεί την επιθετική εξισορρόπηση φόρτου πριν το δίκτυο φτάσει σε κορεσμό.

4.8.2 Εύρος Υστέρησης

Το εύρος υστέρησης 5% επιλέχθηκε για να εξισορροπήσει μεταξύ σταθερότητας και ανταπόκρισης. Μικρότερο εύρος θα οδηγούσε σε συχνότερες μεταβάσεις, ενώ μεγαλύτερο εύρος θα καθυστέρουσε την προσαρμογή σε μεταβαλλόμενες συνθήκες.

4.8.3 Βάθος Πληροφοριών Γειτόνων

Στη φάση υψηλής συμφόρησης, ο APAR εξετάζει γείτονες δεύτερου βαθμού. Η επέκταση σε μεγαλύτερο βάθος θα αύξανε την ακρίβεια αλλά και το υπολογιστικό κόστος. Η επιλογή του δεύτερου βαθμού αποτελεί συμβιβασμό μεταξύ απόδοσης και πολυπλοκότητας και συνάδει με τη γενικότερη ιδέα της περιφερειακής επίγνωσης συμφόρησης (regional congestion awareness) που έχει μελετηθεί στη βιβλιογραφία [9].

5 Πειραματικά Αποτελέσματα

5.1 Σύνοψη πειραμάτων και εγκυρότητα δεδομένων

Η τελική αξιολόγηση βασίστηκε σε **5670** πειράματα προσομοίωσης ακριβείας κύκλου ρολογιού (cycle-accurate) και καλύπτει όλους τους συνδυασμούς της πειραματικής μήτρας (βλ. Πίνακας 5).

Η πειραματική μήτρα περιλαμβάνει: - αλγόριθμοι (5: **XY, West-First, Odd-Even, CARL, APAR**) - μεγέθη πλέγματος (3: 4×4, 8×8, 16×16) - πρότυπα κυκλοφορίας (7: Uniform, Transpose, Hotspot, Bit-Complement, Bit-Reverse, Neighbor, Tornado· βλ. Πίνακας 3) - ρυθμοί έγχυσης (18 επίπεδα, 0.01–0.50 flits/node/cycle) - επαναλήψεις ανά διαμόρφωση (**n=3**)

Ως ρυθμός έγχυσης (Injection Rate, **IR**) ορίζεται ο ρυθμός με τον οποίο κάθε κόμβος επιχειρεί να εισάγει flits στο δίκτυο (flits/node/cycle). Κάθε run περιλαμβάνει warmup 200 κύκλους, παράθυρο μέτρησης 2000 κύκλων και φάση εκκένωσης (drain), όπως περιγράφεται στο Κεφάλαιο 3.

Στο τελικό dataset όλες οι εκτελέσεις είναι έγκυρες: **0 drain timeouts** και **100% no-drop** (Injected flits = Delivered flits).

Οι τιμές που παρουσιάζονται παρακάτω είναι ο **μέσος όρος των n=3 επαναλήψεων**. Όπου χρησιμοποιείται “+/-”, πρόκειται για **95% διάστημα εμπιστοσύνης (95% CI)** γύρω από τον μέσο όρο.

5.2 Μέγιστη διεκπεραιωτική ικανότητα (*peak throughput*) στο 8×8

Ως μέγιστη διεκπεραιωτική ικανότητα (peak throughput) ορίζεται η μέγιστη τιμή της μέσης διεκπεραιωτικής ικανότητας (throughput_mean) που παρατηρείται στο sweep (σάρωση) των IR (0.01–0.50) για ένα συγκεκριμένο πρότυπο κυκλοφορίας. Ο Πίνακας 7 συνοψίζει τις τιμές για πλέγμα 8×8.

Πίνακας 7: Μέγιστη διεκπεραιωτική ικανότητα (peak throughput) για 8×8 (flits/node/cycle). Η στήλη “APAR vs XY” είναι ποσοστιαία μεταβολή ($APAR/XY - 1$).

Πρότυπο	XY	West-First	Odd-Even	CARL	APAR	APAR vs XY
Uniform (ομοιόμορφο)	0.3574	0.3272	0.3102	0.2545	0.2901	-18.8%
Transpose	0.2022	0.2342	0.2075	0.1699	0.1958	-3.2%
Hotspot	0.1622	0.1572	0.1525	0.1385	0.1413	-12.9%
Bit-Complement	0.1905	0.1804	0.1589	0.1173	0.1477	-22.5%
Bit-Reverse	0.1530	0.1920	0.2440	0.1874	0.2043	+33.5%
Neighbour	0.5034	0.5034	0.5034	0.5034	0.5034	+0.0%
Tornado	0.3625	0.3625	0.3625	0.3625	0.3625	+0.0%

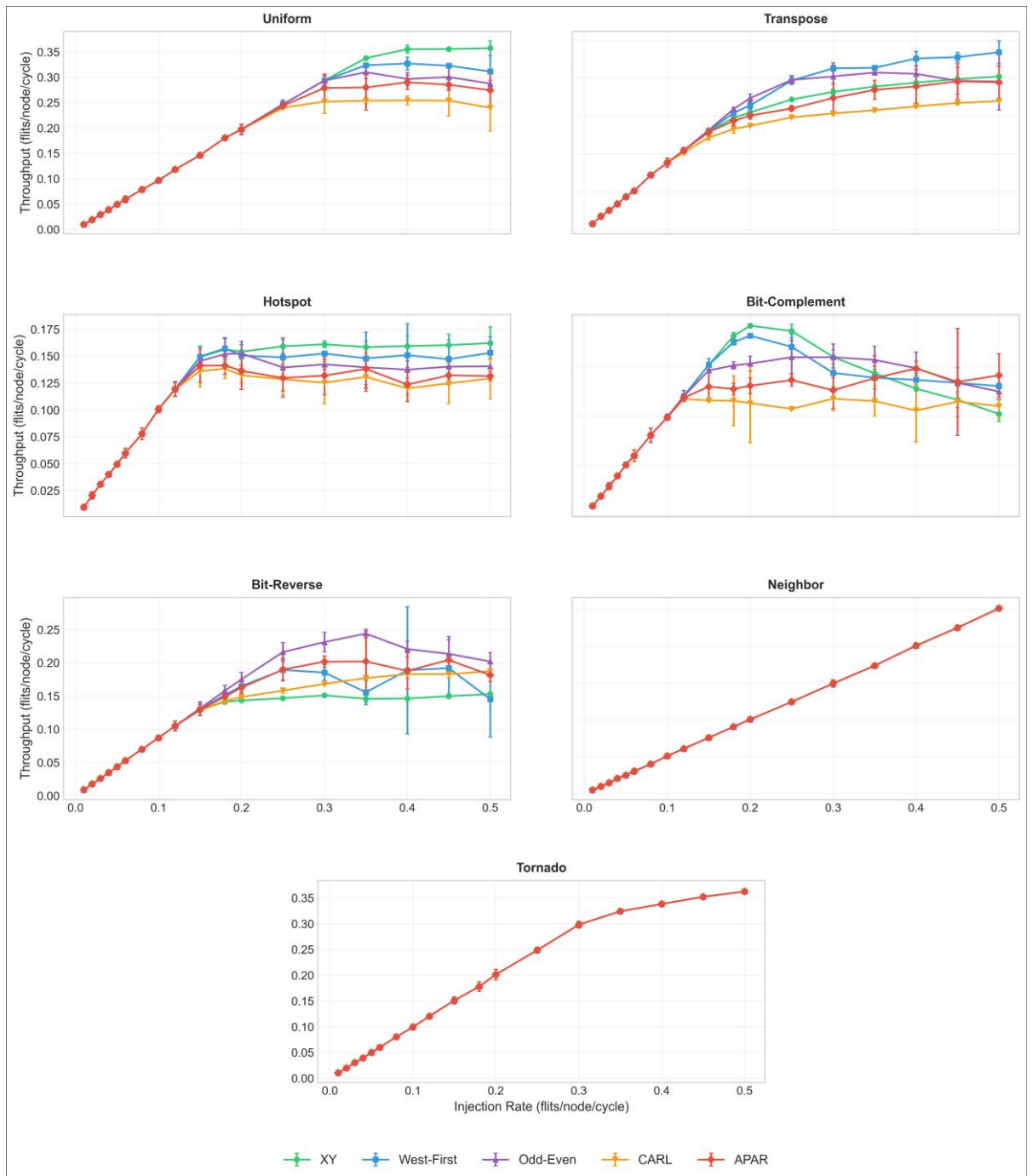
Το βασικό εύρημα του Πίνακα 7 είναι ότι η απόδοση είναι έντονα εξαρτώμενη από το πρότυπο κυκλοφορίας. Για παράδειγμα: - Στο **Bit-Reverse** ο APAR υπερέχει έναντι του XY στο 8×8 (+33.5% peak throughput), υποδεικνύοντας ότι η επίγνωση φάσης μπορεί να ωφελήσει σε πρότυπα με έντονη χωρική δομή. - Στο **Uniform** ο XY παραμένει ισχυρότερος, καθώς η κυκλοφορία δεν δημιουργεί σταθερές «περιοχές» συμφόρησης που να ευνοούν εναλλακτικές επιλογές διαδρομής.

Πίνακας 8: Ποσοστιαία μεταβολή peak throughput του APAR έναντι των αλγορίθμων αναφοράς (8×8).

Πρότυπο	APAR vs XY	APAR vs West-First	APAR vs Odd-Even	APAR vs CARL
Uniform	-18.8%	-11.3%	-6.5%	+14.0%
Transpose	-3.2%	-16.4%	-5.6%	+15.2%
Hotspot	-12.9%	-10.1%	-7.3%	+2.0%
Bit-Complement	-22.5%	-18.2%	-7.1%	+25.9%
Bit-Reverse	+33.5%	+6.4%	-16.3%	+9.0%
Neighbor	+0.0%	+0.0%	+0.0%	+0.0%
Tornado	+0.0%	+0.0%	+0.0%	+0.0%

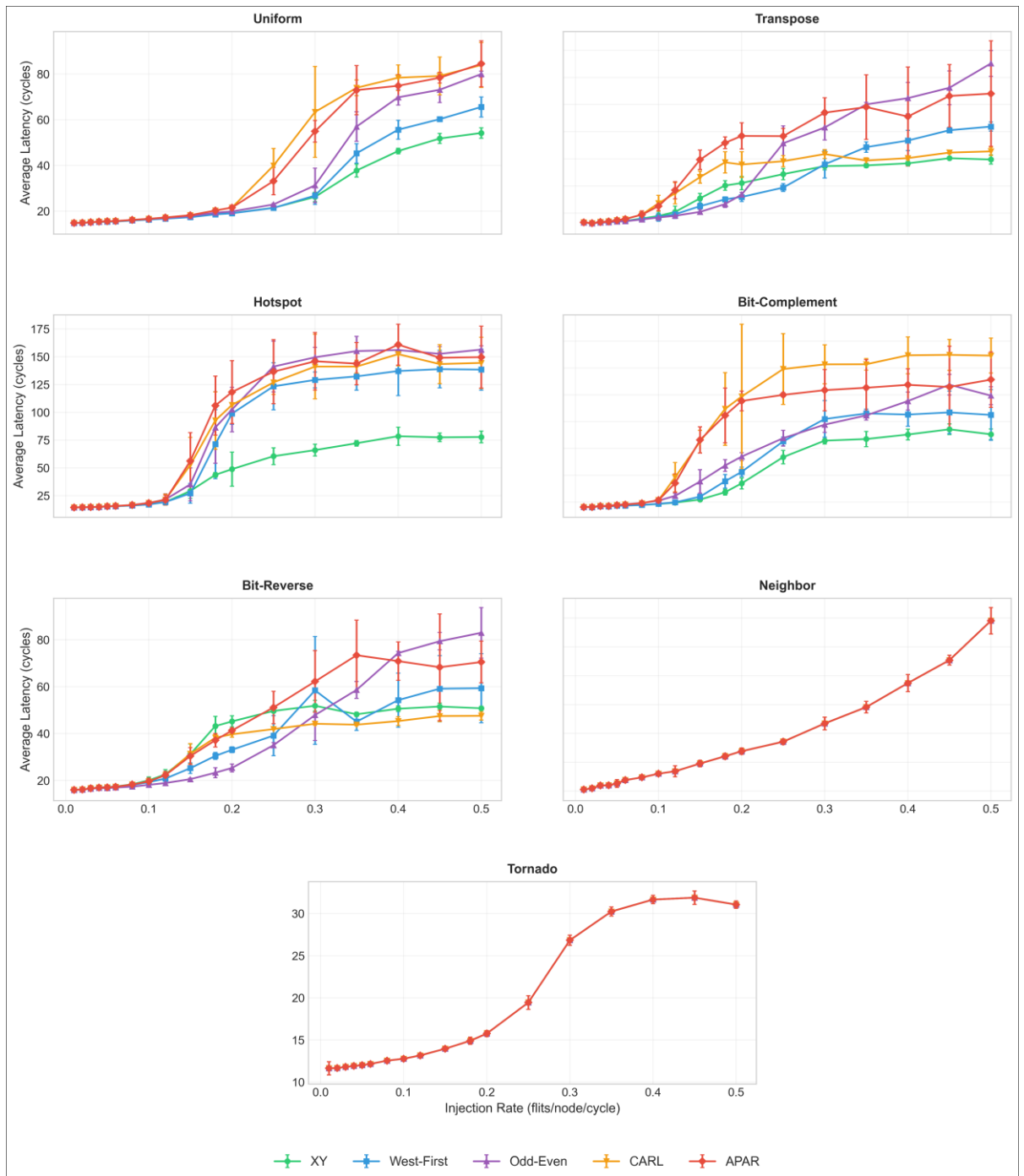
5.3 Καμπύλες throughput/καθυστερήσης ανά πρότυπο κυκλοφορίας (8×8)

Στο Διάγραμμα 7 παρουσιάζεται η διεκπεραιωτική ικανότητα (throughput) ως προς τον ρυθμό έγχυσης (IR) για όλα τα πρότυπα κυκλοφορίας στο 8×8. Οι μπάρες σφάλματος αντιστοιχούν σε 95% CI (n=3).



Διάγραμμα 7: Διεκπεραιωτική ικανότητα ως προς τον ρυθμό έγχυσης για πλέγμα 8×8 και 7 πρότυπα κυκλοφορίας (XY, West-First, Odd-Even, CARL, APAR).

Αντίστοιχα, στο Διάγραμμα 8 παρουσιάζεται η μέση καθυστέρηση (latency) σε κύκλους (cycles).



Διάγραμμα 8: Μέση καθυστέρηση ως προς τον ρυθμό έγχυσης για πλέγμα 8×8 και 7 πρότυπα κυκλοφορίας (XY, West-First, Odd-Even, CARL, APAR).

Η ερμηνεία των καμπυλών γίνεται ανά πρότυπο (βλ. Πίνακας 3 για τους ορισμούς): -

Uniform: οι καμπύλες throughput διατηρούνται υψηλότερες για τον XY, ενώ ο APAR δεν εμφανίζει κέρδος σε peak throughput. - **Transpose:** ο West-First εμφανίζει υψηλότερο throughput σε μεγάλο μέρος του sweep, καθώς το μοντέλο στροφών ταιριάζει στη δομή των

ροών. - **Hotspot:** η κίνηση πολλοί-προς-έναν (many-to-one) δημιουργεί στενωπό (bottleneck) κοντά στο hotspot, με αποτέλεσμα μεγάλη αύξηση καθυστέρησης σε υψηλούς IR για όλους τους αλγορίθμους. - **Bit-Complement / Bit-Reverse:** το Bit-Reverse αναδεικνύει το βασικό πλεονέκτημα του APAR σε throughput έναντι XY, ενώ το Bit-Complement παραμένει δυσμενές για τον APAR στο peak throughput. - **Neighbor / Tornado:** οι καμπύλες συγκλίνουν, καθώς η κυκλοφορία είναι είτε έντονα τοπική (Neighbor) είτε δομημένη με συμμετρικό τρόπο (Tornado), και οι επιλογές διαδρομής δεν διαφοροποιούν ουσιαστικά τη μέγιστη διεκπεραιωτική ικανότητα.

5.4 Σύγκριση σε σταθερό προσφερόμενο φόρτο ($IR=0.30, 8 \times 8$)

Για δίκαιη σύγκριση σε κοινό προσφερόμενο φόρτο (offered load), εξετάζεται ενδεικτικά το $IR=0.30$ στο 8×8 . Στους Πίνακες 9–10 παρουσιάζονται throughput και latency ως μέση τιμή $\pm 95\%$ CI.

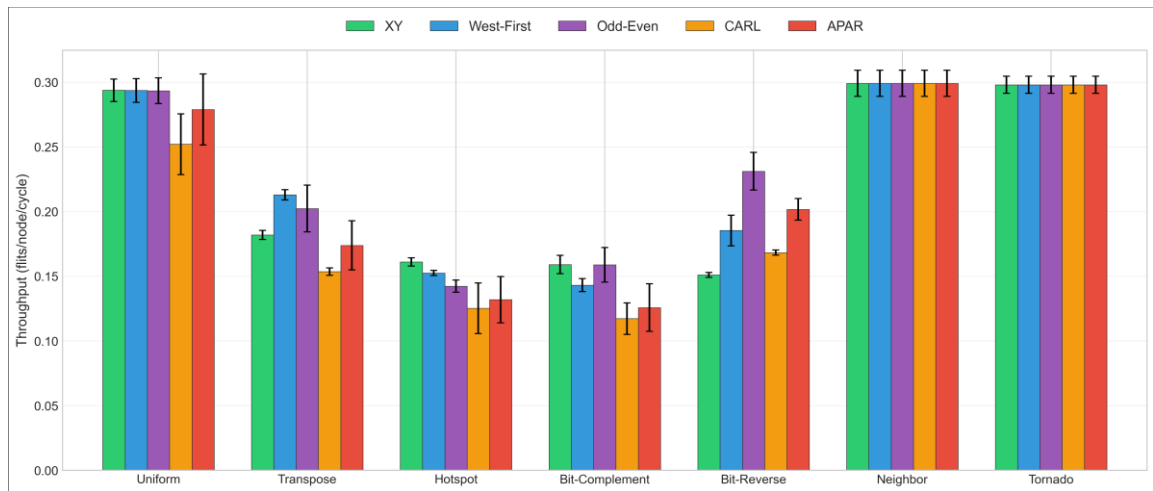
Πίνακας 9: Throughput στο $IR=0.30$ για 8×8 (mean $\pm 95\%$ CI).

Πρότυπο	XY	West-First	Odd-Even	CARL	APAR
Uniform	0.2938 $\pm$ 0.0087	0.2937 $\pm$ 0.0092	0.2935 $\pm$ 0.0099	0.2522 $\pm$ 0.0235	0.2789 $\pm$ 0.0275
Transpos e	0.1820 $\pm$ 0.0035	0.2130 $\pm$ 0.0040	0.2024 $\pm$ 0.0181	0.1537 $\pm$ 0.0029	0.1739 $\pm$ 0.0190
Hotspot	0.1611 $\pm$ 0.0032	0.1525 $\pm$ 0.0020	0.1424 $\pm$ 0.0047	0.1253 $\pm$ 0.0195	0.1319 $\pm$ 0.0179
Bit- Comple ment	0.1591 $\pm$ 0.0071	0.1432 $\pm$ 0.0050	0.1589 $\pm$ 0.0133	0.1173 $\pm$ 0.0122	0.1258 $\pm$ 0.0184
Bit- Reverse	0.1511 $\pm$ 0.0019	0.1854 $\pm$ 0.0119	0.2312 $\pm$ 0.0145	0.1683 $\pm$ 0.0020	0.2018 $\pm$ 0.0084
Neighbor	0.2992 $\pm$ 0.0101	0.2992 $\pm$ 0.0101	0.2992 $\pm$ 0.0101	0.2992 $\pm$ 0.0101	0.2992 $\pm$ 0.0101
Tornado	0.2981 $\pm$ 0.0066	0.2981 $\pm$ 0.0066	0.2981 $\pm$ 0.0066	0.2981 $\pm$ 0.0066	0.2981 $\pm$ 0.0066

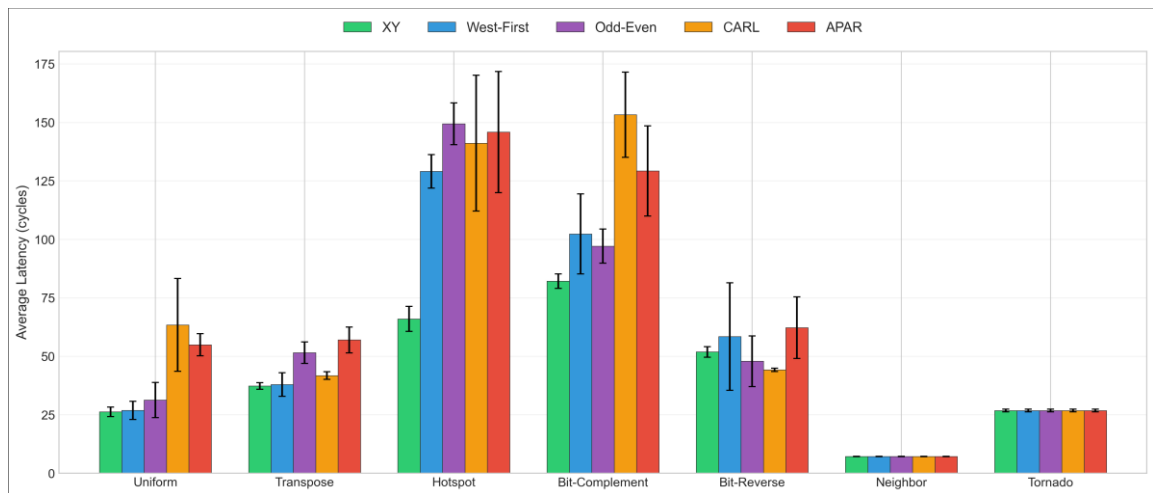
Πίνακας 10: Latency στο IR=0.30 για 8×8 (cycles, mean +/- 95% CI).

Πρότυπο	XY	West-First	Odd-Even	CARL	APAR
Uniform	26.27 +/- 1.99	26.85 +/- 3.88	31.32 +/- 7.53	63.42 +/- 19.84	54.94 +/- 4.74
Transpos e	37.30 +/- 1.40	37.97 +/- 5.04	51.55 +/- 4.62	41.75 +/- 1.65	57.02 +/- 5.52
Hotspot	65.98 +/- 5.34	129.11 +/- 7.16	149.46 +/- 8.93	141.14 +/- 29.06	145.91 +/- 25.88
Bit- Comple ment	82.17 +/- 3.11	102.34 +/- 17.12	97.12 +/- 7.30	153.33 +/- 18.20	129.26 +/- 19.25
Bit- Reverse	51.88 +/- 2.26	58.42 +/- 22.99	47.90 +/- 10.85	44.20 +/- 0.68	62.22 +/- 13.18
Neighbor	7.17 +/- 0.11	7.17 +/- 0.11	7.17 +/- 0.11	7.17 +/- 0.11	7.17 +/- 0.11
Tornado	26.84 +/- 0.61	26.84 +/- 0.61	26.84 +/- 0.61	26.84 +/- 0.61	26.84 +/- 0.61

Τα Διαγράμματα 9–10 συνοψίζουν τα ίδια αποτελέσματα του IR=0.30 ανά πρότυπο κυκλοφορίας.



Διάγραμμα 9: Σύγκριση throughput ανά πρότυπο κυκλοφορίας στο $IR=0.30$ για πλέγμα 8×8 .



Διάγραμμα 10: Σύγκριση latency ανά πρότυπο κυκλοφορίας στο $IR=0.30$ για πλέγμα 8×8 .

Πίνακας 11: Ποσοστιαία μεταβολή latency του APAR έναντι των αλγορίθμων αναφοράς στο $IR=0.30$ (8×8). (Θετικό ποσοστό = μεγαλύτερη καθυστέρηση.)

Πρότυπο	APAR vs XY	APAR vs West-First	APAR vs Odd-Even	APAR vs CARL
Uniform	+109.2%	+104.6%	+75.4%	-13.4%
Transpose	+52.9%	+50.2%	+10.6%	+36.6%
Hotspot	+121.1%	+13.0%	-2.4%	+3.4%
Bit-Complement	+57.3%	+26.3%	+33.1%	-15.7%

Πρότυπο	APAR vs XY	APAR vs West-First	APAR vs Odd-Even	APAR vs CARL
Bit-Reverse	+19.9%	+6.5%	+29.9%	+40.8%
Neighbor	+0.0%	+0.0%	+0.0%	+0.0%
Tornado	+0.0%	+0.0%	+0.0%	+0.0%

5.5 Συμπεριφορά φάσεων APAR (8×8)

Η συμπεριφορά του APAR ως προς τις φάσεις (low/medium/high) καταγράφεται μέσω μετρικών καταγραφής (instrumentation): ποσοστό χαμηλής φάσης (low-phase ratio), αριθμός μεταβάσεων φάσης (phase changes) και ποσοστό αποφάσεων ανά φάση. Σε χαμηλό φόρτο (π.χ. IR=0.05) ο APAR παραμένει πρακτικά αποκλειστικά σε χαμηλή φάση (low-phase) για όλα τα πρότυπα.

Πίνακας 12: Μετρικές φάσεων του APAR στο IR=0.30 για 8×8.

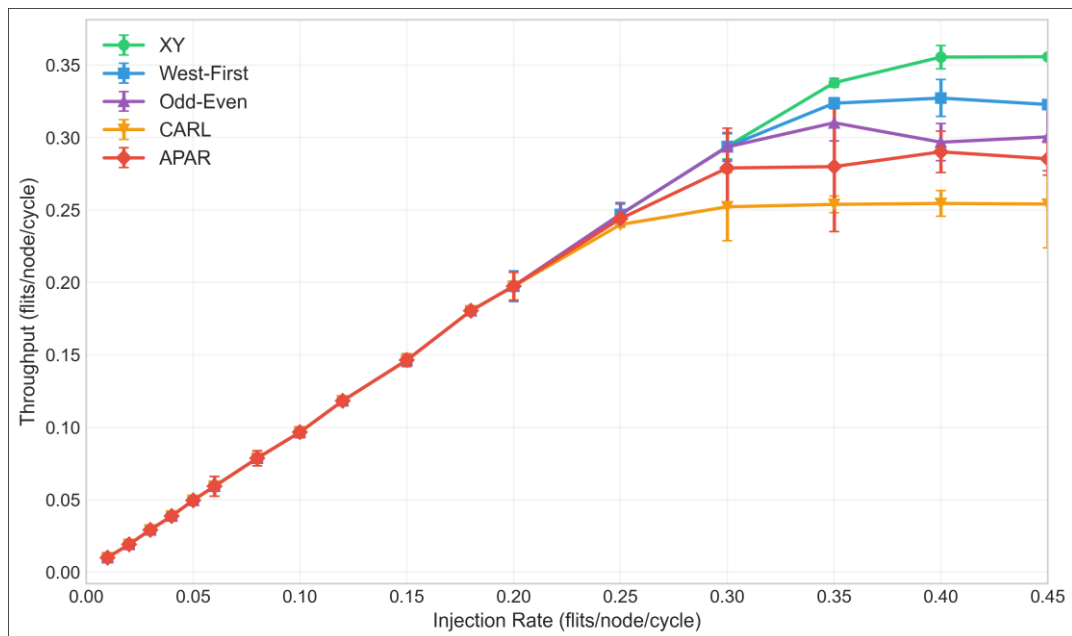
Πρότυπο	Ποσοστό low-phase	Phase changes	Αποφάσεις Low (XY)	Αποφάσεις Medium	Αποφάσεις High
Uniform	0.590	602.7	25.7%	56.0%	18.3%
Transpos e	0.681	220.0	47.2%	48.1%	4.7%
Hotspot	0.349	368.7	4.8%	43.4%	51.8%
Bit-Comple ment	0.326	456.7	7.3%	32.4%	60.2%
Bit-Reverse	0.635	379.0	40.6%	46.6%	12.8%
Neighbor	1.000	0.0	100.0%	0.0%	0.0%
Tornado	0.949	89.0	87.4%	12.6%	0.0%

Η παραπάνω εικόνα είναι συνεπής με τις διαισθητικές ιδιότητες των προτύπων: - **Neighbor:** δεν αναπτύσσεται σημαντική συμφόρηση, οπότε ο APAR μένει σε χαμηλή φάση (low-

phase). - **Hotspot / Bit-Complement:** εμφανίζεται έντονη συμφόρηση, και ο APAR ενεργοποιεί σημαντικό ποσοστό αποφάσεων υψηλής φάσης (high-phase).

5.6 Σημείο κορεσμού και εστίαση στην προκορεσμική περιοχή

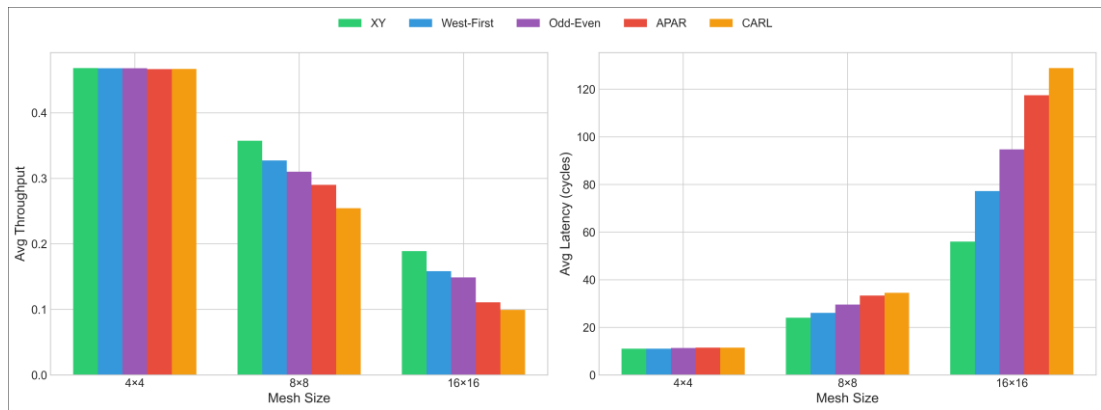
Το σημείο κορεσμού (saturation point) συνδέεται εμπειρικά με δύο παρατηρήσεις: (α) η διεκπεραιωτική ικανότητα (throughput) παύει να αυξάνεται ουσιαστικά και τείνει να σταθεροποιείται, και (β) η καθυστέρηση αυξάνεται απότομα καθώς αυξάνει ο προσφερόμενος φόρτος. Για να αποτυπωθεί καθαρότερα η συμπεριφορά λίγο πριν από τον κορεσμό, το Διάγραμμα 11 εστιάζει στην περίπτωση Uniform (8×8) έως $IR=0.45$.



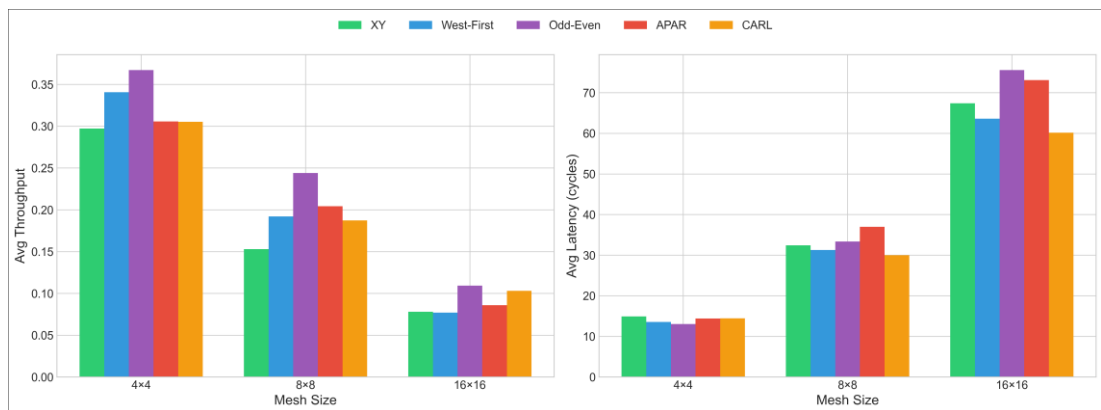
Διάγραμμα 11: Throughput ως προς IR για Uniform (8×8) έως $IR=0.45$, με έμφαση στην περιοχή πριν και κοντά στον κορεσμό.

5.7 Κλιμακωσιμότητα (4×4 , 8×8 , 16×16)

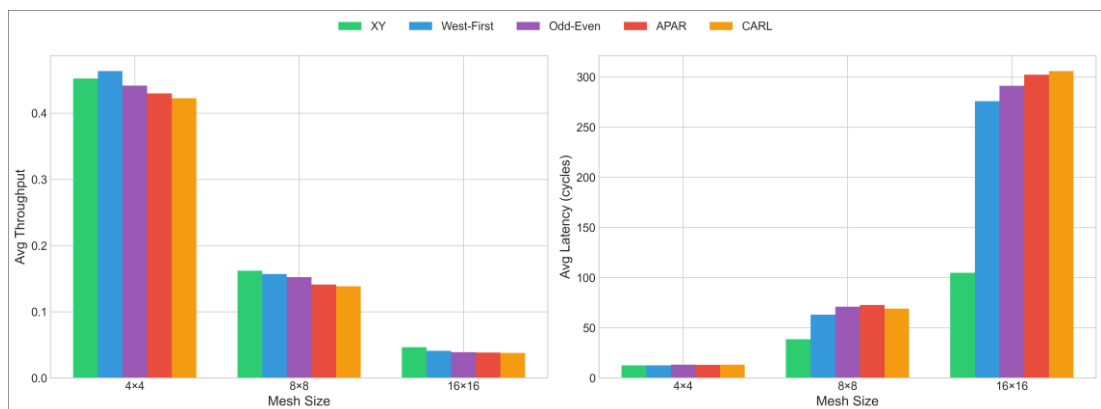
Για να εξεταστεί η κλιμακωσιμότητα (scalability), συγκρίνονται τα πλέγματα 4×4 , 8×8 και 16×16 . Τα Διαγράμματα 12–14 παρουσιάζουν ενδεικτικά αποτελέσματα για τρία χαρακτηριστικά πρότυπα.



Διάγραμμα 12: Κλιμακωσιμότητα στο Uniform, με peak throughput και μέση καθυστέρηση ως προς το μέγεθος πλέγματος.



Διάγραμμα 13: Κλιμακωσιμότητα στο Bit-Reverse, με peak throughput και μέση καθυστέρηση ως προς το μέγεθος πλέγματος.



Διάγραμμα 14: Κλιμακωσιμότητα στο Hotspot, με peak throughput και μέση καθυστέρηση ως προς το μέγεθος πλέγματος.

5.8 Συνολική αποτίμηση των αποτελεσμάτων

Η συνολική εικόνα που προκύπτει από το Κεφάλαιο 5 είναι σαφής: τα 5670 πειράματα είναι έγκυρα, δεν παρατηρήθηκαν drain timeouts και το dataset επιτρέπει ασφαλή σύγκριση των αλγορίθμων υπό κοινές παραδοχές. Αυτό είναι σημαντικό, γιατί τα συμπεράσματα δεν στηρίζονται σε μεμονωμένες ενδείξεις αλλά σε συστηματική σάρωση πολλών προτύπων κυκλοφορίας, ρυθμών έγχυσης και μεγεθών πλέγματος.

Παράλληλα, τα αποτελέσματα δείχνουν ότι δεν υπάρχει ένας αλγόριθμος που να υπερέχει σταθερά σε κάθε συνθήκη. Ο APAR εμφανίζει ουσιαστικό κέρδος σε δομημένα πρότυπα όπως το Bit-Reverse, όπου αξιοποιεί καλύτερα την επίγνωση φάσης, αλλά δεν ξεπερνά τον XY σε πιο ομοιόμορφες ροές όπως το Uniform. Με άλλα λόγια, η συμβολή του APAR δεν είναι ότι προσφέρει μια καθολικά καλύτερη λύση, αλλά ότι προσαρμόζει τη συμπεριφορά του σε σενάρια όπου η συμφόρηση αποκτά σαφή χωρική δομή.

Τέλος, οι μετρικές φάσεων δείχνουν ότι ο μηχανισμός ανίχνευσης φάσης δεν λειτουργεί απλώς θεωρητικά αλλά επηρεάζει έμπρακτα τη συμπεριφορά του αλγορίθμου. Σε χαμηλό φόρτο ο APAR παραμένει κυρίως σε low phase, ενώ σε πιο απαιτητικά πρότυπα αυξάνει τη χρήση medium/high αποφάσεων. Έτσι, τα αποτελέσματα του κεφαλαίου στηρίζουν τη βασική θέση της εργασίας: η επίγνωση φάσης μπορεί να είναι χρήσιμη, αρκεί να αξιολογείται ως μέρος ενός ευρύτερου συμβιβασμού μεταξύ throughput, latency και σταθερότητας.

6 Συμπεράσματα και Μελλοντική Εργασία

6.1 Σύνοψη Εργασίας

Η παρούσα εργασία παρουσίασε τον αλγόριθμο APAR (Adaptive Phase-Aware Routing / Προσαρμοστική Δρομολόγηση με Επίγνωση Φάσης), έναν υβριδικό αλγόριθμο δρομολόγησης για Ολοκληρωμένα Ενδο-συνδεδεμένα Δίκτυα (ΟΕΔ, Networks-on-Chip) που προσαρμόζει την πολιτική επιλογής διαδρομής ανάλογα με τη φάση συμφόρησης. Η βασική ιδέα του APAR είναι ότι ένα δίκτυο εμφανίζει διακριτές καταστάσεις (χαμηλή/μέτρια/υψηλή συμφόρηση) και ότι η καταλληλότερη πολιτική δρομολόγησης μπορεί να διαφέρει ανά φάση.

Η εργασία υλοποίησε ένα πειραματικό πλαίσιο προσομοίωσης ακριβείας κύκλου (cycle-accurate simulation framework) σε C++17, στο οποίο ενσωματώθηκαν ο APAR και τέσσερις αλγόριθμοι αναφοράς (XY, West-First, Odd-Even, CARL), με στόχο τη δίκαιη συγκριτική αξιολόγηση υπό κοινές παραδοχές (και όχι την ανάπτυξη ενός γενικού/βιομηχανικού simulator). Ο πηγαίος κώδικας, τα datasets και τα scripts παρατίθενται στον συνοδευτικό φάκελο της εργασίας (βλ. Παράρτημα Α).

Η πειραματική αξιολόγηση πραγματοποιήθηκε με **5670** πειράματα σε τρία μεγέθη πλέγματος (4×4 , 8×8 , 16×16), επτά πρότυπα κυκλοφορίας και 18 ρυθμούς έγχυσης, με $n=3$ επαναλήψεις ανά διαμόρφωση και έλεγχο εγκυρότητας (drain/no-drop).

6.2 Κύρια Ευρήματα

6.2.1 Βελτίωση Απόδοσης

Τα πειραματικά αποτελέσματα του Κεφαλαίου 5 ανέδειξαν ότι η απόδοση είναι έντονα εξαρτώμενη από το πρότυπο κυκλοφορίας (traffic pattern) και το επίπεδο φόρτου. Ενδεικτικά, στο **Bit-Reverse** (8×8) ο APAR εμφανίζει **+33.5%** υψηλότερο peak throughput έναντι XY (Πίνακας 7), ενώ σε άλλες περιπτώσεις (π.χ. Uniform) ο XY παραμένει ισχυρότερος σε peak throughput.

Συνεπώς, το κύριο συμπέρασμα δεν είναι ότι ένας αλγόριθμος «κερδίζει παντού», αλλά ότι η επιλογή δρομολόγησης πρέπει να λαμβάνει υπόψη το πρότυπο κυκλοφορίας και τους στόχους (διεκπεραιωτική ικανότητα/καθυστέρηση/σταθερότητα).

Σε ό,τι αφορά την καθυστέρηση (latency), δεν παρατηρήθηκε γενικευμένη μείωση για τον APAR σε υψηλό προσφερόμενο φόρτο (π.χ. $IR=0.30$ στο 8×8). Αντίθετα, τα αποτελέσματα αναδεικνύουν έναν σαφή συμβιβασμό (trade-off) μεταξύ throughput και latency σε ορισμένα πρότυπα (βλ. Πίνακες 10–11).

6.2.2 Αποτελεσματικότητα Μηχανισμού Ανίχνευσης Φάσης

Ο μηχανισμός ανίχνευσης φάσης αποδείχθηκε λειτουργικός στην ταξινόμηση της κατάστασης του δικτύου. Οι μετρικές καταγραφής (instrumentation) του APAR (π.χ. ποσοστό low-phase, μεταβάσεις φάσης και κατανομή αποφάσεων ανά φάση) επιβεβαιώνουν ότι σε χαμηλό φόρτο παραμένει κατά κύριο λόγο σε χαμηλή φάση, ενώ σε υψηλότερο φόρτο ενεργοποιεί περισσότερο τις πολιτικές μέτριας/υψηλής συμφόρησης. (βλ. Πίνακας 12 στο Κεφάλαιο 5).

6.2.3 Σταθερότητα Μέσω Υστέρησης

Ο μηχανισμός υστέρησης αποδείχθηκε κρίσιμος για τη σταθερότητα του αλγορίθμου, καθώς αποτρέπει ταλαντώσεις μεταξύ φάσεων σε οριακές συνθήκες και εξασφαλίζει ομαλές μεταβάσεις πολιτικών.

6.3 Συνεισφορές

Η παρούσα εργασία συνεισφέρει στον τομέα των Ολοκληρωμένων Ενδο-συνδεόμενων Δικτύων με τους ακόλουθους τρόπους:

Πρώτον, προτείνει έναν νέο υβριδικό αλγόριθμο δρομολόγησης που συνδυάζει ντετερμινιστική και προσαρμοστική δρομολόγηση με βάση την ανίχνευση φάσης. Αυτή η προσέγγιση διαφέρει από τις υπάρχουσες μεθόδους που χρησιμοποιούν είτε αμιγώς ντετερμινιστική είτε αμιγώς προσαρμοστική δρομολόγηση.

Δεύτερον, εισάγει έναν μηχανισμό υστέρησης για ομαλές μεταβάσεις μεταξύ φάσεων, αντιμετωπίζοντας ένα σημαντικό πρόβλημα των υβριδικών σχημάτων που εναλλάσσονται απότομα μεταξύ λειτουργιών.

Τρίτον, παρέχει εκτεταμένη πειραματική αξιολόγηση (5670 runs) που αναδεικνύει τόσο τις περιπτώσεις όπου ο APAR εμφανίζει κέρδος (π.χ. Bit-Reverse) όσο και τις περιπτώσεις όπου οι κλασικοί αλγόριθμοι παραμένουν ισχυροί (π.χ. Uniform), τεκμηριώνοντας τους σχετικούς συμβιβασμούς throughput/latency.

6.4 Περιορισμοί και Απειλές στην Εγκυρότητα (Threats to Validity)

Παρότι η μεθοδολογία σχεδιάστηκε ώστε να παρέχει δίκαιες και αναπαραγώγιμες συγκρίσεις (warmup/measurement/drain = προθέρμανση/παράθυρο μέτρησης/εκκένωση, έλεγχος no-drop, πολιτική common random numbers (CRN) και 95% CI), υπάρχουν αναπόφευκτοι περιορισμοί και απειλές εγκυρότητας που πρέπει να ληφθούν υπόψη κατά την ερμηνεία των αποτελεσμάτων.

6.4.1 Εσωτερική εγκυρότητα (Internal validity): ορθότητα/πιστότητα προσομοιωτή

Η αξιολόγηση βασίζεται σε προσομοίωση και όχι σε πραγματικό υλικό. Παρότι το πειραματικό πλαίσιο είναι ακριβείας κύκλου ρολογιού και υλοποιεί έλεγχο ροής με πιστώσεις (credit-based flow control) και 4-στάδια διοχέτευση (pipeline) δρομολογητή, παραμένουν υποθέσεις που μπορούν να επηρεάσουν την ακρίβεια των απόλυτων αριθμών:

- Η καθυστέρηση ζεύξης μοντελοποιείται ως σταθερή (1 κύκλος) και δεν αποτυπώνει φυσικές επιδράσεις (καθυστέρηση καλωδίων / wire delay, φυσική χωροθέτηση / physical layout, χρονισμός / timing closure).
- Δεν μοντελοποιούνται σφάλματα, μεταβλητές συχνότητες, ή αλληλεπιδράσεις με το υπόλοιπο SoC (πεδία ρολογιού/ισχύος / clock-power domains, DVFS).

Μείωση κινδύνου: εφαρμόστηκε drain φάση χωρίς έναρξη νέων πακέτων, καθώς και έλεγχος no-drop μετά το drain. Στο τελικό dataset δεν εμφανίστηκαν drain timeouts και ο έλεγχος no-drop επιβεβαιώθηκε σε όλα τα runs.

6.4.2 Κατασκευαστική εγκυρότητα (Construct validity): μετρικές, ορισμοί και παράθυρο μέτρησης

Οι βασικές μετρικές (καθυστέρηση / latency, διεκπεραιωτική ικανότητα / throughput, άλματα / hops) υπολογίζονται αποκλειστικά στο παράθυρο μέτρησης, μετά από warmup (προθέρμανση). Αυτό μειώνει τα μεταβατικά φαινόμενα (transients), αλλά επιβάλλει περιορισμούς:

- Ο ρυθμός έγχυσης ορίζεται ως προσφερόμενος φόρτος (offered load, flits/node/cycle) και η έγχυση γίνεται ανά flit (flit-by-flit) με περιορισμό στη διεπαφή δικτύου ($NI, \leq 1$ flit/κόμβο/κύκλος). Η επιλογή αυτή είναι ρεαλιστική για wormhole/VC μοντέλα, αλλά επηρεάζει τη χαρτογράφηση σε συγκεκριμένες σχεδιάσεις υλικού (hardware designs).

- Το σημείο κορεσμού (saturation point) δεν είναι μοναδικά ορισμένο στη βιβλιογραφία. Στην εργασία χρησιμοποιήθηκε μια συντηρητική, αυτοματοποιημένη ευρετική (πλάτωμα throughput + αύξηση latency), ενώ παρέχονται και οι πλήρεις καμπύλες (curves) ώστε να μπορεί να εφαρμοστεί εναλλακτικός ορισμός.

6.4.3 Εξωτερική εγκυρότητα (External validity): γενίκευση σε πραγματικές εφαρμογές/συστήματα

Τα πρότυπα κυκλοφορίας είναι συνθετικά (Uniform/Transpose/Hotspot/Bit-Complement/Bit-Reverse/Neighbor/Tornado) και λειτουργούν ως «δοκιμές πίεσης» (stress tests) για διαφορετικές μορφές συμφόρησης. Ωστόσο, πραγματικές εφαρμογές μπορούν να εμφανίζουν χρονικά μεταβαλλόμενες φάσεις (phases) και συσχετίσεις που δεν αποτυπώνονται πλήρως από συνθετικά πρότυπα. Η γενίκευση θα ενισχυθεί με μελλοντική αξιολόγηση σε ίχνη εφαρμογών (application traces), π.χ. PARSEC/SPLASH-2 ή traces από GEM5.

Επιπλέον, η εργασία αξιολόγησε μόνο δισδιάστατη τοπολογία πλέγματος (2D mesh) και ένα συγκεκριμένο μικρο-αρχιτεκτονικό σημείο (VCs/buffers/packet size). Συνεπώς, τα αποτελέσματα πρέπει να ερμηνεύονται ως **σχετικές** συγκρίσεις υπό κοινές παραδοχές και όχι ως καθολικές αποδόσεις για κάθε σχεδιασμό ΟΕΔ.

6.4.4 Συμπερασματική εγκυρότητα (Conclusion validity): στατιστική ισχύς

Κάθε διαμόρφωση εκτελέστηκε με $n=3$ επαναλήψεις. Για να αντιμετωπιστεί η αυξημένη αβεβαιότητα μικρού δείγματος, υπολογίστηκαν **95% διαστήματα εμπιστοσύνης** με Student-t και εφαρμόστηκε πολιτική common random numbers (CRN) ώστε οι συγκρίσεις να είναι ζευγαρωμένες (ίδιο seed για όλους τους αλγορίθμους σε ίδιο (*mesh, traffic, IR, run*)). Παρόλα αυτά, μεγαλύτερο n θα αύξανε τη στατιστική ισχύ, ειδικά κοντά στον κορεσμό όπου η διακύμανση αυξάνεται.

6.4.5 Ευαισθησία και ρύθμιση παραμέτρων (APAR/CARL)

Οι παράμετροι του APAR (κατώφλια 30%/70%, υστέρηση $\pm 5\%$, ελάχιστοι κύκλοι πριν από μετάβαση) κρατήθηκαν σταθερές σε όλα τα πειράματα για να αποφεύγεται προσαρμογή ανά πρότυπο (per-pattern tuning). Παρ' όλα αυτά, διαφορετικές αρχιτεκτονικές

(VCs/buffers/link delays) μπορεί να μετατοπίσουν τη βέλτιστη ρύθμιση. Μια συστηματική ανάλυση ευαισθησίας ή/και αυτόματη προσαρμογή κατωφλίων αποτελεί φυσική επέκταση.

6.4.6 Ενεργειακό μοντέλο (όπου χρησιμοποιείται)

Οι ενεργειακές εκτιμήσεις βασίζονται σε απλοποιημένο μοντέλο γεγονότων (event-based) (buffer reads/writes, switch, link traversal) και δεν είναι βαθμονομημένες σε συγκεκριμένη τεχνολογία (π.χ. 7nm/5nm). Άρα, οι απόλυτες τιμές πρέπει να αντιμετωπίζονται ως ενδεικτικές, ενώ οι συγκρίσεις μεταξύ αλγορίθμων είναι πιο ασφαλείς στο μέτρο που κρατούνται κοινές οι παραδοχές.

6.5 Μελλοντική Εργασία

Η παρούσα εργασία ανοίγει πολλές κατευθύνσεις για μελλοντική έρευνα.

Μια επιπλέον κατεύθυνση είναι η σύγκριση του APAR με πιο πρόσφατες προσεγγίσεις βαθιάς ενισχυτικής μάθησης για δρομολόγηση (π.χ. DRLAR [26], DeepNR [27]) ή/και με νεότερες στρατηγικές επιλογής διαδρομής/εξόδου που στοχεύουν στη μείωση της καθυστέρησης [28], αξιοποιώντας ως οδηγό τις πρόσφατες ανασκοπήσεις της βιβλιογραφίας [29], [30].

- **Υλοποίηση σε υλικό:** Η μεταφορά του APAR σε γλώσσα περιγραφής υλικού, όπως Verilog ή VHDL, θα επέτρεπε την αποτίμηση του πραγματικού κόστους σε εμβαδόν, κατανάλωση ενέργειας και κρίσιμη διαδρομή. Μια τέτοια υλοποίηση θα έδειχνε πιο καθαρά αν τα κέρδη σε επίπεδο δρομολόγησης δικαιολογούν την πρόσθετη λογική του αλγορίθμου.
- **Επέκταση σε άλλες τοπολογίες:** Η εφαρμογή του APAR σε τόρο, τρισδιάστατο πλέγμα ή ιεραρχικές τοπολογίες θα έδειχνε αν η λογική της επίγνωσης φάσης παραμένει χρήσιμη και πέρα από το 2D mesh. Κάθε τοπολογία αλλάζει τόσο τα μοτίβα συμφόρησης όσο και τους περιορισμούς δρομολόγησης, άρα απαιτείται νέα προσαρμογή και εκ νέου αξιολόγηση.
- **Χρήση πραγματικών φορτίων εργασίας:** Η αξιολόγηση με ίχνη εφαρμογών, αντί μόνο με συνθετικά πρότυπα, θα παρείχε πιο ρεαλιστική εικόνα της συμπεριφοράς του APAR σε πρακτικά σενάρια. Έτσι θα ήταν δυνατό να φανεί αν τα οφέλη που εμφανίζονται σε δομημένα patterns μεταφέρονται και σε πραγματικά workloads.

- **Προσαρμογή παραμέτρων και μάθηση:** Μια φυσική συνέχεια είναι η αντικατάσταση των σταθερών κατωφλίων φάσης από μηχανισμό αυτόματης προσαρμογής ή από ελαφριά μορφή μάθησης. Αυτό θα επέτρεπε στον αλγόριθμο να προσαρμόζεται σε διαφορετικά δίκτυα και επίπεδα φόρτου χωρίς χειροκίνητη ρύθμιση.
- **Ανοχή σφαλμάτων, ενέργεια και QoS:** Η ενσωμάτωση μηχανισμών ανοχής σφαλμάτων, επίγνωσης ενέργειας και ποιότητας υπηρεσίας (QoS) θα μετέτρεπε τον APAR από ερευνητικό αλγόριθμο σύγκρισης σε πιο ολοκληρωμένη αρχιτεκτονική πρόταση. Οι κατευθύνσεις αυτές είναι ιδιαίτερα σημαντικές για μελλοντικά πολυπύρηννα συστήματα, όπου η απόδοση δεν αρκεί από μόνη της χωρίς αξιοπιστία και ελεγχόμενη κατανάλωση.

6.6 Τελικές Παρατηρήσεις

Η παρούσα εργασία έδειξε ότι η δυναμική προσαρμογή της στρατηγικής δρομολόγησης βάσει της φάσης λειτουργίας του δικτύου είναι μια ρεαλιστική προσέγγιση, η οποία μπορεί να αποδώσει οφέλη σε συγκεκριμένα πρότυπα κυκλοφορίας (π.χ. Bit-Reverse) και να παρέχει ελεγχόμενη συμπεριφορά μέσω υστέρησης. Τα αποτελέσματα δεν υποστηρίζουν ισχυρισμό «σημαντικής μείωσης καθυστέρησης» για όλες τις περιπτώσεις· αντίθετα, αναδεικνύουν σαφείς συμβιβασμούς throughput/latency υπό διαφορετικά πρότυπα και επίπεδα φόρτου.

Καθώς οι πολυπύρηννοι επεξεργαστές συνεχίζουν να αυξάνουν τον αριθμό των πυρήνων τους, η σημασία αποδοτικών υποδομών διασύνδεσης θα αυξάνεται. Ο APAR αποτελεί ένα βήμα προς αυτή την κατεύθυνση, προσφέροντας μια πρακτική λύση που μπορεί να ενσωματωθεί σε υπάρχουσες αρχιτεκτονικές NoC με ελάχιστες τροποποιήσεις.

7 Επίλογος

Η παρούσα πτυχιακή εργασία ασχολήθηκε με τη μελέτη και την ανάπτυξη αλγορίθμων δρομολόγησης σε αρχιτεκτονικές πολυπύρηνων επεξεργαστών, με έμφαση στα Ολοκληρωμένα Ενδο-συνδεδεμένα Δίκτυα. Στο πλαίσιο αυτό σχεδιάστηκε και υλοποιήθηκε ο αλγόριθμος APAR, αναπτύχθηκε πειραματικό περιβάλλον προσομοίωσης σε C++17 και πραγματοποιήθηκε εκτεταμένη συγκριτική αξιολόγηση με αλγορίθμους αναφοράς σε διαφορετικά μεγέθη πλέγματος, πρότυπα κυκλοφορίας και επίπεδα φόρτου.

Τα αποτελέσματα έδειξαν ότι η αποδοτικότητα ενός αλγορίθμου δρομολόγησης δεν μπορεί να εξετάζεται απομονωμένα, αλλά σε συνάρτηση με τις συνθήκες λειτουργίας του δικτύου και τους στόχους βελτιστοποίησης. Ο APAR ανέδειξε ενδιαφέροντα πλεονεκτήματα σε συγκεκριμένα σενάρια, ενώ παράλληλα επιβεβαιώθηκε ότι υπάρχουν σαφείς συμβιβασμοί μεταξύ διεκπεραιωτικής ικανότητας και καθυστέρησης, οι οποίοι απαιτούν προσεκτική ερμηνεία.

Συνολικά, η εργασία επιδίωξε να συμβάλει ουσιαστικά στη μελέτη της προσαρμοστικής δρομολόγησης σε NoC συστήματα, συνδυάζοντας θεωρητική τεκμηρίωση, υλοποίηση και πειραματική αξιολόγηση. Επομένως, η πτυχιακή αυτή δεν ολοκληρώνεται μόνο με την παρουσίαση ενός αλγορίθμου, αλλά με την ανάδειξη της ανάγκης για πιο ευέλικτες και τεκμηριωμένες προσεγγίσεις δρομολόγησης στα μελλοντικά πολυπύρηννα συστήματα.

Παράρτημα Α

Αναπαραγωγιμότητα και Υλικά Εργασίας

Στο παρόν παράρτημα παρατίθενται τα artefacts (κώδικας, δεδομένα, scripts) και οι εντολές που επιτρέπουν την αναπαραγωγή των πειραμάτων και των διαγραμμάτων της εργασίας.

Συνοδευτικός φάκελος του έργου

Ο πηγαίος κώδικας του προσομοιωτή, οι υλοποιήσεις των αλγορίθμων, τα scripts αυτοματοποίησης και τα παραγόμενα datasets παραδίδονται μαζί με την εργασία στον συνοδευτικό φάκελο του project (ρίζα: `proc_arch/`). Ο φάκελος αυτός περιλαμβάνει:

- `src/`: υλοποίηση προσομοιωτή και αλγορίθμων (C++17)
- `scripts/`: αυτοματοποίηση πειραμάτων/ανάλυσης/διαγραμμάτων (PowerShell/Python)
- `results/`: datasets και παράγωγα CSV
- `thesis/figures/`: τελικά διαγράμματα/εικόνες

Dataset (δεδομένα πειραμάτων)

Τα βασικά αρχεία που χρησιμοποιούνται στο Κεφάλαιο 5 είναι:

- `results/final_experiments/results_20251223_170145.csv` (raw αποτελέσματα)
- `results/final_experiments/results_20251223_170145_summary_ci.csv` (σύννοψη: mean/std/95% CI)
- `results/final_experiments/results_20251223_170145_saturation_points.csv` (peak throughput / ενδεικτικά σημεία κορεσμού)
- `thesis/figures/final/` (τελικά διαγράμματα σε PNG/PDF)

Αυτοματοποίηση πειραμάτων και ανάλυσης

Η εκτέλεση της πλήρους πειραματικής μήτρας αυτοματοποιείται μέσω PowerShell/Python scripts:

- `scripts/run_final_experiments.ps1`: εκτέλεση όλων των συνδυασμών (*mesh, traffic, IR, run*) και αποθήκευση raw αποτελεσμάτων σε CSV.
- `scripts/analyze_final_experiments.py`: υπολογισμός σύννοψης (mean/std/95% CI) και βοηθητικών μετρικών (peak throughput / ενδεικτικά saturation points).
- `scripts/generate_final_charts.py`: παραγωγή διαγραμμάτων (throughput/latency/scalability) από το dataset.
- `scripts/generate_thesis_extra_figures.py`: παραγωγή πρόσθετων διαγραμμάτων για το Κεφάλαιο 5 (multi-panel όλων των προτύπων, συγκρίσεις ανά πρότυπο και εστίαση κοντά στον κορεσμό).

Εντολές αναπαραγωγής

(Α) Αναπαραγωγή διαγραμμάτων/πινάκων από το dataset του συνοδευτικού φακέλου

```
# Build simulator + install Python deps
.\build.ps1 release
python -m pip install -r .\scripts\requirements.txt
```

```
# Παραγωγή συνοπτικών CSV (mean/std/95% CI) και διαγραμμάτων από το dataset
python .\scripts\analyze_final_experiments.py results\final_experiments\results_20251223_170145.csv
python .\scripts\generate_final_charts.py results\final_experiments\results_20251223_170145.csv
python .\scripts\generate_thesis_extra_figures.py results\final_experiments\results_20251223_170145_summary_ci.csv
```

(B) Πλήρης επανεκτέλεση πειραμάτων (παράγει νέο timestamped CSV)

```
powershell -ExecutionPolicy Bypass -File .\scripts\run_final_experiments.ps1
python .\scripts\analyze_final_experiments.py
python .\scripts\generate_final_charts.py
python .\scripts\generate_thesis_extra_figures.py
```

Πολιτική σπόρων τυχειότητας (common random numbers)

Για κάθε διαμόρφωση (*mesh, traffic, IR, run*) χρησιμοποιείται σταθερός seed, ο οποίος παραμένει ίδιος για όλους τους αλγορίθμους που συγκρίνονται. Πρακτικά, αυτό σημαίνει ότι σε ένα συγκεκριμένο (*mesh, traffic, IR, run*) όλα τα πειράματα “βλέπουν” την ίδια ακολουθία τυχειότητας (π.χ. επιλογές προορισμών/ενέσεων όπου υπάρχει τυχειότητα), ώστε οι διαφορές στα αποτελέσματα να οφείλονται στον αλγόριθμο δρομολόγησης και όχι σε διαφορετικό δείγμα τυχειότητας. Έτσι οι συγκρίσεις είναι **ζευγαρωμένες** (paired) και μειώνεται η διακύμανση. Η πολιτική seeds υλοποιείται στο scripts/run_final_experiments.ps1.

Βιβλιογραφία

- [1] W. J. Dally and B. Towles, “Route packets, not wires: On-chip interconnection networks,” in *Proceedings of the 38th Design Automation Conference (DAC)*, Las Vegas, NV, USA, 2001, pp. 684-689.
- [2] L. Benini and G. De Micheli, “Networks on chips: A new SoC paradigm,” *Computer*, vol. 35, no. 1, pp. 70-78, Jan. 2002.
- [3] C. J. Glass and L. M. Ni, “The turn model for adaptive routing,” *Journal of the ACM*, vol. 41, no. 5, pp. 874-902, Sep. 1994.
- [4] G.-M. Chiu, “The odd-even turn model for adaptive routing,” *IEEE Transactions on Parallel and Distributed Systems*, vol. 11, no. 7, pp. 729-738, Jul. 2000.
- [5] W. J. Dally and B. Towles, *Principles and Practices of Interconnection Networks*. San Francisco, CA, USA: Morgan Kaufmann, 2004.
- [6] J. Duato, S. Yalamanchili, and L. Ni, *Interconnection Networks: An Engineering Approach*. San Francisco, CA, USA: Morgan Kaufmann, 2003.
- [7] M. Li, Q.-A. Zeng, and W.-B. Jones, “DyXY: A proximity congestion-aware deadlock-free dynamic routing method for network on chip,” in *Proceedings of the 43rd Design Automation Conference (DAC)*, San Francisco, CA, USA, 2006, pp. 849-852.
- [8] J. Hu and R. Marculescu, “DyAD: Dynamic arbitration and routing for network-on-chip,” in *Proceedings of the Design, Automation and Test in Europe Conference (DATE)*, Paris, France, 2004, pp. 260-265.
- [9] P. Gratz, B. Grot, and S. W. Keckler, “Regional congestion awareness for load balance in networks-on-chip,” in *Proceedings of the 14th International Symposium on High Performance Computer Architecture (HPCA)*, Salt Lake City, UT, USA, 2008, pp. 203-214.
- [10] Y. Feng, Z. Lu, and A. Jantsch, “Q-learning based fault-tolerant routing for network-on-chip,” in *Proceedings of the 2010 IEEE International Symposium on Circuits and Systems (ISCAS)*, Paris, France, 2010, pp. 3285-3288.
- [11] F. Farahnakian, M. Ebrahimi, M. Daneshtalab, P. Liljeberg, and J. Plosila, “Q-learning based congestion-aware routing algorithm for on-chip network,” in *Proceedings of the 2011 IEEE International Conference on Networking, Architecture and Storage (NAS)*, Dalian, China, 2011, pp. 392-397.
- [12] K. Zheng and A. Louri, “An energy-efficient network-on-chip design using reinforcement learning,” in *Proceedings of the 56th ACM/IEEE Design Automation Conference (DAC)*, Las Vegas, NV, USA, 2019, pp. 1-6.
- [13] K. Wang and A. Louri, “CURE: A high-performance, low-power, and reliable network-on-chip design using reinforcement learning,” *IEEE Transactions on Parallel and Distributed Systems*, vol. 31, no. 9, pp. 2125-2138, Sep. 2020.
- [14] S. Ma, N. E. Jerger, and Z. Wang, “DBAR: An efficient routing algorithm to support multiple concurrent applications in networks-on-chip,” in *Proceedings of the 38th Annual International Symposium on Computer Architecture (ISCA)*, San Jose, CA, USA, 2011, pp. 413-424.

- [15] A. K. Mishra, R. Das, S. Eachempati, R. Iyer, N. Vijaykrishnan, and C. R. Das, "A case for dynamic frequency tuning in on-chip networks," in *Proceedings of the 42nd Annual IEEE/ACM International Symposium on Microarchitecture (MICRO)*, New York, NY, USA, 2009, pp. 292-303.
- [16] D. Wentzlaff et al., "On-chip interconnection architecture of the Tile Processor," *IEEE Micro*, vol. 27, no. 5, pp. 15-31, Sep.-Oct. 2007.
- [17] B. Grot, J. Hestness, S. W. Keckler, and O. Mutlu, "Express cube topologies for on-chip interconnects," in *Proceedings of the 15th International Symposium on High Performance Computer Architecture (HPCA)*, Raleigh, NC, USA, 2009, pp. 163-174.
- [18] A. Kumar, L.-S. Peh, P. Kundu, and N. K. Jha, "Express virtual channels: Towards the ideal interconnection fabric," in *Proceedings of the 34th Annual International Symposium on Computer Architecture (ISCA)*, San Diego, CA, USA, 2007, pp. 150-161.
- [19] N. Jiang, D. U. Becker, G. Michelogiannakis, J. Balfour, B. Towles, D. E. Shaw, J. Kim, and W. J. Dally, "A detailed and flexible cycle-accurate network-on-chip simulator," in *Proceedings of the 2013 IEEE International Symposium on Performance Analysis of Systems and Software (ISPASS)*, Austin, TX, USA, 2013, pp. 86-96.
- [20] A. B. Kahng, B. Li, L.-S. Peh, and K. Samadi, "ORION 2.0: A power-area simulator for interconnection networks," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 20, no. 1, pp. 191-196, Jan. 2012.
- [21] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*, 2nd ed. Cambridge, MA, USA: MIT Press, 2018.
- [22] V. Mnih et al., "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, pp. 529-533, Feb. 2015.
- [23] J. Kim, D. Park, T. Theocharides, N. Vijaykrishnan, and C. R. Das, "A low latency router supporting adaptivity for on-chip interconnects," in *Proceedings of the 42nd Design Automation Conference (DAC)*, Anaheim, CA, USA, 2005, pp. 559-564.
- [24] R. Mullins, A. West, and S. Moore, "Low-latency virtual-channel routers for on-chip networks," in *Proceedings of the 31st Annual International Symposium on Computer Architecture (ISCA)*, Munich, Germany, 2004, pp. 188-197.
- [25] L.-S. Peh and W. J. Dally, "A delay model and speculative architecture for pipelined routers," in *Proceedings of the 7th International Symposium on High-Performance Computer Architecture (HPCA)*, Monterrey, Mexico, 2001, pp. 255-266.
- [26] S. Wang, X. Zhang, C. Wang, K. Wu, C. Li, and D. Dong, "DRLAR: A deep reinforcement learning-based adaptive routing framework for network-on-chips," *Computer Networks*, vol. 246, Art. no. 110419, 2024, doi: 10.1016/j.comnet.2024.110419.
- [27] R. Raj R.S. et al., "DeepNR: An adaptive deep reinforcement learning based NoC routing algorithm," *Microprocessors and Microsystems*, vol. 90, Art. no. 104485, 2022, doi: 10.1016/j.micpro.2022.104485.
- [28] M. Trik, H. Akhavan, A. M. Bidgoli, A. M. Norouzzadeh Gil Molk, H. Vashani, and S. Pour Mozaffari, "A new adaptive selection strategy for reducing latency in networks on chip," *Integration*, vol. 89, pp. 9-24, 2023, doi: 10.1016/j.vlsi.2022.11.004.
- [29] L. Kondoth, R. Shankaran, Q. Z. Sheng, E. Kuantama, and W. Ni, "A Survey of Adaptive Routing Techniques in Two-Dimensional Network-on-Chip Architectures," *IEEE*

Communications Surveys & Tutorials, vol. 28, pp. 3195-3234, 2026, doi: 10.1109/COMST.2025.3636503.

[30] Y. Zhang, Z. Jing, Q. Yang, N. Cheng, H. Gu, and K. S. Kwak, "A survey on routing algorithm and router microarchitecture of three-dimensional Network-on-Chip," *Journal of Systems Architecture*, vol. 164, Art. no. 103429, 2025, doi: 10.1016/j.sysarc.2025.103429.