



**ΔΗΜΟΚΡΙΤΕΙΟ
ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΡΑΚΗΣ**

**ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ
ΜΗΧΑΝΙΚΩΝ & ΜΗΧΑΝΙΚΩΝ
ΥΠΟΛΟΓΙΣΤΩΝ**

**ΤΟΜΕΑΣ ΛΟΓΙΣΜΙΚΟΥ ΚΑΙ ΑΝΑΠΤΥΞΗΣ
ΕΦΑΡΜΟΓΩΝ**

**ΔΙΑΔΙΚΤΥΟ ΚΑΤ' ΑΠΑΙΤΗΣΗ: ΓΕΩΓΡΑΦΙΚΗ ΕΠΕΚΤΑΣΗ ΤΟΥ
ΔΙΑΔΙΚΤΥΟΥ**

Χρήστος Νάτσης

ΑΜ:56013

Επιβλέπων καθηγητής: Βασίλειος Τσαουσίδης

**Ξάνθη, Μάρτιος
2020**

**Αφιερώνεται στους γονείς μου, οι οποίοι με στήριξαν με κάθε πρόσφορο
μέσο, στην όλη διαδρομή**

Ευχαριστίες

Θα επιθυμούσα πρωτίστως να ευχαριστήσω τον καθηγητή κ. Βασίλειο Τσαουσίδα, για την εμπιστοσύνη που μου έδειξε κατά την ανάθεση του θέματος. Η διδασκαλία βασικών δικτυακών εννοιών από μέρους του, αποτέλεσε πηγή έμπνευσης για την περαιτέρω ενασχόληση μου, με τον τομέα των Δικτύων.

Επιπλέον, ευχαριστώ ενθέρμως τον υποψήφιο διδάκτορα κ. Αλέξανδρο Σάρρο, για την ανεξάντλητη αρωγή, στήριξη και συνδρομή, κατά την εκπόνηση της διπλωματικής εργασίας.

Πίνακας περιεχομένων

| | | |
|--------|--|----|
| 1 | Εισαγωγή | 1 |
| 1.1 | Το Διαδίκτυο..... | 1 |
| 1.2 | Μετάβαση στο πληροφοριό – κεντρικό Διαδίκτυο (ICN) | 3 |
| 1.2.1 | Επικέντρωση στην Κινητικότητα | 5 |
| 1.2.2 | Επικέντρωση στην Ασφάλεια | 5 |
| 1.3 | Σκοπός | 7 |
| 1.4 | Δομή πτυχιακής..... | 7 |
| 2 | Η μετάβαση στο NDN | 8 |
| 2.1 | Δομή κλεψύδρας (hourglass) | 8 |
| 2.2 | Οριοθετημένος χώρος ονομάτων | 9 |
| 2.3 | Βασικές αρχές σχεδίασης..... | 9 |
| 2.4 | Η αρχιτεκτονική Named Data Networking (NDN) | 10 |
| 2.4.1 | Περαιτέρω ανάλυση του NDN | 11 |
| 2.4.2 | Ασφάλεια..... | 12 |
| 2.5 | Θέση του της IP διαστρωμάτωσης στο NDN μοντέλο | 13 |
| 2.6 | Επεξήγηση πακέτων NDN αρχιτεκτονικής | 14 |
| 2.6.1 | Πακέτο Ενδιαφέροντος – Interest Packet | 15 |
| 2.6.2 | Πακέτο Δεδομένων – Data Packet | 16 |
| 2.6.3 | Αρνητική Επιβεβαίωση (Negative Acknowledgment – NACK) | 17 |
| 2.7 | Ονόματα – Names | 17 |
| 2.8 | Κωδικοποίηση TLV (Type-Length-Value)..... | 18 |
| 2.9 | Εσωτερική δομή και Προώθηση | 19 |
| 2.9.1 | Πίνακας Περιεχομένου – Content Store | 19 |
| 2.9.2 | Πρακτική Εφαρμογή..... | 20 |
| 2.10 | Πίνακας εν αναμονή Ενδιαφερόντων (Pending Interest Table) | 20 |
| 2.10.1 | Καταχώριση PIT (PIT Entry) | 21 |
| 2.10.2 | Εσωτερική καταγραφή (In Record) | 21 |
| 2.10.3 | Εξωτερική Καταγραφή (Out Record)..... | 22 |
| 2.10.4 | Χρονικοί δείκτες – Timers | 22 |
| 2.10.5 | Πίνακας εν αναμονή Ενδιαφερόντων (Pending Interest Table – PIT)..... | 22 |
| 2.11 | Λίστα ανίχνευσης επανάληψης (Dead Nonce List) | 23 |
| 2.12 | Πίνακας Προώθησης Ενδιαφέροντος (Forwarding Information Base) | 24 |
| 2.12.1 | Δομή και Σημασιολογία..... | 24 |
| 2.12.2 | Καταχώριση FIB και καταγραφή επόμενου Βήματος (NextHop)..... | 24 |
| 2.13 | Επιλογή στρατηγικής (Strategy) | 25 |

| | | |
|-------|--|----|
| 2.14 | Πίνακας Μετρήσεων | 26 |
| 3 | Προσομοιωτής ndnSIM και ενσωμάτωση του NFD | 28 |
| 3.1 | Σύστημα Οντοτήτων | 28 |
| 3.1.1 | Επίπεδο αφαίρεσης Face URI | 29 |
| 3.1.2 | Επίπεδο αφαίρεσης Protocol Factory | 30 |
| 3.1.3 | Επίπεδο Αφαίρεσης Αγωγού | 30 |
| 3.2 | Πρώθηση στο NDN | 31 |
| 3.3 | Αγωγοί πρώθησης | 32 |
| 3.4 | Αγωγός Επεξεργασίας Ενδιαφέροντος | 32 |
| 3.5 | Αγωγός Εισερχόμενων Αιτημάτων Ενδιαφέροντος | 33 |
| 3.6 | Αγωγός Επεξεργασίας Δεδομένων | 34 |
| 3.6.1 | Αγωγός Εισερχόμενων Δεδομένων | 35 |
| 3.6.2 | Αγωγός Αυτόκλητων Δεδομένων(Unsolicited) | 36 |
| 3.6.3 | Αγωγός Εξερχόμενων Δεδομένων | 36 |
| 3.7 | Διαχείριση Συμφόρησης στο NDN | 36 |
| 3.8 | Προσομοιωτής ndnSIM | 37 |
| 4 | Σενάρια και Εργαλεία Αξιολόγησης | 39 |
| 4.1 | Εφαρμογές παραγωγής πακέτων Ενδιαφέροντος ndnSIM | 39 |
| 4.2 | Τοπολογία και Υλοποίηση στον ndnSIM | 39 |
| 4.3 | Περιγραφή πειράματος | 45 |
| 4.4 | Προγραμματιστικό περιβάλλον R | 46 |
| 4.5 | Υλοποιημένα R πρότυπα | 47 |
| 4.5.1 | Throughput | 47 |
| 4.5.2 | Επιτυχής Ανάκτηση (CacheHits) | 48 |
| 4.5.3 | Συνολική καθυστέρηση (Delay) | 48 |
| 4.5.4 | Χρόνος ζωής πακέτου Δεδομένων (Cache Entry Lifetime) | 49 |
| 4.5.5 | Βιβλιοθήκες | 49 |
| 5 | Αποτελέσματα | 50 |
| 5.1 | Σενάριο 1° | 51 |
| 5.1.1 | Μεταβλητό μέγεθος πακέτου δεδομένων | 51 |
| 5.1.2 | Μεταβλητή τιμή καθυστέρησης διάδοσης | 60 |
| 5.2 | Σενάριο 2° | 66 |
| 5.2.1 | Μεταβλητό μέγεθος πακέτων δεδομένων | 66 |
| 5.2.2 | Μεταβλητό μέγεθος καθυστέρησης διάδοσης | 72 |
| 5.3 | Σενάριο 3° | 78 |
| 5.3.1 | Μεταβλητό μέγεθος πακέτων δεδομένων | 78 |

| | | |
|-------|--|----|
| 5.3.2 | Μεταβλητό μέγεθος καθυστέρησης διάδοσης..... | 84 |
| 6 | Μεθοδολογία-Συμπεράσματα | 90 |
| 6.1 | Μεθοδολογία | 90 |
| 6.2 | Συμπεράσματα | 91 |
| 6.3 | Προτάσεις..... | 93 |
| 7 | Βιβλιογραφία και ηλεκτρονικές αναφορές..... | 94 |

Πίνακας εικόνων

| | |
|--|-----------|
| Εικόνα 1 Χρονική εξέλιξη των IoT διασυνδεδεμένων συσκευών [2] | 2 |
| Εικόνα 2 Συγκριτική παρουσίαση της δομής hourglass μεταξύ του σημερινού διαδικτυακού μοντέλου και της NDN υλοποίησης [10] | 12 |
| Εικόνα 3 Δομή των πακέτων Ενδιαφέροντος και Δεδομένων της αρχιτεκτονικής Named Data Networking[11]..... | 15 |
| Εικόνα 4 Διαδικασία επεξεργασίας πακέτου Ενδιαφέροντος από τον Πίνακα[PIT] Εν αναμονή Ενδιαφερόντων[15]..... | 21 |
| Εικόνα 5 Αξιοποίηση του Forwarding Information Base για δρομολόγηση των πακέτων Ενδιαφέροντος στις πιθανές οντότητες Προσώπου | 24 |
| Εικόνα 6 Συνοπτική περιγραφή του συστήματος οντοτήτων της αρχιτεκτονικής Named Data Networking | 29 |
| Εικόνα 7 Διαδικασία προώθησης των πακέτων Ενδιαφερόντων και Δεδομένων[15]..... | 32 |
| Εικόνα 8 Απεικόνιση του μονοπατιού επεξεργασίας του πακέτου Ενδιαφέροντος[15]..... | 33 |
| Εικόνα 9 Απεικόνιση του μονοπατιού επεξεργασίας πακέτου Δεδομένου[15]..... | 35 |
| Εικόνα 10 Συνοπτική περιγραφή της εσωτερικής δομής λειτουργίας του ndn Simulator | 37 |
| Εικόνα 11 Απεικόνιση της υλοποιημένης dumbbell τοπολογίας | 40 |
| Εικόνα 12 Στιγμιότυπο του προσομοιωτή ndnSIM, όπου διαφαίνεται η δυνατότητα εισαγωγής πολλαπλών τιμών από τον χρήστη | 45 |
| Εικόνα 13 Επιτυχής εκτέλεση στιγμιότυπου προσομοίωσης..... | 45 |

Πρόλογος

Η ραγδαία ενσωμάτωση της τεχνολογίας στον σύγχρονο τρόπο ζωής αποτελεί αναντίρρητο χαρακτηριστικό της ισχύουσας κοινωνικής δομής. Το Διαδίκτυο ως ακρογωνιαίος λίθος συνετέλεσε στην αλλαγή του τρόπου επικοινωνίας, μέσω των πολλαπλών συσκευών, που αποτέλεσαν κομμάτι του λειτουργικού του μοντέλου. Ειδικότερα, η αυξητική τάση ενεργοποίησης νέων, κινητών και μη συσκευών, σε συνδυασμό με την ανάγκη αξιοποίησης δεδομένων σε συνθήκες πραγματικού χρόνου, οδήγησε νομοτελειακά στον κορεσμό της υπάρχουσας υποδομής.

Συμπερασματικά, ανέκυψε η ανάγκη για περαιτέρω διεύρυνση της υπάρχουσας Δικτυακής αρχιτεκτονικής, βάσει ενός μοντέλου που θα αντιλαμβάνεται ως δομικό του στοιχείο την πληροφορία και όχι την τοποθεσία που αυτή εντοπίζεται.

1 Εισαγωγή

1.1 Το Διαδίκτυο

Τα προβλήματα της υπάρχουσας δομής Διαδικτύου αποτελούν φυσική συνέπεια της αρχιτεκτονικής που σχεδιάστηκε για να αντιμετωπίσει τις ανάγκες επικοινωνίας μιας περασμένης εποχής. Η βασική απαίτηση από πλευράς Διαδικτύου σε εκείνη τη χρονική στιγμή, ήταν η προώθηση πακέτων δεδομένων μεταξύ πολυπληθών στατικών μηχανημάτων. Οι σχεδιαστικές αρχές του Διαδικτύου καθιστούσαν εύκολη τη διασύνδεση νέων δικτύων, επιτρέποντας με τον τρόπο αυτό, την τεράστια ανάπτυξη του. Παράλληλα με την εκτεταμένη ανάπτυξη, προέκυψε ένας αξιοσημείωτος αριθμός καινοτομιών στο επίπεδο των εφαρμογών και των υπηρεσιών, όπως επίσης και στο σκέλος των διάφορων τεχνολογιών του επιπέδου δικτυακής διαστρωμάτωσης.

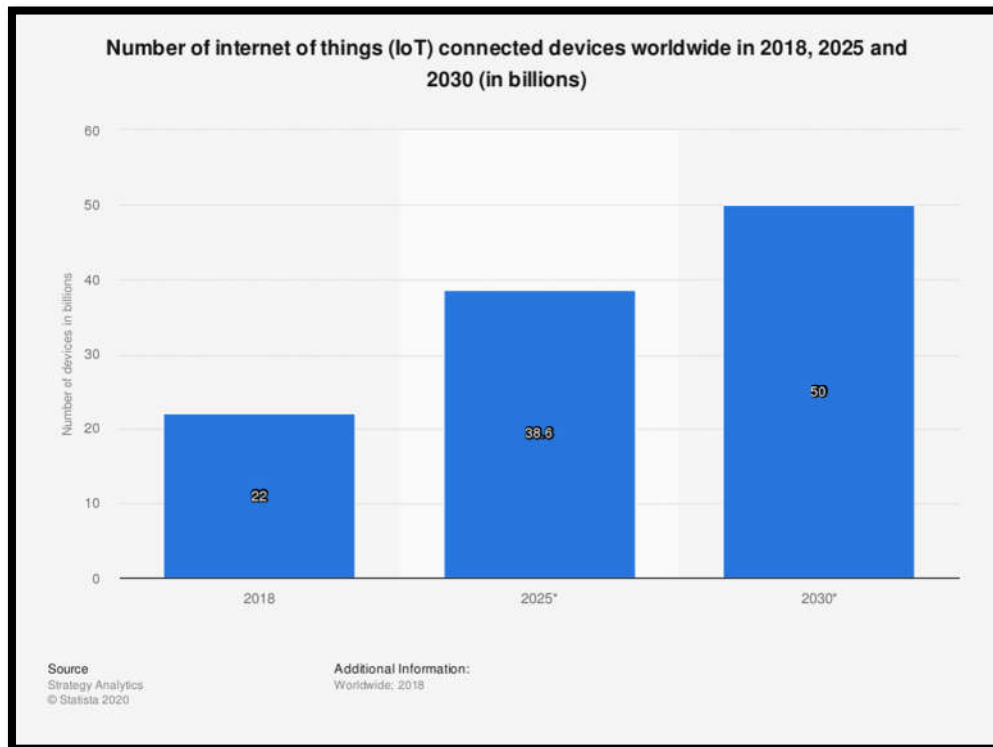
Το Διαδίκτυο όπως το γνωρίζουμε σήμερα βασίστηκε στο προ υπάρχον τηλεφωνικό δίκτυο [1] και σταδιακά μετατράπηκε από ένα δίκτυο βασισμένο στη διάδοση bit, σε ένα δίκτυο βασισμένο στην IP αρχιτεκτονική πακέτων. Το διαδικτυακό μοντέλο στηρίζεται ακόμη στην ίδια θεμελιώδη αρχή, αφού επικεντρώνεται στον εντοπισμό της πληροφορίας και τη διασφάλιση της σύζευξης μεταξύ του χρήστη και της πηγής, δημιουργώντας μια ζεύξη μεταξύ των τερματικών κόμβων.

Η εμπορική διάθεση του Διαδικτύου εντοπίζεται στο έτος 1993. Έκτοτε, δεν πραγματοποιήθηκαν σημαντικές αλλαγές στη βάση της αρχιτεκτονικής δομής. Αυτό συνιστά ότι οι αλλαγές συμβαίνουν εξ ανάγκης για τους κατασκευαστές, είτε για να παραμείνουν ενεργοί στο τεχνολογικό τοπίο είτε για να διασφαλίσουν μια νέα υπηρεσία, προς τους πελάτες. Η αξιοποίηση μιας συγκεκριμένης αρχιτεκτονικής, διατηρώντας παράλληλα τη δυνατότητα της καινοτομίας και της ευελιξίας, αποτέλεσε θεμέλιο λίθο για την ελαχιστοποίηση του λειτουργικού κόστους.

Ο αρχιτεκτονικός σχεδιασμός του πρώιμου Διαδικτύου πραγματοποιήθηκε με τρόπο που ανταποκρινόταν στις τότε επερχόμενες ανάγκες, αξιοποιώντας στο μέγιστο βαθμό τους διαθέσιμους πόρους. Το κόστος αποθήκευσης αποτελούσε πρόκληση, οπότε ο διαμοιρασμός αρχείων υπήρξε ακριβή μέθοδος. Επιπλέον, ο εξοπλισμός ήταν ογκώδης και επομένως στατικός, καθιστώντας όλα τα διαθέσιμα μέσα ημι-μόνιμα και καλωδιωμένα.

Το Διαδίκτυο μετατράπηκε ραγδαία από ένα μικρό επιστημονικό δίκτυο σε παγκόσμιο, ενσωματώνοντας έναν τεράστιο όγκο στατικών και κινητών μονάδων. Το πέρασμα στην εποχή του Διαδικτύου των Πραγμάτων θα οδηγήσει σε δραματική αύξηση των διασυνδεδεμένων συσκευών, αλλάζοντας οριστικά και αμετάκλητα το τοπίο της σημερινής διαδικτυακής πραγματικότητας.

Παρακάτω παρατίθεται η γραφική απεικόνιση των συσκευών IoT κατά το 2018, όπως επίσης η στατιστική πρόβλεψη της επόμενης δεκαετίας :



Εικόνα 1 Χρονική εξέλιξη των IoT διασυνδεδεμένων συσκευών [2]

Η σημερινή δομή διαδικτύου διαφοροποιείται από το υφιστάμενο μοντέλο. Αυτό συμβαίνει λόγω της έλευσης μιας πληθώρας κινητών συσκευών, του χαμηλότερου κόστους απόκτησης αυτών, της κυρίαρχης δημογραφίας των χρηστών, καθώς και των διεργασιών που εκτελούνται από τις διάφορες συσκευές. Οι χρήστες καταναλώνουν εν κινήσει τεράστια ποσά δεδομένων, ενώ ταυτόχρονα παράγουν κατ' αντιστοιχία δεδομένα. Τα δεδομένα που καταναλώνονται, διαμοιράζονται ταυτοχρόνως, όπως στις υπηρεσίες βίντεο διαμοιρασμού, όπου πολλοί χρήστες τα επιζητούν σε συνθήκες πραγματικού χρόνου.

Το χρησιμοποιούμενο μοντέλο διασύνδεσης “τερματικών κόμβων” μεταξύ του πελάτη και του διακομιστή είναι στατικό και αποτελεί μέρος του συνήθους τρόπου επικοινωνίας του τηλεφωνικού δικτύου, μεταξύ των τερματικών σημείων.

Παρότι καθίσταται σαφής η αδιαμφισβήτητη υπεροχή της σημερινής αρχιτεκτονικής Διαδικτύου, όπου η λογική δομή της επικεντρώνεται στο “τι” των στρωμάτων υψηλού επιπέδου συνδυαστικά με το “που” των στρωμάτων χαμηλού επιπέδου, δε διασφαλίζεται η λειτουργία της με βέλτιστο τρόπο.

Η τεράστια ανάπτυξη του, σε συνδυασμό με την εισαγωγή νέων εφαρμογών ικανοποίησης των εντεινόμενων αναγκών, οδήγησε σε αναθεώρηση των απαιτήσεων που αφορούν τη δομή. Η ανάγκη υποστήριξης της διανομής περιεχομένου, της κινητικότητας, της ασφάλειας, της εμπιστοσύνης αποτελούν χαρακτηριστικά παραδείγματα, των νέων απαιτήσεων. Ωστόσο, το Διαδίκτυο δεν σχεδιάστηκε εξ αρχής για την εκτενή ικανοποίηση τους. Η συνεχής εξέλιξη τους προέκυψε μέσα από έναν ατέρμων κύκλο λειτουργικών τροποποιήσεων, όπως λόγου χάριν η εφαρμογή του κινητού πρωτοκόλλου IP(Mobile IP)[3].

Η πλειοψηφία των τροποποιήσεων αύξησαν την πολυπλοκότητα της αρχιτεκτονικής και ως εκ τούτου, αποτέλεσαν προσωρινές λύσεις. Επιπλέον, πολλαπλές πρόσφατες και επείγουσες ανάγκες, δεν μπορούν ακόμη να διευθετηθούν επαρκώς. Επομένως, προέκυψε το λογικό ερώτημα για το αν θα συνεχιστούν οι “τροποποιήσεις” ή θα πρέπει να οδηγηθούμε σε μια νέα αρχιτεκτονική προσέγγιση, που θα είναι σε θέση να διασφαλίζει τις σύγχρονες ανάγκες.

1.2 Μετάβαση στο πληροφοριό – κεντρικό Διαδίκτυο (ICN)

Το Διαδίκτυο καλείται να διαχειριστεί τεράστια ποσά πληροφορίας και κίνησης, λόγω της συνεχούς πορείας μεταστροφής προς την κατεύθυνση των εφαρμογών περιεχομένου. Ωστόσο, η α-γνωστική δομή δεδομένων της αρχιτεκτονικής Διαδικτύου δε διαθέτει εγγενείς μηχανισμούς διαχείρισης συμβάντων διαχείρισης πλήθους(flash crowd) και αποδοτικής λειτουργίας παράδοσης πληροφοριών. Η τρέχουσα δομή Διαδικτύου αντιμετωπίζει τα δεδομένα υπό διαμετακόμιση ως μια σειρά από byte, τα οποία πρέπει να μεταφερθούν από μία συγκεκριμένη πηγή, προς ένα συγκεκριμένο προορισμό. Επομένως, τα στοιχεία του δικτύου δε διαθέτουν γνώση επί των πληροφοριών που μεταφέρονται. Ως εκ τούτου, δεν γίνονται αντιληπτές οι διάφορες

βελτιστοποιήσεις από πλευράς δικτύου, όπως η προσωρινή αποθήκευση και η μηχανική εντοπισμού κίνησης.

Στη θεμελιώδη βάση της αρχιτεκτονικής ICN[4] η πληροφορία καθ' αυτή αποσυνδέεται από τις πηγές παραγωγής. Στη βασική της προσέγγιση, η πληροφορία ονοματοδοτείται, διευθετείται και εντοπίζεται ανεξαρτήτως τοποθεσίας, οπότε και δύναται ο εντοπισμός της σε οποιοδήποτε σημείο του δικτύου. Δηλαδή, η αρχιτεκτονική ICN δεν συγκεκριμενοποιεί το δίδυμο εξυπηρετητής-πηγή περιεχομένου, αλλά ένα δεδομένο κομμάτι πληροφορίας. Η άμεση διαφοροποίηση της μετάβασης από το βασιζόμενο στον εξυπηρετητή μοντέλο σε ένα μοντέλο ονοματολογίας πληροφοριών, βασίζεται στην καθοδήγηση ανάκτησης της εκάστοτε πληροφορίας, από πλευράς δέκτη.

Σε αντίθεση με την υπάρχουσα υλοποίηση όπου οι αποστολείς έχουν τον απόλυτο έλεγχο στα δεδομένα τα οποία παράγουν, στο ICN καμία πληροφορία δεν μπορεί να ληφθεί, εκτός και αν ζητηθεί ρητά από το δέκτη. Η ύπαρξη ενός αιτήματος οδηγείται βάσει της δυνατότητας εντοπισμού της βέλτιστης πηγής παροχής πληροφορίας. Επομένως, η δρομολόγηση των αιτημάτων πληροφορίας επιδιώκει τον εντοπισμό των καλύτερων πηγών πληροφορίας, βάσει του ονόματος και όχι της τοποθεσίας.

Για να αξιοποιηθούν πλήρως οι υφιστάμενες δυνατότητες αποθήκευσης από πλευράς Διαδικτύου, πρέπει να επεκταθούν οι μηχανισμοί ανίχνευσης και ανάκτησης πληροφοριών, με βάση την βέλτιστη τοποθεσία. Η εμφάνιση τέτοιων τεχνικών σε επίπεδο εφαρμογών, μπορεί εκ των προτέρων να επιβεβαιώσει το ενδεχόμενο ύπαρξης μιας διαφορετικής αρχιτεκτονικής Διαδικτύου, μελλοντικά.

Αντ' αυτού, είναι αναγκαία η ανάπτυξη μιας ευρεία υποδομής Διαδικτύου που θα διαθέτει εσωτερικούς μηχανισμούς αποτελεσματικής ανάκτησης της πληροφορίας.

Στη διαδικτυακή αρχιτεκτονική ICN ένα αίτημα πληροφορίας μπορεί να ικανοποιηθεί όχι μόνο βάσει της εντοπισμένης πηγής, αλλά με παράλληλη αξιοποίηση των εσωτερικών προσωρινών μνημών (cache), οι οποίες διαθέτουν αντίγραφα της επιθυμητής πληροφορίας (ή κομμάτια της). Οι αρχιτεκτονικές ICN αντιλαμβάνονται τα πακέτα δεδομένων, κατ' αντιπαραβολή του ονόματος τους με πακέτο που μεταφέρουν. Επομένως, κομμάτια της πληροφορίας μπορούν να αποθηκευτούν προσωρινά και να ανακτηθούν εύκολα, σε αντίθεση με την τωρινή δομή διαδικτύου όπου δαπανηρές τεχνικές όπως η "Ανάλυση πακέτων σε Βάθος" (Deep Packet Inspection),

αξιοποιούνται για την ικανοποίηση αντίστοιχων αναγκών. Επιπλέον, η ονοματισμένη πληροφορία του ICN επιτρέπει τη συγκεντρωτική αποστολή των αιτημάτων για την ίδια πληροφορία, διευκολύνοντας κατά αυτόν τον τρόπο την παράδοσή τους στους αντίστοιχους προορισμούς, μέσω της πολιτικής προώθησης πολλαπλών μεταδόσεων.

Ακόμη μπορούν να εφαρμόζονται δυνητικά, έλεγχοι πρόσβασης στο επίπεδο διαστρωμάτωσης δικτύου, όπως για το ποιος επιτρέπεται να προσπελάσει, ποια δεδομένα. Πλέον, τα απαρτιζόμενα κομμάτια του δικτύου μπορούν να αναγνωρίζουν ποια πληροφορία μεταφέρεται εντός εκάστου πακέτου.

1.2.1 Επικέντρωση στην Κινητικότητα

Το σχέδιο διευθυνσιοδότησης του Διαδικτύου σχεδιάστηκε απευθυνόμενο σε σταθερούς εξυπηρετητές, οπότε η IP διεύθυνση δικαιωματικά πρέπει να ανήκει σε δίκτυο στο οποίο ο εξυπηρετητής εντοπίζεται. Παρόλα αυτά, ο συνεχώς αυξανόμενος αριθμός των μη σταθερών εξυπηρετητών, οδήγησε στην δραματική αύξηση των κινητών συσκευών στο Διαδίκτυο. Οι ασύρματες και κινητές συσκευές μπορούν εύκολα να μεταβαίνουν σε διαφορετικά δίκτυα, αλλάζοντας τη διεύθυνση IP τους και εισάγοντας νέα επικοινωνιακά μοντέλα βασισμένα στη διακοπτόμενη και ευκαιριακή διασύνδεση. Η σημερινή προσέγγιση Διαδικτύου δεν διασφαλίζει συνεχή διασύνδεση, παρά την ολοένα και αυξανόμενη ανάγκη συνεχούς μεταβολής θέσεως.

Από την άλλη πλευρά το Κινητό(Mobile) IP πρωτόκολλο, μια τροποποίηση επίλυσης του προβλήματος εντοπισμού κινητών εξυπηρετητών, εισαγάγει την έννοια της τριγωνικής δρομολόγησης. Η εφαρμογή της τριγωνικής δρομολόγησης μεταφέρει τα δεδομένα από ένα σταθερό εξυπηρετητή σε μια οποιαδήποτε κινητή συσκευή, χωρίς όμως να αξιοποιείται η βέλτιστη διαδρομή.

1.2.2 Επικέντρωση στην Ασφάλεια

Το Διαδίκτυο σχεδιάστηκε ώστε να λειτουργεί ως έμπιστο περιβάλλον. Η πιστοποίηση του χρήστη και των δεδομένων, η διασφάλιση της ακεραιότητας των δεδομένων και η εξασφάλιση της ιδιωτικότητας των χρηστών, δεν ήταν προ απαιτούμενα. Απεναντίας, η σχεδίαση επικεντρώθηκε κυρίως στην ευελιξία προσαρμογής νέων εξυπηρετητών στο δίκτυο. Επιπλέον, το Διαδίκτυο σχεδιάστηκε ώστε να προωθεί κάθε εισαγόμενη κίνηση καταλήγοντας σε ανισορροπία δύναμης μεταξύ αποστολών και παραληπτών. Τα χαρακτηριστικά αυτά επιτρέπουν στους κακόβουλους χρήστες να

πραγματοποιούν επιθέσεις Άρνησης Υπηρεσίας – DoS(Denial-of-service attack) εναντίον της υπάρχουσας Δικτυακής υποδομής ή εναντίον των διάφορων εξυπηρετητών και υπηρεσιών, με παράλληλη διατήρηση της ανωνυμίας. Για την αντιμετώπιση τέτοιων κακόβουλων επιθέσεων απαιτήθηκαν επιπρόσθετες τροποποιήσεις και μηχανισμοί ασφαλείας, όπως για παράδειγμα τα τείχη προστασίας και τα φίλτρα sraml, αλλά ακόμη και νέα πρωτόκολλα ασφαλείας που να συμπληρώνουν τα υπάρχουσα δικτυακά πρωτόκολλα (IPSec). Η απαιτούμενη επιπρόσθετη πληροφορία επεξεργασίας και η παγιωμένη δομή των τερματικών κόμβων, δεν επιτρέπει την βαθύτερη εφαρμογή σε επίπεδο αρχιτεκτονικής δικτύου, όπου η ταυτοποίηση και αποτροπή επιθέσεων καθίσταται αποδοτικότερη. Αρκετά προβλήματα ασφαλείας, οφείλονται κατά κύριο λόγο σε προβλήματα αποσύνδεσης, μεταξύ της διαστρωμάτωσης εφαρμογής και της αδιαφανούς δομής των IP πακέτων δεδομένων.

Αντίθετα, η αρχιτεκτονική ICN οδηγείται βάσει των εκάστοτε αιτημάτων, οπότε η όποια ροή δεδομένων επέρχεται στην περίπτωση που συγκεκριμένος χρήστης αιτηθεί, ένα συγκεκριμένο κομμάτι πληροφορίας. Ως εκ τούτου, αναμένεται να μειωθεί ο αριθμός των άνευ λόγου δεδομένων που παράγονται και να διευκολυνθεί η εγκαθίδρυση μηχανισμών υπευθυνότητας και ασφαλείας στα σημεία του δικτύου που διαχειρίζονται τις επιλογές “διαθεσιμότητας” και “σηματοδότησης ενδιαφέροντος”. Επιπροσθέτως, για τις αρχιτεκτονικές ICN όπου χρησιμοποιούν ονόματα αυτοπιοστοποίησης θα καθίσταται δυνατή η διασφάλιση της υπηρεσίας από εσωτερικούς ενσωματωμένους μηχανισμούς πληροφορίας και φιλτραρίσματος κακόβουλων δεδομένων. Τέλος, η πλειοψηφία των αρχιτεκτονικών ICN προσθέτουν ένα σημείο επικοινωνίας μεταξύ εκείνων των χρηστών που αναζητούν την πληροφορία και των χρηστών που επεξεργάζονται την πληροφορία, αποσυνδέοντας την άμεση επικοινωνία των μερών αυτών, μέσω της εισαγωγής ενός σημείου ανακατεύθυνσης. Η αποσύνδεση αυτή μπορεί να αποτελέσει κατευθυντήρια γραμμή προς την καταπολέμηση των επιθέσεων DoS, καθώς τα εκάστοτε αιτήματα μπορούν να αξιολογηθούν σε πρότερο στάδιο, προτού δηλαδή αφιχθούν στον τελικό προορισμό.

1.3 Σκοπός

Ο σκοπός της παρούσας διπλωματικής εργασίας εδράζεται στη μελέτη των θεμελιωδών αρχών, της πολλά υποσχόμενης αρχιτεκτονικής Named Data Networking, η οποία αποτελεί επέκταση της ICN αρχιτεκτονικής. Στη βασική της δομή πραγματεύεται την επίδραση των μεγάλων μεγεθών πακέτων δεδομένων, στη συνολική απόδοση του δικτύου, βάσει των διάφορων πολιτικών διαχείρισης ουράς. Παράμετροι όπως η καθυστέρηση διάδοσης, διαδραματίζουν καθοριστικό ρόλο, στην επίτευξη της βέλτιστης δικτυακής αποδοτικότητας.

Ως εκ τούτου χρησιμοποιήθηκαν όλα τα διαθέσιμα εργαλεία ανοικτού κώδικα, ώστε να εξεταστούν οι προγραφείσες παράμετροι. Εν τέλει, θα αποσαφηνιστεί κατά πόσο η νέα δικτυακή αρχιτεκτονική μπορεί να ενσαρκώσει επιτυχώς στο λειτουργικό της μοντέλο, κομμάτια δεδομένων μεγαλύτερα του συνηθισμένου.

Το προηγούμενο θα αποτελέσει αξιοσημείωτη παρακαταθήκη, αφού εν γένει μπορεί να χρησιμοποιείται για τη δημιουργία ad hoc δικτύων, σε περιπτώσεις έκτακτων συμβάντων, όπου ο χρόνος και η μαζική επιτυχής αποστολή δεδομένων, διαδραματίζουν καθοριστικό ρόλο στη διαμόρφωση των επικείμενων πολιτικών.

1.4 Δομή πτυχιακής

Κεφάλαιο 1: Εισαγωγή στη δομή του Διαδικτύου και περιγραφή των σύγχρονων αναγκών

Κεφάλαιο 2: Μετάβαση στην NDN αρχιτεκτονική. Πραγματοποιείται θεωρητική ανάλυση των ενσωματωμένων στοιχείων

Κεφάλαιο 3: Περιγραφή των διαδικασιών προώθησης των Πακέτων Ενδιαφέροντος και Δεδομένων, καθώς και του προσομοιωτή ndnSIM

Κεφάλαιο 4: Αναλυτική περιγραφή των σεναρίων, καθώς και των απαιτούμενων τροποποιήσεων

Κεφάλαιο 5: Γραφική αναπαράσταση των αποτελεσμάτων

Κεφάλαιο 6: Ανάλυση και εξαγωγή συμπερασμάτων

Κεφάλαιο 7: Βιβλιογραφία και ηλεκτρονικές αναφορές

2 Η μετάβαση στο NDN

2.1 Δομή κλεψύδρας (hourglass)

Το πρωτόκολλο TCP/IP[5,6] αποτελεί την κοινή βάση των σύγχρονων τηλεπικοινωνιών. Αποτελεί ένα υποβόσκον πρωτόκολλο που χρησιμοποιείται για τη μεταφορά δεδομένων μεταξύ ιστοσελίδων, ανταλλαγή ηλεκτρονικών μηνυμάτων και αρχείων, όπως και λόγους δικτυακής διασύνδεσης εταιρών. Η αναπαράσταση του πραγματοποιείται με χρήση της hourglass δομής, όπου η IP διαστρωμάτωση εντοπίζεται στο μέσο. Νέα πρωτόκολλα μπορούν δυνητικά να εισαχθούν σε οποιοδήποτε επίπεδο του μοντέλου.

Η δομή hourglass καθίσταται αποδοτική για τη δημιουργία μιας κοινής πλατφόρμας διαφορετικών τύπων συσκευών πολλαπλών σκοπών, διευκολύνοντας την εισαγωγή είτε νέων συσκευών στα κατώτερου επιπέδου στρώματα είτε νέων εφαρμογών στα υψηλότερου επιπέδου στρώματα. Οι όποιες αλλαγές, μπορούν να πραγματοποιηθούν σε οποιοδήποτε σημείο της διαστρωμάτωσης δικτύου, χωρίς να απαιτείται ολοκληρωτική αλλαγή της δομής TCP/IP.

Η διαστρωμάτωση δικτύου που σχηματίζεται στη μέση της δομής κλεψύδρας (hourglass) καθίσταται αρκετά διαφανής, ώστε κάθε εφαρμογή να μπορεί να εκτελεστεί σε ανώτερου επιπέδου διαστρωμάτωση. Επιπλέον, ο απλοϊκός σχεδιασμός της διασφαλίζει τη διασύνδεση κάθε τεχνολογίας επιπέδου συνδέσμου.

Ακόμη, επικεντρώνεται σε ένα ευρέως διαδεδομένο επίπεδο διαστρωμάτωσης, το οποίο εφαρμόζει την ελάχιστη απαιτούμενη λειτουργία για τη διασφάλιση της παγκόσμιας διασύνδεσης. Η λεπτή αυτή δομή επέτρεψε την εκρηκτική ανάπτυξη του Διαδικτύου επιτρέποντας την ανεξάρτητη ανάπτυξη των τεχνολογιών κατώτερης και ανώτερης διαστρωμάτωσης. Ωστόσο, το IP σχεδιάστηκε για τη δημιουργία ενός κοινού δικτύου επικοινωνίας, όπου τα πακέτα κατευθύνονται μεταξύ των τερματικών σημείων. Η τεράστια ανάπτυξη των υπηρεσιών ηλεκτρονικού εμπορίου, των ψηφιακών μέσων, των κοινωνικών δικτύων και των κινητών εφαρμογών, οδήγησε στην επικράτηση του Διαδικτύου, ως ένα δίκτυο διανομής. Τα δίκτυα διανομής αποτελούν μια γενικότερη έννοια συγκριτικά με τα τηλεπικοινωνιακά δίκτυα, οπότε και η επίλυση προβλημάτων διανομής μεταξύ των διάφορων τερματικών σημείων διακρίνεται από αυξημένη πολυπλοκότητα και επιρρέπεια σε σφάλματα.

Το μοντέλο IP αποτελεί τη ραχοκοκαλιά του σημερινής Διαδικτυακής υποδομής. Οι αυστηρές και καθορισμένες προδιαγραφές του το καθιστούν αποτελεσματικό και συνάμα παραμετροποιήσιμο.

2.2 Οριοθετημένος χώρος ονομάτων

Το IPv4[7] αποτελεί το συνηθέστερο πρωτόκολλο διαστρωμάτωσης διαδικτύου. Η 32 bit δομή του επιτρέπει την απόδοση 4.294.967.295 IP διευθύνσεων, καθιστώντας τον αριθμό των διαθέσιμων διευθύνσεων μικρότερο από τον σημερινό παγκόσμιο πληθυσμό. Ως εκ τούτου, δεν υπήρξε πρόθεση για διευρυμένη χρήση του διαδικτύου, ούτε για διασύνδεση πολλαπλών συσκευών. Συνεπώς, βρισκόμαστε σε ένα περιοριστικό χρονικό σημείο για τις διαθέσιμες IPv4 διευθύνσεις.

Για την περαιτέρω διεύρυνση του χώρου διευθύνσεων, δημιουργήθηκε η έκδοση IPv6 [8] του πρωτοκόλλου IP. Η 128 bit δομή του επιλύει τα περιοριστικά προβλήματα που δημιουργούνται από την εγγενή έλλειψη διαθέσιμων διευθύνσεων IPv4. Αποτελεί ενσωματωμένο χαρακτηριστικό της διαδικτυακής δομής από τις 14.07.2017, παρότι υπάρχει στο προσκήνιο από το 1998. Το 2016 το 43% των κινητών συσκευών διέθεταν ικανότητα διευθυνσιοδότησης IPv6. Η αργή ενσωμάτωση του IPv6, οφείλεται στις ελλειπείς επενδύσεις από πλευράς τηλεπικοινωνιακών παρόχων, αφού ήταν επιβεβλημένες οι εκτεταμένες αλλαγές στον διαδικτυακό κορμό.

2.3 Βασικές αρχές σχεδίασης

Η αποσαφήνιση της δομής του Διαδικτύου[9] απαριθμείται από τις ανώτερου επιπέδου διαστρωματώσεις, καταλήγοντας στις κατώτερου επιπέδου, όπου κρίνεται αναγκαία η ύπαρξη ενός κοινού βασικού χαρακτηριστικού. Το αναγκαίο αυτό χαρακτηριστικό ονομάζεται IP διαστρωμάτωση.

Οι πρώιμες σχεδιαστικές αποφάσεις αποσκοπούσαν στην ικανοποίηση των σχεδιαστικών απαιτήσεων, ώστε να ικανοποιούνται οι κάτωθι συνθήκες:

- Να διασφαλίζεται αδιάκοπα η διαδικτυακή επικοινωνία, ανεξάρτητα από τις ενδεχόμενες διακοπές που εντοπίζονται μεταξύ των πυλών εισόδου
- Να υποστηρίζονται όλοι οι τύποι υπηρεσιών επικοινωνίας
- Να επιτρέπεται η “στέγαση” πολλαπλών δικτύων διαφορετικών χαρακτηριστικών (ετερογένεια)
- Να διασφαλίζεται η κατανεμημένη διαχείριση των πόρων

- Να προσαρμόζεται άμεσα ο εκάστοτε εξυπηρετητής, με χρήση ελάχιστης καταβαλλόμενης προσπάθειας
- Να καθίστανται ως μετρήσιμη ποσότητα οι αξιοποιήσιμοι πόροι

Η παραπάνω λίστα έχει ταξινομηθεί σε φθίνουσα σειρά βάσει της σημασίας. Η χρήση του πρωτοκόλλου IP ως ενδιάμεσης διαστρωμάτωσης εκπληρώνει τις προγραφείς προδιαγραφές, ενώ παράλληλα δημιουργεί μία κοινή γλώσσα για όλες συσκευές που ενδέχεται να συζευχθούν.

Η επιθυμία ύπαρξης μιας δικτυακής αρχιτεκτονικής που θα ανταποκρίνεται στον τρόπο με τον οποίο χρησιμοποιούμε το διαδίκτυο, αποτελεί το κύριο επιχείρημα για τη μετάβαση σε μία αρχιτεκτονική, η οποία δε θα βασίζεται πλέον στο IP μοντέλο. Η επικείμενη αλλαγή θα δημιουργήσει χώρο για καινοτομία μιας αποκεντρωμένης δικτυακής πολιτικής δικτύου, όπου κάθε υπολογιστική μονάδα θα διαθέτει ικανοποιητική επεξεργαστική δύναμη συνδυαστικά με μεγαλύτερη ευελιξία κινήσεων. Ο στόχος έγκειται στο να μεταμορφωθεί η δομή του σημερινού διαδικτυακού μοντέλου από το μοντέλο IP-διακομιστή στην κατεύθυνση εντοπισμού της πληροφορίας, ανεξάρτητα από την επικείμενη θέση της.

Από δικτυακής σκοπιάς η NDN δομή θα λειτουργεί ως επικάλυψη(overlay), προσομοιώνοντας δηλαδή τον τρόπο που το IP ξεκίνησε. Οπότε, είναι εφικτή η λειτουργία του σε κάθε δικτυακή υποδομή, με τελικό σκοπό την απλοποίηση της μετάφρασης μεταξύ των σημερινών δικτυακών εφαρμογών και της βασισμένης στους εξυπηρετητές δικτυακής πολιτικής, οδηγώντας στη δημιουργία ενός περιεχόμενο-κεντρικού συστήματος.

Η λογική της προηγούμενης πολιτικής εδράζεται στο γεγονός του συνεχώς αυξανόμενου διαθέσιμου χώρου αποθήκευσης, ο οποίος πλέον είναι οικονομικότερος, κατ' αντιπαράβολή με το διαθέσιμο εύρος ζώνης. Επομένως, γίνεται εφικτή η αποθήκευση δεδομένων σε κάθε κόμβο με σκοπό την εξυπηρέτηση των χρηστών που αναζητούν το ίδιο περιεχόμενο. Συμπερασματικά, οδηγούμαστε σε εξοικονόμηση του διαθέσιμου εύρους ζώνης, αφού απαιτούνται λιγότερα άλματα για την απόκτηση των δεδομένων.

2.4 Η αρχιτεκτονική Named Data Networking (NDN)

Στην υλοποίηση του Named Data Networking **[10]** προτάθηκε η μετεξέλιξη της αρχιτεκτονικής IP, ώστε να καθίσταται γενικός ο ρόλος της λεπτής μέσης(waist) μέσω της ονοματοδοσίας των διάφορων πακέτων, παρά των

τερματικών σημείων. Συγκεκριμένα, η αρχιτεκτονική NDN μεταβάλλει τη σημασιολογία της δικτυακής υπηρεσίας, από το μοντέλο της “προσκόμισης ενός πακέτου σε μια συγκεκριμένη τοποθεσία”, στο μοντέλο της “προσκόμισης του εκάστοτε πακέτου δεδομένων συγκεκριμένου ονόματος”. Το όνομα ενός πακέτου NDN μπορεί να είναι οτιδήποτε, όπως για παράδειγμα ένα τερματικό σημείο, ένα τμήμα δεδομένων ενός βιβλίου ή μιας ταινίας, μια εντολή απενεργοποίησης λαμπτήρων.

Αυτή η εννοιολογική και απλή τροποποίηση επιτρέπει την αξιοποίηση ολόκληρης της δομής Διαδικτύου, με τελικό σκοπό την αντιμετώπιση προβλημάτων που προκύπτουν από τις επικοινωνίες τερματικών σημείων, καθώς και προβλημάτων σχετιζόμενων με τη διανομή περιεχομένου. Βασιζόμενοι στην πολυπληθή εμπειρία των τελευταίων τριών δεκαετιών και αναλογιζόμενοι των δυνατών σημείων, όπως και των περιορισμών της υπάρχουσας αρχιτεκτονικής, ο σχεδιασμός του NDN ενσωματώνει συγκεκριμένες πολιτικές ασφαλείας που αξιοποιούν διαδικασίες υπογραφής πακέτων, καθώς και αυτορρύθμισης του δικτυακής συμφόρησης.

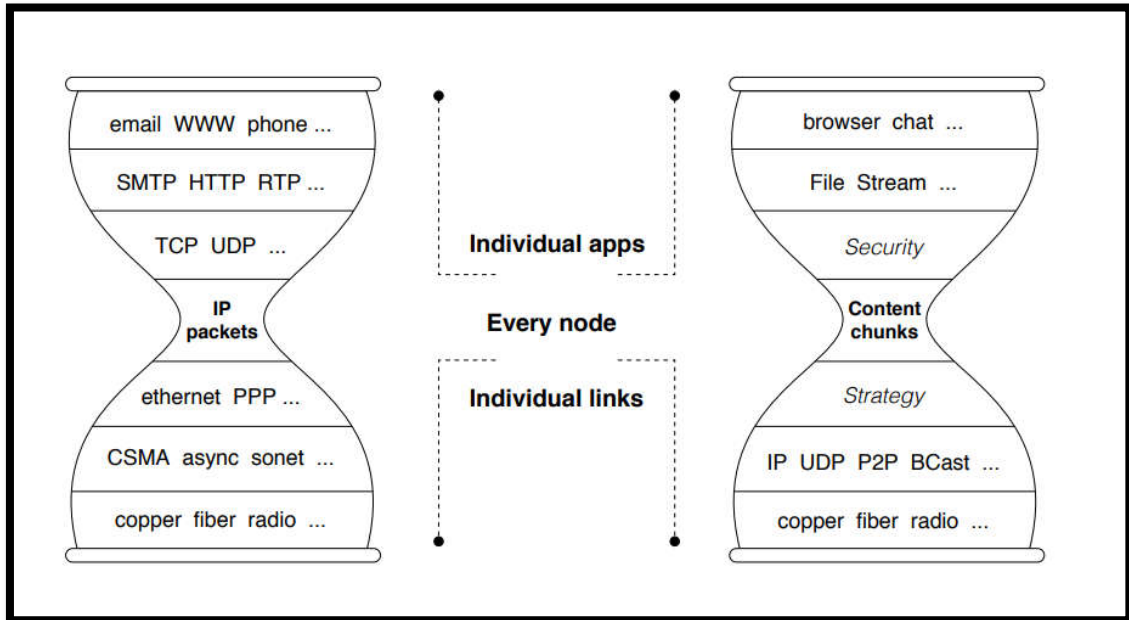
Το NDN αποτελεί μια μόνο υλοποίηση στο πλαίσιο της γενικότερης δικτυακής έρευνας στην κατεύθυνση του ICN, βάσει του οποίου έχουν προκύψει ποικίλες διαφορετικές αρχιτεκτονικές.

2.4.1 Περαιτέρω ανάλυση του NDN

Το NDN αποτελεί μία διαδικτυακή αρχιτεκτονική που προορίζεται για αντικατάσταση του IP πρωτόκολλου, αποτελώντας ένα ενδιάμεσο στάδιο μεταξύ των εφαρμογών και του δικτύου. Μπορεί να αξιοποιεί το υπάρχον IP πρωτόκολλο, λειτουργώντας αφαιρετικά σε ένα παραπάνω επίπεδο.

Η βασική ιδέα δημιουργίας του NDN αφορά τον τρόπο οργάνωσης της κίνησης στο δίκτυο, αφού τα εκάστοτε αιτήματα βασίζονται στην πληροφορία αυτή καθ' αυτή και όχι στη θέση που εντοπίζεται. Κάθε κόμβος διαθέτει τη δυνατότητα αποθήκευσης πακέτων Δεδομένων και Ενδιαφέροντος (Data packets, Interest packets). Στην περίπτωση που ένας δεύτερος καταναλωτής αιτηθεί το ίδιο πακέτο δεδομένων, θα εξυπηρετηθεί από κάποιον ενδιάμεσο κόμβο που θα διαθέτει ήδη την πληροφορία, αντί της πηγής δεδομένων. Κάθε ενδιάμεσος κόμβος διαθέτει πληροφορίες σχετικά με τις διεπαφές που σχετίζονται με κάθε πακέτο, ώστε να τα δρομολογεί στην κατάλληλη κατεύθυνση. Επιπλέον, το πακέτο δεδομένων δύναται να προωθηθεί προς κάθε διαθέσιμη κατεύθυνση.

Η NDN αρχιτεκτονική μεταθέτει το μεγαλύτερο μέρος της ευφυΐας και των ευθυνών στο δίκτυο καθ' αυτό, παρά στις εφαρμογές που εντοπίζονται στα τερματικά σημεία. Ένα ακόμη ιδιάζον χαρακτηριστικό της δομής εντοπίζεται στη δυνατότητα ελέγχου της συμφόρησης, από κάθε κόμβο τοπικά.



Εικόνα 2 Συγκριτική παρουσίαση της δομής *hourglass* μεταξύ του σημερινού διαδικτυακού μοντέλου και της NDN υλοποίησης [10]

2.4.2 Ασφάλεια

Σε αντίθεση με την αρχιτεκτονική TCP/IP όπου η ευθύνη της ασφάλειας (ή η έλλειψη) επαφίεται στα τερματικά σημεία, η NDN αρχιτεκτονική ασφαλίσει τα δεδομένα, απαιτώντας από τους παραγωγούς την κρυπτογραφική υπογραφή του εκάστοτε πακέτου Δεδομένων. Η υπογραφή του παραγωγού διασφαλίζει την ακεραιότητα των Δεδομένων, επιτρέποντας ταυτόχρονα τον προσδιορισμό της προέλευσης τους. Έτσι, αποσυνδέεται η ευθύνη διασφάλισης των δεδομένων από τον Καταναλωτή.

Διαθέτει επίσης μια καλοσχεδιασμένη μέθοδο εμπιστοσύνης, παρέχοντας την δυνατότητα αξιολόγησης από μέρους Καταναλωτών, των κατόχων παραγωγών του δημόσιου κλειδιού, ενός συγκεκριμένου πακέτου δεδομένων.

Η δεδομένο-κεντρική δομή του NDN περιέχει ενσωματωμένες εφαρμογές ελέγχου πρόσβασης περιεχομένου και ασφάλειας υποδομών. Οι εφαρμογές μπορούν να ελέγχουν την πρόσβαση στα δεδομένα μέσω κρυπτογράφησης και διανομής κλειδιών, ως NDN κρυπτογραφημένα δεδομένα, περιορίζοντας την περίμετρο ασφαλείας στα όρια μιας και μοναδικής εφαρμογής. Η απαίτηση

υπογραφών ασφαλείας για τη διαδικασία δρομολόγησης, καθώς και μηνυμάτων ελέγχου, παρέχει στέρεα θεμέλια για τη διασφάλιση πρωτοκόλλων δρομολόγησης, εναντίον ενδεχόμενης παραποίησης. Η χρήση της προώθησης πολλαπλών διαδρομών σε συνδυασμό με τη λειτουργική μονάδα προσαρμοστικής στρατηγικής προώθησης, μετριάζει την προσπάθεια κακόβουλων επιθέσεων. Πλέον οι δρομολογητές μπορούν να ανιχνεύσουν ενδεχόμενο δόλο που προκλήθηκε από hijacks και να προβούν στην ανάκτηση των δεδομένων από εναλλακτικές διαδρομές.

Δεδομένης της εγγενούς δομής του NDN, όπου τα πακέτα αναφέρονται στο περιεχόμενο και όχι στις συσκευές καθίσταται δυσκολότερη η κακόβουλη στόχευση μιας συσκευής. Παρά ταύτα, είναι απαιτητοί μηχανισμοί ασφαλείας, όπως ο έλεγχος συμφόρησης πακέτων Ενδιαφέροντος.

2.5 Θέση του της IP διαστρωμάτωσης στο NDN μοντέλο

Παρακάτω παρουσιάζεται η δομή της αρχιτεκτονικής NDN[10,11]. Αρχικά εξετάζουμε τη νέα διαστρωμάτωση που παρουσιάζεται στην εικόνα που ακολουθεί και στη συνέχεια παρουσιάζονται οι τύποι των πακέτων δεδομένων που αξιοποιούνται από τη νέα αρχιτεκτονική.

Στη μέση της αρχιτεκτονικής εμπεριέχεται η Name Data διαστρωμάτωση, η οποία είναι υπεύθυνη για όλα τα διαφορετικά τμήματα που καθιστούν λειτουργικό το πακέτο Δεδομένων, για τους κόμβους που βρίσκονται στο μέσο και στα άκρα. Παράλληλα, κάθε εφαρμογή έχει καθοριστικό ρόλο στη διασφάλιση της πληροφορίας του εκάστοτε πακέτου.

Η ονοματοδοσία του πακέτου πραγματοποιείται από τη διαστρωμάτωση του ονόματος Δεδομένων. Η NDN αρχιτεκτονική πραγματοποιεί διαχωρισμό των πακέτων σε δύο κατηγορίες:

- Πακέτα Ενδιαφέροντος (Interest Packet)
- Πακέτα Δεδομένων (Data Packet)

Το ελάχιστο όριο ενός πακέτου Ενδιαφέροντος καθορίζεται από το Όνομα και τον δείκτη Nonce, εξασφαλίζοντας έτσι διακριτό διαχωρισμό ενός πακέτου Ενδιαφέροντος, το οποίο επιζητά την ίδια πληροφορία.

Τη στιγμή που ένα πακέτο Ενδιαφέροντος προωθείται από την ανοδική ροή σε έναν Καταναλωτή, αποθηκεύεται σε κάθε ενδιάμεσο κόμβο. Εν συνεχεία, αποφασίζεται από τη στρατηγική προώθησης σε ποιους κόμβους θα προωθηθεί με χρήση των παρακάτω μεθόδων:

- απλής δρομολόγησης
- πολλαπλής δρομολόγησης
- δρομολόγησης πακέτου NACK

Ο κόμβος αποθηκεύει το πακέτο Ενδιαφέροντος στον πίνακα Εν Αναμονή Πακέτων Ενδιαφέροντος (Pending Interest Table – PIT) και στη συνέχεια επιδρά βάσει συγκεκριμένης πολιτικής, ώστε να εντοπιστεί αντιστοιχία του εκάστοτε αιτήματος με οποιοδήποτε πακέτο.

Τα πακέτα Δεδομένων στη συνέχεια ακολουθούν την ανάστροφη διαδρομή προς τον καταναλωτή, διότι η πορεία τους γίνεται στη βάση της καθοδικής πορείας του πακέτου Ενδιαφέροντος. Οι ενδιαμέσοι δρομολογητές διαγράφουν τα πακέτα Ενδιαφέροντος από τον πίνακα Εν Αναμονή Πακέτων(PIT), αποθηκεύοντας ταυτόχρονα ένα αντίγραφο στον Πίνακα Περιεχομένου(Content Store). Με τον τρόπο αυτό διασφαλίζεται η ενδεχόμενη ικανοποίηση ενός μελλοντικού πακέτου(αιτήματος) Ενδιαφέροντος, που αναζητά τα ίδια δεδομένα.

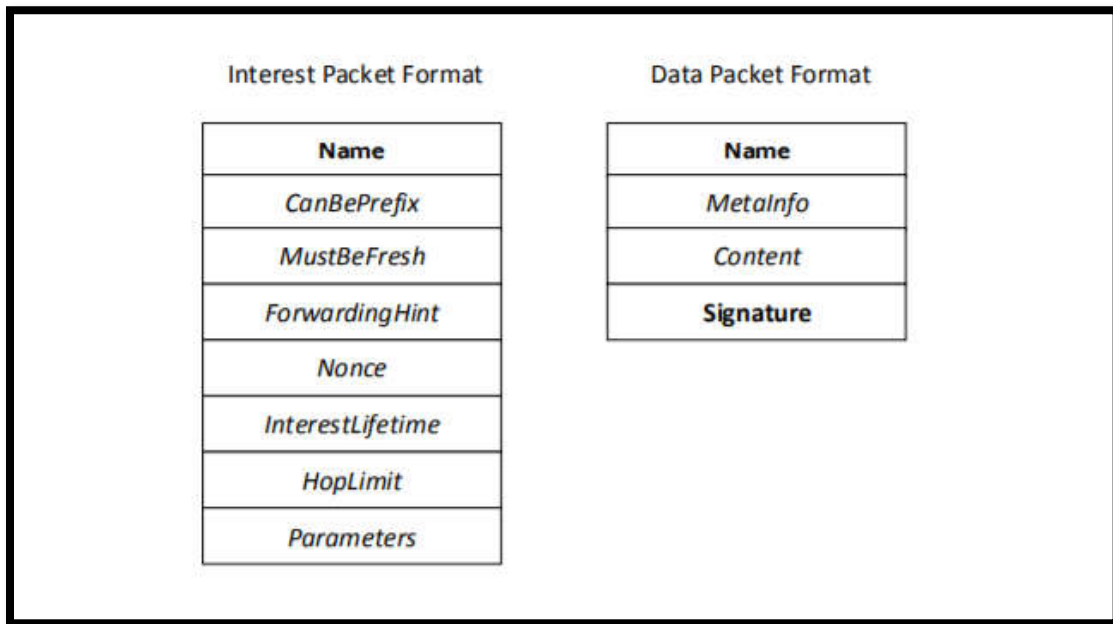
Εν γένει, οι κόμβοι των εφαρμογών μπορούν να προβούν σε συγκεκριμένες δραστηριότητες, που αφορούν τη ρύθμιση του αριθμού πακέτων Ενδιαφέροντος τα οποία και προωθούνται στην ανοδική ροή.

2.6 Επεξήγηση πακέτων NDN αρχιτεκτονικής

Στη βάση της αρχιτεκτονικής εδράζονται δύο διαφορετικοί τύποι πακέτων:

- Πακέτα Ενδιαφέροντος
- Πακέτα Δεδομένων

Κάθε Καταναλωτής που επιθυμεί δεδομένα από κάποιον Παραγωγό, πρέπει να αποστέλλει ένα πακέτο Ενδιαφέροντος για κάθε πακέτο δεδομένων που επιδιώκει. Επομένως, η δομή των πακέτων χαρακτηρίζεται ως αμφιμονοσήμαντη (1-Interest-1-Data). Η δυναμική των συγκεκριμένων πακέτων, συμπεριλαμβανομένων και των NACK, αποτελούν τον πυρήνα της NDN αρχιτεκτονικής.



Εικόνα 3 Δομή των πακέτων Ενδιαφέροντος και Δεδομένων της αρχιτεκτονικής Named Data Networking[11]

2.6.1 Πακέτο Ενδιαφέροντος – Interest Packet

Το πακέτο Ενδιαφέροντος[12] αποτελείται από τις κάτωθι οντότητες:

Name: Αποτελεί το μόνο απαραίτητο πεδίο του πακέτου. Πρόκειται για μια ιεραρχική δομή ονοματολογίας του NDN περιεχομένου.

CanBePrefix: Αποτελεί προαιρετικό πεδίο, του οποίου η τιμή συγκεκριμενοποιεί το προκαθορισμένο(prefix), ακριβές όνομα του ζητούμενου πακέτου Δεδομένων.

MustBeFresh: Αποτελεί προαιρετικό πεδίο, του οποίου η ύπαρξη υποδηλώνει τον χρόνο ζωής (freshness) ενός πακέτου Δεδομένων που βρίσκεται στον Content Store, βάσει του οποίου θα ικανοποιήσει ή όχι ένα αίτημα πακέτου Ενδιαφέροντος. Συνδέεται με την προαιρετική τιμή FreshnessPeriod που πακέτου Δεδομένων. Εφόσον ένα αίτημα περιλαμβάνει το πεδίο MustBeFresh, ένας κόμβος δεν πρέπει να επιστρέψει δεδομένα που έχουν χαρακτηριστεί ως "non-fresh".

Forwarding Hint: Πρόκειται για προαιρετικό πεδίο, του οποίο η ύπαρξη υποδηλώνει τις ενδεικτικές διαδρομές ανάκτησης του πακέτου Δεδομένων, προωθώντας το πακέτο Ενδιαφέροντος κατά μήκος των ενδεικτικών διαδρομών.

Nonce: Το πεδίο διαθέτει μια τυχαία δημιουργηθείσα τιμή 4 byte. Ορίζει συνδυαστικά με το πεδίο Ονόματος, τη μοναδική ταυτότητα του εκάστοτε

πακέτου Ενδιαφέροντος. Η τιμή του Nonce αποτελεί παγκόσμιο αναγνωριστικό, αφού δεν επιτρέπεται η ταυτόχρονη ύπαρξη δύο ίδιων πεδίων Nonce, για ίδιο πακέτο Ενδιαφέροντος.

InterestLifeTime: Το πεδίο υποδηλώνει το υπόλοιπο χρόνου ζωής του πακέτου Ενδιαφέροντος, σε χιλιοστά του δευτερολέπτου [ms]. Το χρονικό όριο σχετίζεται με τη στιγμή άφιξης του πακέτου Ενδιαφέροντος στον τρέχοντα κόμβο. Η προκαθορισμένη τιμή του ορίζεται σε 4 δευτερόλεπτα.

HopLimit: Πρόκειται για προαιρετικό πεδίο, του οποίου η ύπαρξη υποδηλώνει το μέγιστο αριθμό αλμάτων που ένα πακέτο ενδιαφέροντος μπορεί να πραγματοποιήσει. Η τιμή του κωδικοποιείται με μέγεθος 1 byte. Εφόσον η τιμή του είναι 0, το πακέτο πρέπει να απορριφθεί. Στην περίπτωση που έχει τιμή μεγαλύτερη ή ίση με 1, ο κόμβος πρέπει να δεχθεί το πακέτο και να μειώσει την κωδικοποιημένη τιμή κατά μία μονάδα.

Application Parameters: Πρόκειται για προαιρετικό πεδίο, στο οποίο περιλαμβάνεται πιθανή επιπρόσθετη πληροφορία, με σκοπό την παραμετροποίηση του εκάστοτε αιτήματος για Δεδομένα.

2.6.2 Πακέτο Δεδομένων – Data Packet

Το πακέτο Δεδομένων[13] ενσαρκώνει τις κάτωθι οντότητες:

Name: Αποτελεί υποχρεωτικό πεδίο της ιεραρχικής δομής, το οποίο υποδηλώνει το όνομα του εκάστοτε πακέτου Δεδομένων.

MetaInfo: Όλη η πληροφορία είναι προαιρετική.

ContentType: Καθορίζει τον τύπο πληροφορίας που εντοπίζεται στο πακέτο Δεδομένων.

FreshnessPeriod: Πρόκειται για προαιρετικό πεδίο, του οποίου η τιμή υποδηλώνει το χρονικό διάστημα που ένας κόμβος πρέπει να περιμένει, προτού χαρακτηρίσει τα δεδομένα ως “non-fresh”. Η διέλευση του ορίου τιμής, πυροδοτεί την εκ νέου αναζήτηση του πακέτου Δεδομένων, ώστε να ανανεωθεί το περιεχόμενο του CS.

FinalBlockId: Πρόκειται για προαιρετικό πεδίο βάσει του οποίου πραγματοποιείται η ταυτοποίηση του τελευταίου μπλοκ αξιοποιείται για λόγους ταυτοποίησης του τελευταίου μπλοκ μιας σειράς. Πρέπει να εντοπίζεται όχι μόνο στο τελευταίο μπλοκ δεδομένων, αλλά και στα προηγούμενα, ώστε να παράσχει υπηρεσίες προειδοποίησης στους Καταναλωτές.

Content: Το συγκεκριμένο πεδίο μπορεί να μεταφέρει μια τυχαία ακολουθία από bytes.

Signature: Υπογραφή των στοιχείων, που πιστοποιεί την αυθεντικότητα του πακέτου.

2.6.3 Αρνητική Επιβεβαίωση (Negative Acknowledgment – NACK)

Στην περίπτωση που κάποιος δρομολογητής αποφασίσει ότι ένα πακέτο Αιτήματος δεν μπορεί να ικανοποιηθεί, τότε αποστέλλει ένα πακέτο τύπου NACK στην καθοδική ροή που αρχικά μετέδωσε το αίτημα. Μια τέτοιου τύπου αρνητική επιβεβαίωση, μπορεί να οδηγήσει τον αρχικό δρομολογητή στην προώθηση του πακέτου Ενδιαφέροντος σε διαφορετικές διεπαφές, διαφορετικών μονοπατιών.

2.7 Ονόματα – Names

Η αρχιτεκτονική NDN περιεργάζεται τα αιτήματα με όρους περιεχομένου και όχι τοποθεσίας. Τα ονόματα της δομής NDN δεν είναι γνωστά στο δίκτυο, καθώς κάθε εφαρμογή μπορεί να σχεδιάσει τη δομή ονοματολογίας που της αρμόζει, δίχως να προ-απαιτείται κάποια δικτυακή συμβατότητα. Αυτός ο διαχωρισμός καθιστά πιθανή την ανάπτυξη του δικτύου ξέχωρα από τις εφαρμογές, δίνοντας περισσότερη ελευθερία και ευελιξία στην εξέλιξη των μελλοντικών δικτύων.

Ο σχεδιασμός του NDN υιοθετεί ιεραρχικά δομημένα ονόματα[14], όπως για παράδειγμα ένα βίντεο που παράγεται από το τμήμα Ιατρικής του ΔΠΘ, μπορεί να διαθέτει όνομα /duth/videos/corona-outbreak.mpg. Η ιεραρχική αυτή δομή επιτρέπει στις εφαρμογές να αναπαραστήσουν το περιεχόμενο, όπως και τις σχέσεις μεταξύ των οντοτήτων δεδομένων. Παραδείγματος χάριν, το κομμάτι 3 της πρώτης έκδοσης του βίντεο δεδομένων ΔΠΘ, θα μπορούσε να ονομαστεί ως /duth/videos/corona-outbreak.mpg/1/3.

Για την ανάκτηση δυναμικά δημιουργημένων δεδομένων, οι Καταναλωτές πρέπει να είναι σε θέση ντετερμινιστικής κατασκευής του ονόματος για τα επιθυμητά κομμάτια δεδομένων, χωρίς να έχουν οποιαδήποτε προηγούμενη επαφή με το όνομα. Για να πραγματωθεί η προγραφείσα συνθήκη πρέπει ο ντετερμινιστικός αλγόριθμος να επιτρέπει, είτε την ανάκτηση του ίδιου ονόματος από πλευράς Καταναλωτή και Παραγωγού την ίδια στιγμή είτε την ανάκτηση των επιθυμητών δεδομένων από τους Επιλογείς ενδιαφέροντος σε συνδυασμό

με την μεγαλύτερη αντιστοίχιση προθέματος, μέσω μιας ή περισσοτέρων επαναλήψεων.

Βάσει της εμπειρίας μας η αρχιτεκτονική NDN μπορεί να υποστηρίξει την ανάκτηση δεδομένων ακόμη και στην περίπτωση γνώσης ενός τμήματος του ονόματος. Λόγου χάρη, ένας καταναλωτής που αναζητεί την πρώτη έκδοση του corona-virus.mpg βίντεο μπορεί να αναζητήσει βάσει του /duth/videos/corona-virus.mpg/1 και να λάβει ένα κομμάτι του βίντεο /duth/videos/corona-virus.mpg/1/1. Στη συνέχεια μπορεί να αναζητήσει και τα υπόλοιπα κομμάτια της πληροφορίας, βάσει των δεδομένων που θα λάβει από το πρώτο πακέτο.

Τα δεδομένα που είναι παγκοσμίως προσβάσιμα πρέπει να διαθέτουν παγκοσμίως μοναδικά ονόματα, ενώ για τα δεδομένα που αξιοποιούνται για τοπικές επικοινωνίες μπορεί να απαιτείται τοπική δρομολόγηση(ή τοπική εκπομπή) για αντιστοίχιση δεδομένων. Η ύπαρξη μοναδικών ονομάτων βρίθκει ευελιξίας, αφού το εύρος της μπορεί να περιλαμβάνει για παράδειγμα ονόματα για “απενεργοποίηση λαμπτήρα στο δωμάτιο”.

Ο χώρος ονομάτων δεν αποτελεί ενσωματωμένο μέρος της αρχιτεκτονικής NDN, όπως αντίστοιχα ο χώρος διευθύνσεων δεν αποτελεί μέρος της αρχιτεκτονικής IP. Παρόλα αυτά η ονοματολογία αποτελεί το σημαντικότερο τμήμα του σχεδιασμού NDN εφαρμογών. Η ονοματολογία δεδομένων υποστηρίζει λειτουργικότητες όπως η διανομή περιεχομένου ή ευρεία εκπομπή, η κινητικότητα και η ανεκτική στην καθυστέρηση δρομολόγηση.

Η δυνατότητα των προγραμματιστών να σχεδιάζουν το δικό τους χώρο ονομάτων για ανταλλαγή δεδομένων ενσαρκώνει πολλαπλά πλεονεκτήματα, αφού:

- Αυξάνουν τη δυνατότητα χαρτογράφησης εγγύτητας μεταξύ των δεδομένων μιας εφαρμογής και της συνολικής αξιοποίησης δικτύου
- Εκμηδενίζουν την ανάγκη εκ των υστέρων καταχώρησης
- Επεκτείνουν το εύρος αφαίρεσης που δυνητικά μπορεί να αξιοποιηθεί από τους προγραμματιστές

2.8 Κωδικοποίηση TLV (Type-Length-Value)

Η τιμή Type-Length-Value χρησιμοποιείται για λόγους κωδικοποίησης των πακέτων Δεδομένων και Ενδιαφέροντος. Σε γενικές γραμμές το πεδίο TLV είναι αναγκαίο για την διασφάλιση μιας ευέλικτης και πολυποίκιλης ονοματοδοσίας στο NDN.

2.9 Εσωτερική δομή και Προώθηση

2.9.1 Πίνακας Περιεχομένου – Content Store

Ο Content Store [15] (**nfd::cs::Cs**) αποτελεί προσωρινή μονάδα αποθήκευσης του NDN. Αποθηκεύει κάθε πακέτο Δεδομένων που αντιστοιχεί στο πακέτο ενδιαφέροντος του PIT, κάνοντας κατ' αντιστοιχία αναζήτηση ονόματος. Τα αφιχθέντα πακέτα τοποθετούνται στην προσωρινή μνήμη για όσο το δυνατόν μεγαλύτερο χρονικό διάστημα, ώστε να ικανοποιήσουν μελλοντικά αιτήματα Ενδιαφέροντος, που ενδέχεται να απαιτούν τα συγκεκριμένα Δεδομένα. Όπως καθορίζεται από την NDN αρχιτεκτονική, ο Πίνακας Δεδομένων σαρώνεται προτού το εισερχόμενο αίτημα Ενδιαφέροντος δοθεί στη στρατηγική δρομολόγησης, για περαιτέρω επεξεργασία. Με αυτόν τον τρόπο, τα προσωρινώς αποθηκευμένα Δεδομένα, εφόσον είναι διαθέσιμα, χρησιμοποιούνται για την ικανοποίηση των αιτημάτων Ενδιαφέροντος, δίχως να πραγματοποιηθεί οποιαδήποτε ανατιμολόγηση προώθησης του αιτήματος.

Τα πακέτα δεδομένων εισάγονται είτε στον CS (**Cs::insert**), είτε στον αγωγό εισερχομένων Δεδομένων, είτε στον αυτόκλητο αγωγό Δεδομένων, αφού η διαδικασία δρομολόγησης διασφαλίζει την καταλληλότητα να τεθεί υπό περαιτέρω επεξεργασία το πακέτο Δεδομένων.

Πριν την αποθήκευση του πακέτου, πραγματοποιείται αξιολόγηση της πολιτικής εισαγωγής. Στη συνέχεια το πακέτο Δεδομένων αποθηκεύεται ταυτόχρονα με την τιμή του χρονικού πεδίου, μετά την παρέλευση του οποίου θα καταστεί αδρανές. Στην περίπτωση όπου παρέλθει το χρονικό όριο, δε δύνανται να ικανοποιηθεί κανένα πακέτο Ενδιαφέροντος.

Ο πίνακας CS(**Cs::find**) διερευνάται βάσει ενός πακέτου Ενδιαφέροντος, προτού το αίτημα προωθηθεί στον αγωγό εισερχομένων Ενδιαφερόντων. Ο αλγόριθμος αναζήτησης επιστρέφει το πακέτο Δεδομένων που αντιστοιχίζεται καλύτερα βάσει του Ενδιαφέροντος, οπότε είτε παραδίδεται στον Αγωγό Ευστοχίας Περιεχομένου (**ContentStore Hit**), είτε στον Αγωγό Αστοχίας Περιεχομένου (**ContentStore miss**), εφόσον δεν υπάρχει κάποια αντιστοιχία.

Ο CS έχει περιορισμένη χωρητικότητα, η οποία οροθετείται βάσει του αριθμού των πακέτων. Η χρήση πολιτικών διαχείρισης όπως λόγου χάριν - **Cs::setLimit** – επιτρέπει την τροποποίηση της χωρητικότητας. Η εφαρμογή της πολιτικής του εκάστοτε Πίνακα Δεδομένων πρέπει να διασφαλίζει ότι ο αριθμός των πακέτων Ενδιαφέροντος δε θα υπερβεί την αναμενόμενη χωρητικότητα.

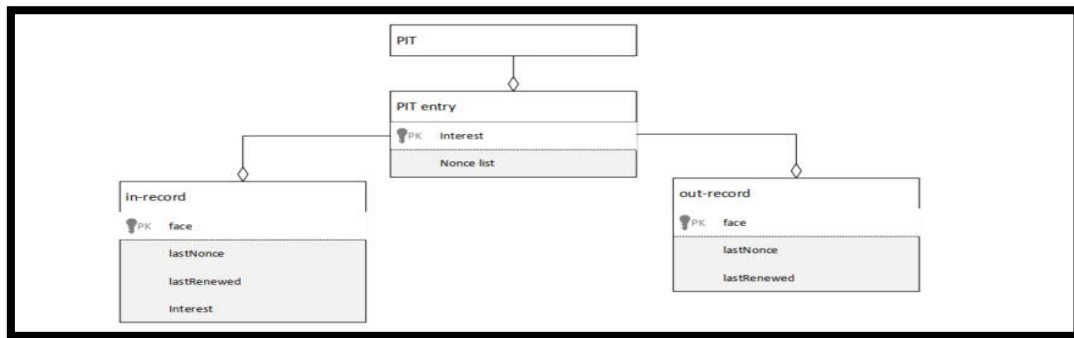
2.9.2 Πρακτική Εφαρμογή

Η επίδοση του CS έχει τεράστιο αντίκτυπο στη συνολική απόδοση του NFD, καθώς αποθηκεύει έναν τεράστιο αριθμό πακέτων. Η επιλογή της υποβόσκουσας δομής δεδομένων για λόγους αποδοτικής αναζήτησης, εισαγωγής, διαγραφής και αντικατάστασης είναι ζωτικής σημασίας, έχοντας ως αποτέλεσμα να μεγιστοποιηθεί η απόδοση της εσωτερικής δομής προσωρινής αποθήκευσης.

2.10 Πίνακας εν αναμονή Ενδιαφερόντων (Pending Interest Table)

Ο Πίνακας Ενδιαφέροντος καταγράφει τα πακέτα Ενδιαφέροντος που προωθήθηκαν στην ανοδική ροή ανεύρεσης περιεχομένου, ώστε τα δεδομένα να μπορούν να επιστραφούν κατά την καθοδική πορεία στον αρχικό χρήστη. Επιπλέον, περιέχονται τα προσφάτως ικανοποιημένα Ενδιαφέροντα για αποφυγή ατέρμονης επανάληψης, καθώς και για λόγους μετρήσεων. Αυτή η δομή δεδομένων καλείται ως Πίνακας εν αναμονή Αιτημάτων (Pending Interest Table). Παρά ταύτα, περιέχει ταυτοχρόνως τα εν αναμονή Ενδιαφέροντα όπως και τα προσφάτως ικανοποιημένα αιτήματα. Επομένως ο όρος Πίνακας Ενδιαφερόντων (Interest Table), αποτελεί πιο περιγραφική σύμπτυξη του όρου Pending Interest Table.

Στην περίπτωση που το πακέτο είναι ήδη αποθηκευμένο στον PIT, δεν πραγματοποιείται η προώθηση του στον FIB. Ένεκα αυτού, διαγράφεται με σκοπό να αποφευχθεί η ύπαρξη διπλότυπου πακέτου στο δίκτυο. Ο κόμβος θα ταυτοποιήσει την ύπαρξη δύο ενδιαφερόμενων Καταναλωτών για το ίδιο πακέτο, σημειώνοντας την ύπαρξη νέου Καταναλωτή, στην παλιά PIT καταχώριση. Στη συνέχεια, ο πίνακας PIT αξιοποιείται κατά την επιστροφή του πακέτου Δεδομένων, ώστε να αναγνωριστούν οι σχετιζόμενοι Καταναλωτές και εν τέλει για να αποσταλεί το πακέτο Δεδομένων σε όλα τα ενδιαφερόμενα μέρη της ανοδικής ροής. Όταν ένα πακέτο Δεδομένων εισαχθεί στον κόμβο, θα συγκριθεί βάσει των PIT καταχωρίσεων. Ενδεχόμενη μη ύπαρξη του πακέτου στους κόλπους του PIT οδηγεί σε αυτόματη απόρριψη του, ενώ σε αντίθετη περίπτωση προωθείται στους ενδιαφερόμενους χρήστες-Καταναλωτές. Ο PIT θα προβεί σε διαγραφή των καταχωρίσεων του πακέτου Ενδιαφέροντος, όταν τα ταυτιζόμενα δεδομένα αντιστοιχηθούν με την εσωτερική καταχώριση και στη συνέχεια δρομολογηθούν προς τους ενδιαφερόμενους Καταναλωτές.



Εικόνα 4 Διαδικασία επεξεργασίας πακέτου Ενδιαφέροντος από τον Πίνακα[PIT] Εν αναμονή Ενδιαφερόντων[15]

Συνοπτικά, η δομή PIT αποτελεί μια συλλογή καταχωρίσεων που αξιοποιούνται για λόγους δρομολόγησης.

2.10.1 Καταχώριση PIT (PIT Entry)

Μια καταχώριση PIT (**nfd::pit::Entry**) αναπαριστά είτε ένα εν αναμονή πακέτο Ενδιαφέροντος είτε ένα προσφάτως ικανοποιημένο. Δύο πακέτα Ενδιαφέροντος θεωρούνται “πανομοιότυπα”, εφόσον έχουν το ίδιο όνομα και τους ίδιους Επιλογείς. Πολλαπλά πακέτα(αιτήματα) Ενδιαφέροντος μοιράζονται την ίδια καταχώριση PIT.

Κάθε PIT καταχώριση αντιστοιχίζεται σε ένα πακέτο Ενδιαφέροντος. Όλα τα υπόλοιπα πεδία του Αιτήματος, εκτός από τα Όνομα και Επιλογείς, είναι ασήμαντα. Επιπλέον, κάθε PIT καταχώριση περιέχει μια συλλογή εσωτερικών καταγραφών, μια συλλογή εξωτερικών καταγραφών και δύο δείκτες χρόνου. Επιπροσθέτως, η στρατηγική δρομολόγησης επιτρέπει την αποθήκευση τυχαίων πληροφοριών στον πίνακα PIT.

2.10.2 Εσωτερική καταγραφή (In Record)

Μια καταγραφή In Record(**nfd::pit::InRecord**) αντιπροσωπεύει μια οντότητα Προσώπου καθόδου, ενός πακέτου Ενδιαφέροντος. Μια οντότητα Προσώπου καθόδου αποτελεί ένα αίτημα περιεχομένου, αφού το Ενδιαφέρον προέχεται από την κάθοδο και τα Δεδομένα καταλήγουν στην κάθοδο. Η εσωτερική καταγραφή διατηρεί:

- μια αναφορά στην οντότητα(face)
- το Nonce(μοναδικό αναγνωριστικό) στο τελευταίο πακέτο ενδιαφέροντος της οντότητας
- τη χρονική σήμανση για το που κατευθύνεται το τελευταίο πακέτο ενδιαφέροντος της συγκεκριμένης οντότητας
- το τελευταίο πακέτο ενδιαφέροντος

Μια εσωτερική καταγραφή εισάγεται ή ενημερώνεται από τον αγωγό εισερχομένων αιτημάτων. Όλες οι καταγραφές διαγράφονται από τον αγωγό εισερχομένων Δεδομένων, όταν ένα εν αναμονή πακέτο Ενδιαφέροντος ικανοποιείται.

2.10.3 Εξωτερική Καταγραφή (Out Record)

Μια καταγραφή εξόδου (**nfd::pit::OutRecord**) αναπαριστά μια οντότητα Προσώπου ανόδου ενός Ενδιαφέροντος. Η οντότητα ανόδου αποτελεί την ενδεχόμενη πηγή περιεχομένου: Το πακέτο Ενδιαφέροντος προωθείται στην άνοδο και τα Δεδομένα έρχονται από την άνοδο. Μια καταγραφή εξόδου διατηρεί:

- Μια αναφορά στην οντότητα
- το Nonce(μοναδικό αναγνωριστικό) στο τελευταίο πακέτο ενδιαφέροντος της οντότητας
- τη χρονική σήμανση για το που κατευθύνεται το τελευταίο πακέτο ενδιαφέροντος της συγκεκριμένης οντότητας

2.10.4 Χρονικοί δείκτες – Timers

Υπάρχουν δύο χρονικοί δείκτες σε μια καταχώριση PIT, που χρησιμοποιούνται από τους αγωγούς δρομολόγησης:

- χρόνος μη ικανοποίησης, όταν ο χρόνος καταχώρισης του PIT παρέλθει
- straggler timer, όταν μια καταχώριση του PIT μπορεί να διαγραφεί, είτε επειδή ικανοποιήθηκε είτε επειδή απορρίφθηκε, οπότε πλέον δεν απαιτείται για λόγους ανίχνευσης επανάληψης και μετρήσεων

2.10.5 Πίνακας εν αναμονή Ενδιαφερόντων (Pending Interest Table – PIT)

Ο PIT (**nfd::Pit**) είναι ο πίνακας που περιέχει τις καταχωρίσεις PIT, ευρετηριοποιημένος από τη δυάδα <Όνομα, Επιλογείς>. Οι συνήθεις λειτουργίες εισαγωγής και διαγραφής υποστηρίζονται. Η μέθοδος (**Pit::insert**) αναζητά αρχικά μια PIT καταχώριση για κάποιο παρόμοιο πακέτο Ενδιαφέροντος και εισάγει τιμή μόνο σε περίπτωση αποτυχίας ανεύρεσης.

Ο αλγόριθμος Αντιστοιχίας Δεδομένων (**Pit::findAllDataMatches**) αναζητά όλα τα πακέτα Ενδιαφέροντος που μπορούν να ικανοποιηθούν από ένα πακέτο Δεδομένων. Ως παράμετρος εισάγεται ένα πακέτο Δεδομένων. Η επιστρεφόμενη τιμή είναι μια συλλογή από PIT καταχωρίσεις που μπορούν να

ικανοποιηθούν από το εκάστοτε πακέτο Δεδομένων. Ο συγκεκριμένος αλγόριθμος δεν προβαίνει σε καμία διαγραφή PIT καταχώρισης.

2.11 Λίστα ανίχνευσης επανάληψης (Dead Nonce List)

Η λίστα Dead Nonce αποτελεί μια δομή δεδομένων που συμπληρώνει τον PIT, για λόγους ανίχνευσης επανάληψης.

Τον Αύγουστο του 2014, ανιχνεύθηκε ένα επίμονο πρόβλημα επανάληψης όταν ο χρόνος ζωής του πακέτου Ενδιαφέροντος ήταν πάρα πολύ μικρός. Η ανίχνευση επανάληψης σε προηγούμενες εκδόσεις χρησιμοποιούσε μόνο τα πεδία Nonce που ήταν αποθηκευμένα σε κάθε PIT καταχώριση. Εφόσον ένα πακέτο Ενδιαφέροντος δεν ικανοποιείται εντός ορισμένου χρονικού διαστήματος, η καταχώριση PIT διαγράφεται κατά τη λήξη του χρόνου ζωής. Όταν το δίκτυο περιέχει έναν κύκλο του οποίου η καθυστέρηση είναι μεγαλύτερη από το χρόνο ζωής του αιτήματος Ενδιαφέροντος, το πρόβλημα επανάληψης που σχετίζεται με τον συγκεκριμένο κύκλο δεν μπορεί να ανιχνευθεί, επειδή η PIT καταχώριση διεγράφη προτού το πακέτο Ενδιαφέροντος επιστρέψει.

Μια πρόχειρη λύση στο πρόβλημα αυτό αποτέλεσε η διατήρηση της PIT καταχώρισης για μεγαλύτερο χρονικό διάστημα. Παρόλα αυτά εντοπίστηκε τεράστια κατανάλωση μνήμης, αφού μια PIT καταχώριση περιλαμβάνει πολλαπλές τιμές, εκτός του αναγνωριστικού Nonce. Επομένως, η λίστα Dead Nonce εισήχθη, ώστε να σηματοδοτούνται “dead” Nonce καταχωρίσεις του PIT.

Η λίστα Dead Nonce αποτελεί μια πιθανοκρατική δομή δεδομένων, αφού κάθε καταχώριση αποθηκεύεται ως ένα 64-bit hash του ονόματος και του Nonce, οδηγώντας σε αξιοσημείωτη μείωση της κατανάλωσης μνήμης της δομής δεδομένων. Την ίδια χρονική στιγμή υπάρχει μηδενική πιθανότητα hash σύγκρουσης, που αναπόφευκτα οδηγεί σε ανίχνευση ψευδούς θετικής κατάστασης: μη επαναληπτικά αιτήματα μπερδεύονται με την κατάσταση επανάληψης.

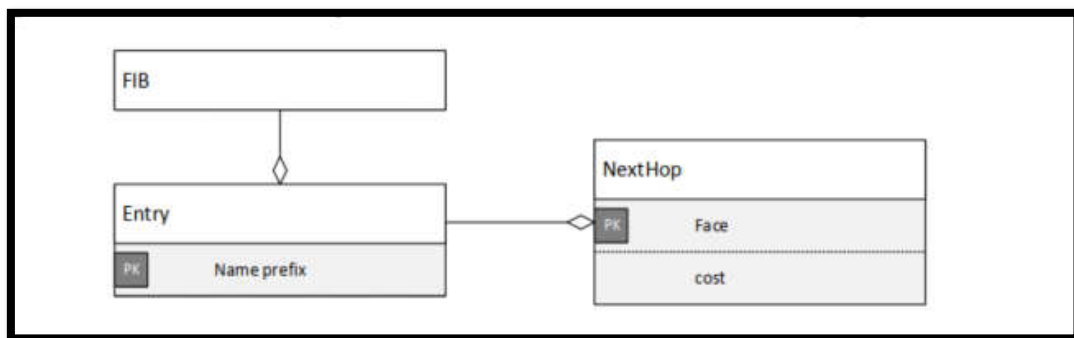
Ο τρόπος ανάκτησης των ψευδώς θετικών πακέτων Ενδιαφέροντος, περιγράφεται ως εξής: “Ο Καταναλωτής μπορεί να αναμεταδώσει το πακέτο Ενδιαφέροντος με ένα καινούριο Nonce, το οποίο θα αποδώσει καινούριο hash, που δε θα δημιουργήσει ζητήματα σύγκρουσης”. Θεωρείται ότι τα πλεονεκτήματα από την εξοικονόμηση μνήμης υπερτερούν τα προβλήματα που δημιουργούν τα ψευδώς θετικά πακέτα.

2.12 Πίνακας Προώθησης Ενδιαφέροντος (Forwarding Information Base)

Ο FIB χρησιμοποιείται για την προώθηση των πακέτων Ενδιαφέροντος, προς τις πιθανές πηγές αντιστοιχίας δεδομένων. Για κάθε πακέτο Ενδιαφέροντος που πρέπει να προωθηθεί, πραγματοποιείται αναζήτηση εκτενούς προθέματος(longest prefix) εντός του πίνακα FIB. Επομένως, η λίστα των εξερχόμενων οντοτήτων Προσώπου που εντοπίζονται στον πίνακα FIB αποτελούν ισχυρή αναφορά για την προώθηση.

2.12.1 Δομή και Σημασιολογία

Στην κάτωθι εικόνα διαφαίνονται οι συσχετίσεις λογικού περιεχομένου μεταξύ του FIB, των FIB καταχωρίσεων, και των καταγραφών επόμενου Βήματος(NextHop).



Εικόνα 5 Αξιοποίηση του Forwarding Information Base για δρομολόγηση των πακέτων Ενδιαφέροντος στις πιθανές οντότητες Προσώπου

2.12.2 Καταχώριση FIB και καταγραφή επόμενου Βήματος (NextHop)

Η καταχώριση του πίνακα FIB (**nfd::fib::Entry**) περιέχει το πρόθεμα ονόματος και μια ενεργή συλλογή καταγραφών επόμενου βήματος (**NextHop**). Η εισαγωγή ενός προκαθορισμένου ονόματος στον πίνακα FIB μεταφράζεται στο εξής : για ένα συγκεκριμένο πακέτο ονόματος με το συγκεκριμένο πρόθεμα (prefix), μια πιθανή πηγή αντιστοιχίας δεδομένων μπορεί να ανακτηθεί μέσω συγκεκριμένων οντοτήτων Προσώπου, που ορίζονται από τις υπάρχουσες καταγραφές Επόμενου Βήματος, των FIB καταχωρίσεων.

Κάθε καταγραφή Επόμενου Βήματος (**nfd::fib::NextHop**) περιέχει μια εξερχόμενη οντότητα Προσώπου με κατεύθυνση προς μία πιθανή πηγή περιεχομένου, όπως επίσης και το κόστος δρομολόγησης. Μια FIB καταχώριση μπορεί να περιλαμβάνει το πολύ ένα επόμενο βήμα, με κατεύθυνση στην ίδια οντότητα Προσώπου. Εντός του FIB, οι καταχωρίσεις των επόμενων Βημάτων πραγματοποιούνται σε αύξουσα σειρά.

Ο πίνακας FIB αποτελεί μια συλλογή καταχωρίσεων, που ευρετηριοποιούνται βάσει προθέματος ονόματος. Οι διεργασίες που επιτρέπονται είναι αυτές της συνήθους εισαγωγής, διαγραφής και ακριβούς αντιστοιχίας. Ο αλγόριθμος Longest Prefix Match αντιστοιχίζει μια FIB καταχώριση, η οποία δυνητικά χρησιμοποιείται για την προώθηση του πακέτου Ενδιαφέροντος.

Ως παράμετρος εισαγωγής λαμβάνεται ένα όνομα, το οποίο αποτελεί το πεδίο ονόματος του πακέτου Ενδιαφέροντος. Η επιστρεφόμενη τιμή αποτελεί μια καταχώριση FIB, της οποίας το όνομα αποτελεί πρόθεμα της παραμέτρου εισόδου. Η επιστροφή κενής τιμής NULL πραγματοποιείται, εφόσον δεν εντοπιστεί οποιαδήποτε αντιστοιχία ονόματος.

Επιπλέον, η μέθοδος (**Fib::removeNextHopFromAllEntries**) προβαίνει σε σάρωση όλων των FIB καταχωρίσεων, αφαιρώντας τις καταχωρίσεις Επόμενου Βήματος μιας Οντότητας συγκεκριμένου προσώπου. Εφόσον μια καταχώριση του πίνακα FIB περιέχει αναγκαία μια τουλάχιστον τιμή, στην περίπτωση που οδηγηθούμε σε αφαίρεση της τελευταίας καταχώρισης Επόμενου Βήματος, πρέπει να προβαίνουμε και σε ταυτόχρονη αφαίρεση της FIB καταχώρισης. Η συγκεκριμένη μεθοδολογία αποτελεί χρήσιμο εργαλείο στην περίπτωση που κάποια οντότητα Προσώπου διαγραφεί.

2.13 Επιλογή στρατηγικής (Strategy)

Ο Πίνακας Επιλογής Στρατηγικής περιέχει την στρατηγική προώθησης για κάθε χώρο ονομάτων. Ο παρών πίνακας αποτελεί μια νέα προσθήκη της δομής NDN. Από θεωρητικής σκοπιάς η στρατηγική προώθησης αποτελεί ένα πρόγραμμα που αποθηκεύεται με τη μορφή μιας FIB καταχώρισης. Πρακτικά όμως, καθίσταται πιο βολική η αποθήκευσης των στρατηγικών προώθησης σε έναν ξεχωριστό πίνακα.

Ο τρόπος προώθησης και αποθήκευσης του εκάστοτε πακέτου αποφασίζεται από τη διαστρωμάτωση στρατηγικής. Η διαστρωμάτωση στρατηγικής διαχειρίζεται τον τρόπο αντίδρασης στα εισερχόμενα και εξερχόμενα πακέτα Ενδιαφέροντος, Δεδομένων και NACK. Το συγκεκριμένο επίπεδο διαστρωμάτωσης μπορεί να οριοθετήσει την πολιτική αποστολής σε μία ή πολλαπλές διεπαφές και να καθορίσει ταυτόχρονα τα διάφορα καθήκοντα μετάδοσης μεταξύ των επιλογών unicast/multicast/broadcast.

Ο Πίνακας Επιλογής Στρατηγικής αποτελεί μια συλλογή από τις διαθέσιμες(εγκατεστημένες) στρατηγικές. Ο έλεγχος του πραγματοποιείται από τον διαχειριστή Επιλογής Στρατηγικής, κάθε φορά που σηματοδοτείται από μία εντολή. Επομένως, απαιτείται η εγκατάσταση (**StrategyChoice::install**) της μεθόδου, ώστε να γνωστοποιηθεί μια παραμετροποιημένη στρατηγική προώθησης του NFD, η οποία θα χρησιμοποιείται σε συνδυασμό με τη στρατηγική του χώρου ονομάτων. Λαμβάνουμε υπόψη πως κάθε νέα εγκατεστημένη στρατηγική πρέπει να διαθέτει μοναδικό όνομα, αφού σε κάθε άλλη περίπτωση θα ανακύψει σφάλμα πραγματικού χρόνου.

Η λειτουργία εισαγωγής (**StrategyChoice::insert**) είτε εισάγει μια καταχώριση τύπου Strategy Choice είτε ανανεώνει την επιλεχθείσα στρατηγική. Η λειτουργία διαγραφής(**StrategyChoice::erase**) απαλείφει μια καταχώριση στρατηγικής. Επιπλέον, υποστηρίζεται η λειτουργία αντιστοίχισης.

Ο αλγόριθμος Ανεύρεσης Αποτελεσματικότερης Στρατηγικής (Find Effective Strategy) προβαίνει σε εντοπισμό της κατάλληλης στρατηγικής για την προώθηση του πακέτου Ενδιαφέροντος. Η αποδοτικότερη στρατηγική μπορεί να καθοριστεί με βάση τα παρακάτω:

- Εφόσον ο χώρος ονομάτων σχετίζεται ρητά με τη στρατηγική, τότε αυτή καθίσταται ως η αποτελεσματικότερη
- Διαφορετικά, ο πρώτος γονικός χώρος ονομάτων για τον οποίο έχει καθοριστεί σαφώς η στρατηγική, αποτελεί την αποτελεσματικότερη στρατηγική

Για την εξεύρεση της καταλληλότερης στρατηγικής χρησιμοποιούνται ως παράμετροι εισόδου, τα πεδία Ονόματος και PIT καταχώρισης.

Η επιλογή Στρατηγικής καταχώρισης (**nfd::strategy_choice::Entry**) περιέχει ένα πρόθεμα Ονόματος και το Όνομα της στρατηγικής δρομολόγησης, για το συγκεκριμένο χώρο ονομάτων. Για την παρούσα υλοποίηση δεν περιλαμβάνονται άλλες παράμετροι.

2.14 Πίνακας Μετρήσεων

Ο πίνακας μετρήσεων αξιοποιείται από τις στρατηγικές προώθησης για να αποθηκεύσει πληροφορίες σχετιζόμενες με μετρήσεις του προθέματος ονόματος. Η στρατηγική μπορεί να αποθηκεύσει τυχαία πληροφορία στον PIT και στις Μετρήσεις. Ο πίνακας Μετρήσεων μπορεί να στοιχειοθετηθεί βάσει του χώρου ονομάτων, οπότε καθίσταται κατάλληλη η αποθήκευση πληροφορίας

σχετική με τον εκάστοτε χώρο ονομάτων, αλλά όχι με ένα συγκεκριμένο Ενδιαφέρον.

Παρακάτω παρέχεται η δομή του Πίνακα Μετρήσεων:

Measurements Entry: Μια καταχώριση Μέτρησης περιέχει ένα Όνομα και ένα API για τη στρατηγική, ώστε να αποθηκεύσει και να ανακτήσει τυχαία πληροφορία. Είναι εφικτή η προσθήκη κάποιων συγκεκριμένων μετρικών που μπορούν να μοιραστούν μεταξύ των στρατηγικών, όπως ο χρόνος επιστροφής, η καθυστέρηση και η τιμή της διακύμανσης καθυστέρησης(jitter). Η προσθήκη καθορισμένων μετρικών αποτελεί μη αναγκαία επιβάρυνση(overhead), καθώς δεν χρησιμοποιούνται απαραίτητα από την στρατηγική.

Πίνακας Μετρήσεων: Ο Πίνακας Μετρήσεων(**nfd::Measurements**) αποτελεί μια συλλογή καταχωρίσεων Πίνακα. Η μέθοδος (**Measurements::get**) εντοπίζει ή εισάγει μια καταχώριση Μέτρησης με παραμέτρους το Όνομα, την καταχώριση FIB, ή την καταχώρηση PIT. Κατά την εφαρμογή του Πίνακα Μετρήσεων καθίσταται πιο αποδοτική η εισαγωγή μιας καταχώρισης FIB ή PIT, από τη χρήση κάποιου Ονόματος. Η μέθοδος (**Measurements::getParent**) εντοπίζει ή εισάγει καταχωρίσεις Μετρήσεων του γονικού χώρου Ονομάτων.

Σε αντίθεση με άλλους πίνακες, δεν υπάρχει λειτουργία διαγραφής. Κάθε καταχώριση διαθέτει συγκεκριμένο χρόνο ζωής, οπότε διαγράφεται αυτόματα αφού επέλθει λήξη του χρονικού ορίου. Για να πραγματοποιηθεί επέκταση του χρονικού ορίου απαιτείται η κλήση της μεθόδου (**Measurements::extendLifetime**). Επιπλέον, για λόγους ανάκτησης υποστηρίζονται αναζητήσεις ακριβούς αντιστοίχισης και μεγαλύτερου προθέματος(Longest Prefix).

3 Προσομοιωτής ndnSIM και ενσωμάτωση του NFD

3.1 Σύστημα Οντοτήτων

Το σύστημα οντότητας του NFD [15] χωρίζεται σε τρεις λογικές κατηγορίες:

- πρωτόκολλα (protocol factories)
- κανάλια (channels)
- οντότητες προσώπου (faces)

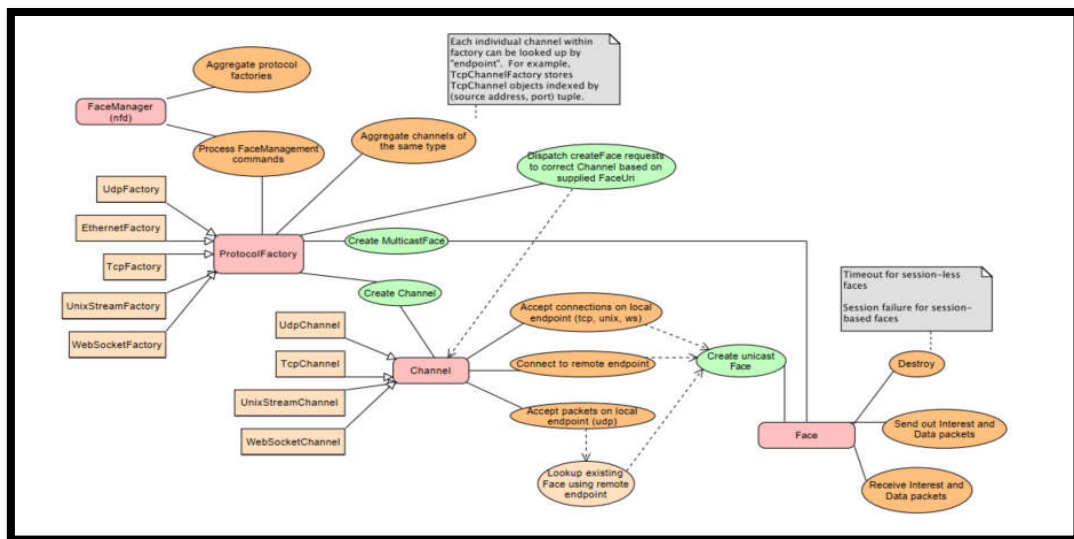
Ένα πρωτόκολλο (protocol factory) δημιουργεί κανάλια ή οντότητες προσώπου, συγκεκριμένων πρωτοκόλλων. Λόγου χάριν, το πρωτόκολλο TCP αξιοποιείται για τη δημιουργία και αρχικοποίηση TCP οντοτήτων.

Ένας αγωγός αντιπροσωπεύει την τερματική πλευρά του NFD για ευρεία επικοινωνία (unicast), όπως για παράδειγμα η υποδοχή επικοινωνίας ή η υποδοχή που ενδέχεται να εγκαθιδρυθεί η σύνδεση. Ονομάζουμε το αρχικό τερματικό σημείο “τοπικό”, ενώ το “απομακρυσμένο” τερματικό αντιπροσωπεύει την άλλη πλευρά της επικοινωνίας (π.χ. ένας κόμβος δικτύου, μια εφαρμογή, ή ένα σύνολο απομακρυσμένων εταίρων).

Μια Οντότητα Προσώπου αποτελεί μια αφαίρεση που εφαρμόζει θεμελιακά στοιχεία της διαδικασίας επικοινωνίας, ώστε να πραγματοποιείται αποστολή και λήψη πακέτων Ενδιαφέροντος και Δεδομένων. Βασιζόμενοι στη φύση της επικοινωνίας, μια οντότητα Προσώπου μπορεί να αναπαριστά ελαφρώς διαφοροποιημένα στοιχεία. Για διευρυμένες (unicast) επικοινωνίες (TCP, unicast UDP, UNIX sockets, WebSocket), μια οντότητα Προσώπου αναπαριστά μια διασύνδεση μεταξύ των τοπικών και απομακρυσμένων τερματικών για το συγκεκριμένο πρωτόκολλο, όπως για παράδειγμα μια σύνδεση μεταξύ των διαφορετικών NFD δικτυακών ή τοπικών κόμβων και της τοπικής εφαρμογής. Για επικοινωνίες πολλαπλής πρόσβασης (Ethernet-IP multicast), μια οντότητα Προσώπου αναπαριστά μια εικονική διασύνδεση μεταξύ του λογικού τερματικού και κανενός, ή περισσότερων εταίρων.

Η συνολική αλληλεπίδραση μεταξύ των αφαιρετικών εννοιών παρουσιάζονται στην παρακάτω εικόνα. Συνοπτικά οι αλληλεπιδράσεις μπορούν να συνοψιστούν ως εξής:

- Τα protocol factories, που προβαίνουν στην δημιουργία αγωγών
- Τους αγωγούς, που δημιουργούν οντότητες Προσώπου
- Τις οντότητες Προσώπου, που είναι υπεύθυνες για την αποστολή και λήψη των πακέτων Ενδιαφέροντος και Δεδομένων, κάνοντας χρήση της υπόγειας διασύνδεσης(tunnel) του συγκεκριμένου πρωτοκόλλου



Εικόνα 6 Συνοπτική περιγραφή του συστήματος οντοτήτων της αρχιτεκτονικής Named Data Networking

3.1.1 Επίπεδο αφαίρεσης Face URI

Το Face URI (nfd::FaceUri κλάση) χρησιμοποιείται για αναγνώριση ενός τερματικού πρωτοκόλλου και αξιοποιείται στις αφαιρέσεις Οντότητας και Καναλιού. Έχει κοινά χαρακτηριστικά με το απλό URI και αποτελείται από το όνομα του πρωτοκόλλου και κάθε άλλη πληροφορία που σχετίζεται με αυτό (πχ IP διεύθυνση και πόρτα).

Ορισμένοι τύποι Οντοτήτων διαθέτουν διαφορετικούς τύπους Face URI για τα τοπικά και απομακρυσμένα τερματικά. Αυτό εξαρτάται από τον εννοιολογικό συμβολισμό των εκάστοτε τερματικών σημείων. Παραδείγματος χάριν, το UnixStream Face διαθέτει ένα UNIX Stream τύπου Οντότητας URI, για λόγους αναπαράστασης του τοπικού τερματικού σημείου.

3.1.2 Επίπεδο αφαίρεσης Protocol Factory

Το protocol factory αποτελεί υψηλότερο επίπεδο αφαίρεσης στο σύστημα οντότητας. Κάθε protocol factory διαχειρίζεται ένα συγκεκριμένο πρωτόκολλο, το οποίο που υποστηρίζεται εγγενώς από τον NFD.

Τα δύο βασικά καθήκοντα για τα οποία έχει σχεδιαστεί το protocol factory είναι:

- Η δημιουργία και η διαχείριση των καναλιών(**createChannel**)
- Η δημιουργία και η διαχείριση των πολλαπλών(multicast) Οντοτήτων(**createMulticastFace**). Οι δύο λειτουργίες υποστηρίζονται από την πλειοψηφία των protocol factories. Ορισμένα όμως ενδέχεται να υποστηρίζουν μόνο τη δυνατότητα δημιουργίας καναλιών ή μόνο την δημιουργία πολλαπλών (multicast) Οντοτήτων.

3.1.3 Επίπεδο Αφαίρεσης Αγωγού

Ο σκοπός της αφαίρεσης αγωγού αφορά την ενσωμάτωση λειτουργιών που απαιτούνται, είτε για λόγους δημιουργίας εισερχόμενων διασυνδέσεων είτε αρχικοποίησης εξερχόμενων διασυνδέσεων, με σκοπό τη δημιουργία μιας οντότητας προσώπου, όταν η σύνδεση καταστεί επιτυχής. Οι αγωγοί δημιουργούνται μέσω της μεθόδου (**createChannel**), η οποία ενσωματώνεται στα πρωτόκολλα (protocol factories). Η χρήση της μεθόδου δεσμεύει και επιστρέφει έναν αγωγό που μπορεί να “ακούει” και να “αποδέχεται” εισερχόμενες συνδέσεις ,στο συγκεκριμένο τοπικό τερματικό σημείο.

Ένα τερματικό σημείο αποτελεί έναν τύπο συγκεκριμένου πρωτοκόλλου που περιλαμβάνει όλη την απαραίτητη πληροφορία, ώστε να αντιστοιχηθεί μοναδικά ένα τερματικό σημείο, σε ένα μηχανήμα. Λόγου χάριν, ας θεωρήσουμε την περίπτωση του TCP, όπου ως τερματικό σημείο ορίζεται το σύνολο(host, port). Πολλαπλοί αγωγοί μπορούν να δημιουργηθούν από το ίδιο factory protocol, αλλά κάθε κανάλι πρέπει να σχετίζεται με ένα διαφορετικό τοπικό τερματικό σημείο.

Συνήθως κατά τη δημιουργία ενός αγωγού δεν απαιτείται δέσμευση πόρων, οπότε η εντολή “ακοής” πρέπει να εκτελεστεί κατά το στιγμιότυπο καναλιού, ώστε να γίνει κατάλληλη προετοιμασία του για αποδοχή διασυνδέσεων. Η μέθοδος αυτή αναλαμβάνει την ανακατανομή των απαιτούμενων πόρων λειτουργικού συστήματος (sockets,ports).

3.2 Προώθηση στο NDN

Η NDN προώθηση περιγράφει την απαραίτητη διεργασία που πραγματοποιείται από πλευράς Καταναλωτή για την αποστολή ενός πακέτου Ενδιαφέροντος, καθώς και για τα ανταποκρινόμενα πακέτα Δεδομένων, που εν τέλει επιστρέφονται στον Καταναλωτή.

Το NDN δεν αξιοποιεί την τερματική διασύνδεση για την εγκαθίδρυση επικοινωνίας, όπως συμβαίνει στο TCP. Επομένως, για τον Παραγωγό δεν απαιτείται η συνεχής ανίχνευση των πακέτων Δεδομένων που στέλνονται στην προσωρινή μνήμη. Ο Παραγωγός δεν καθίσταται σε καμία περίπτωση υπεύθυνος, είτε για την επιτυχή παράδοση των πακέτων Δεδομένων στον Καταναλωτή είτε για τη διατήρηση της μεταξύ τους σύζευξης.

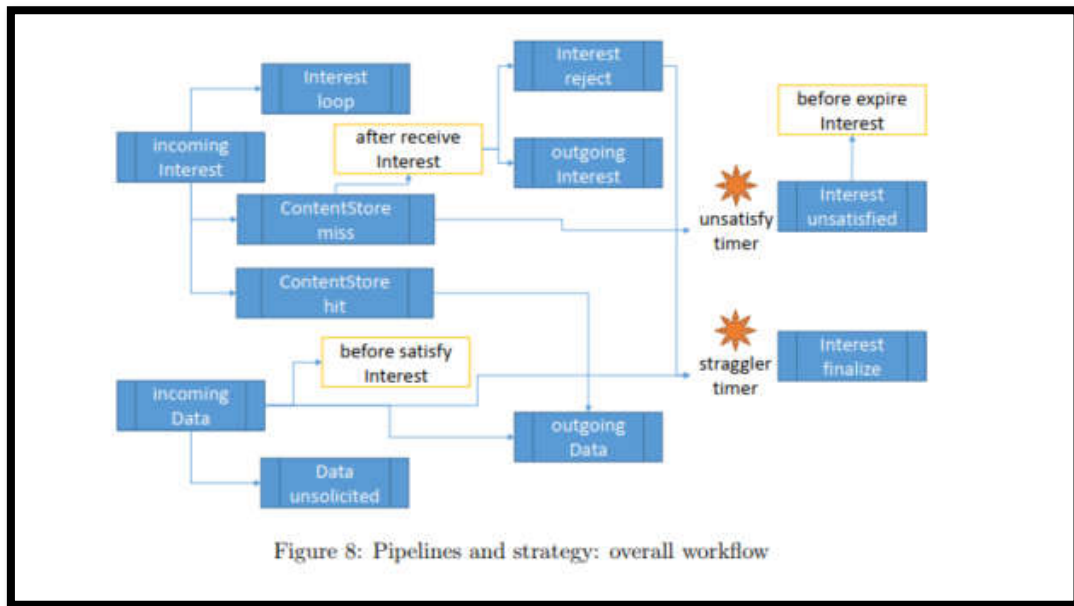
Επιπλέον, κάθε κόμβος είναι καθ' εαυτού υπεύθυνος, είτε για την αποστολή ενός πακέτου Ενδιαφέροντος στον Παραγωγό είτε για την αποστολή ενός πακέτου NACK στον προηγούμενο κόμβο. Κάθε διαβάθμιση πραγματοποιείται βασιζόμενη στην χωρητικότητα των συνδέσεων του εκάστοτε κόμβου και όχι του όλου δικτύου. Ακόμη, πραγματοποιείται ανανέωση της κατάστασης των συνδέσεων βασιζόμενη στην απεριόριστη (implicit) πληροφορία που αποκτάται, είτε από το πακέτο Δεδομένων είτε από την επιστροφή της επιβεβαίωσης.

Η επεξεργασία πακέτων στον NFD πραγματοποιείται από τους αγωγούς και τις στρατηγικές προώθησης. Ένας αγωγός προώθησης αποτελεί μια σειρά βημάτων που δρουν σε ένα πακέτο ή μια καταχώριση PIT και ενεργοποιείται βάσει ενός συγκεκριμένου γεγονότος:

- τη λήψη ενός αιτήματος Ενδιαφέροντος
- την ανίχνευση προβλήματος επανάληψης
- την προώθηση του πακέτου σε μια οντότητα Προσώπου

Μια στρατηγική προώθησης περιγράφεται ως μια απόφαση που σχετίζεται με την προώθηση ενός πακέτου Ενδιαφέροντος. Δηλαδή, η στρατηγική αποφασίζει το “πότε και το που” θα προωθηθεί ένα αίτημα, ενώ ο αγωγός τροφοδοτεί τα πακέτα με την κατάλληλη στρατηγική, καθώς και με την αναγκαία πληροφορία για τη λήψη αποφάσεων.

Η παρακάτω εικόνα παρουσιάζει το διάγραμμα ροής ενός αγωγού και μιας στρατηγικής δρομολόγησης. Τα μπλε κουτιά αναπαριστούν τους αγωγούς, ενώ τα άσπρα τις αποφάσεις της εκάστοτε στρατηγικής.



Εικόνα 7 Διαδικασία προώθησης των πακέτων Ενδιαφερόντων και Δεδομένων[15]

3.3 Αγωγοί προώθησης

Οι αγωγοί προώθησης πακέτων λειτουργούν στη διαστρωμάτωση δικτύου. Κάθε πακέτο διέρχεται από έναν αγωγό στον επόμενο, μέχρις ότου ολοκληρωθεί η διαδικασία. Η επεξεργασία εντός των αγωγών χρησιμοποιεί τους PIT, CS, FIB και Strategy Choice πίνακες.

Η επεξεργασία των πακέτων Ενδιαφέροντος και Δεδομένων διαφοροποιείται ελάχιστα, αφού το πρώτο εξυπηρετεί το αίτημα καθ' αυτό, ενώ το δεύτερο ικανοποιεί τα εν αναμονή πακέτα Ενδιαφέροντος. Οπότε πραγματοποιείται διαχωρισμός των αγωγών, σε μονοπάτια επεξεργασίας Ενδιαφέροντος και σε μονοπάτια επεξεργασίας Δεδομένων.

3.4 Αγωγός Επεξεργασίας Ενδιαφέροντος

Ο NFD διαχωρίζει την επεξεργασία Ενδιαφέροντος στους ακόλουθους αγωγούς:

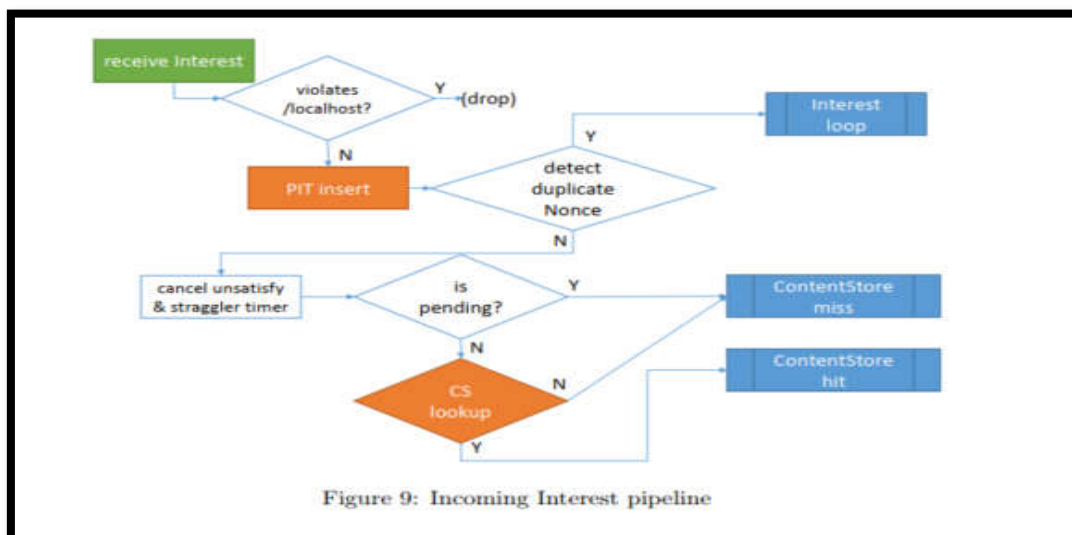
- Εισερχόμενου Αιτήματος: επεξεργασία εισερχομένων αιτημάτων Ενδιαφέροντος
- Ανίχνευσης επανάληψης Ενδιαφέροντος
- Αστοχίας – ContentStore Miss: αιτήματα που δεν μπορούν να ικανοποιηθούν από την προσωρινή μνήμη
- Ανεύρεσης – ContentStore Hit: αιτήματα που ικανοποιούνται από την προσωρινή μνήμη
- Εξερχόμενο αίτημα: προετοιμασία και αποστολή αιτημάτων

- Απόρριψη αιτήματος: επεξεργασία καταχωρίσεων PIT που απορρίπτονται από την στρατηγική
- Μη ικανοποιημένα αιτήματα: επεξεργασία καταχωρίσεων PIT που δεν έχουν ικανοποιηθεί, προτού επέλθει χρονική λήξη των καθοδικών ροών
- Τελικό αίτημα Ενδιαφέροντος: διαγραφή καταχώρισης

3.5 Αγωγός Εισερχόμενων Αιτημάτων Ενδιαφέροντος

Για την εφαρμογή του αγωγού εισερχόμενων αιτημάτων εφαρμόζεται η μέθοδος (**Forwarder::onIncomingInterest**). Οι παράμετροι εισόδου του αγωγού εισερχόμενων αιτημάτων περιλαμβάνουν τα προσφάτως αφιχθέντα πακέτα Ενδιαφέροντος, καθώς και μια αναφορά στην οντότητα Προσώπου από την οποία στάλθηκε το αίτημα.

Τα βήματα της προγραφείσας διαδικασίας συνοψίζονται παρακάτω:



Εικόνα 8 Απεικόνιση του μονοπατιού επεξεργασίας του πακέτου Ενδιαφέροντος[15]

- Ως πρώτο βήμα ορίζεται ο έλεγχος για οποιαδήποτε τοπική παραβίαση, του αιτήματος προτού εισαχθεί στον αγωγό Ενδιαφέροντος. Συγκεκριμένα, ένα αίτημα από μία μη τοπική οντότητα Προσώπου, δεν επιτρέπεται να διαθέτει όνομα αρχικοποιημένο με το πρόθεμα ονόματος /localhost, καθώς αυτό αξιοποιείται μόνο για τοπική επικοινωνία
- Σε επόμενο βήμα είτε αναζητείται μια προ υπάρχουσα καταχώριση είτε δημιουργείται νέα στον PIT, χρησιμοποιώντας το όνομα και τους Επιλογείς του πακέτου Ενδιαφέροντος.

- Προτού το αίτημα τεθεί υπό περαιτέρω επεξεργασία πραγματοποιείται έλεγχος του πεδίου Nonce μεταξύ των υπολοίπων Nonce καταχωρίσεων του πίνακα PIT, καθώς και της λίστας Dead Nonce. Εφόσον υπάρξει αντιστοίχιση, το εισερχόμενο αίτημα Ενδιαφέροντος θεωρείται αντίγραφο, είτε λόγω ανίχνευσης επανάληψης είτε λόγω πολλαπλών αφίξεων, οπότε και δίδεται στο για περαιτέρω επεξεργασία στον αγωγό ανίχνευσης επανάληψης. Εφόσον δε βρεθεί κάτι, συνεχίζεται η περαιτέρω επεξεργασία.
- Σε επόμενο βήμα ο χρονοδιακόπτης μη ικανοποίησης και ο straggler χρονοδιακόπτης της PIT καταχώρισης ακυρώνονται, αφού ένα νέο αίτημα καταφθάνει, οπότε και επεκτείνεται το χρονικό όριο ζωής της PIT καταχώρισης. Οι χρονοδιακόπτες θα μπορούσαν να επανακαθοριστούν αργότερα, στο μονοπάτι επεξεργασίας Ενδιαφέροντος.
- Ο αγωγός στη συνέχεια εξετάζει εάν κάποιο αίτημα βρίσκεται σε αναμονή, όπως στην περίπτωση που ο PIT διαθέτει μια ίδια ή από οποιαδήποτε άλλη οντότητα Προσώπου, ως εσωτερική καταγραφή. Υπενθυμίζουμε ότι η PIT καταχώριση μπορεί να αναπαριστά ακόμη και ένα πακέτο Ενδιαφέροντος που έχει ικανοποιηθεί πρόσφατα.
- Εφόσον δεν εντοπίζεται κάποια εκκρεμότητα, πραγματοποιείται αντιστοίχιση με το πακέτο Δεδομένων του ContentStore(**Cs::find**).

3.6 Αγωγός Επεξεργασίας Δεδομένων

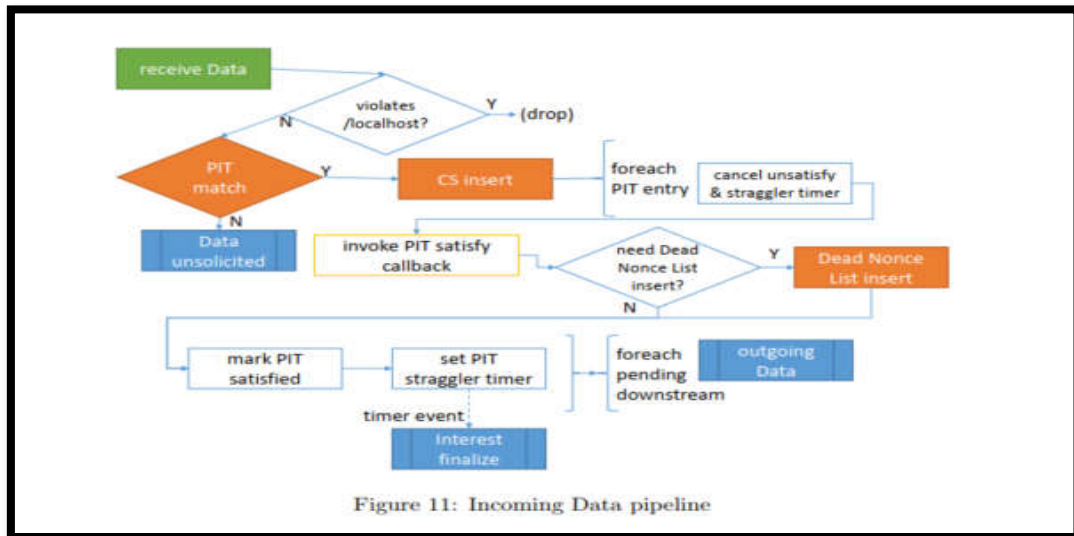
Ο αγωγός επεξεργασίας των Δεδομένων χωρίζεται στα κάτωθι μέρη:

- Εισερχόμενα Δεδομένα
- Μη ζητηθέντα(αυτόκλητα) Δεδομένα
- Εξερχόμενα Δεδομένα

Η ανάλυση των προγραφέντων μερών, πραγματοποιείται διεξοδικά στις επόμενες υπο-ενότητες.

3.6.1 Αγωγός Εισερχόμενων Δεδομένων

Τα βήματα της διαδικασίας μπορούν να συνοψιστούν ως εξής:



Εικόνα 9 Απεικόνιση του μονοπατιού επεξεργασίας πακέτου Δεδομένου[15]

- Στο πρώτο βήμα ελέγχεται το ενδεχόμενο τοπικής παραβίασης. Πρόκειται για δικλείδα ασφαλείας, έναντι κακόβουλων αποστολών
- Στη συνέχεια γίνεται χρήση του αλγορίθμου ταυτοποίησης στον πίνακα PIT. Εφόσον δε βρεθεί κάποια PIT καταχώριση, τα Δεδομένα θεωρούνται αυτόκλητα, όποτε παρέχονται στον αγωγό Αυτόκλητων Δεδομένων (Data unsolicited pipeline)
- Εάν μια ή περισσότερες καταχωρίσεις PIT εντοπιστούν, τα δεδομένα εισάγονται στον CS. Το χρονικό διάστημα κατά το οποίο τα δεδομένα θα παραμείνουν στον Πίνακα Αποθήκευσης καθορίζεται αποκλειστικά από την πολιτική του
- Σε επόμενο βήμα πραγματοποιείται ακύρωση του χρονοδιακόπτη μη ικανοποίησης και του χρονοδιακόπτη straggler, για κάθε εντοπισμένη καταχώριση του PIT, αφού το εν αναμονή αίτημα ικανοποιείται
- Η καταχώριση PIT στη συνέχεια θεωρείται επιτυχής, οπότε πραγματοποιείται διαγραφή κάθε εσωτερικής και εξωτερικής καταχώρισης που ανταποκρίνεται στην οντότητα εισερχομένου Προσώπου των Δεδομένων.

3.6.2 Αγωγή Αυτόκλητων Δεδομένων(Unsolicited)

Η υλοποίηση του αγωγού εφαρμόζεται βάσει της μέθοδου (**Forwarder::onDataUnsolicited**), όταν ένα πακέτο Δεδομένων θεωρηθεί αυτόκλητο. Οι εισαγόμενοι στον αγωγό παράμετροι περιλαμβάνουν ένα πακέτο Δεδομένων και την εισερχόμενη οντότητα Προσώπου.

Κατά γενική ομολογία τα αυτόκλητα πακέτα Δεδομένων πρέπει να απορρίπτονται αφού ενέχουν πιθανούς κινδύνους. Παρόλα αυτά, υπάρχουν περιπτώσεις δεδομένων τα οποία πρέπει να γίνουν δεκτά από τον Content Store.

Ειδικότερα, επιτρέπεται από την τρέχουσα υλοποίηση η προσωρινή αποθήκευση ενός αυτόκλητου πακέτου Δεδομένων, εφόσον αυτό αφιχθεί από μια τοπική οντότητα Προσώπου.

3.6.3 Αγωγή Εξερχόμενων Δεδομένων

Η υλοποίηση του αγωγού πραγματοποιείται από τη μέθοδο (**Forwarder::onOutgoingData**). Όταν τα δεδομένα εντοπιστούν στον Πίνακα Αποθήκευσης(CS) χρησιμοποιείται ο αγωγός Εισερχόμενων Ενδιαφερόντων. Εφόσον τα δεδομένα αντιστοιχίζονται με μία ή περισσότερες καταχωρίσεις PIT, χρησιμοποιείται ο αγωγός Εισερχόμενων Δεδομένων.

Οι παράμετροι εισόδου στον αγωγό περιλαμβάνουν ένα πακέτο Δεδομένων, καθώς και την εξερχόμενη οντότητα Προσώπου. Τα βήματα που ακολουθούνται για την προγραφείσα υλοποίηση, έχουν ως εξής:

- Αρχικά τα δεδομένα ελέγχονται για οποιαδήποτε παραβίαση /localhost, αφού τα πακέτα του συγκεκριμένου προθέματος δεν μπορούν να αποσταλούν σε μη τοπικές Οντότητες Προσώπου
- Το επόμενο βήμα δεσμεύεται για τον διαχειριστή κυκλοφορίας δεδομένων, η υλοποίηση του οποίου δεν έχει ακόμη ενσωματωθεί
- Τελικά, τα πακέτα Δεδομένων αποστέλλονται μέσω της εξερχόμενης Οντότητας Προσώπου

3.7 Διαχείριση Συμφόρησης στο NDN

Στην αρχιτεκτονική NDN οι εκάστοτε Καταναλωτές ρυθμίζουν τον αριθμό πακέτων Δεδομένων που θα αποσταλούν στο δίκτυο. Ο ενδιαμέσος κόμβος και ο Κόμβος Παραγωγού μπορούν οποιαδήποτε στιγμή να επιστρέψουν ένα πακέτο σήμανσης, ώστε να μετριάσει ο αριθμός αποστολής από πλευράς Καταναλωτή. Η δομή του NDN δεν επιτρέπει τη διαχείριση του ελέγχου των εισερχόμενων πακέτων Ενδιαφέροντος, από πλευράς Παραγωγού.

Βέβαια, οι ενδιαμέσσοι κόμβοι μπορούν να κάνουν στραγγαλίσουν την κίνηση επιλέγοντας τον επιθυμητό ρυθμό αποστολής αιτημάτων Ενδιαφέροντος, στα οποία θα ανταποκριθούν. Παρόλα αυτά, σε κάποια δεδομένη χρονική στιγμή θα επέλθει κορεσμός του πίνακα PIT.

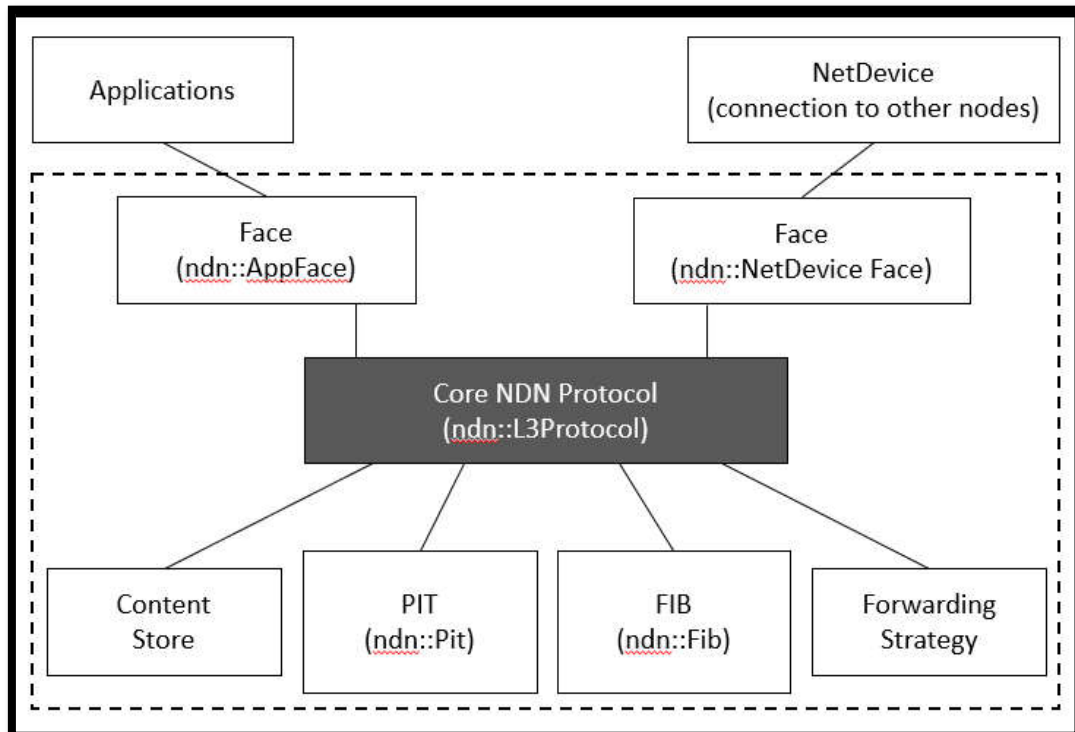
Συνοπτικά υπάρχουν δύο διαθέσιμοι μέθοδοι ελέγχου συμφόρησης της αρχιτεκτονικής NDN, όπου πραγματοποιείται:

- πρότερος έλεγχος του αριθμού αιτημάτων από πλευράς Καταναλωτή
- στραγγαλισμός του αριθμού αιτημάτων από τους ενδιαμέσους δρομολογητές-κόμβους

Μια τρίτη υβριδική λύση αφορά τη συνδυαστική χρήση των δύο προαναφερθέντων επιλογών.

3.8 Προσομοιωτής ndnSIM

Στις προηγούμενες ενότητες έγινε αναλυτική περιγραφή της αρχιτεκτονικής Named Data Networking. Για λόγους απλοποίησης πραγματοποιείται παράθεση του κάτωθι μπλοκ διαγράμματος, το οποίο αποτελεί συνοπτική απεικόνιση της προγραφείσας ανάλυσης :



Εικόνα 10 Συνοπτική περιγραφή της εσωτερικής δομής λειτουργίας του ndn Simulator

Ο ndn-SIMulator αποτελεί ένα ερευνητικό εργαλείο που αποσκοπεί την ενσωμάτωση και αξιοποίηση της σχεδιαστικής υλοποίησης Named Data Networking Forwarding Daemon.

Στη βάση του ο προσομοιωτής ndnSIM χαρακτηρίζεται ως ένα σύνολο συνδεδεμένων στοιχείων, τα οποία προσδίδουν τη δυνατότητα τροποποίησης ή αντικατάστασης σε έναν ερευνητή, με παράλληλο μετριάσμό των επιπτώσεων στα υπόλοιπα ενσωματωμένα μέρη του.

Η υλοποίηση του επεκτείνει τον ήδη προ-υπάρχων NS-3 προσομοιωτή, διατηρώντας ταυτόχρονα την ανοιχτού κώδικα δομή του. Η υλοποίηση προσομοιώσεων μεγάλης κλίμακας παρέχει τα αναγκαία εργαλεία κατανόησης δομικών εννοιών λειτουργίας, των NDN πρωτοκόλλων.

4 Σενάρια και Εργαλεία Αξιολόγησης

4.1 Εφαρμογές παραγωγής πακέτων Ενδιαφέροντος ndnSIM

ConsumerCbr

Αποτελεί βασική υλοποίηση εφαρμογής καταναλωτή[16]. Μεταδίδει πακέτα ενδιαφέροντος βάσει προ καθορισμένης συχνότητας, χωρίς να απαιτείται η λήψη κάποιου πακέτου NACK ή κάποιας ενδεχόμενης σήμανσης συμφόρησης.

Consumer Zipf Mandelbrot

Λειτουργεί στη βάση του κυλιόμενου παραθύρου[16]. Το παράθυρο δεν καθορίζεται από τον αριθμό των πακέτων εν αναμονή, όπως συμβαίνει στην TCP υλοποίηση, αλλά από τον αριθμό των πακέτων Ενδιαφέροντος που αποστέλλονται συνολικά στο δίκτυο. Η συγκεκριμένη εφαρμογή ενσωματώνει την κατανομή Zipf-Mandelbrot [17].

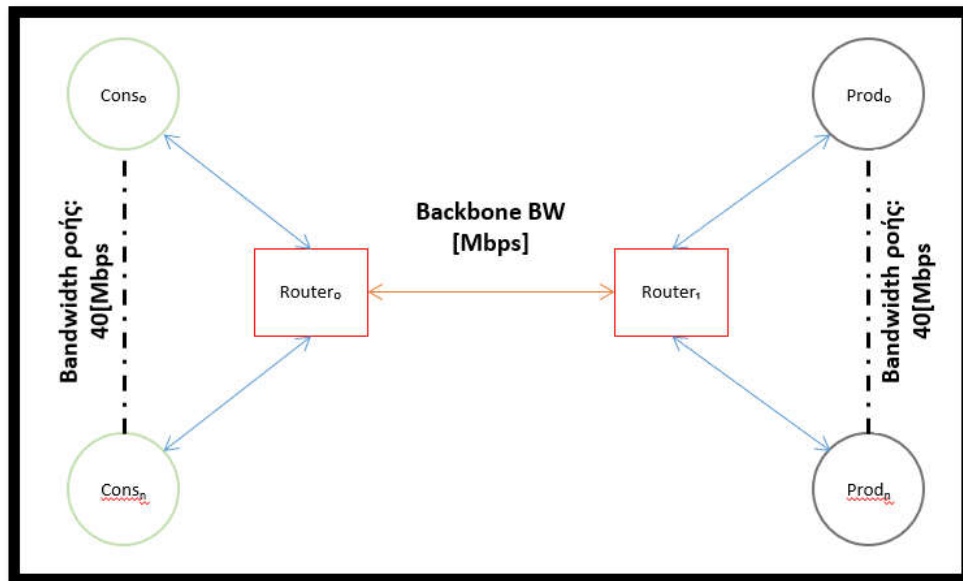
Ο νόμος Zipf αποτελεί έναν εμπειρικό νόμο της στατιστικής, όπου διάφορα δεδομένα, είτε επιστημονικού είτε κοινωνικού χαρακτήρα, περιγράφονται βάσει μιας πιθανοκρατικής κατανομής (power law probability distribution). Ο νόμος Zipf συνοψίζεται ως εξής:

“Η αριθμητική θέση μιας λέξης σε μια ταξινομημένη λίστα σχετίζεται με την μειωμένη συχνότητα εμφάνισης της”

Επεξηγηματικά, η συχνότητα της ν-οστής πιο συχνά εμφανιζόμενης λέξης είναι $1/n$ φορές η συχνότητα της πιο συχνής.

4.2 Τοπολογία και Υλοποίηση στον ndnSIM

Για τους σκοπούς της παρούσας εργασίας χρησιμοποιήθηκε η dumbbell τοπολογία, με χρήση point-to-point διασυνδέσεων, μεταξύ των Καταναλωτών και των Παραγωγών. Καθότι το αντικείμενο της πραγματευόταν την αξιολόγηση της απόδοσης της δικτυακής υποδομής, για μεγάλα μεταβλητά μεγέθη πακέτων, ευρισκόμενο υπό μεταβλητές συνθήκες διαθέσιμου εύρους ζώνης και καθυστέρησης διάδοσης, χρησιμοποιήθηκαν δύο κόμβοι έκαστοι, σε κάθε άκρο.



Εικόνα 11 Απεικόνιση της υλοποιημένης dumbbell τοπολογίας

Παραθέτουμε τον υλοποιημένο σε C++ κώδικα, για την αρχικοποίηση της dumbbell τοπολογίας:

- Αρχικά, ορίζουμε τις αναγκαίες διασυνδέσεις (modules) για την επιτυχή υλοποίηση της τοπολογίας :

```
#include "ns3/core-module.h"
#include "ns3/network-module.h"
#include "ns3/point-to-point-module.h"
#include "ns3/ndnSIM-module.h"
#include "ns3/point-to-point-net-device.h"
```

- Εν συνεχεία μεταβαίνουμε στο κυρίως μέρος. Παρακάτω εντοπίζεται η συνάρτηση που καλείται για τον εντοπισμό του χρόνου παραμονής της εκάστοτε καταχώρισης, στον πίνακα Content Store :

```
namespace ns3 {
//Χρόνος ζωής εκάστοτε καταχώρισης στον Πίνακα Περιεχομένων (CS)
void
CacheEntryRemoved(std::string context, Ptr<const ndn::cs::Entry> entry, Time
Lifetime)
{
    std::cout << entry->GetName() << " " << Lifetime.ToDouble(Time::S) << "s" <<
    std::endl;
}
}
```

- Μετέπειτα ορίζουμε τις αναγκαίες για την τοπολογία προϋποθέσεις, προβαίνοντας σε αρχικοποίηση συγκεκριμένων μεταβλητών:

```
Int main(int argc, char* argv[])
{
    //Παραμετροποίηση μεγέθους buffer
    Config::SetDefault ("ns3::QueueBase::MaxSize", StringValue ("200p"));
    int consumernodes = 2; // αρχικοποίηση του αριθμού κόμβου καταναλωτή
    int producernodes = 2; // αρχικοποίηση του αριθμού κόμβου παραγωγού
    int routers = 2 ; // αρχικοποίηση του αριθμού κόμβου δρομολογητών

    //Αρχικοποίηση των μεταβλητών που εισάγονται από το πληκτρολόγιο
    std::string cachingStrategy = "LRU";
    std::string numCacheEntries = "1000";
    std::string packetsize = "5000";
    std::string interests = "900";
    std::string delayvalue = "10ms";
}
```

- Το επόμενο κομμάτι επιτρέπει την εισαγωγή τιμών από πλευράς χρήστη:

```
// Εισαγωγή των τιμών από το πληκτρολόγιο
CommandLine cmd;
cmd.AddValue ("csStrategy", "cachingStrategy(LRU,LFU,FIFO,RANDOM,nocache)",
cachingStrategy);
cmd.AddValue ("cacheEntries", "numCacheEntries", numCacheEntries);
cmd.AddValue ("packetsize", "packetsize", packetsize);
cmd.AddValue ("interests", "interests pers second", interests);
cmd.AddValue ("delayvalue", "channel in [ms]", delayvalue);
cmd.Parse(argc, argv);
```

- Επιπλέον προβαίνουμε σε αρχικοποίηση του εκάστοτε αντικειμένου-κόμβου, όπως επίσης και στην αναγκαία ονοματοδοσία:

```
//Δημιουργία Node Containers
NodeContainer consumer;
NodeContainer router;
NodeContainer producer;
//Δημιουργία των Κόμβων
consumer.Create(consumernodes);
producer.Create(producernodes);
router.Create(routers);
//Ονοματοδοσία
Names::Add ("Consumer1", consumer.Get (0));
Names::Add ("Consumer2", consumer.Get (1));
Names::Add ("Rtr1", router.Get(0));
Names::Add ("Rtr2", router.Get(1));
Names::Add ("Producer1", producer.Get(0));
```

```
Names::Add ("Producer2", producer.Get(1));
```

- Θέτουμε τις αναγκαίες για την point-to-point διασύνδεση παραμέτρους:

```
//Διασύνδεση με χρήση point to point καναλιών
//Αρχικοποίηση των καναλιών-αντικειμένων, για τα διασυνδεόμενα κανάλια μεταξύ
καταναλωτών- παραγωγών και δρομολογητή, όπως και μεταξύ των δύο δρομολογητών
PointToPointHelper p2p;
p2p.SetChannelAttribute ("Delay", StringValue (delayvalue));
p2p.SetDeviceAttribute ("DataRate", StringValue ("40Mbps"));
/*p2p.SetQueue("ns3::DropTailQueue", "MaxPackets", UintegerValue(50)); */
PointToPointHelper rp2p;
rp2p.SetChannelAttribute ("Delay", StringValue (delayvalue));
rp2p.SetDeviceAttribute ("DataRate", StringValue ("60Mbps"));
//rp2p.SetQueue("ns3::DropTailQueue", "MaxPackets", UintegerValue(80));
//Εγκατάσταση του απαραίτητου συνδέσμου-καναλιού
p2p.Install(consumer.Get(0), router.Get(0));
p2p.Install(consumer.Get(1), router.Get(0));
rp2p.Install(router.Get(0), router.Get(1));
p2p.Install(router.Get(1), producer.Get(0));
p2p.Install(router.Get(1), producer.Get(1));
```

- Παρακάτω πραγματοποιείται εφαρμογή της κατάλληλης πολιτικής διαχείρισης ουράς με παράλληλη αρχικοποίηση του Content Store:

```
// Δημιουργία του NDN Stack
ndn::StackHelper ndnHelper;
// Χρήση της παλαιάς εφαρμογής Content Store
if (cachingStrategy == "nocache"){
    std::cout << "Setting cache of routers to no-cache" << std::endl;
    ndnHelper.SetOldContentStore("");
    ndnHelper.setCsSize(1); // disable content store
}
else if (cachingStrategy == "LRU") {
    std::cout << "Using LRU caching strategies on routers " << std::endl;
    ndnHelper.setCsSize(0); // use old content store
    ndnHelper.SetOldContentStore ("ns3::ndn::cs::Stats::Lru", "MaxSize",
numCacheEntries);
    ndnHelper.Install(consumer);
    ndnHelper.Install(producer);
    ndnHelper.Install(router);
//Χρόνος ζωής της εκάστοτε καταχώρισης πακέτου Δεδομένων
Config::Connect("/NodeList/5/$ns3::ndn::cs::Stats::Lru/WillRemoveEntry",
MakeCallback(CacheEntryRemoved));
}
else if (cachingStrategy == "LFU"){
    std::cout << "Using LFU caching strategies on routers " << std::endl;
```

```

    ndnHelper.setCsSize(0); // use old content store
    ndnHelper.SetOldContentStore      ("ns3::ndn::cs::Stats::Lfu", "MaxSize",
numCacheEntries);
    ndnHelper.Install(consumer);
    ndnHelper.Install(producer);
    ndnHelper.Install(router);
    //Χρόνος ζωής της εκάστοτε καταχώρισης πακέτου Δεδομένων
    Config::Connect("/NodeList/5/$ns3::ndn::cs::Stats::Lfu/WillRemoveEntry",
MakeCallback(CacheEntryRemoved));
}
else if (cachingStrategy == "FIFO") {
    std::cout << "Using FIFO caching strategies on routers " << std::endl;
    ndnHelper.setCsSize(0); // use old content store
    ndnHelper.SetOldContentStore      ("ns3::ndn::cs::Stats::Fifo", "MaxSize",
numCacheEntries);
    ndnHelper.Install(consumer);
    ndnHelper.Install(producer);
    ndnHelper.Install(router);
    //Χρόνος ζωής της εκάστοτε καταχώρισης πακέτου Δεδομένων
    Config::Connect("/NodeList/5/$ns3::ndn::cs::Stats::Fifo/WillRemoveEntry",
MakeCallback(CacheEntryRemoved));
}
else if (cachingStrategy == "RANDOM") {
    std::cout << "Using RANDOM caching strategies on routers " << std::endl;
    ndnHelper.setCsSize(0); // use old content store
    ndnHelper.SetOldContentStore      ("ns3::ndn::cs::Stats::Random", "MaxSize",
numCacheEntries);
    ndnHelper.Install(consumer);
    ndnHelper.Install(producer);
    ndnHelper.Install(router);
    //Χρόνος ζωής της εκάστοτε καταχώρισης πακέτου Δεδομένων
    Config::Connect("/NodeList/5/$ns3::ndn::cs::Stats::Random/WillRemoveEntry",
MakeCallback(CacheEntryRemoved));
}
else {
    fprintf(stderr, "No proper caching strategy selected!");
    return 1;
}
}

```

- Επιλέγουμε την στρατηγική προώθησης και επιπλέον εγκαθιστούμε τις εφαρμογές Καταναλωτή και Πελάτη, στους αντίστοιχους κόμβους:

```

//Επιλογή στρατηγικής προώθησηςChoosing
    ndn::StrategyChoiceHelper::InstallAll("/prefix",
"/localhost/nfd/strategy/best-route");
// Εγκατάσταση καθολικών διεπαφών δρομολόγησης για όλους τους κόμβους

```

```

ndn::GlobalRoutingHelper ndnGlobalRoutingHelper;
ndnGlobalRoutingHelper.InstallAll();
//Εγκατάσταση εφαρμογής Καταναλωτή
ndn::AppHelper::consumerHelper("ns3::ndn::ConsumerZipfMandelbrot");
consumerHelper.SetAttribute("Frequency", StringValue(interests));
consumerHelper.SetAttribute("NumberOfContents", StringValue("2000"));//
Ορισμός αριθμού διαφορετικού περιεχομένου για την ZipfMandelbrot
// Πρόθεμα Καταναλωτή
consumerHelper.SetPrefix("/dst1");
consumerHelper.Install(consumer.Get(0));
consumerHelper.SetPrefix("/dst2");
consumerHelper.Install(consumer.Get(1));
// Εγκατάσταση Εφαρμογής Παραγωγού
ndn::AppHelper producerHelper("ns3::ndn::Producer");
producerHelper.SetAttribute("PayloadSize", StringValue(packetSize));
ndnGlobalRoutingHelper.AddOrigins("/dst1", producer.Get(0));
producerHelper.SetPrefix("/dst1");
producerHelper.Install(producer.Get(0));
ndnGlobalRoutingHelper.AddOrigins("/dst2", producer.Get(1));
producerHelper.SetPrefix("/dst2");
producerHelper.Install(producer.Get(1));
//Υπολογισμός και εγκατάσταση των διαδρομών FIBs
ndn::GlobalRoutingHelper::CalculateRoutes();

```

- Εν κατακλείδι, παρατίθεται το τελευταίο τμήμα του κώδικα, όπου κατά κύριο λόγο περιλαμβάνονται μηνύματα οθόνης, οι ανιχνευτές δεδομένων[18] όπως και ο χρόνος προσομοίωσης:

```

std::cout << "Starting simulation :)" << std::endl;
Simulator::Stop(Seconds(30.0));
//Κλήση των ενσωματωμένων ανιχνευτών L3RateTracer, CsTracer, AppDelayTracer
ndn::L3RateTracer::InstallAll("through.txt", Seconds(0.5));
ndn::CsTracer::InstallAll("cs.txt", Seconds(0.5));
ndn::AppDelayTracer::InstallAll("delay.txt");
Simulator::Run();
Simulator::Destroy();
std::cout << "Simulation ended" << std::endl;
return 0;
}
}
// namespace ns3
int
main(int argc, char* argv[])
{
    return ns3::main(argc, argv);
}

```

Επιπροσθέτως, παρέχονται στιγμιότυπα από τον προσομοιωτή ndnSIM, κατά τη στιγμή του compile και της εκτέλεσης ενός στιγμιότυπου προσομοίωσης:

```
christos@christos-Inspiron-5570:~/ndn/scenario$ ./waf --run "test1 --PrintHelp"
Waf: Entering directory '/home/christos/ndn/scenario/build'
Waf: Leaving directory '/home/christos/ndn/scenario/build'
'build' finished successfully (0.104s)
test1 [Program Options] [General Arguments]

Program Options:
  --csStrategy:      cachingStrategy(LRU,LFU,FIFO,RANDOM,nocache) [LRU]
  --cacheEntries:    numCacheEntries [1000]
  --packetSize:      packetSize [5000]
  --interests:       interests pers second [900]
  --delayValue:      channel in [ms] [10ms]

General Arguments:
  --PrintGlobals:      Print the list of globals.
  --PrintGroups:       Print the list of groups.
  --PrintGroup=[group]: Print all TypeIds of group.
  --PrintTypeIds:      Print all TypeIds.
  --PrintAttributes=[typeid]: Print all attributes of typeid.
  --PrintHelp:         Print this help message.

christos@christos-Inspiron-5570:~/ndn/scenario$
```

Εικόνα 12 Στιγμιότυπο του προσομοιωτή ndnSIM, όπου διαφαίνεται η δυνατότητα εισαγωγής πολλαπλών τιμών από τον χρήστη

```
christos@christos-Inspiron-5570:~/ndn/scenario$ ./waf --run "test1 --csStrategy=LFU --packetSize=10000 --interests=450 --delayValue=10ms"
Waf: Entering directory '/home/christos/ndn/scenario/build'
Waf: Leaving directory '/home/christos/ndn/scenario/build'
'build' finished successfully (0.100s)
Using LFU caching strategies on routers
Starting simulation :)
Simulation ended
christos@christos-Inspiron-5570:~/ndn/scenario$
```

Εικόνα 13 Επιτυχής εκτέλεση στιγμιότυπου προσομοίωσης

4.3 Περιγραφή πειράματος

Για την εκτέλεση του πειραματικού μέρους χρησιμοποιήθηκαν 3 διαφορετικά σενάρια συμφόρησης, τα οποία περιγράφονται στο παρακάτω:

| Σενάριο | BW [Mbps] Backbone | BW [Mbps] Καταναλωτών-Παραγωγών |
|----------------|--------------------|---------------------------------|
| 1 ^ο | 160 | 40 |
| 2 ^ο | 80 | 40 |
| 3 ^ο | 60 | 40 |

Για ένα σενάριο μελετήθηκε η επίδραση του μεγέθους πακέτου στη συνολική απόδοση του δικτύου. Η συσχέτιση Πακέτων Ενδιαφέροντος και Μεγέθους Πακέτου Δεδομένων, δίδεται παρακάτω:

| Πακέτα Ενδιαφέροντος | Μέγεθος Πακέτου Δεδομένων [KB] |
|----------------------|--------------------------------|
| 900 | 5 |
| 450 | 10 |
| 225 | 20 |
| 112 | 40 |
| 56 | 80 |
| 42 | 120 |
| 28 | 160 |

Στη συνέχεια παρουσιάζεται η επίδραση της μεταβλητής καθυστέρησης διάδοσης του βέλτιστου μεγέθους πακέτου Δεδομένων, συναρτήσει μεταβλητής καθυστέρησης διάδοσης, για τις τέσσερις διαφορετικές πολιτικές διαχείρισης ουράς του πίνακα Περιεχομένου:

| Καθυστέρηση διάδοσης [ms] για πακέτο Δεδομένων 120 [KB] | | | | | | |
|---|----|----|----|-----|-----|-----|
| 10 | 20 | 40 | 80 | 160 | 200 | 400 |

Επιπροσθέτως πραγματοποιήθηκαν μικρές τροποποιήσεις που αφορούν την δυνατότητα διαχείρισης μεγάλων μεγεθών πακέτων από τον προσομοιωτή ndnSIM:

Στην τοποθεσία [ndn/ns-3/src/ndnSIM/ndn-cxx/ndn-cxx/encoding/tlv.hpp](https://github.com/ndn-cxx/ndn-cxx/blob/master/ndn-cxx/encoding/tlv.hpp) έγινε τροποποίηση της μεταβλητής MAX_NDN_PACKET_SIZE, στην τιμή των 100[MB], για την υποστήριξη μεγάλων πακέτων Δεδομένων. Η αρχικοποιημένη τιμή της μεταβλητής εντοπιζόταν στα 8800[Byte].

Εν τέλει, οι προσδιδόμενες από τις παραπάνω υλοποιήσεις μετρικές επεξεργάστηκαν με χρήση του γραφικού περιβάλλοντος, της προγραμματιστικής γλώσσας R.

4.4 Προγραμματιστικό περιβάλλον R

Για την επεξεργασία των πολυπληθών μετρικών χρησιμοποιήθηκε το γραφικό περιβάλλον R Studio. Η γλώσσα R[19] αποτελεί ένα σύγχρονο περιβάλλον υπολογιστικής στατιστικής, καθότι παρέχει ευρεία ποικιλία επεκτάσιμων στατιστικών και γραφικών τεχνικών.

Το γραφικό περιβάλλον της R διαθέτει δυνατότητα διαχείρισης και υπολογισμού δεδομένων, όπως και γραφικής απεικόνισης. Παρακάτω συνοψίζονται τα βασικά χαρακτηριστικά του, τα οποία περιλαμβάνουν:

- Αποτελεσματική διαχείριση και αποθήκευση δεδομένων
- Εκτέλεση πολλαπλών και πολύπλοκων υπολογισμών για σειρές δεδομένων

- Πολυποίκιλη, συναφή και ενσωματωμένη συλλογή επεξεργαστικών εργαλείων για λόγους στατιστικής ανάλυσης
- Δυνατότητα γραφικής απεικόνισης σε πραγματικό χρόνο

Επιπρόσθετα η σαφώς ορισμένη, απλή και αποδοτική δομή, επιτρέπει την ύπαρξη πολυπληθών προγραμματιστικών συνθηκών και συναρτήσεων. Τέλος, δίδεται η δυνατότητα μαζικής εισαγωγής και εξαγωγής δεδομένων.

Τα παραπάνω συνιστούν στην ταξινόμηση της R, ως ένα πλήρως οργανωμένο υπολογιστικό περιβάλλον, εν αντιθέσει με την κοινότυπη άποψη, που της προσδίδει μόνο την ιδιότητα της στατιστικής υπολογιστικής.

4.5 Υλοποιημένα R πρότυπα

Για την επιτυχή ανάλυση των μετρικών χρησιμοποιήθηκαν τα κάτωθι R προγραμματιστικά πρότυπα:

4.5.1 Throughput

```
#setwd("~/Desktop ") #Ορισμός προορισμού θέσης

through <- list() #Αρχικοποίηση λίστας
list_throughput<- dir(pattern = "*.txt") #Δημιουργία της λίστας όλων .txt αρχείων στη θέση προορισμού
{
  through[[k]] <- read.table(list_throughput[k],header=T)
}
for (k in 1:length(list_throughput)) #byte to bit conversion
{
  through[[k]]$Kilobits = through[[k]]$Kilobytes * 8
}
for (k in 1:length(list_throughput)) #Δημιουργία υποσυνόλου απαραίτητων στηλών
{
  through[[k]] = subset( through[[k]], Type %in% c( "InData" ))
  through[[k]] = subset( through[[k]], Node %in% c("Rtr1"))
  through[[k]]$ID = k
  through[[k]] = summaryBy(. ~ Time + Node + Type + ID, data = through[[k]], FUN=sum)
  through[[k]] = subset( through[[k]], select = -c(FacId.sum,Node, Type, Packets.sum, Kilobytes.sum,
KilobytesRaw.sum))
}
through_final = bind_rows(through)
through_final = summarise(group_by(through_final, ID), value = mean(Kilobits.sum))
write_tsv(through_final, path = "throughput.txt") #Εξαγωγή σε αρχείο κατάληξης .txt
```

4.5.2 Επιτυχής Ανάκτηση (CacheHits)

```
setwd("~/Desktop/ ") #Ορισμός προορισμού θέσης
#freq = c(900,450,225,112,56,40,28) #Δημιουργία πίνακα με τον αριθμό των πακέτων Ενδιαφέροντος ανά
#προσομοίωση
cs <- list() #Αρχικοποίηση λίστας
list_cs<- dir(pattern = "*.txt") #Δημιουργία της λίστας όλων .txt αρχείων στη θέση προορισμού
for (k in 1:length(list_cs))
{
  cs[[k]] = read.table(list_cs[k],header=T)
}
for (k in 1:length(list_cs)) #Δημιουργία υποσυνόλου απαραίτητων στηλών
{
  cs[[k]] = subset(cs[[k]], Type %in% c("CacheHits"))
  cs[[k]] = subset(cs[[k]], Node %in% c("Rtr1","Consumer1","Consumer2"))
  cs[[k]] = subset(cs[[k]], select = -c(Time, Type)) #Node
  cs[[k]]$ID = k
}
for (k in 1:length(list_cs)) #Υπολογισμός της συχνότητας CacheHits ανά προσομοίωση μεγέθους πακέτου.
#Η τιμή έχει εύρος [0-1].
{
  # cs[[k]]$Packets = cs[[k]]$Packets/freq[[k]]
  cs[[k]]$Packets = cs[[k]]$Packets/freq
}
cs_finale = bind_rows(cs)
cs_finale = cs_finale[cs_finale$Packets != 0, ] #Έλεγχος μηδενικών τιμών στήλης
cs_finale = summarise(group_by(cs_finale, ID, Node), value= mean(Packets))
write_tsv(cs_finale, path = "cachehits.txt" #Εξαγωγή σε αρχείο κατάληξης .txt
```

4.5.3 Συνολική καθυστέρηση (Delay)

```
#setwd("~/Desktop/") #Ορισμός προορισμού θέσης
delay <- list() #Αρχικοποίηση λίστας
list_delay<- dir(pattern = "*.txt") #Δημιουργία της λίστας όλων .txt αρχείων στη θέση προορισμού
for (k in 1:length(list_delay))
{
  delay[[k]] <- read.table(list_delay[k],header=T)
}
for (k in 1:length(list_delay)) #Δημιουργία υποσυνόλου απαραίτητων στηλών(FullDelay)
{
  delay[[k]] = subset(delay[[k]], Type %in% c("FullDelay" ))
  delay[[k]] = subset(delay[[k]], select = -c(Time,Appld, SeqNo, Type, DelayUS, RetxCount, HopCount))
  delay[[k]]$ID = k
}
```

```

delay_finale = bind_rows(delay)
delay_finale = summaryBy(. ~ID+Node, data=delay_finale, FUN=mean)
write_tsv(delay_finale, path = "delaymean.txt") #Εξαγωγή σε αρχείο κατάληξης .txt

```

4.5.4 Χρόνος ζωής πακέτου Δεδομένων (Cache Entry Lifetime)

```

#setwd("~/Desktop/ ") #Ορισμός προορισμού θέσης

cstime<- list() #Αρχικοποίηση λίστας
list_cstime<- dir(pattern = ".*") #Δημιουργία της λίστας όλων .txt αρχείων στη θέση προορισμού

for (k in 1:length(list_cstime))
{
  cstime[[k]] <- read.table(list_cstime[k],header=F)
  # cstime[[k]] <- lapply(cstime[[k]], setNames, colnames)
  cstime[[k]]$ID <- k
}

csfinale <- bind_rows(cstime)
csfinale=csfinale[!grepl("/localhost", csfinale$V1),] #Αφαίρεση τιμών χαρακτήρων "localhost"
csfinale=csfinale[!grepl("Simulation", csfinale$V1),] #Αφαίρεση τιμών χαρακτήρων "Simulation"

entries = csfinale %>% group_by(ID,V2) %>% tally() #Ταξινόμηση
entries$V2 = gsub("s","",entries$V2) #Αφαίρεση τιμών χαρακτήρα "s"
entries$V2 = sapply(entries$V2, as.numeric) #Αριθμητική μετατροπή

lifetime_finale=summarise(group_by(entries, ID), value = weighted.mean(V2,n)) #Υπολογισμός του σταθμισμένου μέσου όρου

write_tsv(lifetime_finale, path = "entries.txt") #Εξαγωγή σε αρχείο κατάληξης .txt

```

4.5.5 Βιβλιοθήκες

Η εκτέλεση των προγραμμάτων script ήταν εφικτή με την ενσωμάτωση των παρακάτω βιβλιοθηκών:

- **ggplot2**: Η χρήση της βιβλιοθήκης αποσκοπεί στη δημιουργία γραφημάτων
- **scales**: Η χρήση της βιβλιοθήκης αποσκοπεί στην παροχή μεθόδων παραμετροποίησης γραφημάτων
- **doBy**: Η χρήση της βιβλιοθήκης επιτρέπει διάφορες διαδικασίες ομαδοποιημένων δεδομένων
- **dplyr**: Η χρήση της βιβλιοθήκης επιτρέπει τον χειρισμό δεδομένων με χρήση απλών ονοματολογικά συναρτήσεων
- **readr**: Η χρήση της βιβλιοθήκης επιτρέπει την ταχύτατη και φιλική στο χρήστη ανάγνωση δεδομένων

5 Αποτελέσματα

Η κάτωθι ενότητα προβαίνει σε αναλυτική παρουσίαση των εξαχθέντων δεδομένων, τα οποία αντλήθηκαν από μία σειρά πολυπληθών και συνάμα στοχευμένων, προσομοιώσεων.

Όπως ήδη αναφέρθηκε, οι προσομοιώσεις εκτελέστηκαν με διαφορετικές συνθήκες συμφόρησης (BW), συναρτήσει διαφορετικών-μεταβλητών- μεγεθών πακέτων Δεδομένων. Εκ παραλλήλου, αξιοποιούνται τέσσερις διαφορετικές ουρές επεξεργασίας(FIFO, LFU, LRU, RANDOM), οι οποίες διαδραματίζουν αξιοσημείωτο ρόλο στην συνολική απόδοση δικτύου.

Οι τρεις επόμενες υπό-ενότητες στο πρώτο σκέλος τους αφορούν το μεταβλητό [BW] της backbone διασύνδεσης, με παράλληλη μεταβολή του μεγέθους πακέτων Δεδομένων, για τις εφαρμογές παραγωγής πακέτων Ενδιαφέροντος ConsumerCbr και ConsumerZipfMandelbrot.

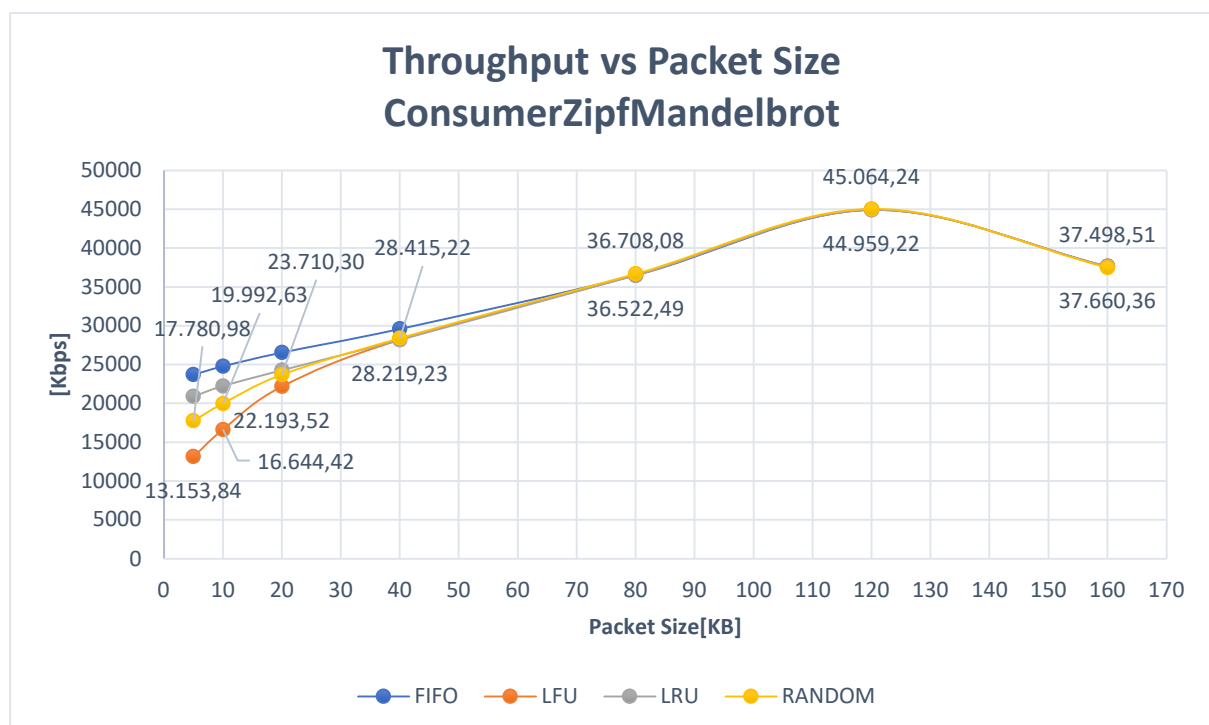
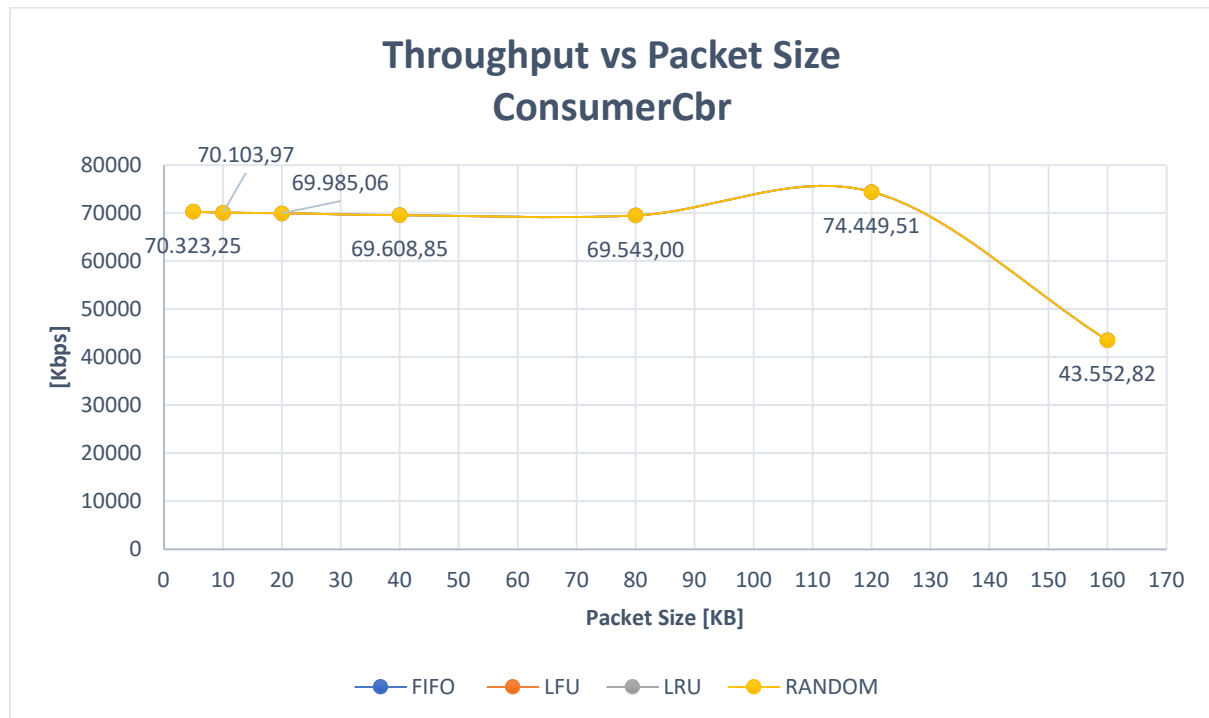
Στο δεύτερο σκέλος, αφού αποσαφηνιστεί το βέλτιστο μέγεθος πακέτου Δεδομένων, προβαίνουμε σε διαφοροποίηση των τιμών της καθυστέρησης διάδοσης, ώστε να εξεταστεί η επίδραση στη συνολική απόδοση του δικτύου, δεδομένων των τεσσάρων διαφορετικών πολιτικών επεξεργασίας ουράς.

5.1 Σενάριο 1^ο

5.1.1 Μεταβλητό μέγεθος πακέτου δεδομένων

5.1.1.1 Throughput

Οι κάτωθι γραφικές αναπαραστάσεις αποτελούν συγκριτική απεικόνιση μεταξύ των δύο διαφορετικών εφαρμογών παραγωγής πακέτων Ενδιαφέροντος, ConsumerCbr και ConsumerZipfMandelbrot, για μεταβλητό μέγεθος πακέτων Δεδομένων.



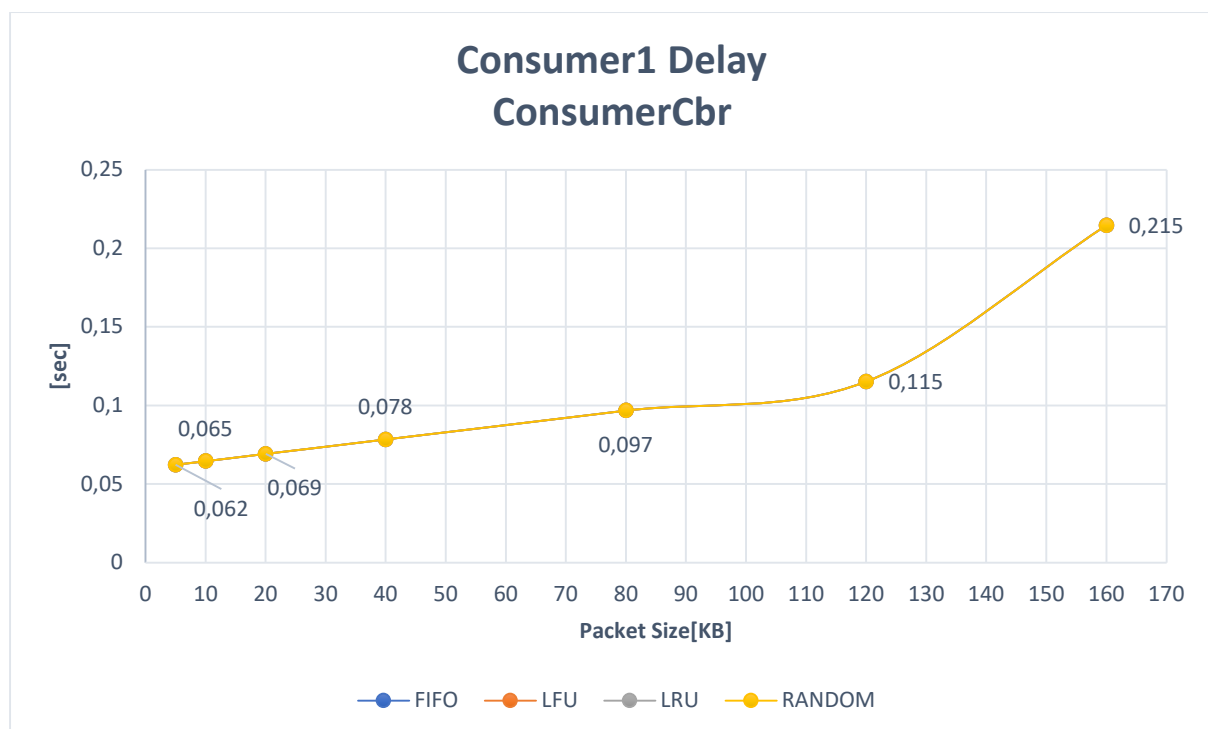
Οι προσομοιώσεις έχουν πραγματοποιηθεί για τέσσερις διαφορετικές πολιτικές επεξεργασίας ουράς.

Παρατηρούμε πως το throughput παραμένει σχετικά σταθερό κατά την αύξηση του μεγέθους πακέτου, στην περίπτωση της αποστολής δεδομένων με σταθερό ρυθμό -ConsumerCbr-, ανεξάρτητα από την πολιτική διαχείρισης ουράς(cache) που χρησιμοποιείται.

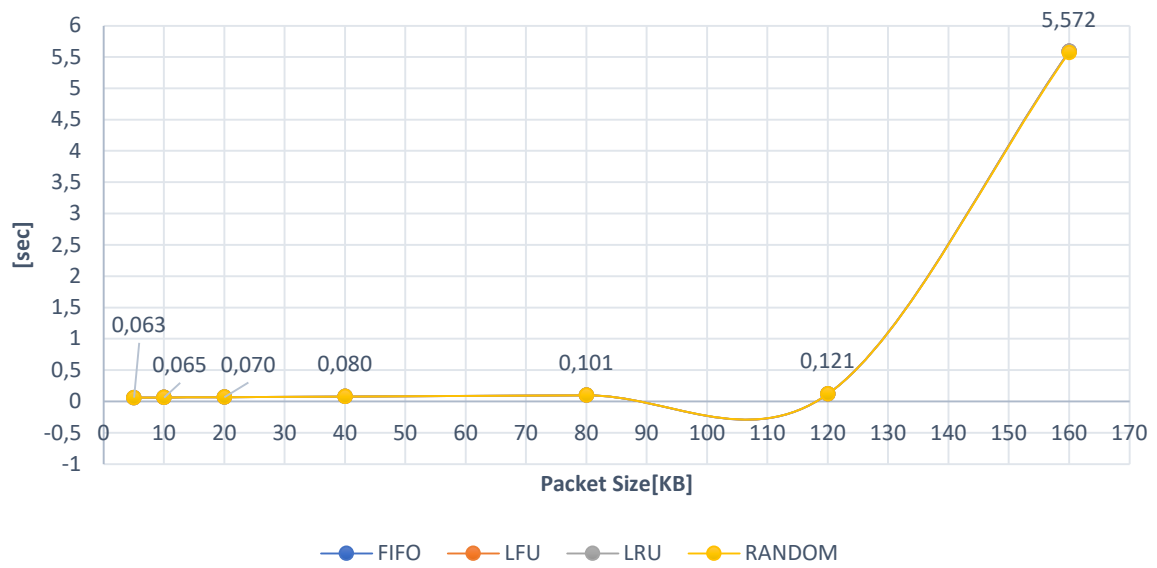
Αντίθετα, στην περίπτωση δημιουργίας κίνησης με την κατανομή ConsumerZipfMandelbrot, καθώς αυξάνεται το μέγεθος πακέτου, αυξάνεται αναλογικά και το throughput. Βάσει του προηγούμενου και δεδομένων των ίδιων συνθηκών δημιουργίας κίνησης με την ConsumerCbr, όπου ο αριθμός των αιτημάτων ανά δευτερόλεπτο παρέμενε ίδιος, η αυξητική πορεία αξιοποίησης του καναλιού οφείλεται στην συνεχόμενη ζήτηση διαφορετικών πακέτων Δεδομένων από τους Καταναλωτές.

5.1.1.2 Συνολική καθυστέρηση – Delay

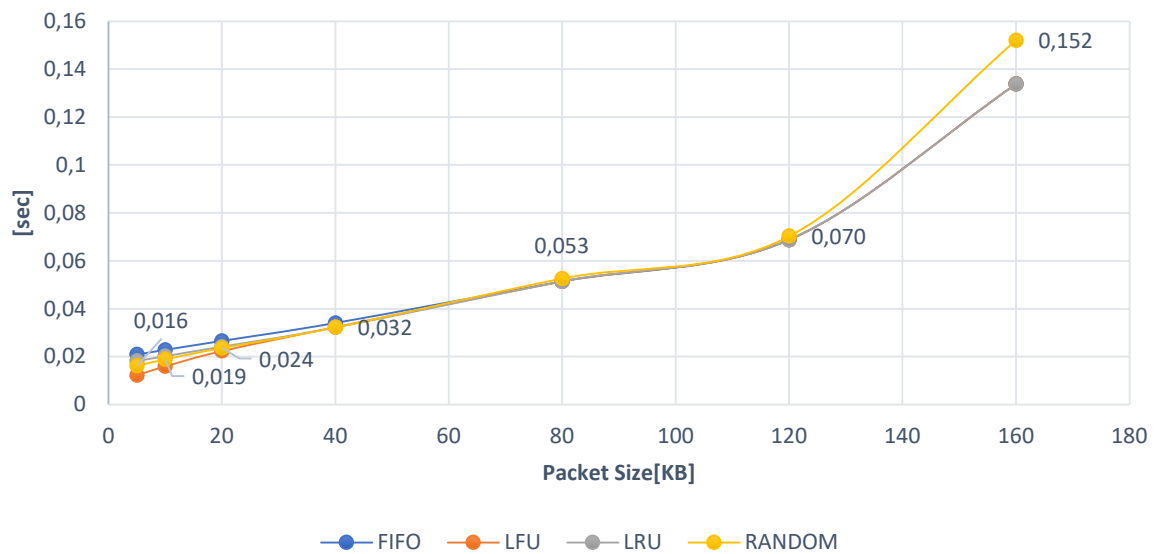
Οι κάτωθι γραφικές αναπαραστάσεις αποτελούν συγκριτική απεικόνιση μεταξύ των δύο διαφορετικών εφαρμογών παραγωγής πακέτων Ενδιαφέροντος, για μεταβλητό μέγεθος πακέτων Δεδομένων, στους δύο κόμβους Καταναλωτή.

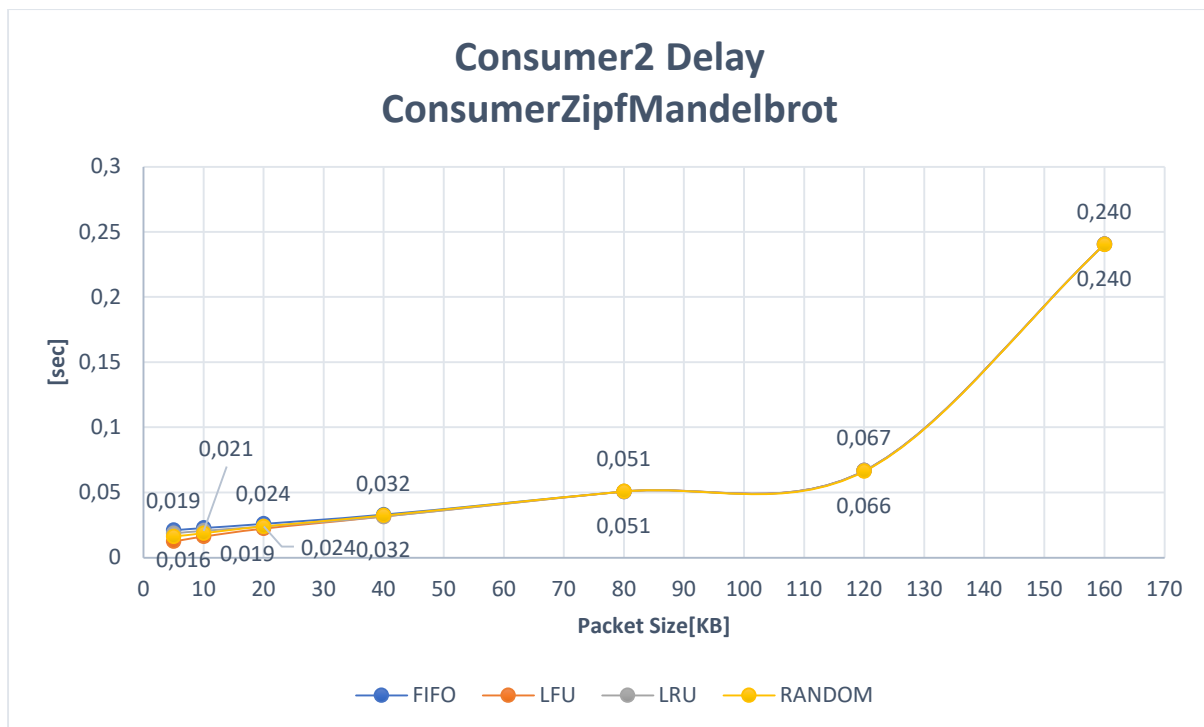


Consumer2 Delay ConsumerCbr



Consumer1 Delay ConsumerZipfMandelbrot





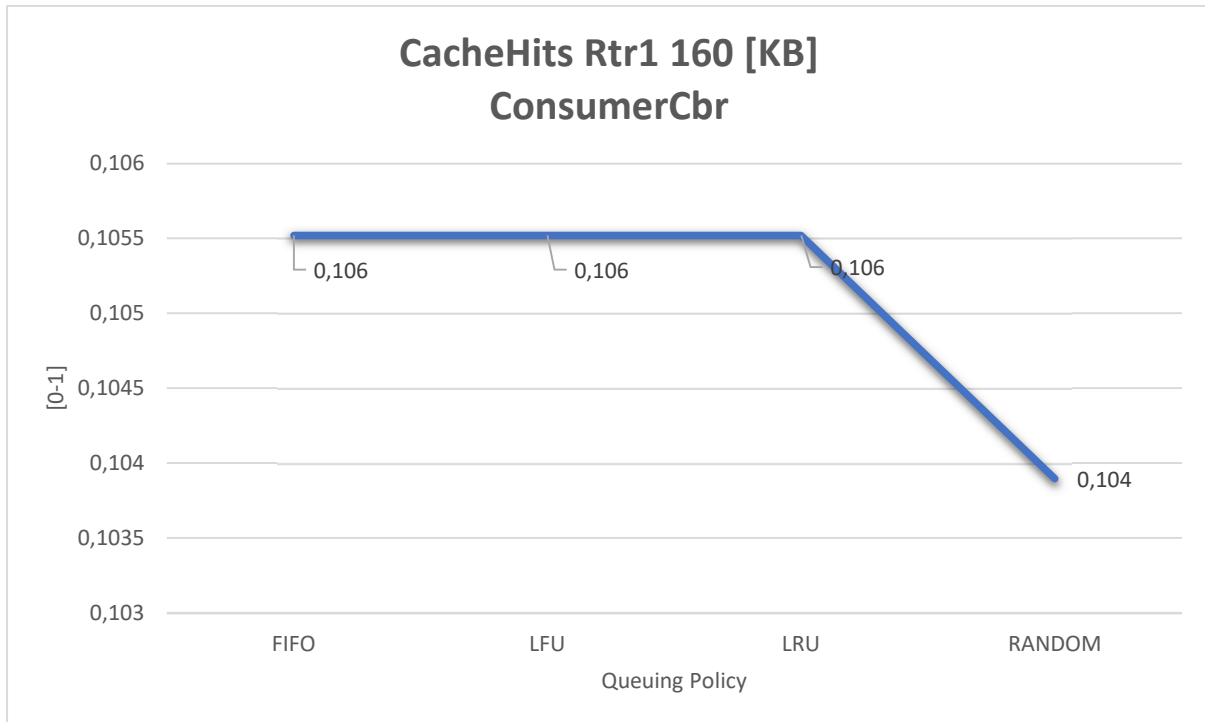
Επομένως, στην παρούσα υπό-παράγραφο εντοπίζεται η μέση τιμή της συνολικής καθυστέρησης που παρατηρείται, από την αποστολή ενός πακέτου Ενδιαφέροντος έως την άφιξη ενός πακέτου Δεδομένων, για τις τέσσερις διαφορετικές πολιτικές επεξεργασίας ουράς, συναρτήσει του μεταβλητού μεγέθους πακέτων Δεδομένων. Από στατιστικής σκοπιάς αξιοποιήθηκαν οι τιμές της στήλης **Delays**.

Συμπερασματικά, δεν διαφαίνεται επαρκής διαφορά σε ότι αφορά την εκάστοτε πολιτική επεξεργασίας ουράς. Ως εκ τούτου, οι όποιες διαφοροποιήσεις των τιμών συνολικής καθυστέρησης, οφείλονται στο μεταβλητό μέγεθος πακέτου Δεδομένων.

Ωστόσο, παρατηρούμε ότι σε αντίθεση με το throughput, η καθυστέρηση διάδοσης αυξάνεται σταθερά με την αυξητική μεταβολή του μεγέθους πακέτου, ανεξάρτητα από την πολιτική επεξεργασίας ουράς(cache) που χρησιμοποιείται.

5.1.1.3 Επιτυχής ανάκτηση – CacheHits

Οι κάτωθι γραφικές αναπαραστάσεις αποτελούν συγκριτική απεικόνιση μεταξύ των δύο διαφορετικών εφαρμογών παραγωγής πακέτων Ενδιαφέροντος, για μεταβλητό μέγεθος πακέτων Δεδομένων.

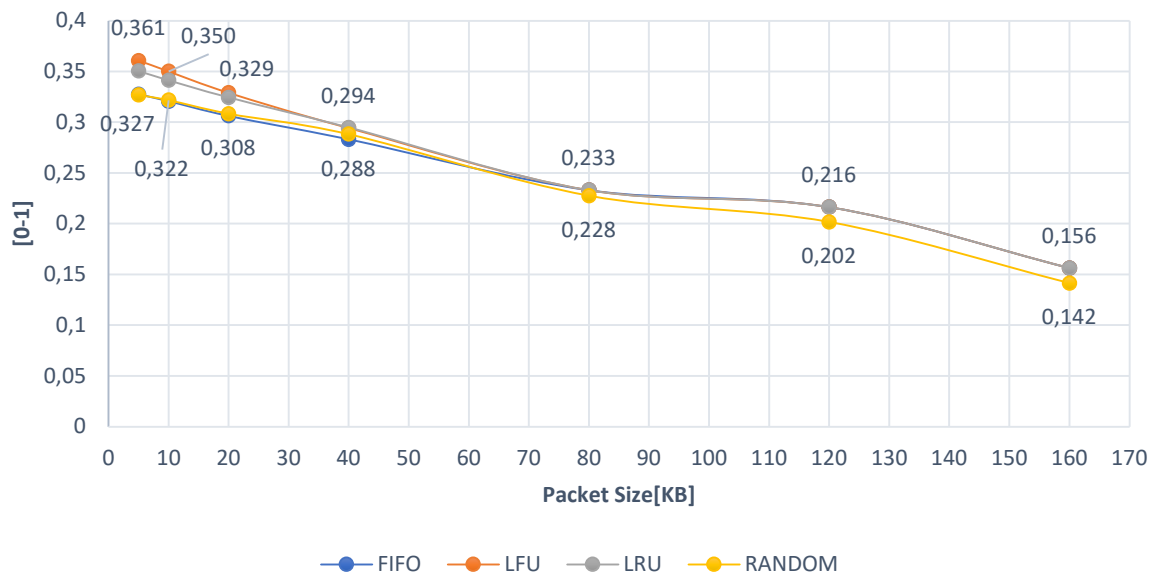


Το προηγούμενο γράφημα αντιπροσωπεύει τα ποσοστά ανάκτησης κατά την υλοποίηση της εφαρμογής ConsumerCbr.

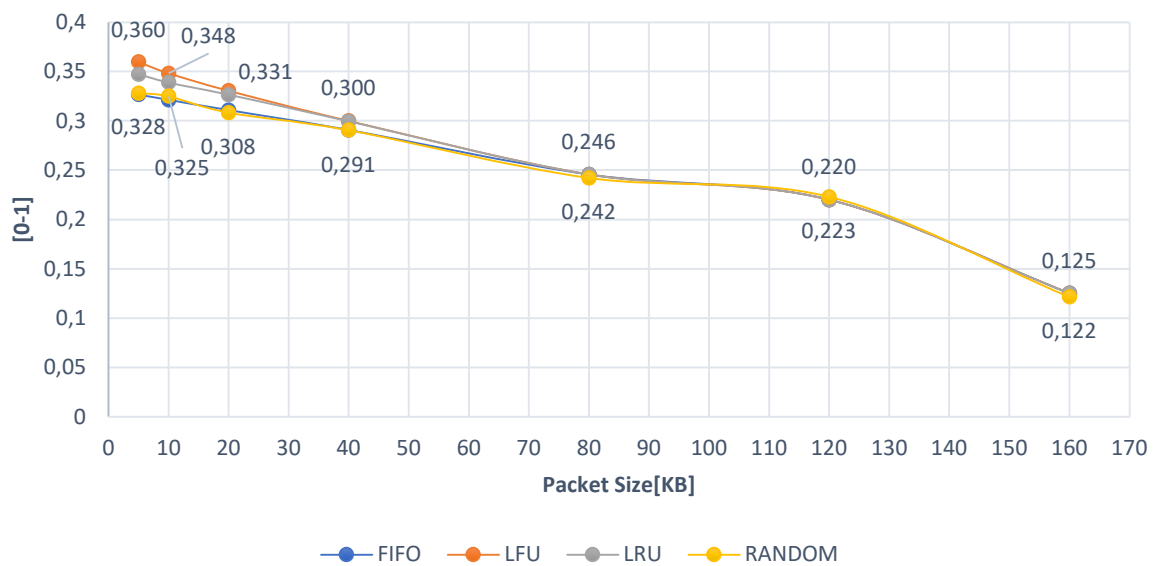
Λόγω της σχεδιαστικής δομής της, η ConsumerCbr επιζητά συνεχώς καινούρια πακέτα Δεδομένων, στα οποία προσδίδει διαφορετικό πρόθεμα ονόματος. Επομένως, δεν είναι εφικτή η ανάκτηση πακέτων Δεδομένων από τον πίνακα Content Store, αφού κανένα “πανομοιότυπο” πακέτο, δεν πρόκειται να απαιτηθεί.

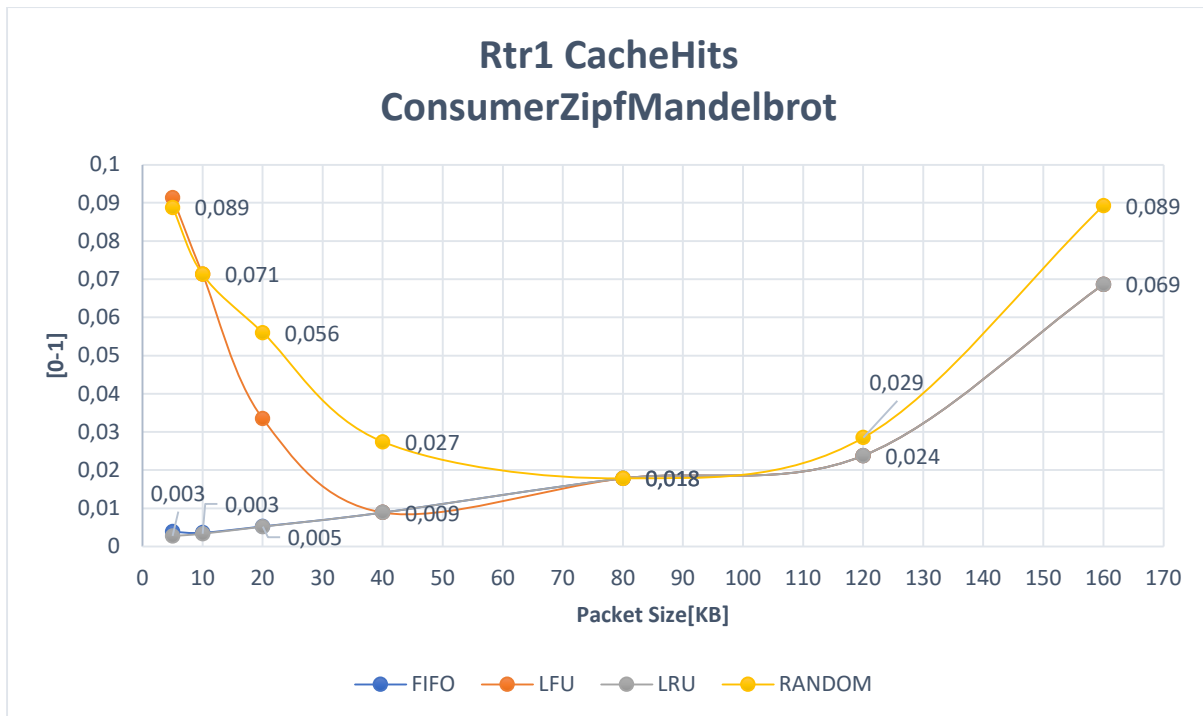
Όμως, μόνο κατά την προσέγγιση του σημείου κορεσμού, όπου παρατηρείται τμηματική κατάρρευση της τοπολογίας, παρατηρήθηκε ανάκτηση πακέτων Δεδομένων από τον Content Store.

Consumer1 CacheHits ConsumerZipfMandelbrot



Consumer2 CacheHits ConsumerZipfMandelbrot





Συμπερασματικά, στην παρούσα υπό-παράγραφο παρουσιάζεται ποσοστιαία ο αριθμός επιτυχούς ανάκτησης πακέτων Δεδομένων, από τον πίνακα Περιεχομένου – Content Store.

Από τη μία πλευρά η εγγενής φύση της εφαρμογής ConsumerCbr δεν επιτρέπει την αίτηση του ίδιου πακέτου Δεδομένων, καθότι το εκάστοτε πρόθεμα πακέτου Ενδιαφέροντος διαφοροποιείται.

Από την άλλη, η ίδια εγγενής δομή της εφαρμογής ConsumerZipfMandelbrot που προσεγγίζει τον τρόπο αίτησης νέων πακέτων Δεδομένων βάσει πραγματικών συνθηκών, επιτρέπει την στοιχειοθέτηση δεδομένων που αφορούν τη διαδικασία επιτυχούς ανάκτησης από τον πίνακα Περιεχομένου.

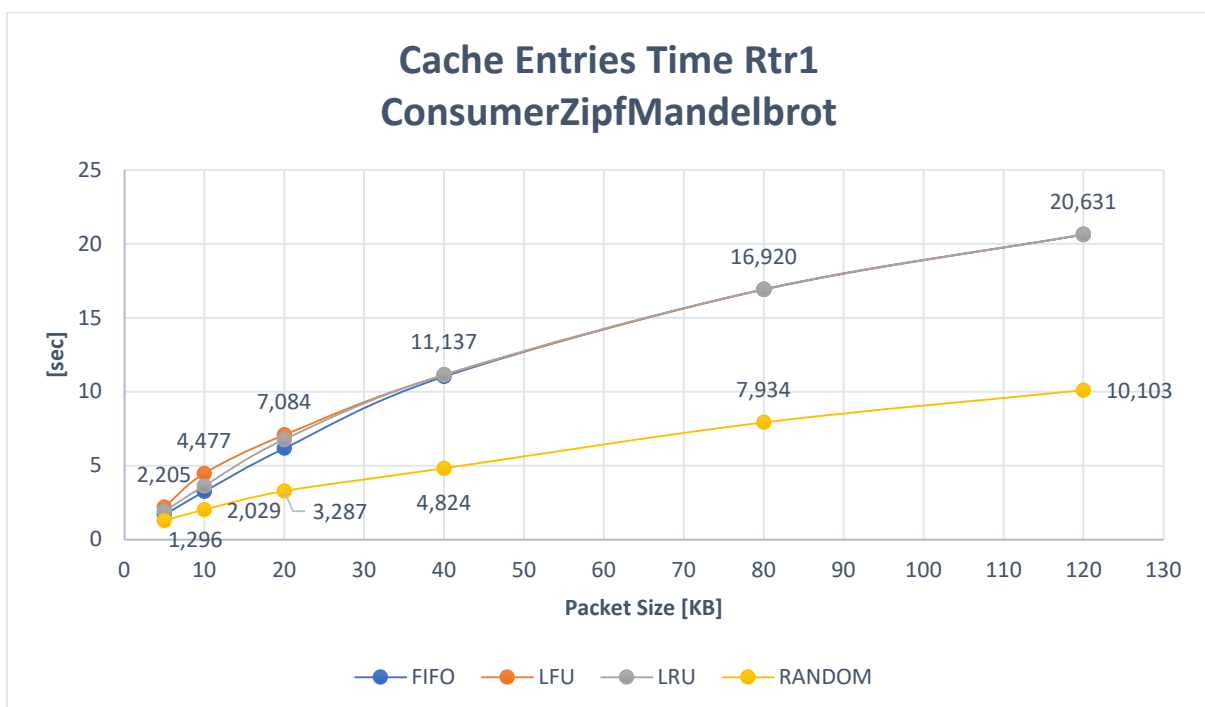
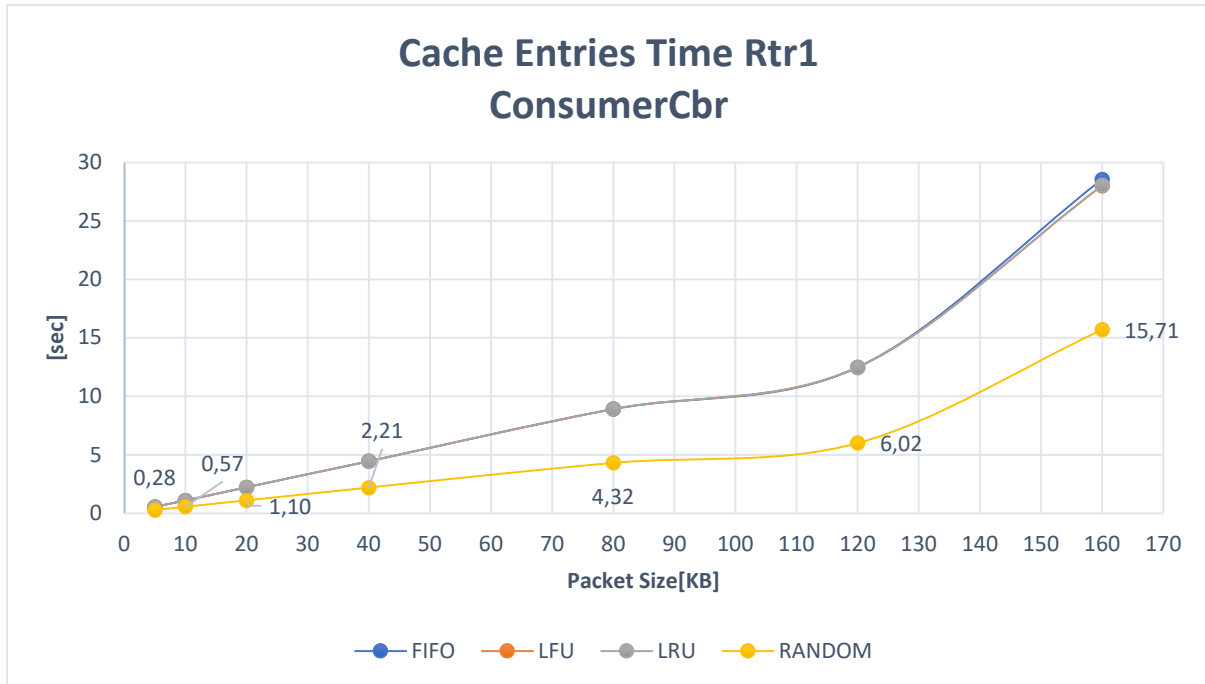
Μολονότι πολλαπλά πακέτα Δεδομένων ενδέχεται να αιτούνται από τον ίδιο χρήστη σε μεταγενέστερη χρονική στιγμή, εξετάζονται τα ποσοστά επιτυχούς ανάκτησης στους κόμβους και των δύο Καταναλωτών, καθώς και στον πλησιέστερο (πρώτο) δρομολογητή.

Η συνεχώς μειούμενη τιμή ανάκτησης πακέτων Δεδομένων από τους κόμβους Καταναλωτή, οφείλεται στην ποικιλία διαφορετικού περιεχομένου την οποία μπορεί να στην προσωρινή μνήμη.

Από στατιστικής σκοπιάς αξιοποιήθηκαν οι τιμές της στήλης **Packets** του ενσωματωμένου ανιχνευτή δεδομένων.

5.1.1.4 Χρόνος ζωής καταχωρίσεων - Entries Life Time

Οι κάτωθι γραφικές αναπαραστάσεις αποτελούν συγκριτική απεικόνιση μεταξύ των δύο διαφορετικών εφαρμογών παραγωγής πακέτων Ενδιαφέροντος, για μεταβλητό μέγεθος πακέτων Δεδομένων, στον κόμβο δρομολογητή-**Rtr1**.



Επομένως, παρουσιάζεται ο μέσος χρόνος αναμονής των πακέτων Δεδομένων, στον Πίνακα Περιεχομένου(CS). Για λόγους μεγαλύτερης ακρίβειας και επειδή διαφορετικά πακέτα της ίδιας προσομοίωσης, παραμένουν

διαφορετικά χρονικά διαστήματα στην προσωρινή μνήμη, χρησιμοποιήθηκε η μέθοδος του σταθμισμένου μέσου όρου.

Για μικρότερα μεγέθη πακέτων Δεδομένων ο χρόνος παραμονής στην προσωρινή μνήμη είναι μικρότερος, σε σχέση με τα πακέτα μεγαλύτερου μεγέθους. Είναι εμφανές ότι λόγω του περιορισμένου χώρου της εκάστοτε προσωρινής μνήμης και της δημιουργίας πολλαπλών αιτημάτων Ενδιαφέροντος για τα μικρά μεγέθη πακέτων Δεδομένων, η δικτυακή δομή οδηγήθηκε σε αντικατάσταση των παλαιότερων πακέτων.

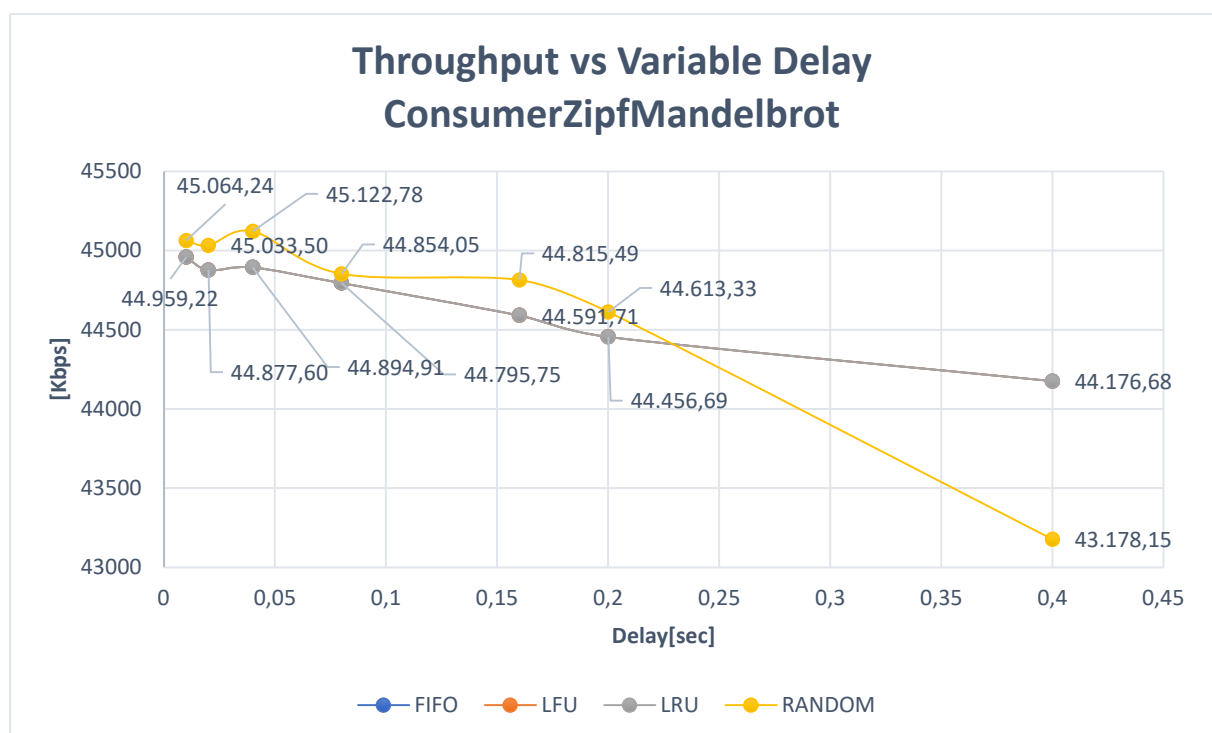
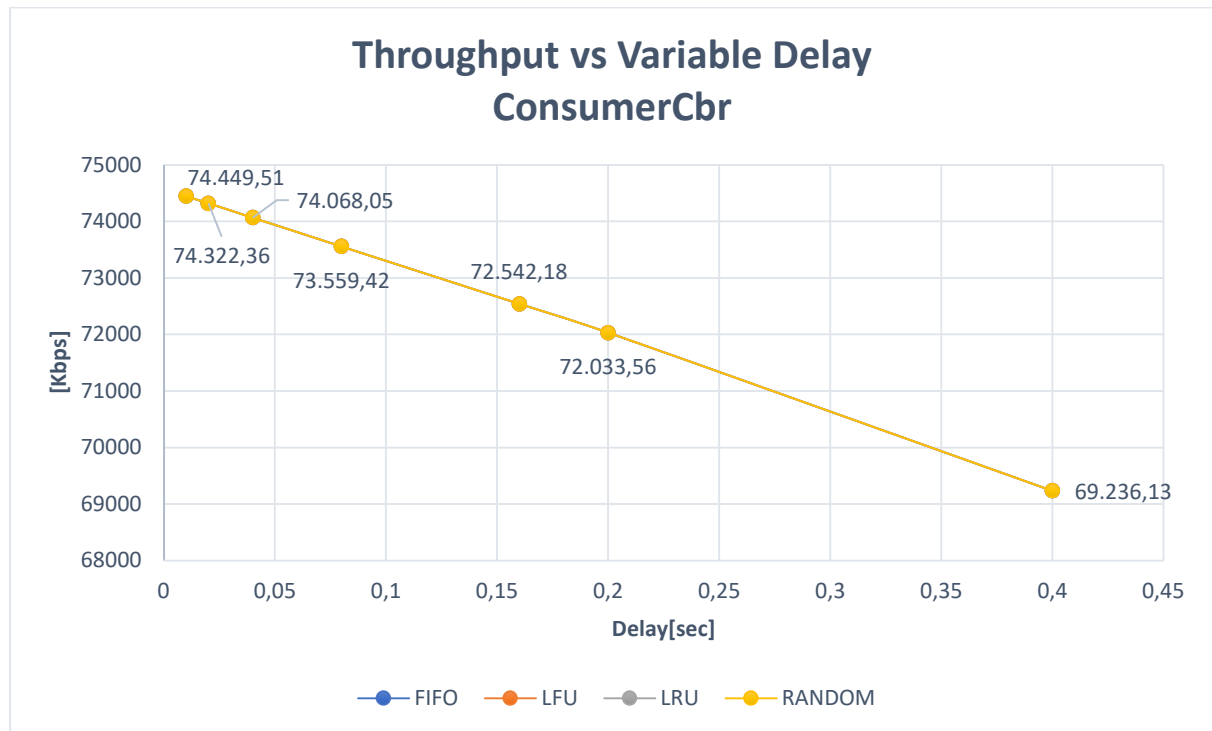
Στο σημείο αυτό πρέπει να αποσαφηνιστεί ότι μονάδα μέτρησης του αποθηκευτικού χώρου στην προσωρινή μνήμη είναι ο αριθμός πακέτων. Καθίσταται σαφές ότι δεν υπάρχει εξάρτηση από το μέγεθος του.

Η μεθοδολογία ακολουθεί την ίδια ακριβώς δομή, αφού εξετάζεται η συνολική αποδοτικότητα του δικτύου για τις τέσσερις διαφορετικές πολιτικές επεξεργασίας, συναρτήσει του μεταβλητού μεγέθους πακέτου Δεδομένων.

5.1.2 Μεταβλητή τιμή καθυστέρησης διάδοσης

5.1.2.1 Throughput

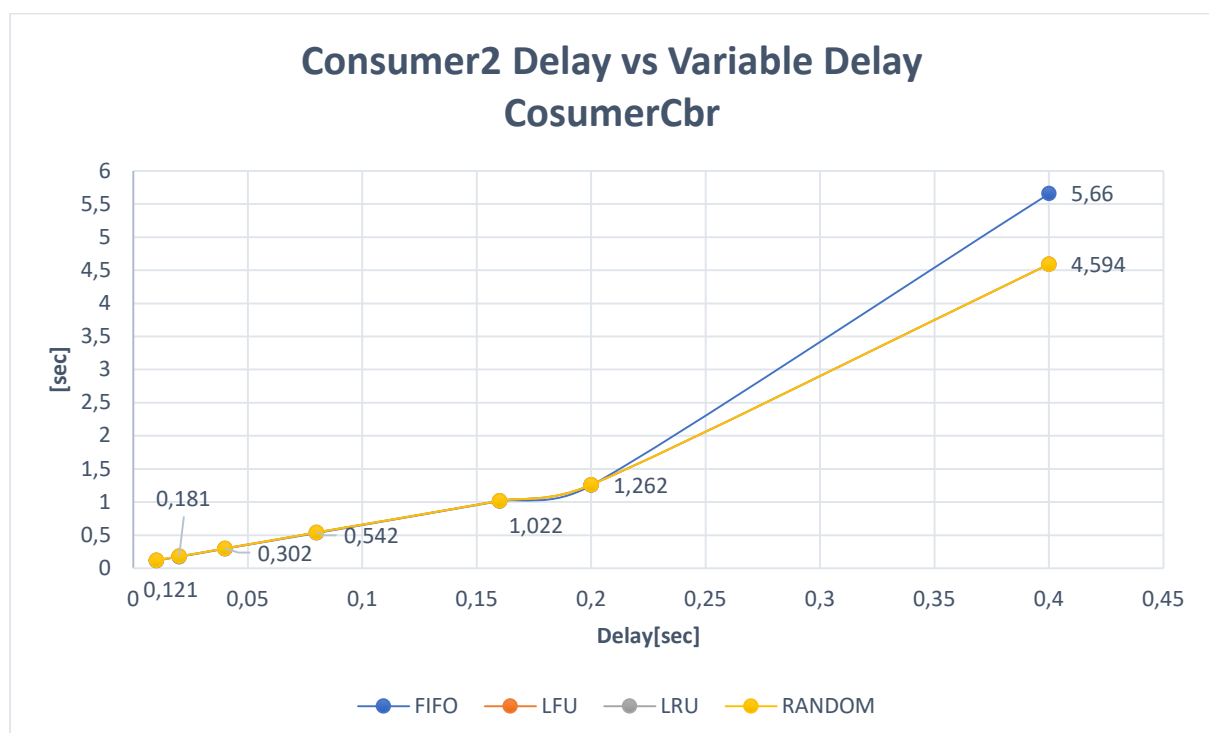
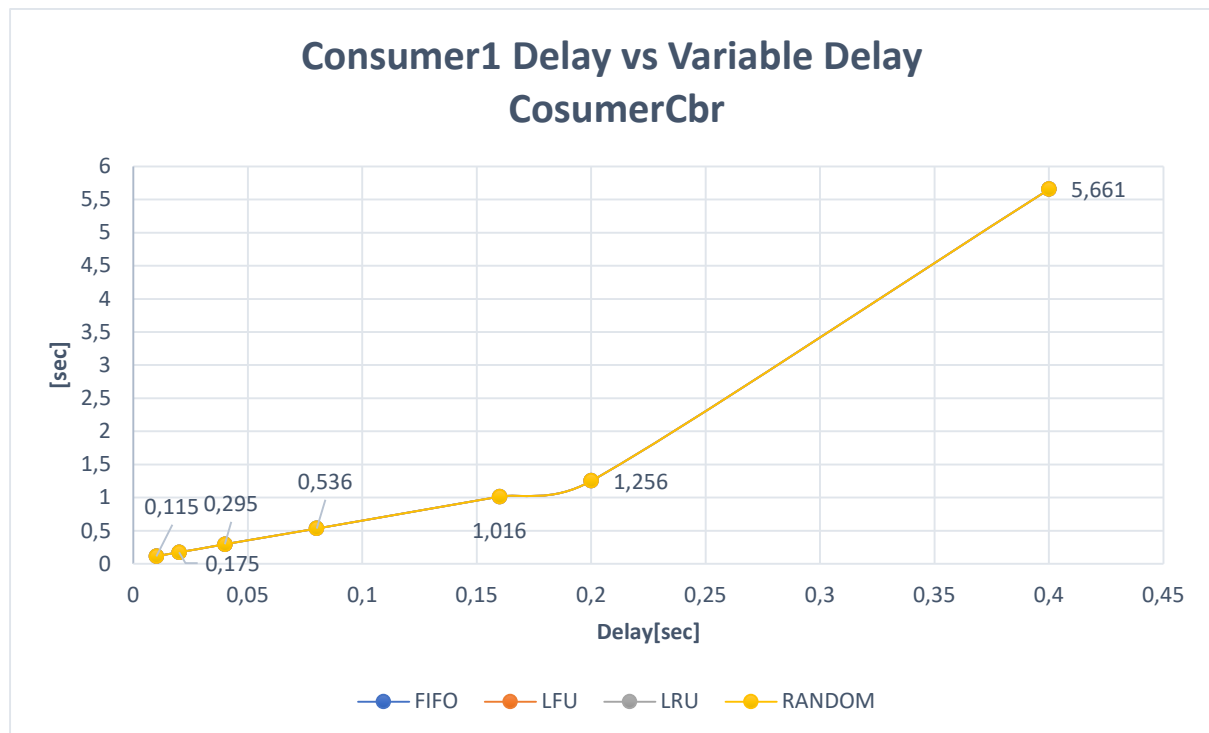
Οι κάτωθι γραφικές αναπαραστάσεις αποτελούν την συγκριτική απεικόνιση μεταξύ των δύο διαφορετικών εφαρμογών παραγωγής πακέτων Ενδιαφέροντος, ConsumerCbr και ConsumerZipfMandelbrot, για μεταβλητή τιμή καθυστέρησης διάδοσης.

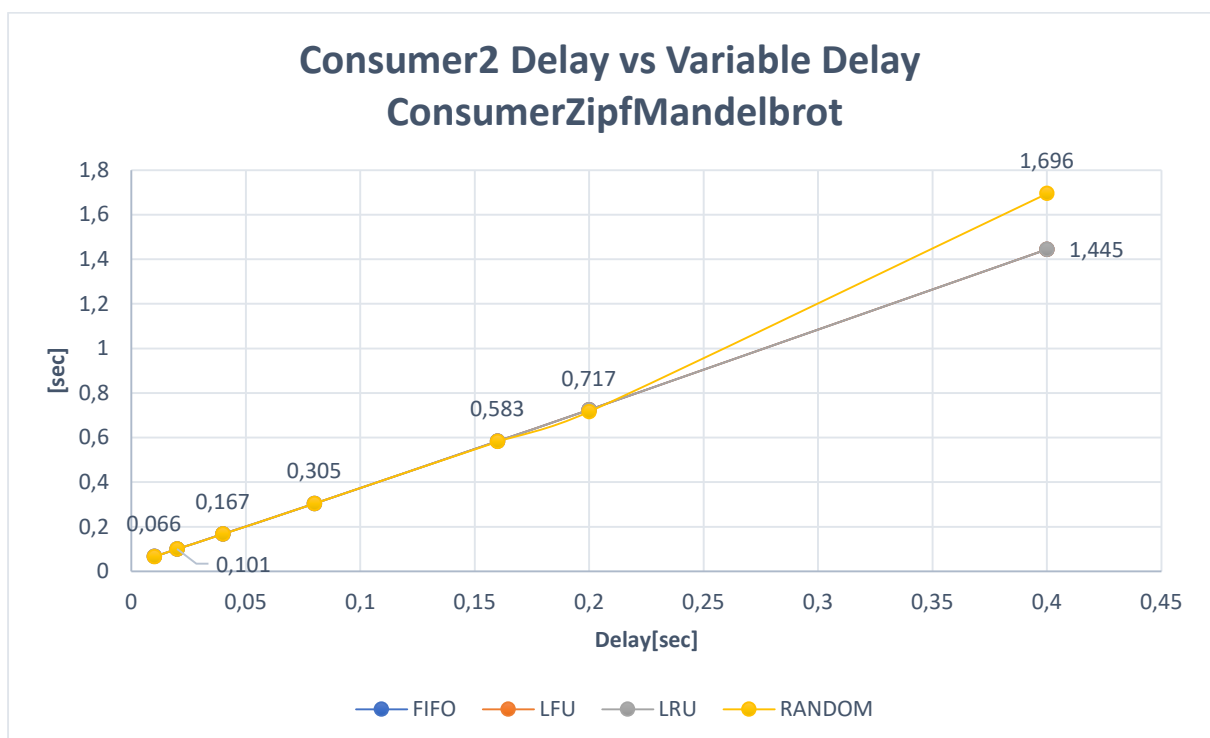
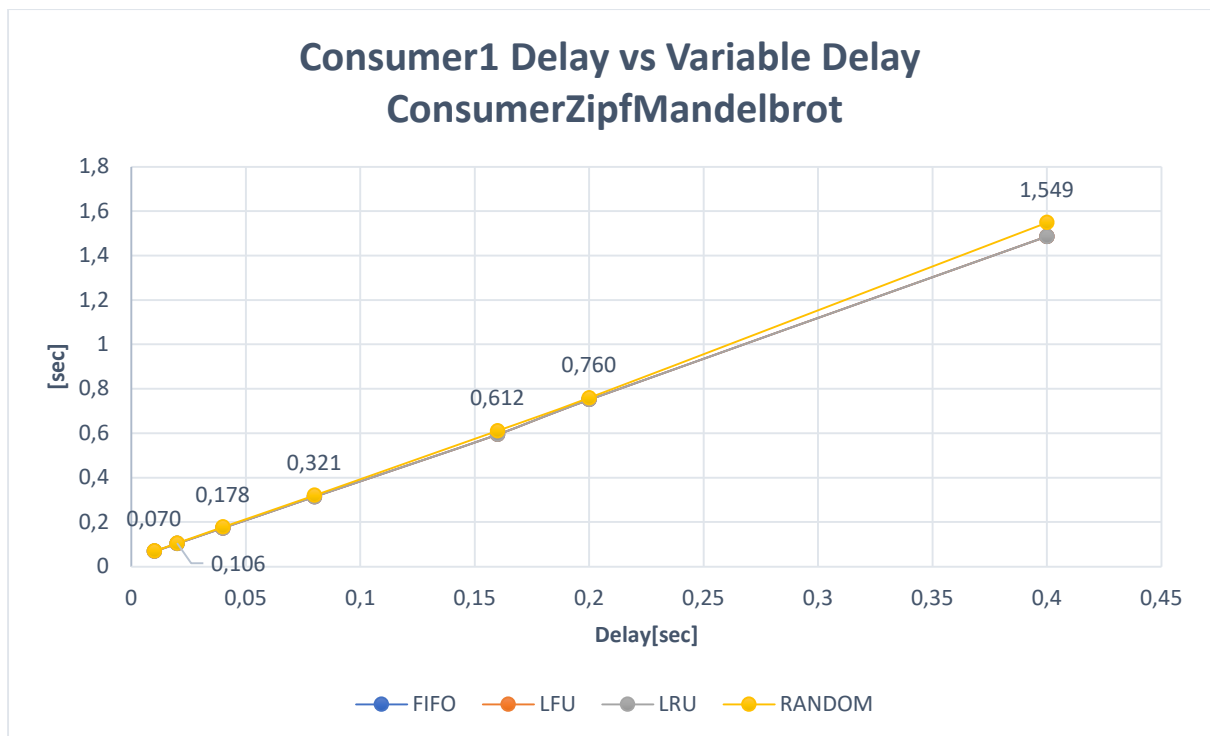


Ως εκ τούτου, παρουσιάζονται τα δεδομένα, τα οποία αποδίδουν επαρκώς την επίδραση της αυξημένης τιμής καθυστέρησης διάδοσης, αφού η τιμή του throughput βαίνει μειούμενη και στις δύο υλοποιήσεις εφαρμογών.

5.1.2.2 Συνολική καθυστέρηση – Delay

Οι κάτωθι γραφικές αναπαραστάσεις αποτελούν συγκριτική απεικόνιση στους κόμβους Καταναλωτή, για μεταβλητή τιμή καθυστέρησης διάδοσης.

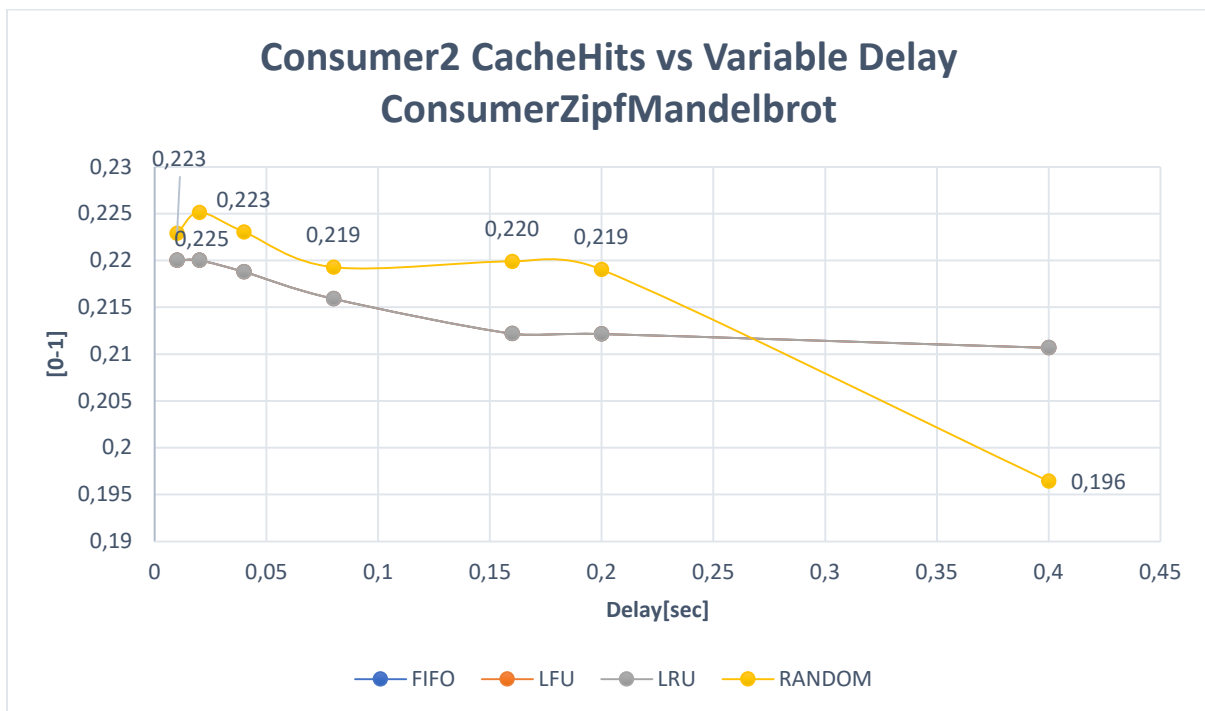
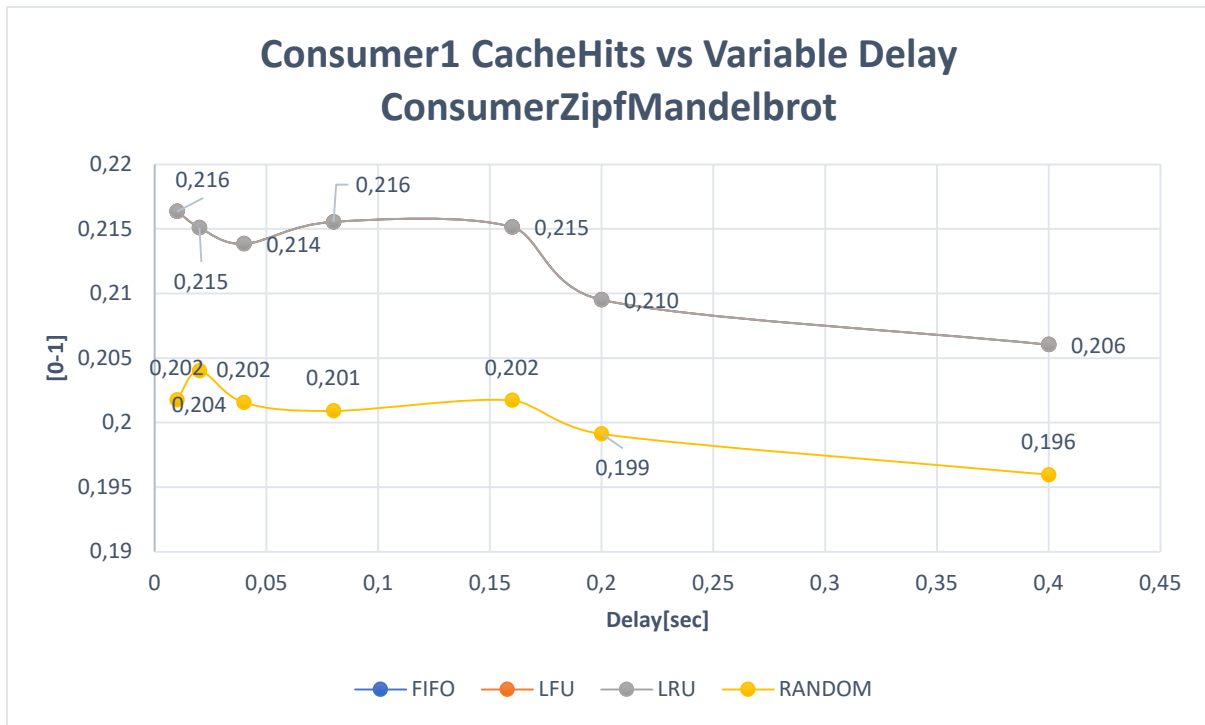


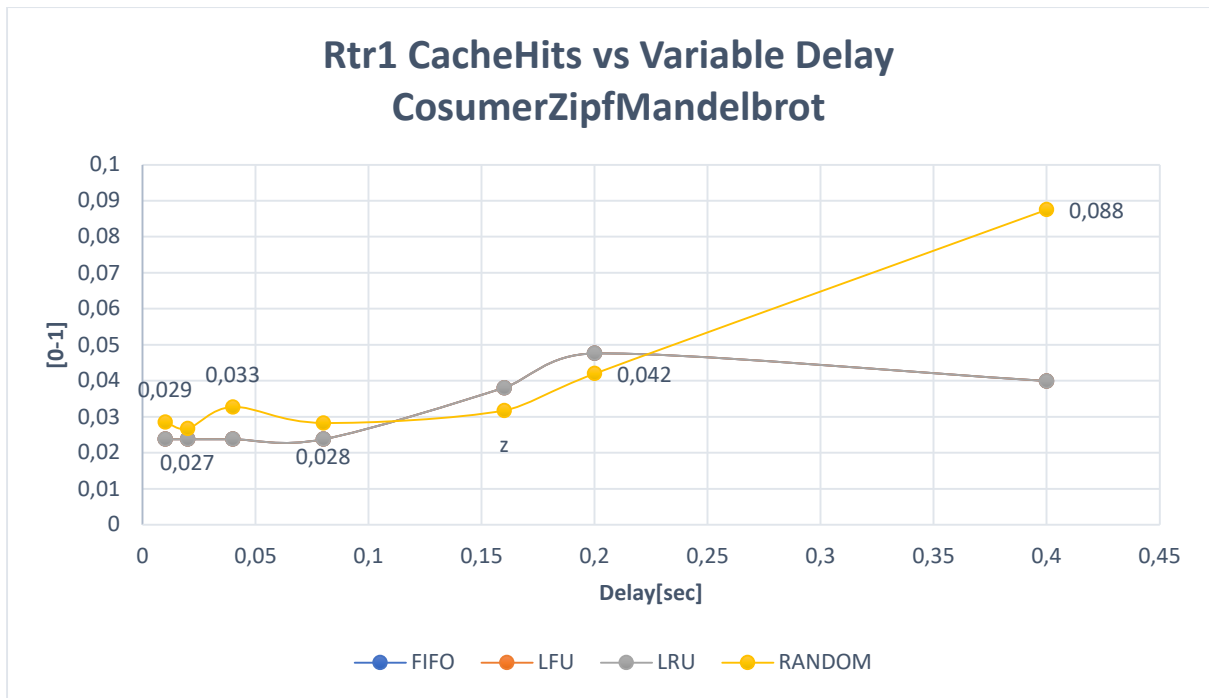


Συμπερασματικά, στην παρούσα υπό-ενότητα απεικονίζεται η μέση τιμή του χρόνου άφιξης του πακέτου στον αρχικό Καταναλωτή, με σημείο βάσης την αρχική αποστολή του πακέτου Ενδιαφέροντος, για μεταβλητή τιμή του χρόνου καθυστέρησης διάδοσης. Η αύξηση της συνολικής καθυστέρησης, αποτελεί αναντίρρητο χαρακτηριστικό, το οποίο επιβεβαιώνεται για κάθε πολιτική επεξεργασίας ουράς, σε εκάστοτε εφαρμογή παραγωγής κίνησης.

5.1.2.3 Επιτυχής ανάκτηση – CacheHits

Οι κάτωθι γραφικές αναπαραστάσεις αποτελούν συγκριτική απεικόνιση της εφαρμογής ConsumerZipf, για μεταβλητή τιμή καθυστέρησης διάδοσης, στους κόμβους Καταναλωτή. Για την εφαρμογή ConsumerCbr, δεν παρουσιάζονται δεδομένα ανάκτησης, αφού αυτά εμφανίζονται μόνο σε ενδεχόμενη κατάρρευση.

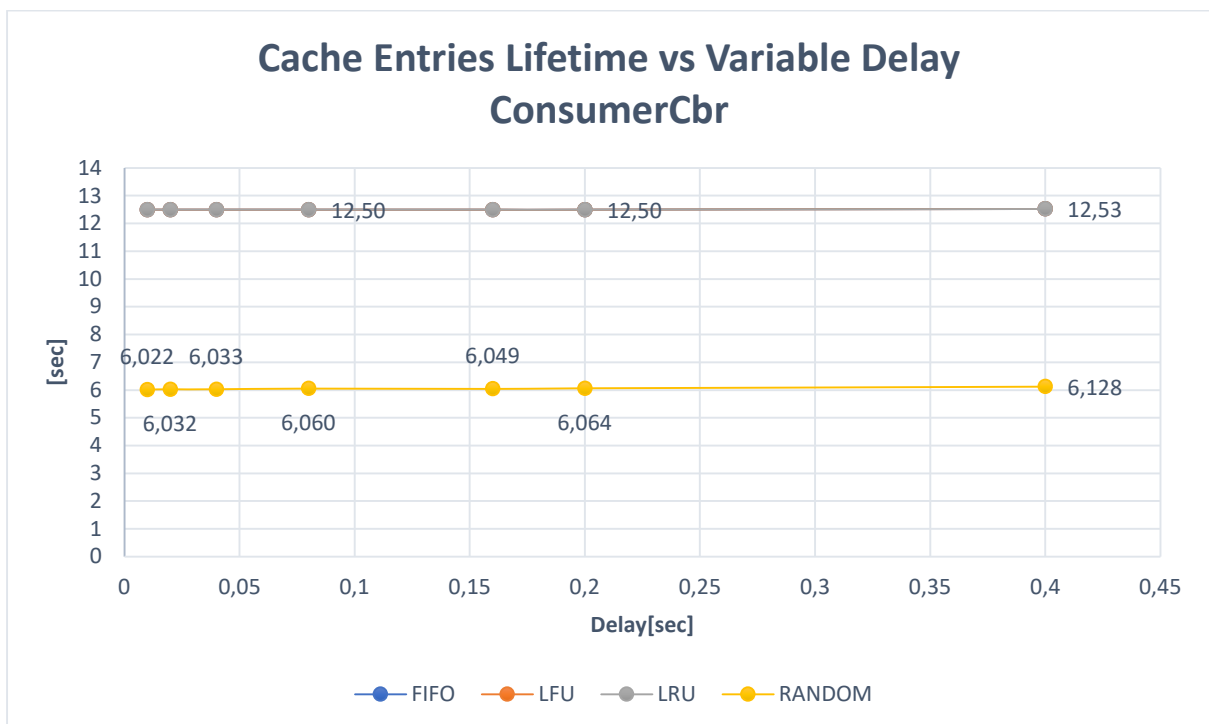


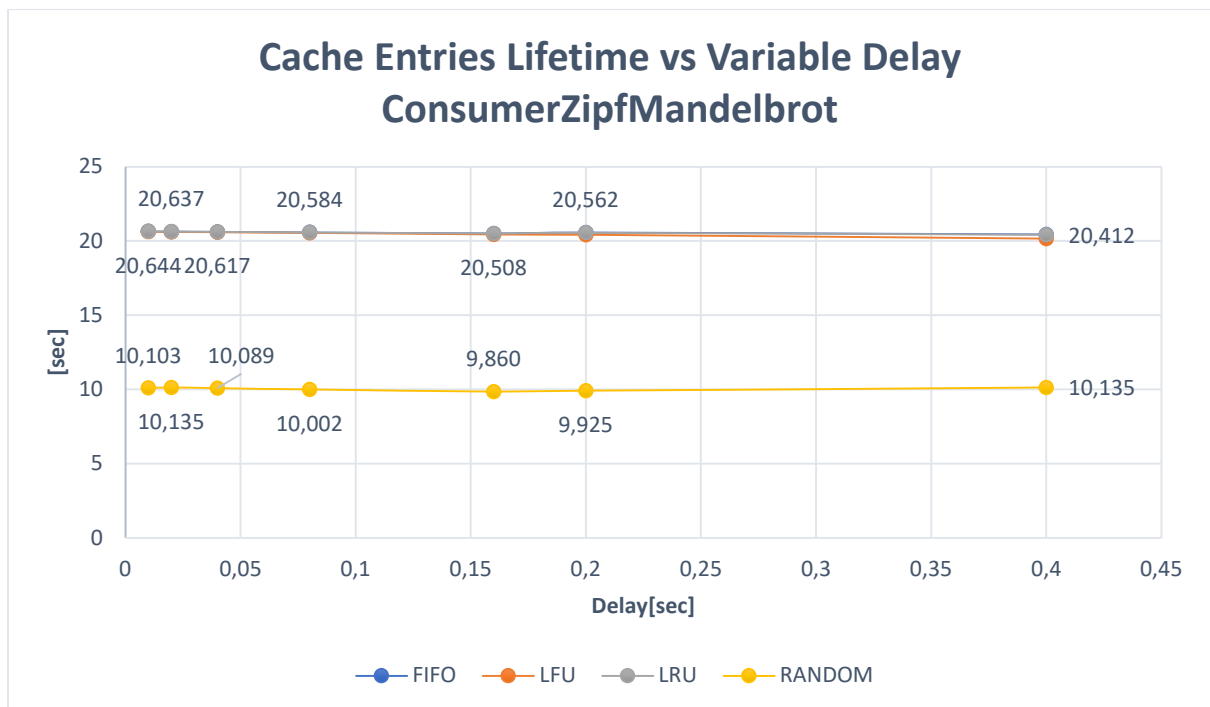


Επομένως, παρουσιάζονται τα συνολικά αποτελέσματα για την εφαρμογή παραγωγής Ενδιαφερόντων ConsumerZipfMandelbrot, για μεταβλητό μέγεθος τιμής της καθυστέρησης διάδοσης. Ως εκ τούτου, τα ποσοστά ανάκτησης για οποιαδήποτε πολιτική επεξεργασίας ουράς παραμένουν σταθερά.

5.1.2.4 Χρόνος Ζωής Καταχωρίσεων - Cache Entries Lifetime

Οι κάτωθι γραφικές αναπαραστάσεις αποτελούν συγκριτική απεικόνιση, για μεταβλητή τιμή καθυστέρησης διάδοσης, στον κόμβο δρομολογητή-**Rtr1**-.





Ως εκ τούτου, παρουσιάζεται ο σταθμισμένος μέσος χρόνος παραμονής του εκάστοτε πακέτου Δεδομένων στον πίνακα Περιεχομένου για τις τέσσερις διαφορετικές πολιτικές επεξεργασίας ουράς, συναρτήσει του μεταβλητού χρόνου καθυστέρησης διάδοσης.

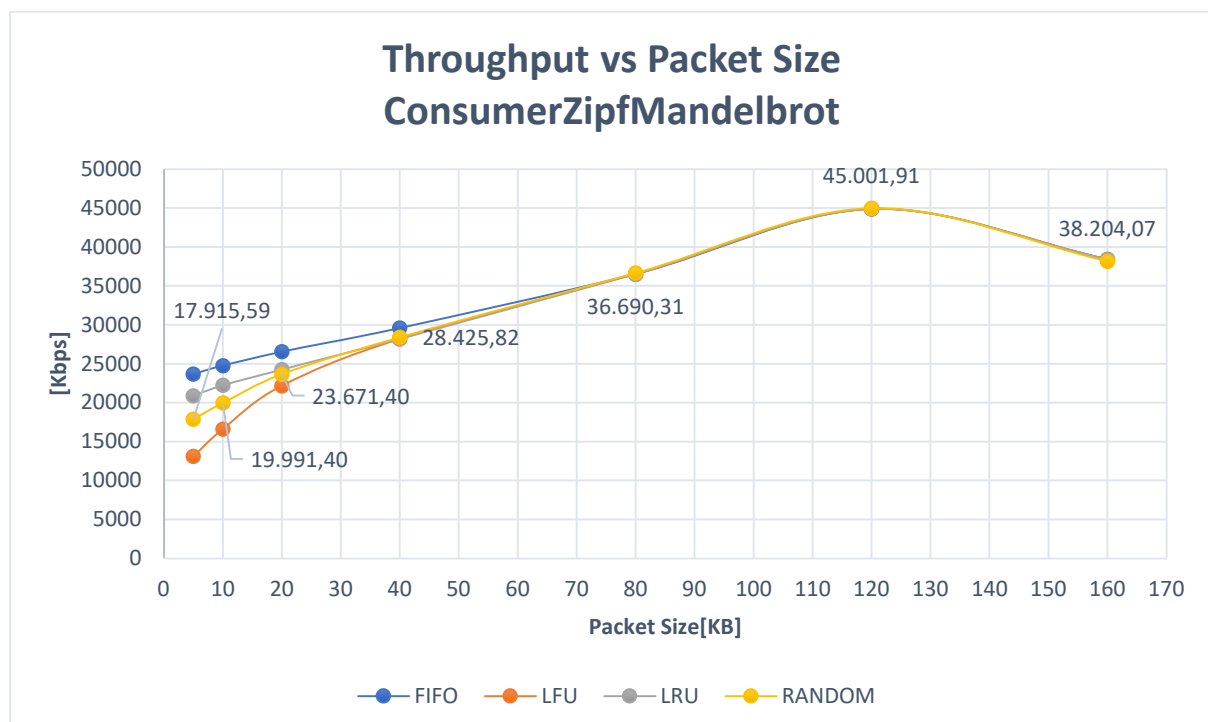
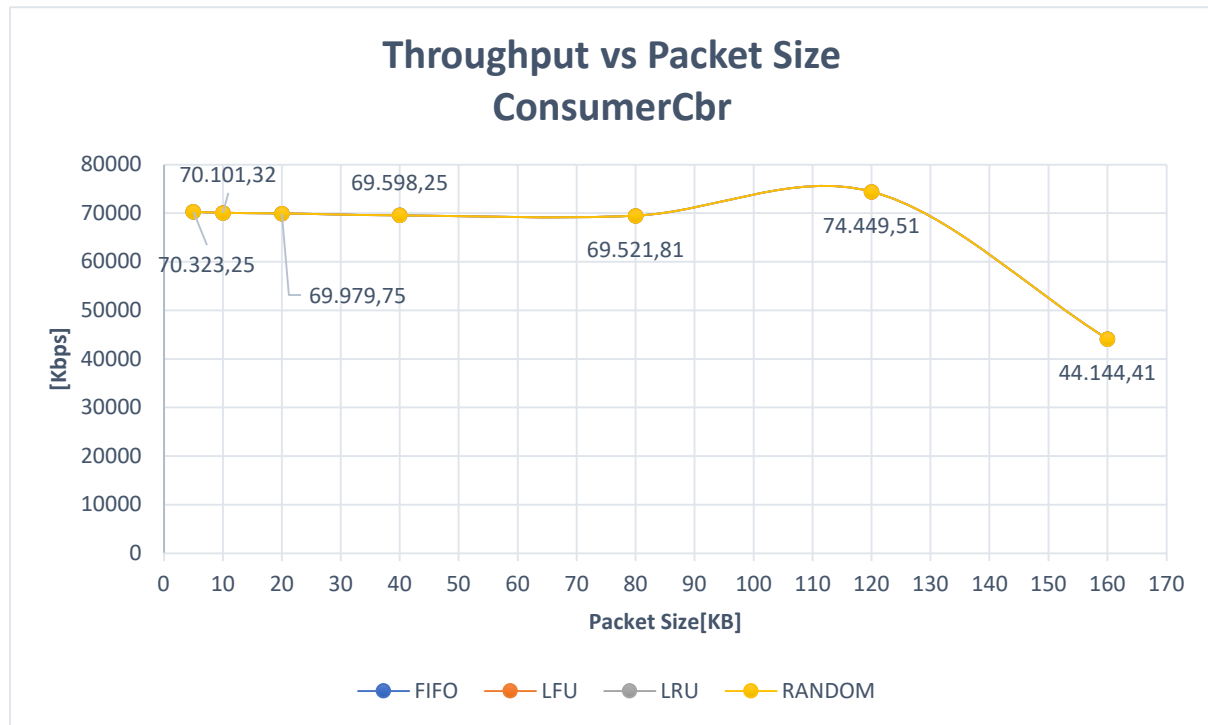
Συμπερασματικά, καθίσταται εμφανής η μειωμένη χρονική τιμή, για τα πακέτα που επεξεργάζονται βάσει της RANDOM πολιτικής.

5.2 Σενάριο 2^ο

Η μεθοδολογία εξέτασης των ανακτηθέντων δεδομένων ακολουθεί επακριβώς την συλλογιστική πορεία, η οποία αναλύθηκε ενδελεχώς στην πρώτη ενότητα. Η διαφοροποίηση έγκειται στο διαθέσιμο backbone Bandwidth, για το οποίο προβαίνουμε σε μείωση, κατά το ήμισυ.

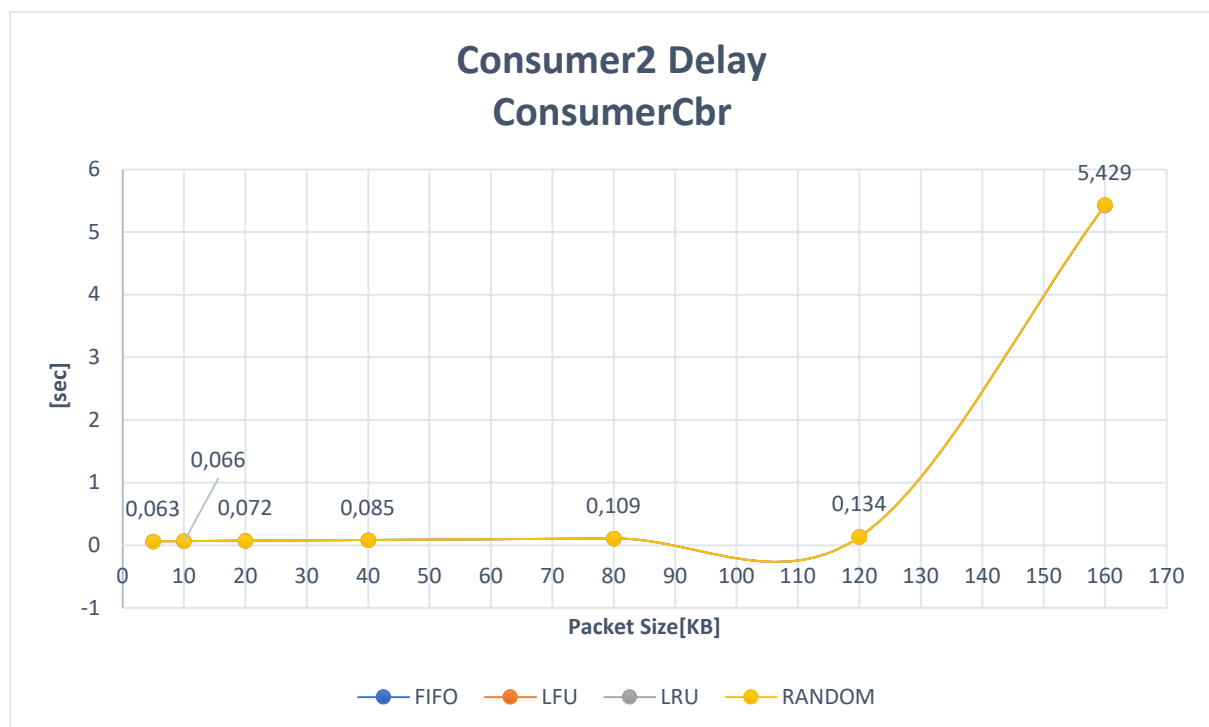
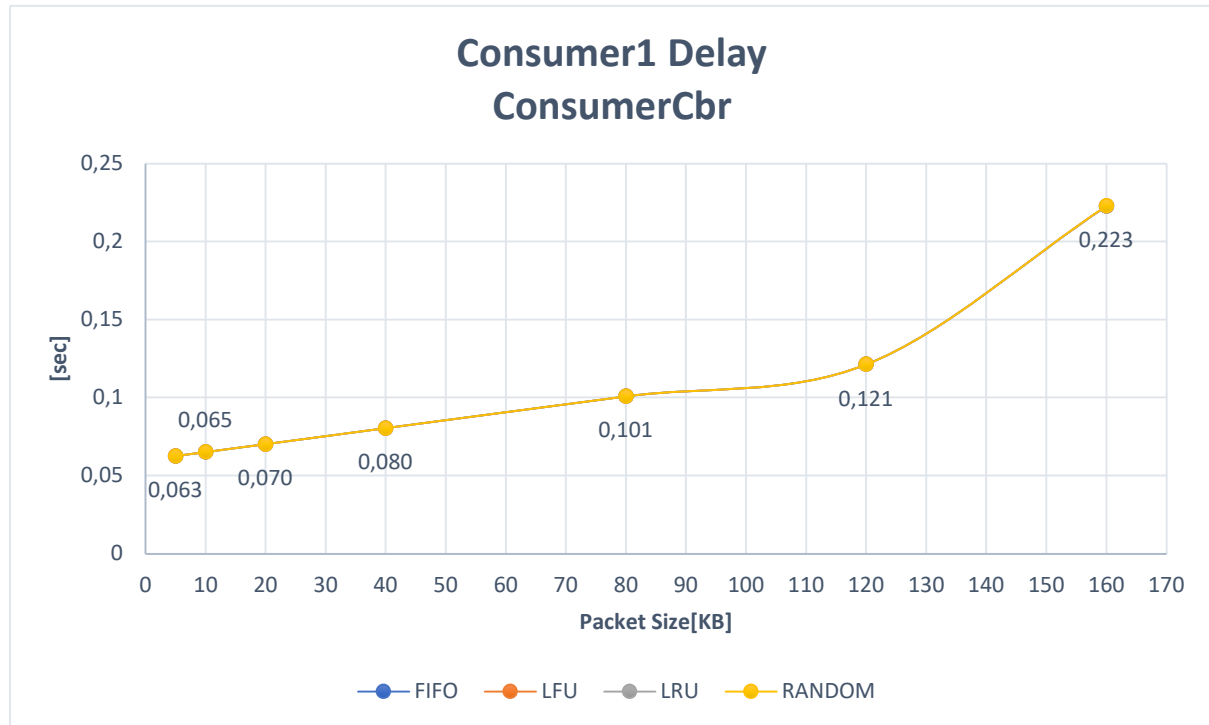
5.2.1 Μεταβλητό μέγεθος πακέτων δεδομένων

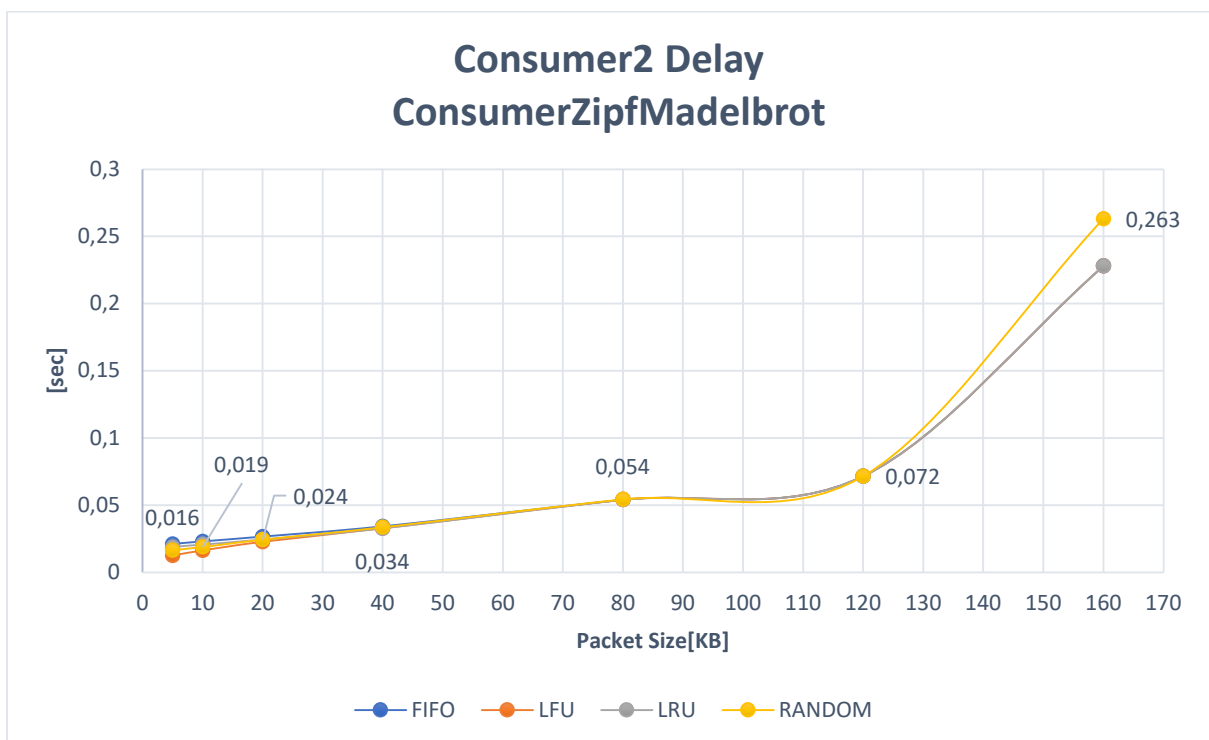
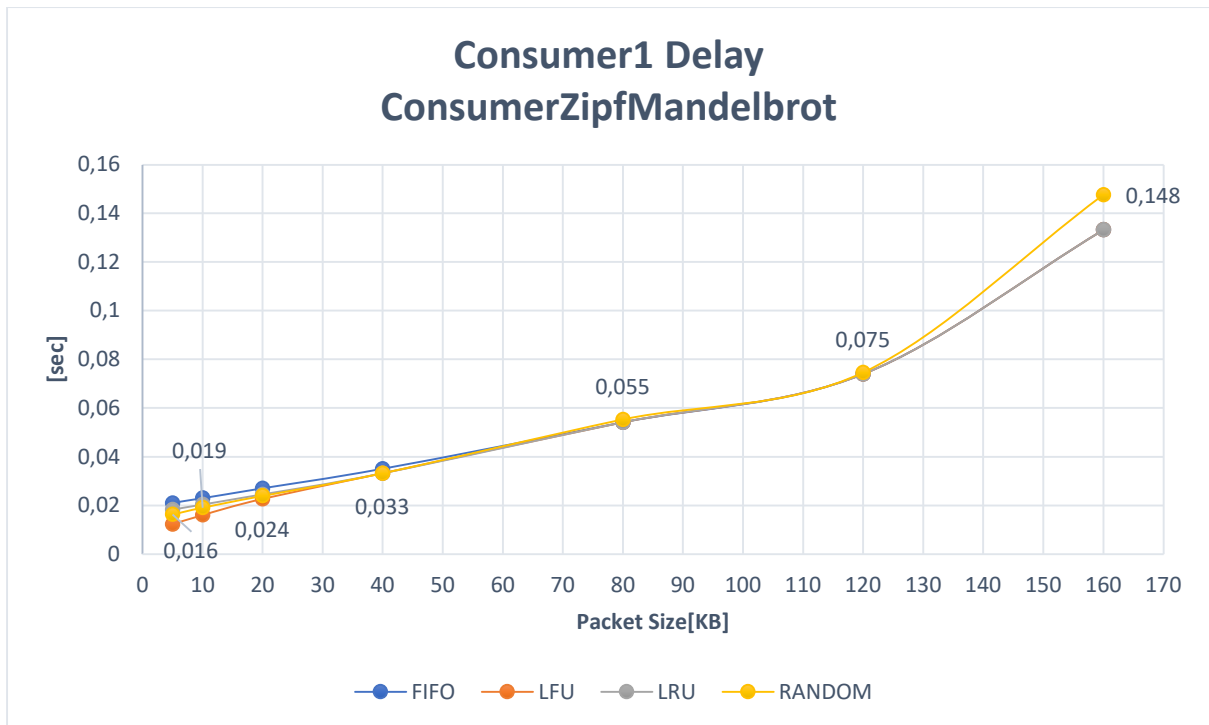
5.2.1.1 Throughput



Η μείωση του συνολικού διαθέσιμου εύρους ζώνης οδήγησε σε ανεπαίσθητη επίδραση της συνολικής αποδοτικότητας. Το μοντέλο εξέλιξης του αξιοποιήσιμου [BW] προσομοιάζει αυτό της αντίστοιχης πρώτης υπό-ενότητας.

5.2.1.2 Συνολική καθυστέρηση - Delay

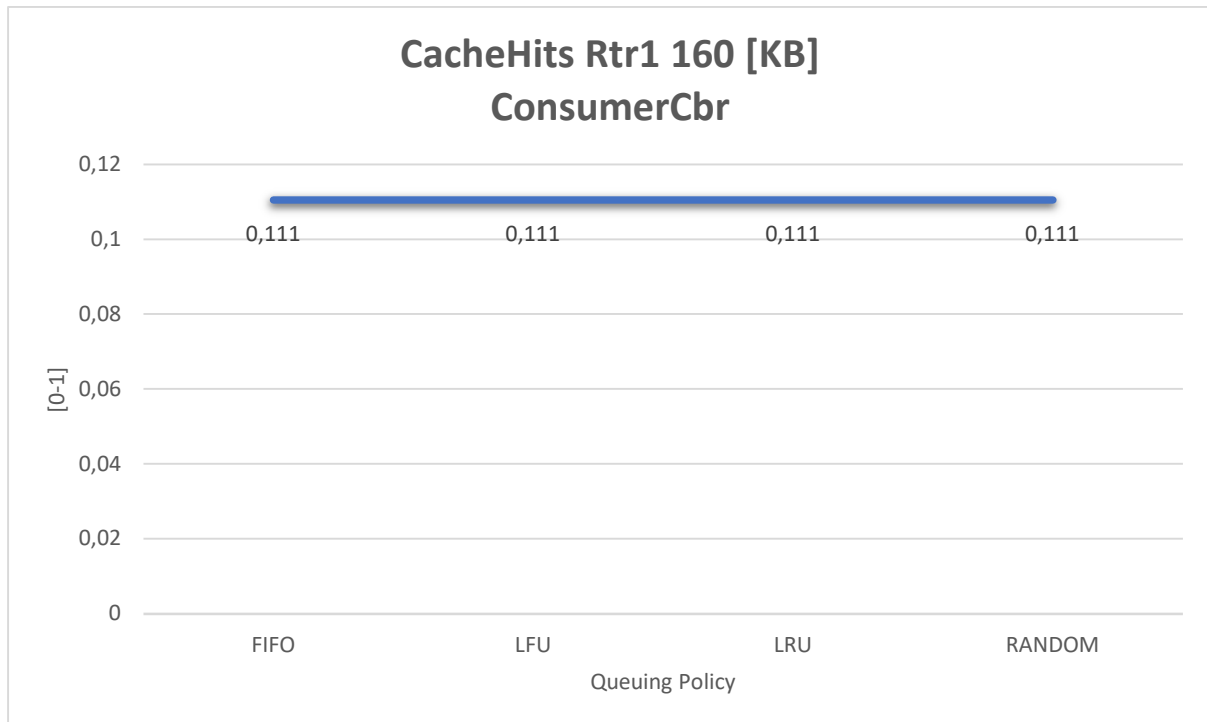




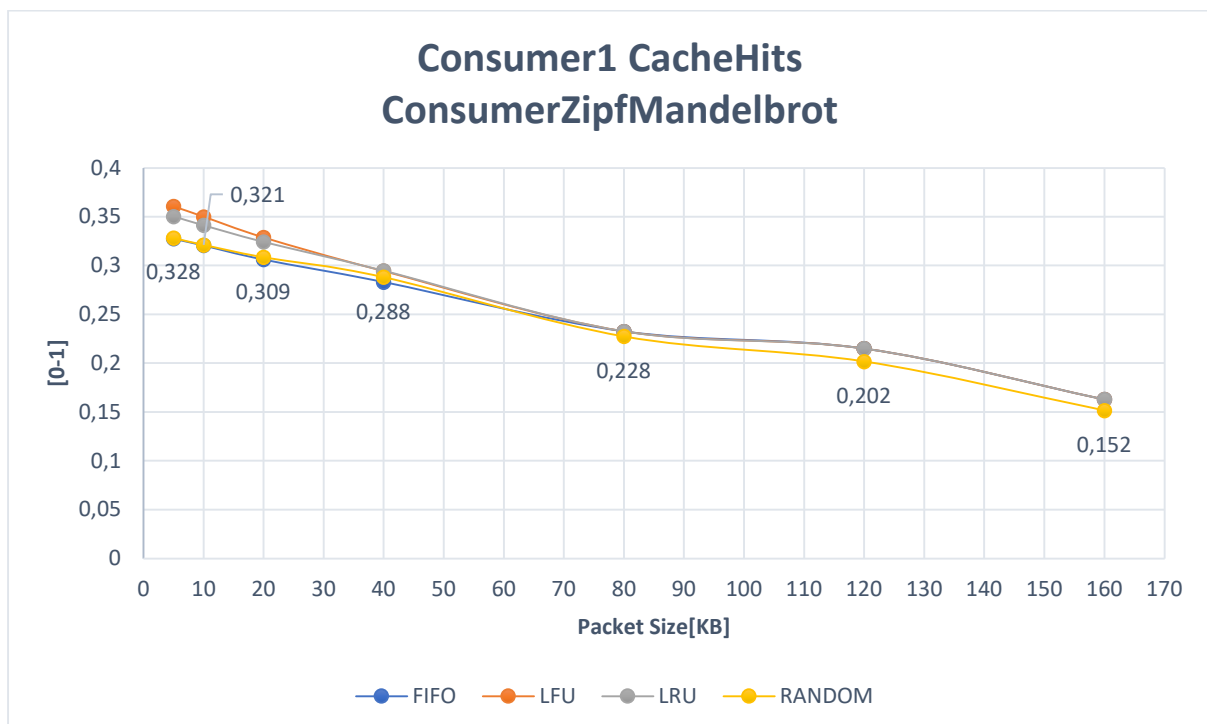
Τα αποτελέσματα παρουσιάζουν μικρή αυξητική τάση συγκριτικά με τα αντίστοιχα της πρώτης υπό-ενότητας. Καθίσταται λοιπόν σαφές, ότι η συνολική καθυστέρηση διάδοσης είναι ανεξάρτητη της πολιτικής επεξεργασίας ουράς, ενώ παράλληλα εξαρτάται πλήρως από το μεταβλητό και αυξανόμενο μέγεθος πακέτου Δεδομένων.

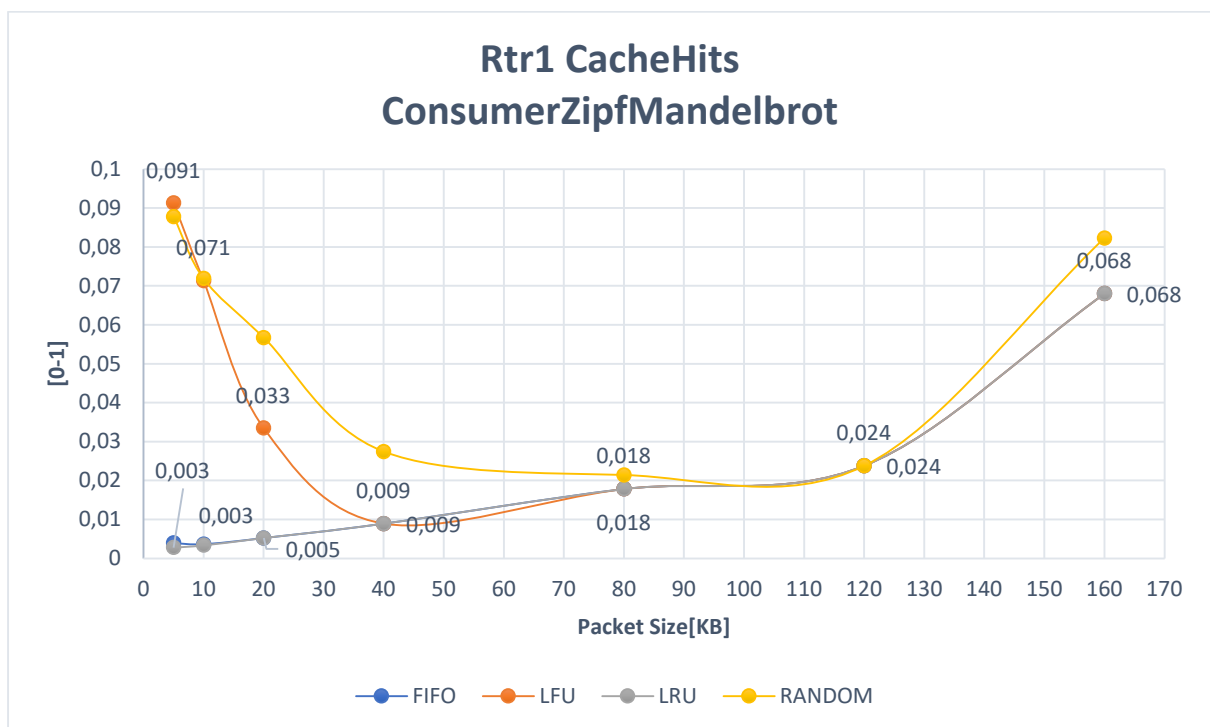
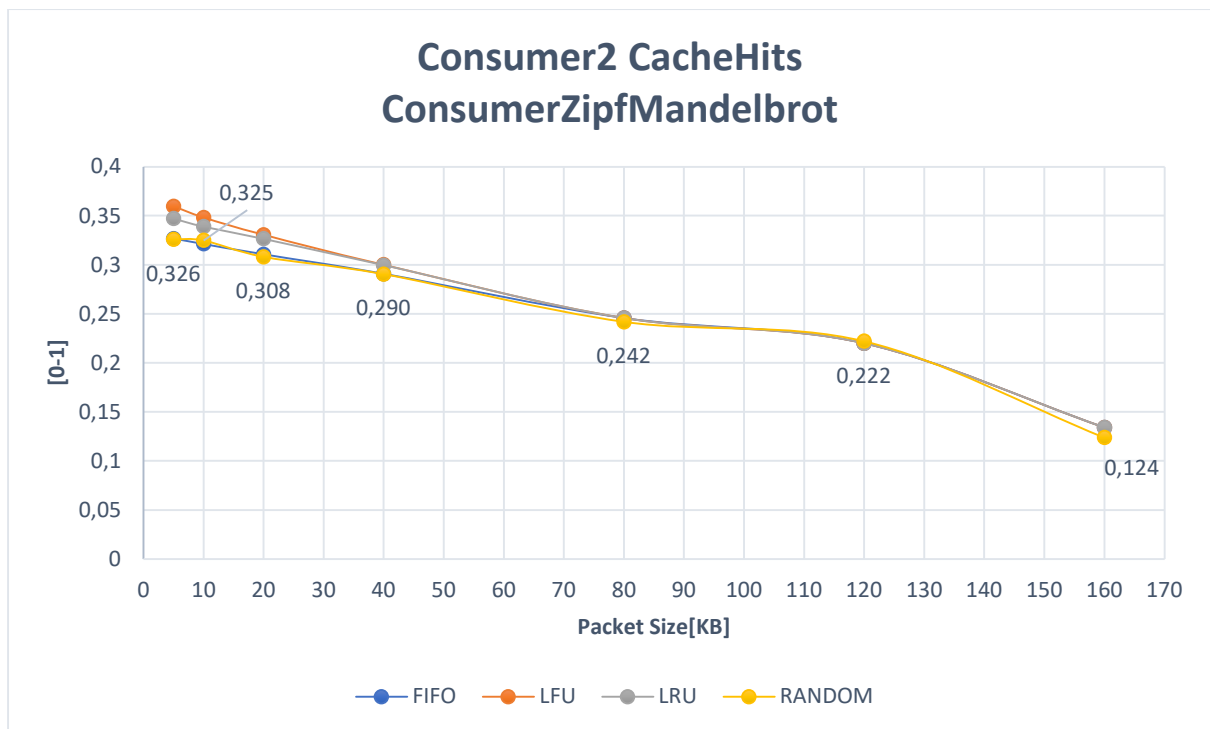
5.2.1.3 Επιτυχής Ανάκτηση – CacheHits

Το επόμενο γράφημα αντιπροσωπεύει τα ποσοστά ανάκτησης κατά την υλοποίηση της εφαρμογής ConsumerCbr.



Επιπλέον, ακολουθούν οι γραφικές αναπαραστάσεις των CacheHits για την εφαρμογή ConsumerZipfMandelbrot.

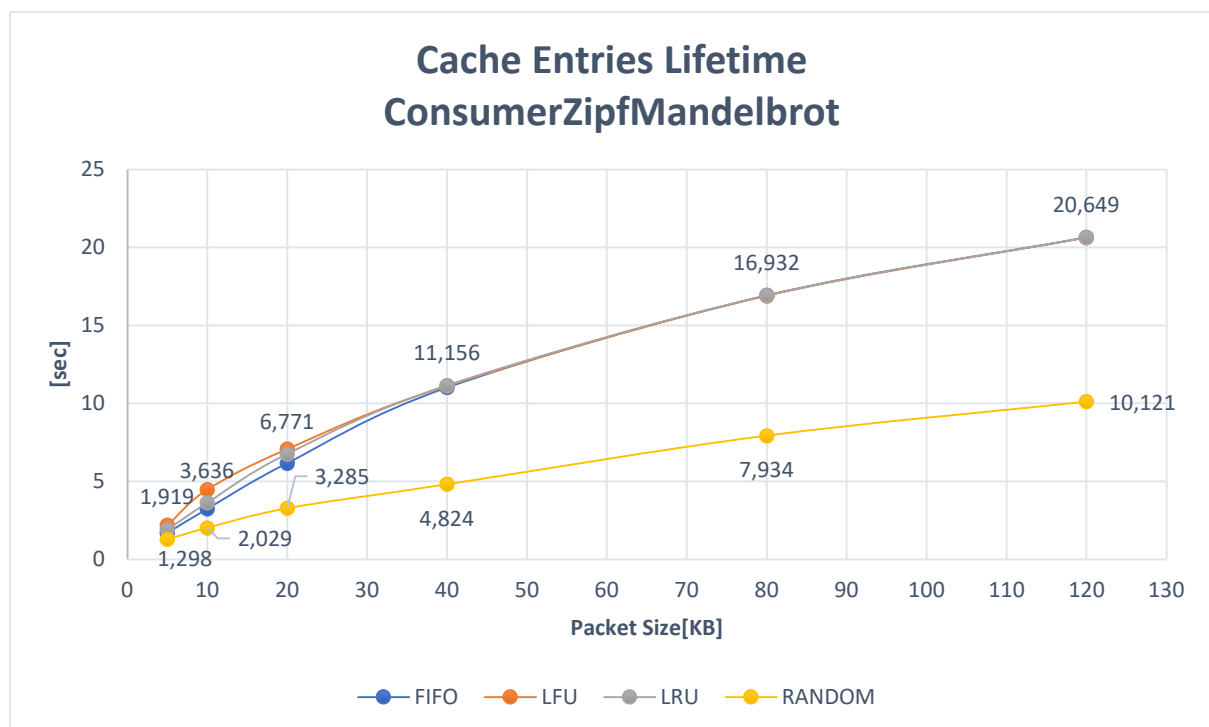
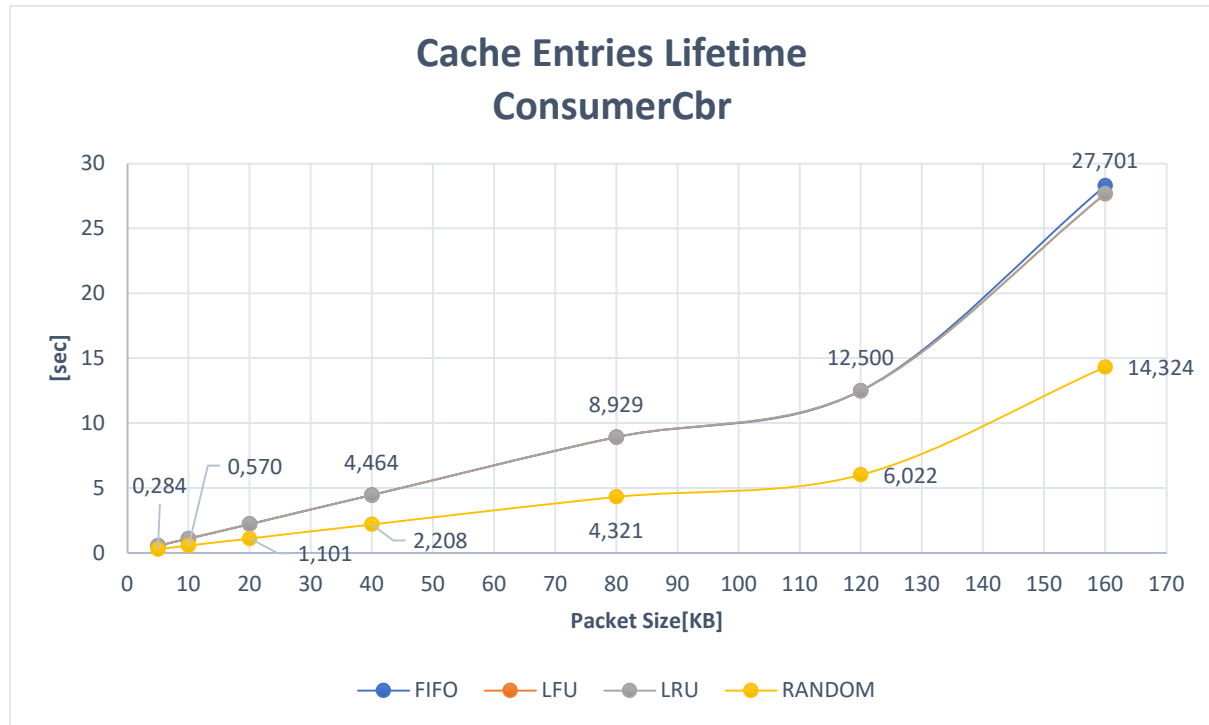




Τα αποτελέσματα παρουσιάζουν ομοιογένεια με τα κατ' αντιπαράβολή συγκριτικά της πρώτης υπό-ενότητας. Η μεταβολή του backbone εύρους ζώνης [BW], οδήγησε σε ανεπαισθητή μείωση των ποσοστών ανάκτησης. Ως εκ τούτου, η τοπολογία λειτούργησε επαρκώς σε οριακές συνθήκες.

5.2.1.4 Χρόνος ζωής καταχωρίσεων - Entries Life Time

Στα κάτωθι γραφήματα παρουσιάζεται η συγκριτική απεικόνιση των δύο διαφορετικών εφαρμογών παραγωγής πακέτων Ενδιαφέροντος, για μεταβλητό μέγεθος πακέτων Δεδομένων, στον κόμβο δρομολογητή-**Rtr1**.

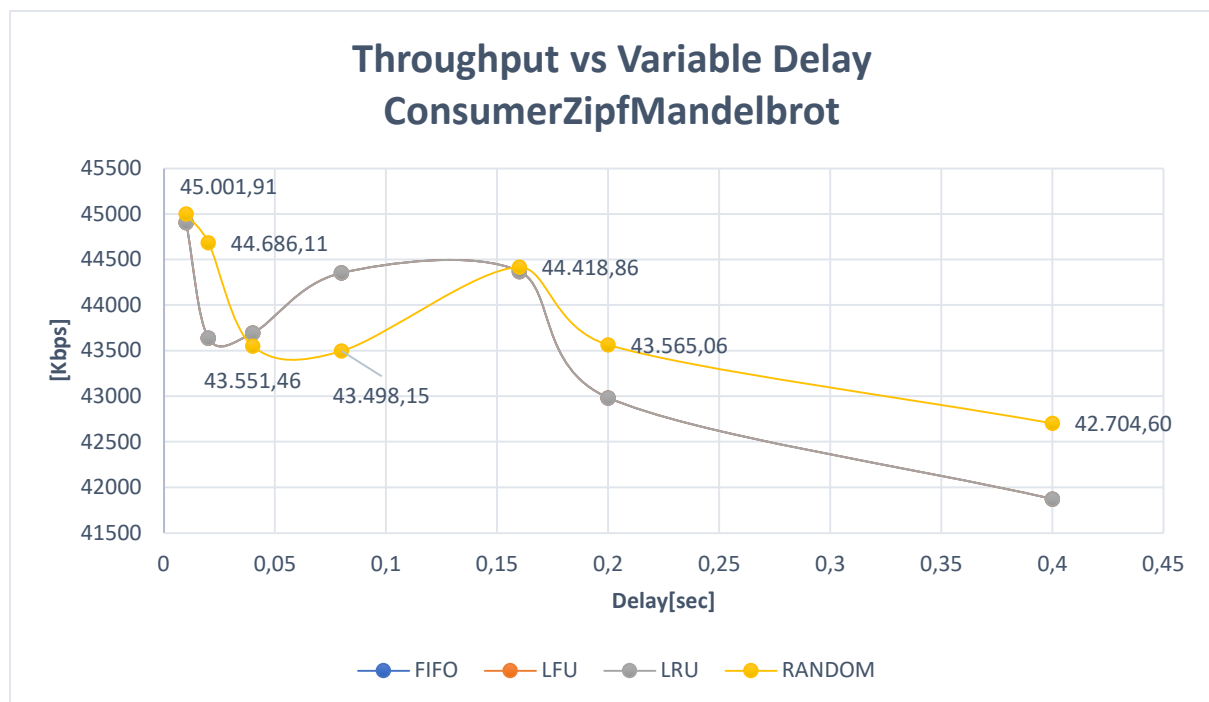
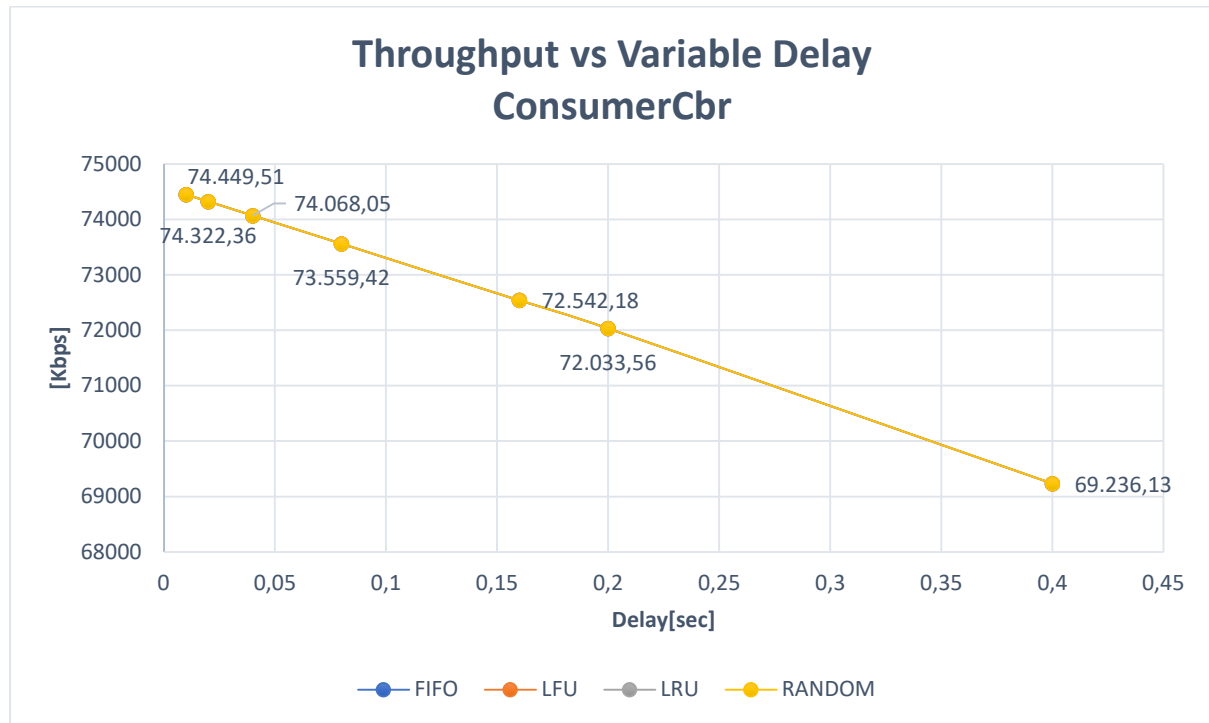


Η πολιτική διαχείρισης ουράς RANDOM καθίσταται αποδοτικότερη, επιβεβαιώνοντας τα δεδομένα της πρώτης ενότητας.

5.2.2 Μεταβλητό μέγεθος καθυστέρησης διάδοσης

5.2.2.1 Throughput

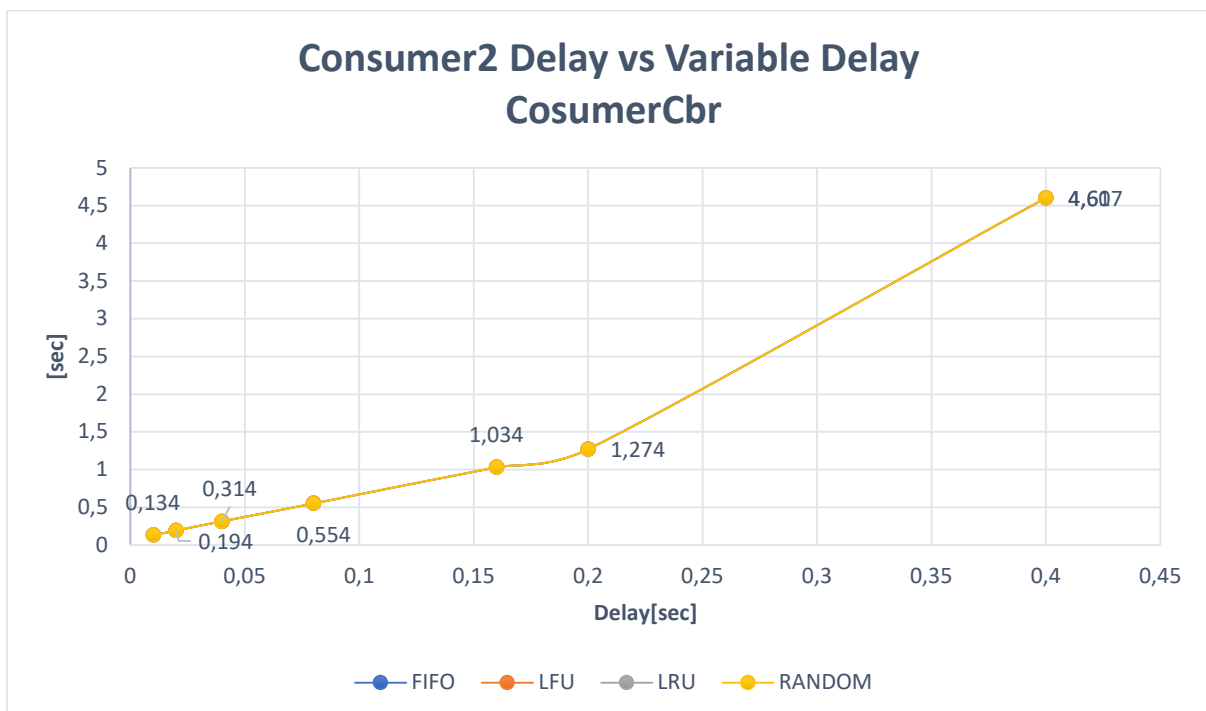
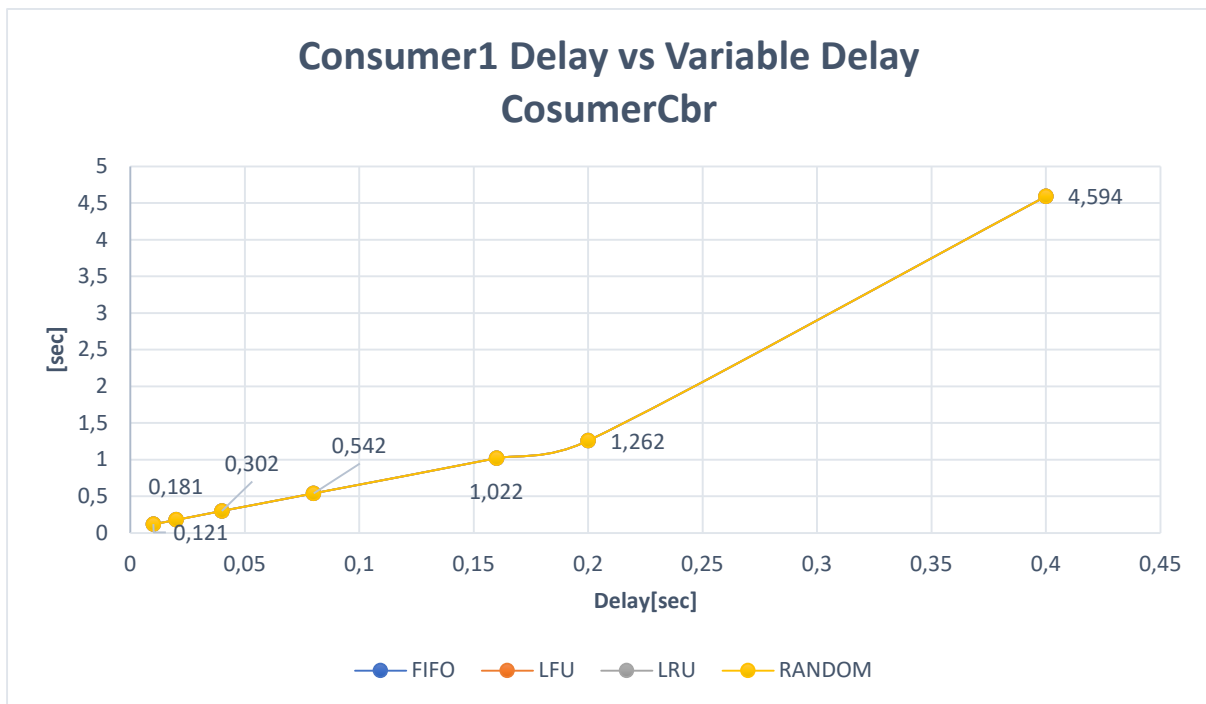
Στις κάτωθι γραφικές αναπαραστάσεις παρουσιάζεται η συνολική επίδραση της μεταβλητής τιμής καθυστέρησης διάδοσης, για το συνολικό Throughput.



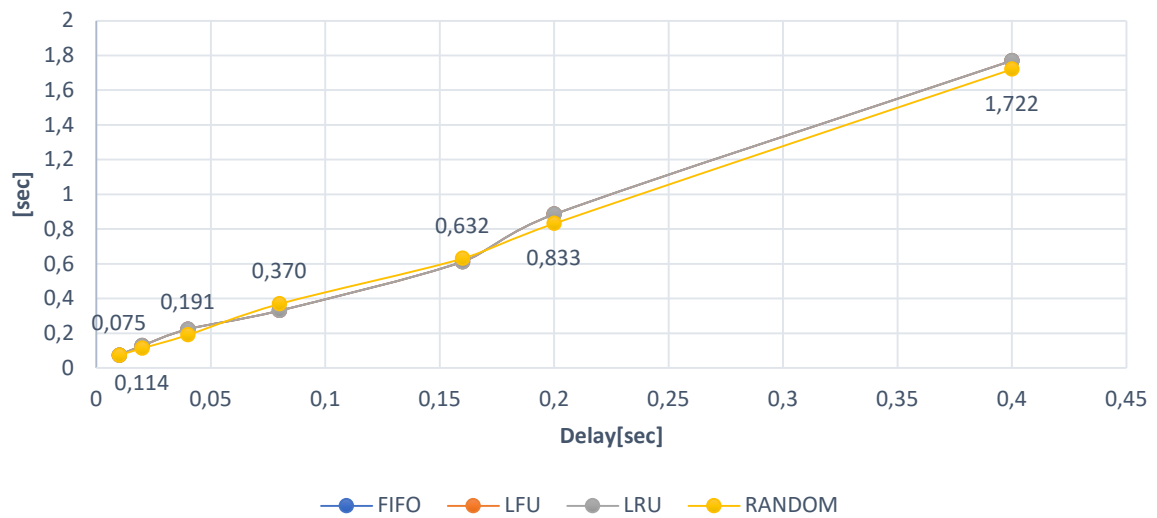
Η συνεχής αύξηση της καθυστέρησης διάδοσης, οδηγεί σε αναλογική μείωση του αξιοποιήσιμου εύρους ζώνης.

5.2.2.2 Συνολική καθυστέρηση – Delay

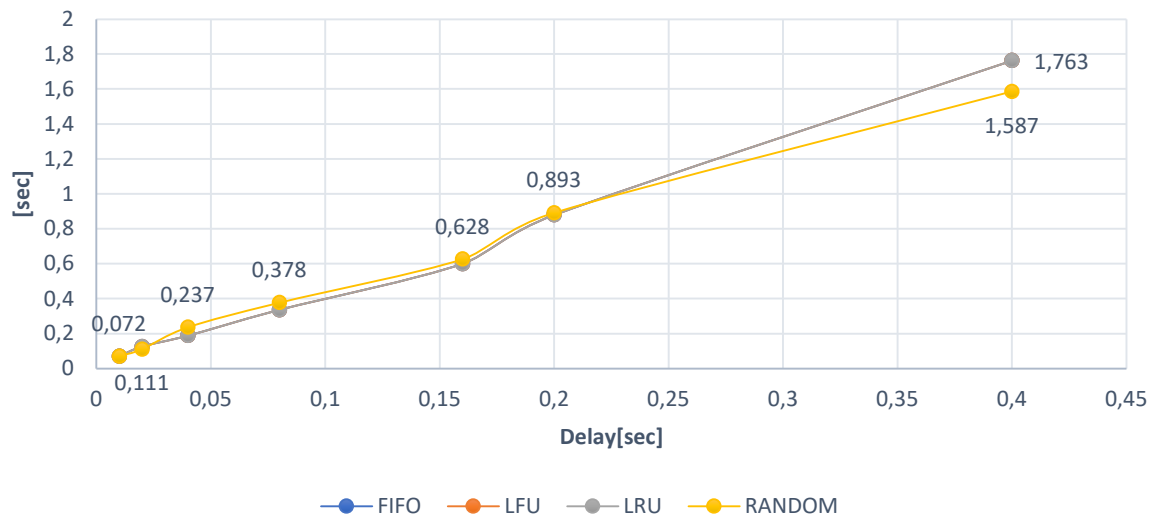
Οι κάτωθι γραφικές αναπαραστάσεις αποτελούν συγκριτική απεικόνιση στους κόμβους Καταναλωτή, για μεταβλητή τιμή καθυστέρησης διάδοσης. Παρατηρείται μικρή αύξηση, σε σχέση με τις μετρικές της αντίστοιχης πρώτης υπό-ενότητας. Η πολιτική διαχείρισης ουράς, δε διαδραματίζει καθοριστικό ρόλο.



Consumer1 Delay vs Variable Delay ConsumerZipfMandelbrot

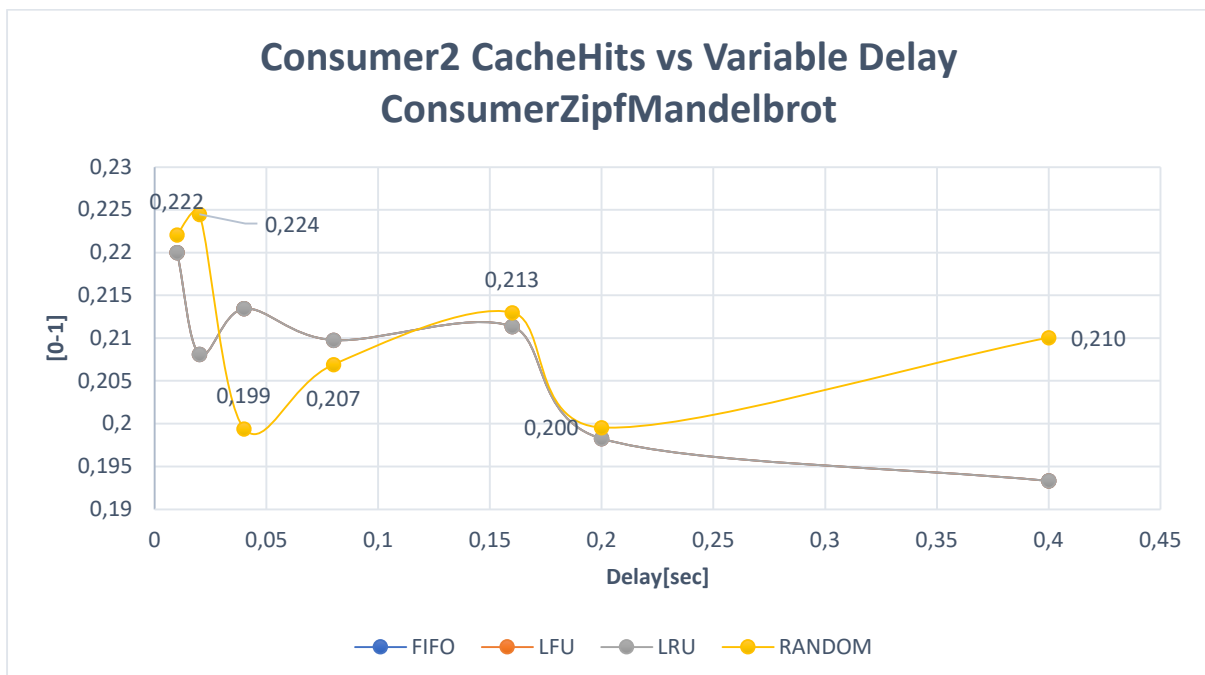
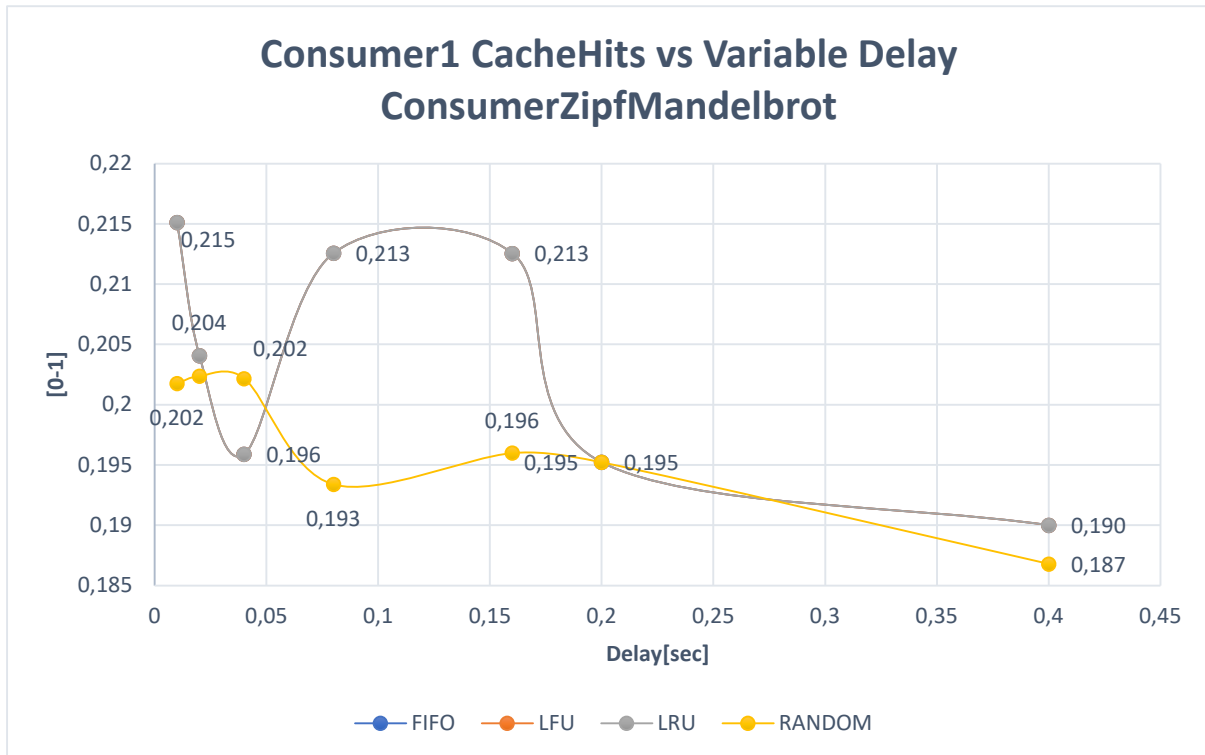


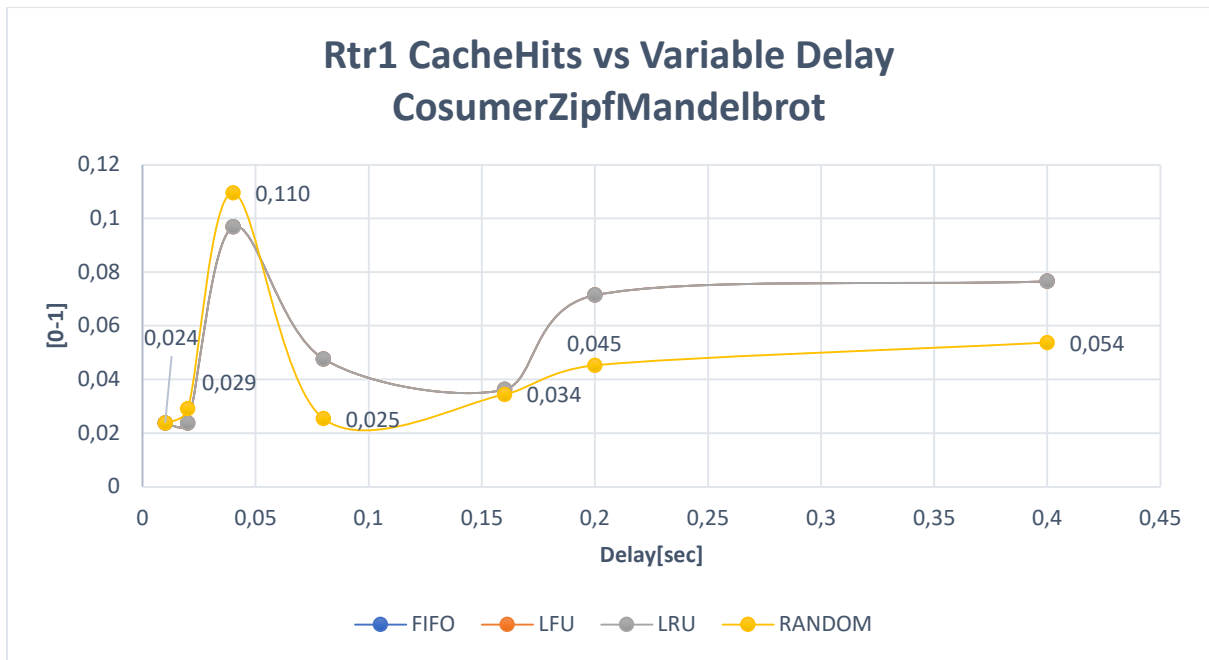
Consumer2 Delay vs Variable Delay ConsumerZipfMandelbrot



5.2.2.3 Επιτυχής ανάκτηση – CacheHits

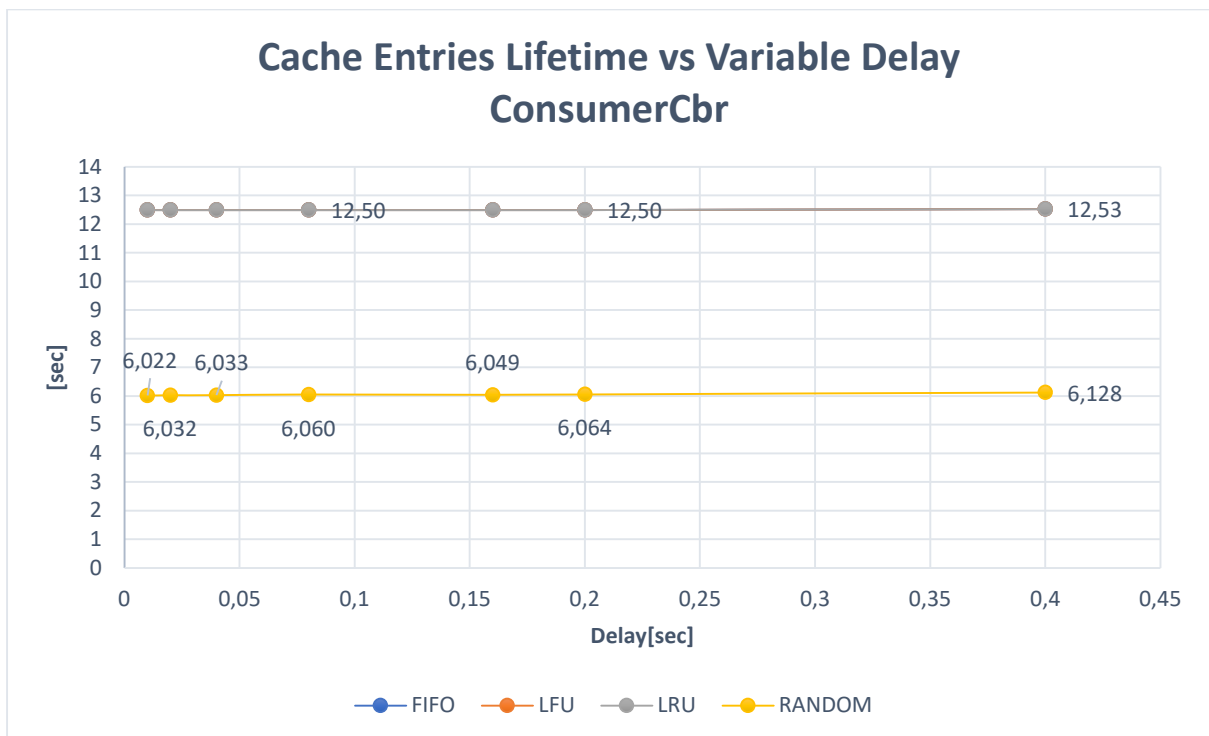
Για την εφαρμογή ConsumerCbr, δεν παρουσιάζονται δεδομένα ανάκτησης, αφού αυτά εμφανίζονται μόνο σε ενδεχόμενη κατάρρευση. Οι μετρικές επιτυχούς ανάκτησης πακέτων Δεδομένων από τον πίνακα Content Store διαφοροποιούνται σε μικρό βαθμό, σε σχέση με αυτές της πρώτης υπό-ενότητας. Τα ποσοστά ανάκτησης διατηρούνται σχεδόν σταθερά, μόνο για τη διαχείριση πολιτικής ουράς RANDOM.

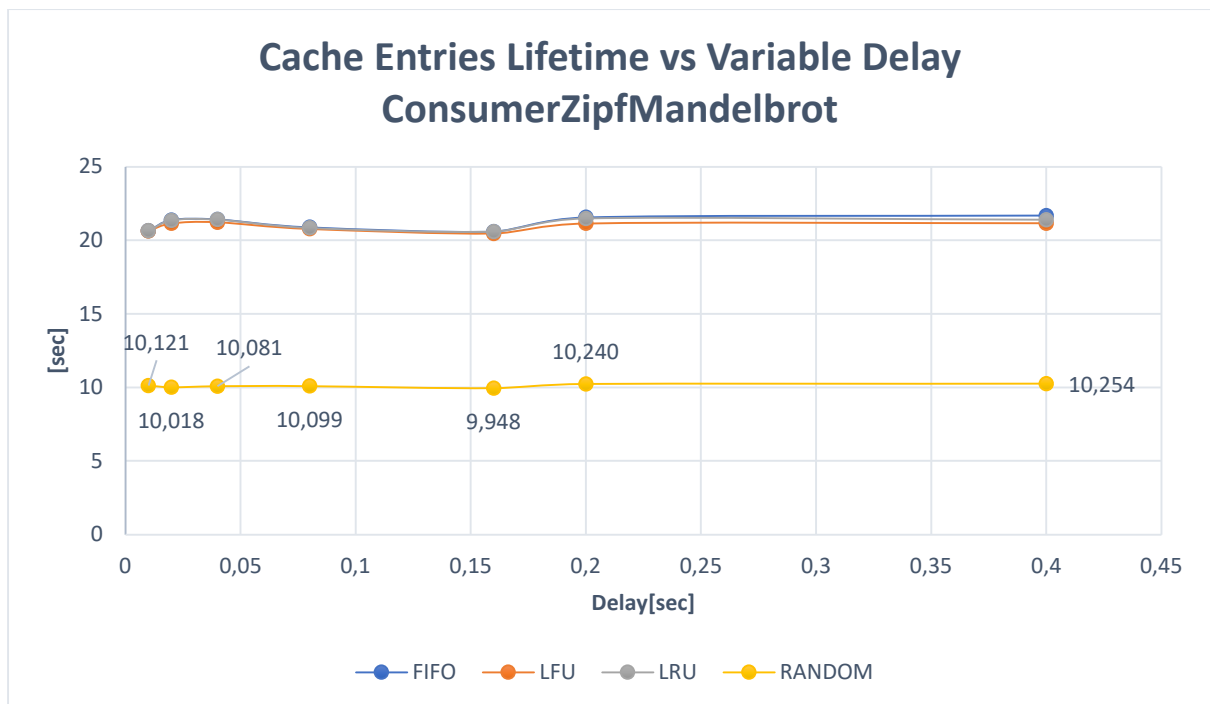




5.2.2.4 Χρόνος ζωής καταχωρίσεων - Entries Life Time

Οι κάτωθι γραφικές αναπαραστάσεις αποτελούν συγκριτική απεικόνιση μεταξύ των δύο διαφορετικών εφαρμογών παραγωγής πακέτων Ενδιαφέροντος, για μεταβλητή τιμή καθυστέρησης διάδοσης, στον κόμβο δρομολογητή-**Rtr1**-.





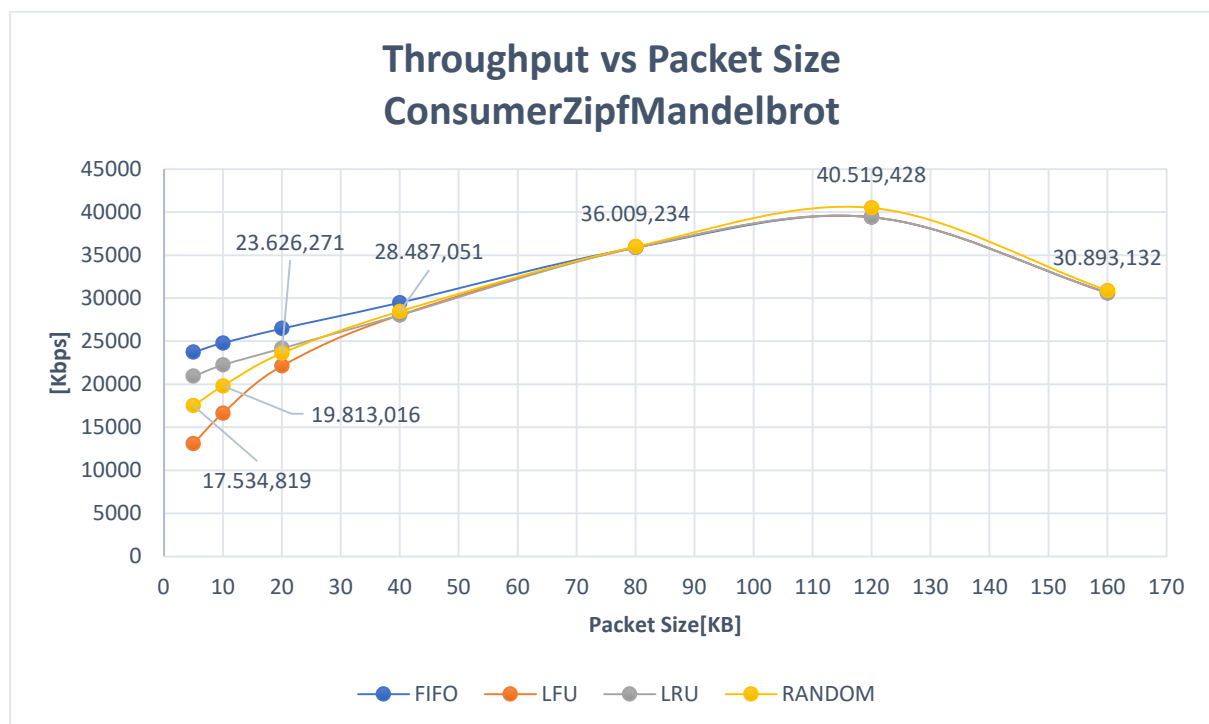
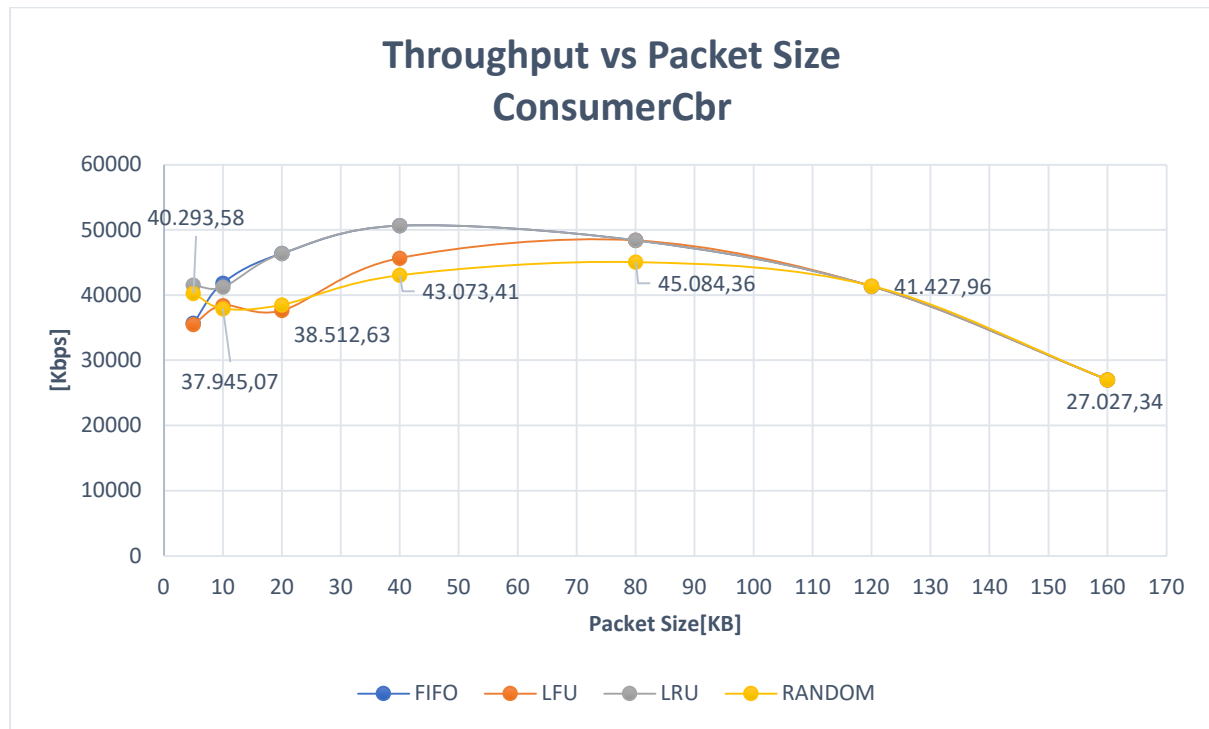
Κατ' αντιπαράβολή, οι παρούσες μετρικές δεν διαφοροποιούνται σε μεγάλο βαθμό σε σύγκριση με τις αντίστοιχες της πρώτης υπό-ενότητας. Η πολιτική διαχείρισης ουράς RANDOM επιφέρει βέλτιστα αποτελέσματα, συγκριτικά με τις υπόλοιπες, ανεξαρτήτως της καθυστέρησης διάδοσης.

5.3 Σενάριο 3^ο

Η διαφοροποίηση της παρούσας ενότητας έγκειται στο διαθέσιμο backbone εύρος ζώνης(BW), για το οποίο προβαίνουμε σε περαιτέρω μείωση, δημιουργώντας συνθήκες έντονης συμφόρησης.

5.3.1 Μεταβλητό μέγεθος πακέτων δεδομένων

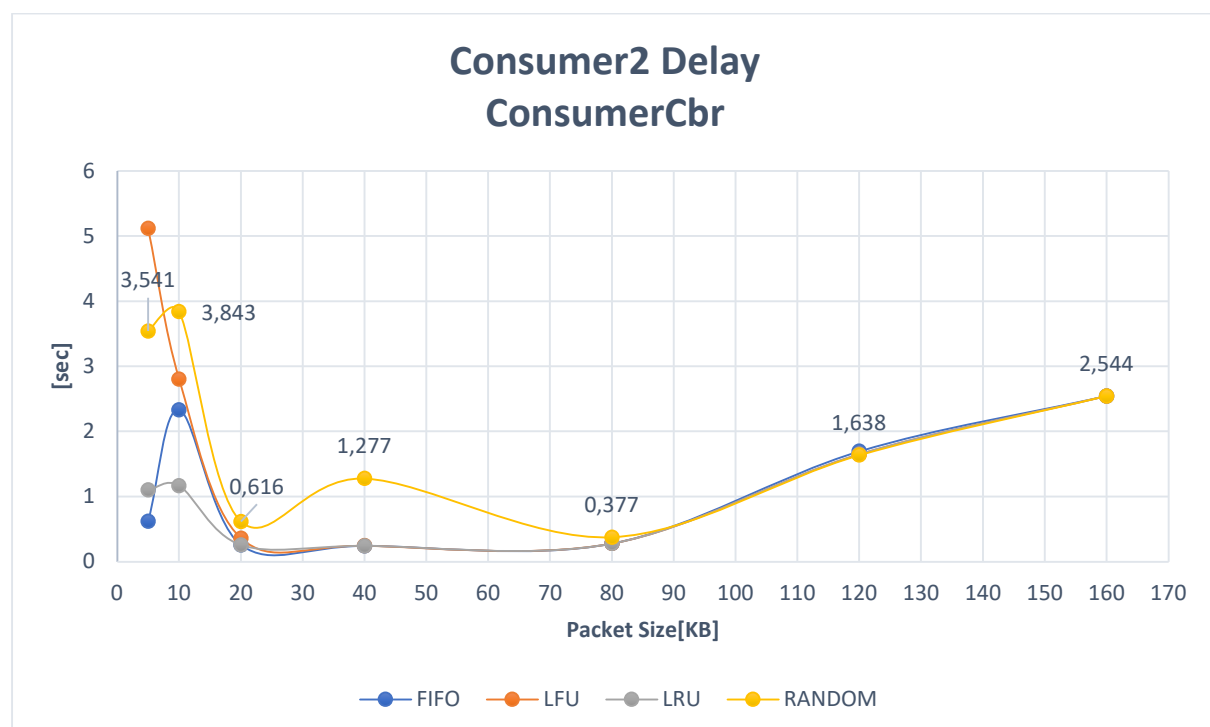
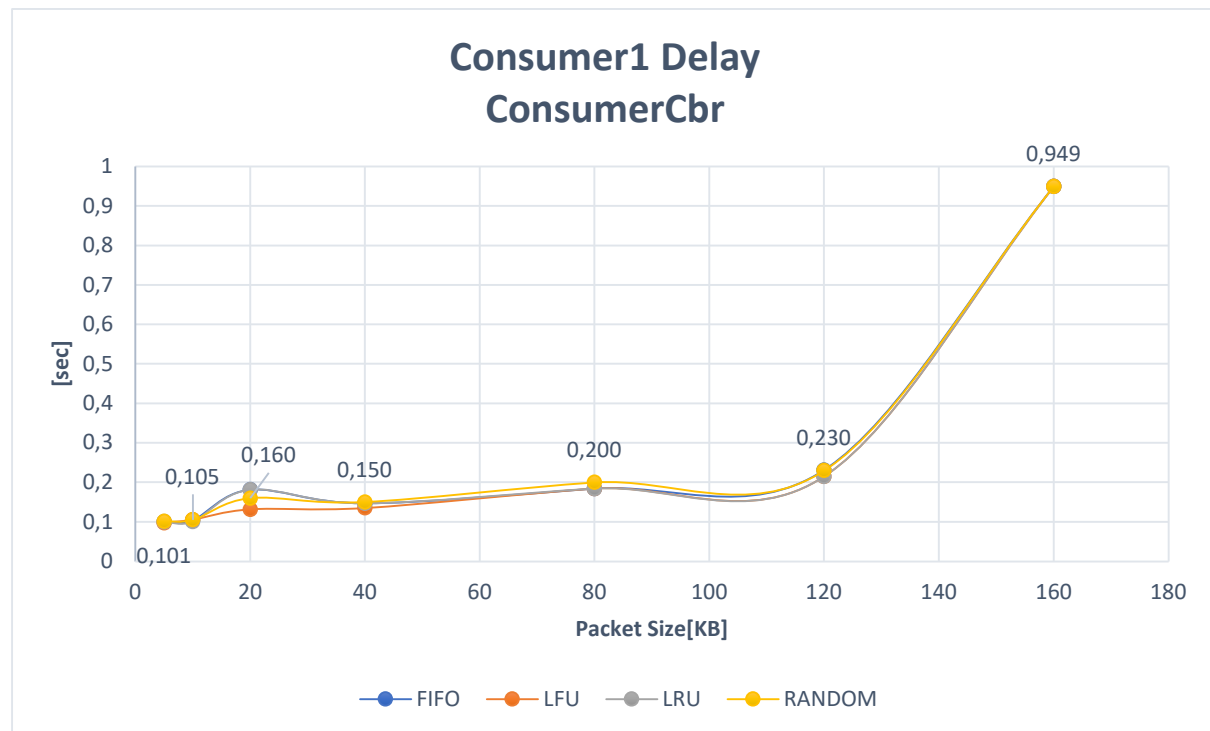
5.3.1.1 Throughput

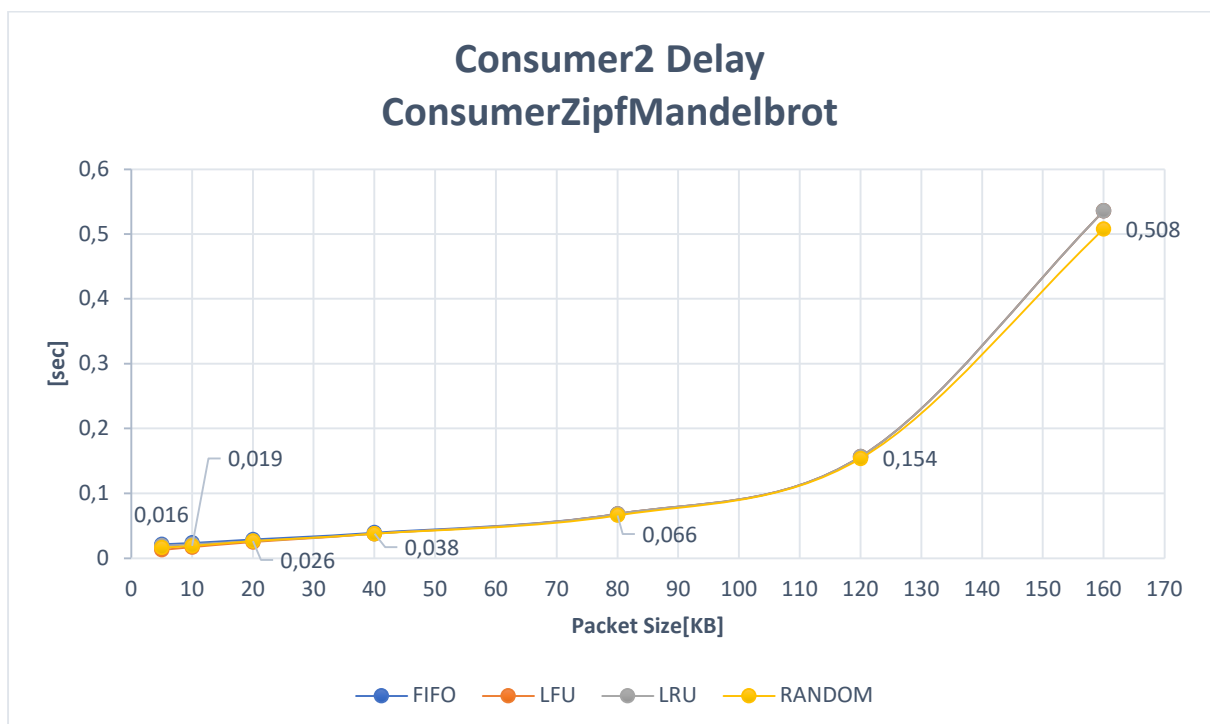
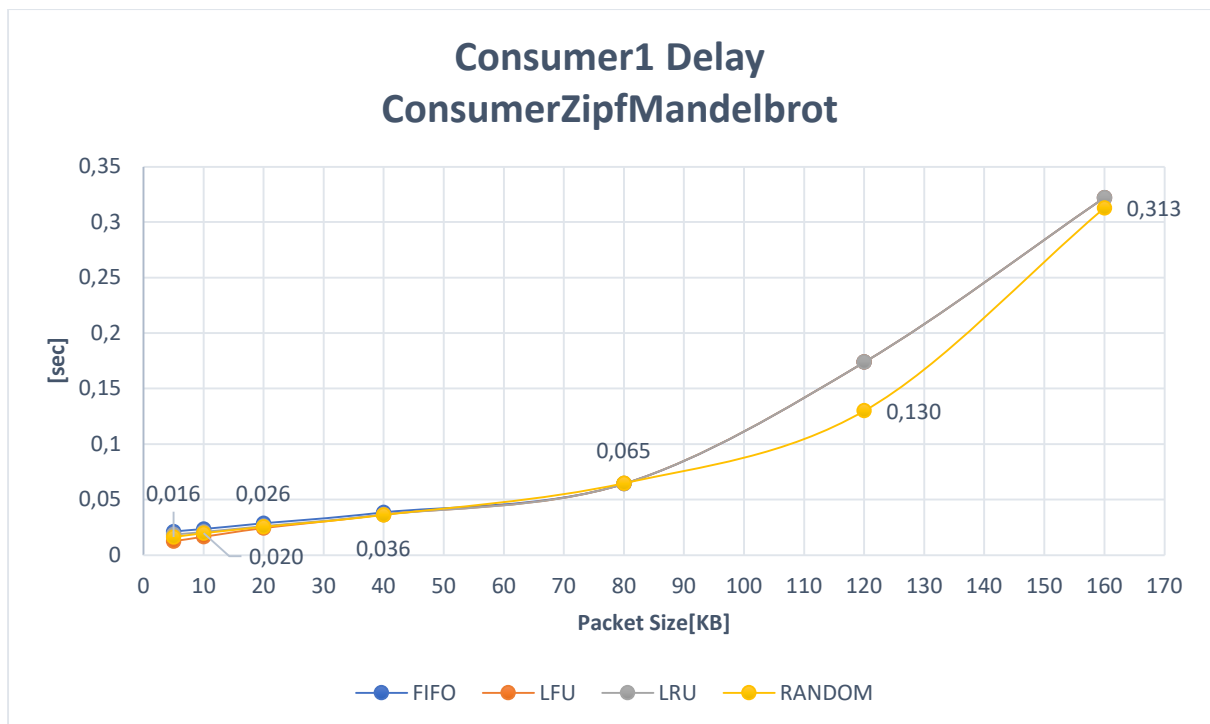


Η αισθητή μείωση του διαθέσιμου εύρους ζώνης οδήγησε σε μεταβολή των συνολικών μετρικών. Ειδικότερα, για την εφαρμογή ConsumerCbr παρατηρήθηκε έντονη μείωση του συνολικού Throughput, το οποίο παρέμεινε σταθερό στα σενάρια των προηγούμενων ενοτήτων.

Η συνολική επίδραση στην ConsumerZipf, καθίσταται μετριασμένη.

5.3.1.2 Συνολική καθυστέρηση – Delay



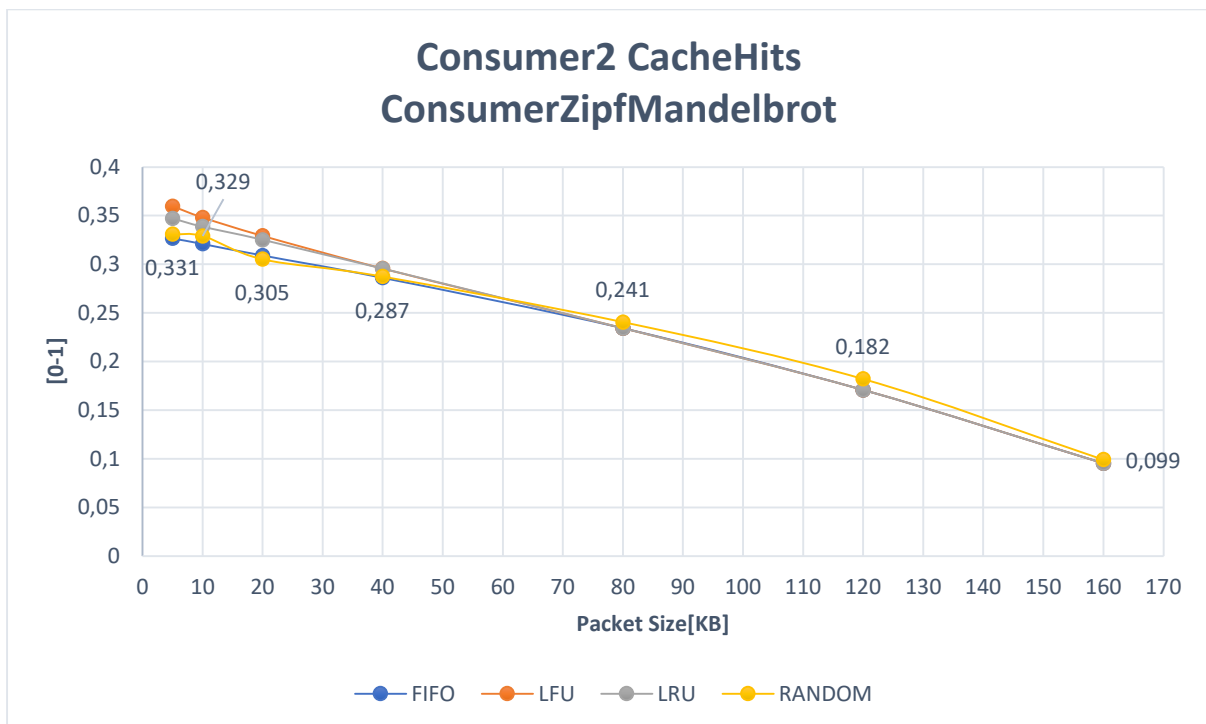
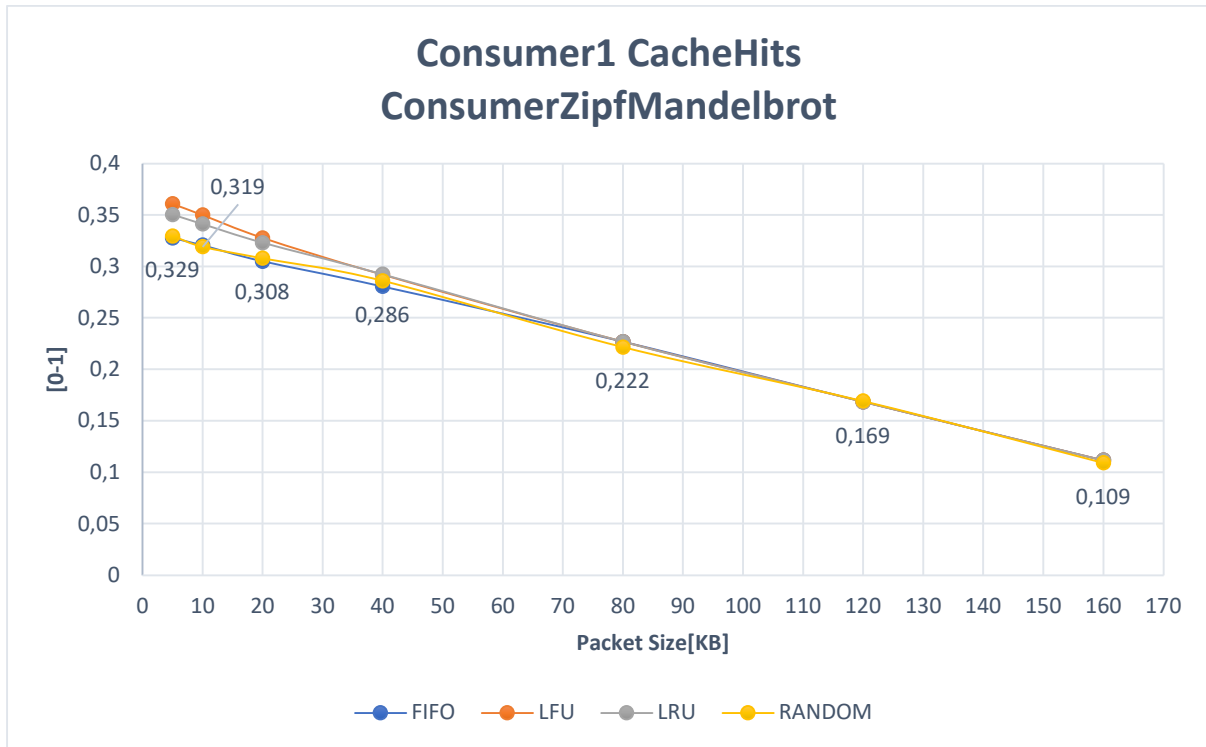


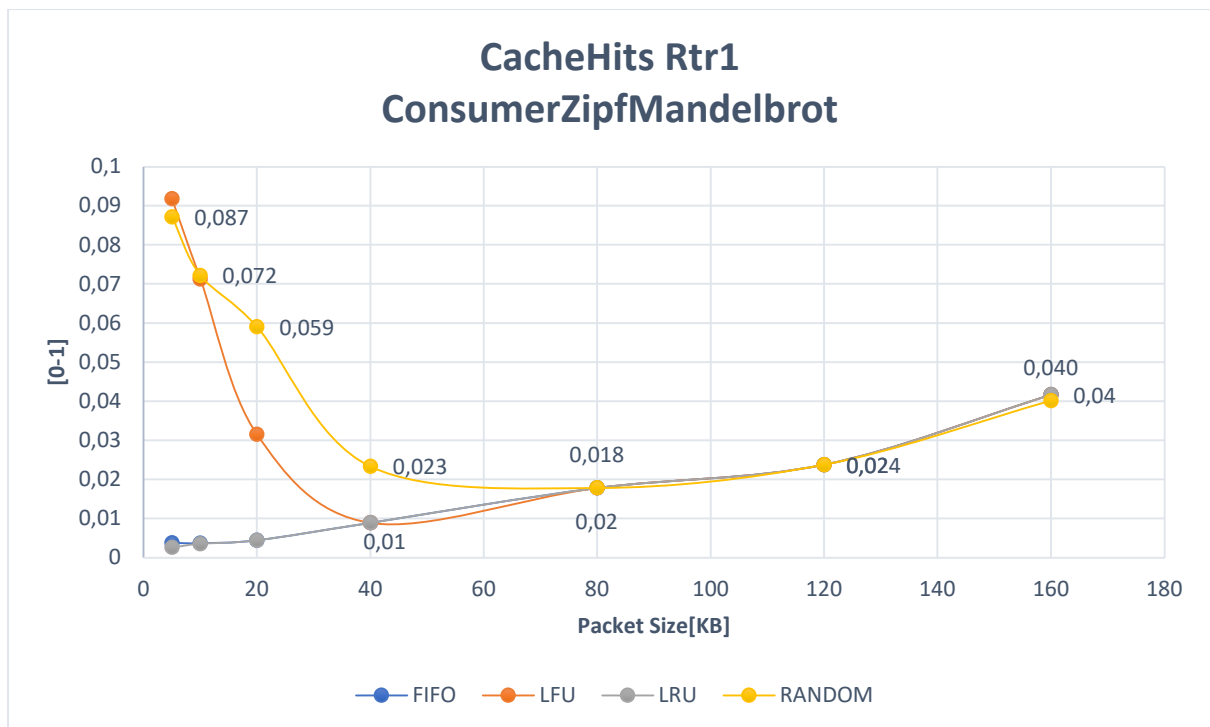
Οι προηγούμενες γραφικές αναπαραστάσεις αποτέλεσαν τη συγκριτική αντιπαραβολή των μετρικών για τους κόμβους Καταναλωτή, κατά τη μεταβολή του μεγέθους πακέτου Δεδομένων.

Συμπεραίνουμε ότι οι τιμές της συνολικής καθυστέρησης, στην περίπτωση της έντονης καθυστέρησης, έχουν αισθητή επίδραση στη συνολική απόδοση της δικτυακής υποδομής.

5.3.1.3 Επιτυχής ανάκτηση – CacheHits

Οι κάτωθι γραφικές αναπαραστάσεις αποτελούν συγκριτική απεικόνιση μεταξύ των δύο διαφορετικών εφαρμογών παραγωγής πακέτων Ενδιαφέροντος, για μεταβλητό μέγεθος πακέτων Δεδομένων.

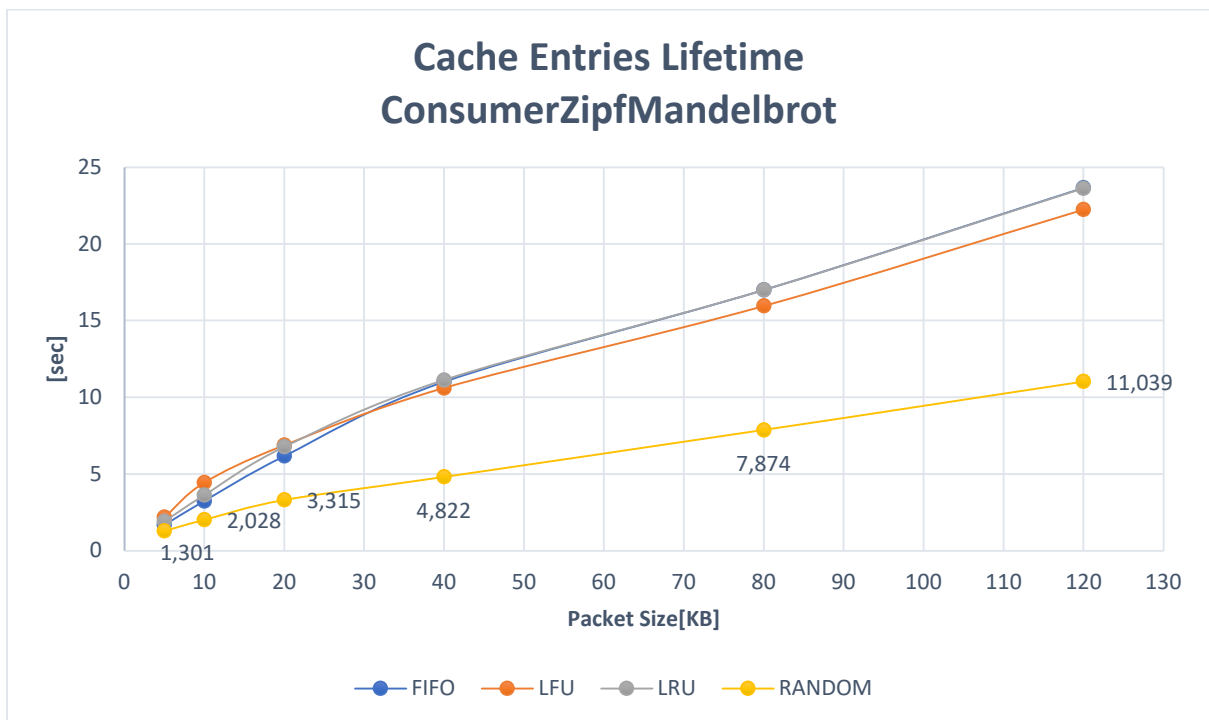
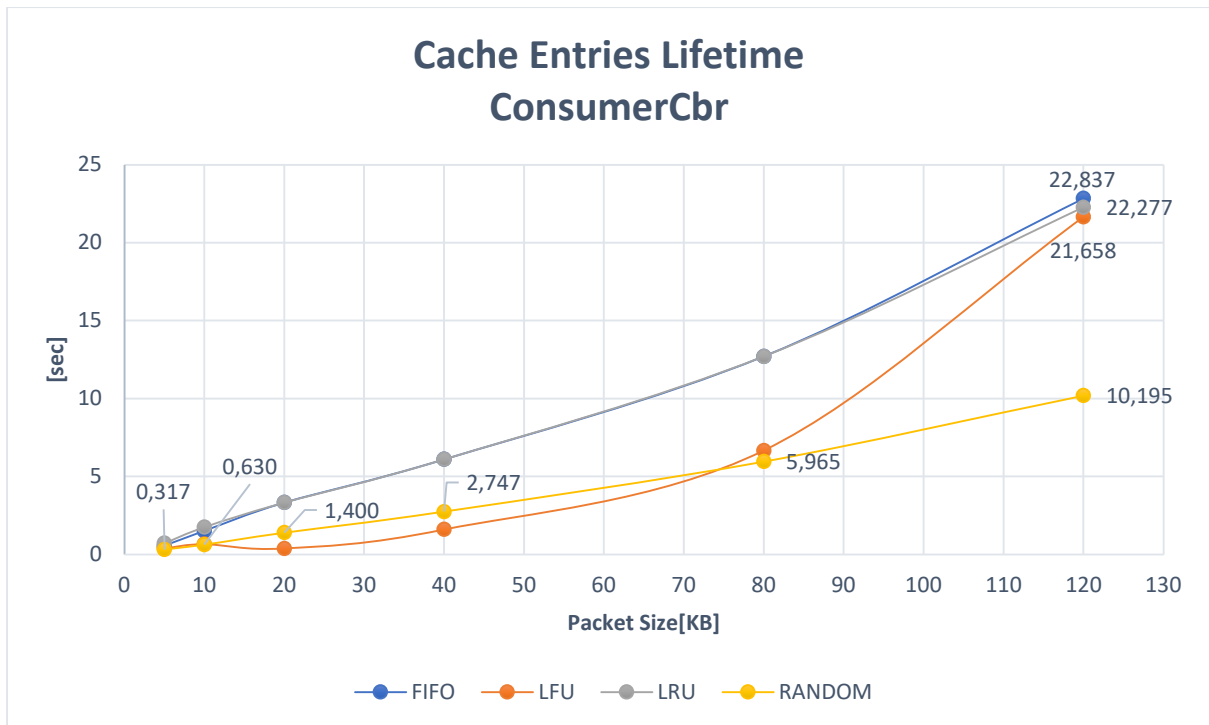




Συμπερασματικά, παρουσιάζεται μείωση των ποσοστών ανάκτησης πακέτων από τον πίνακα Content Store, σε σχέση με την προηγούμενη υπό-ενότητα.

5.3.1.4 Χρόνος ζωής καταχωρίσεων - Entries Life Time

Παρακάτω ακολουθεί η συγκριτική απεικόνιση των δύο διαφορετικών εφαρμογών παραγωγής πακέτων Ενδιαφέροντος, για μεταβλητό μέγεθος πακέτων Δεδομένων, στον κόμβο δρομολογητή-**Rtr1**.

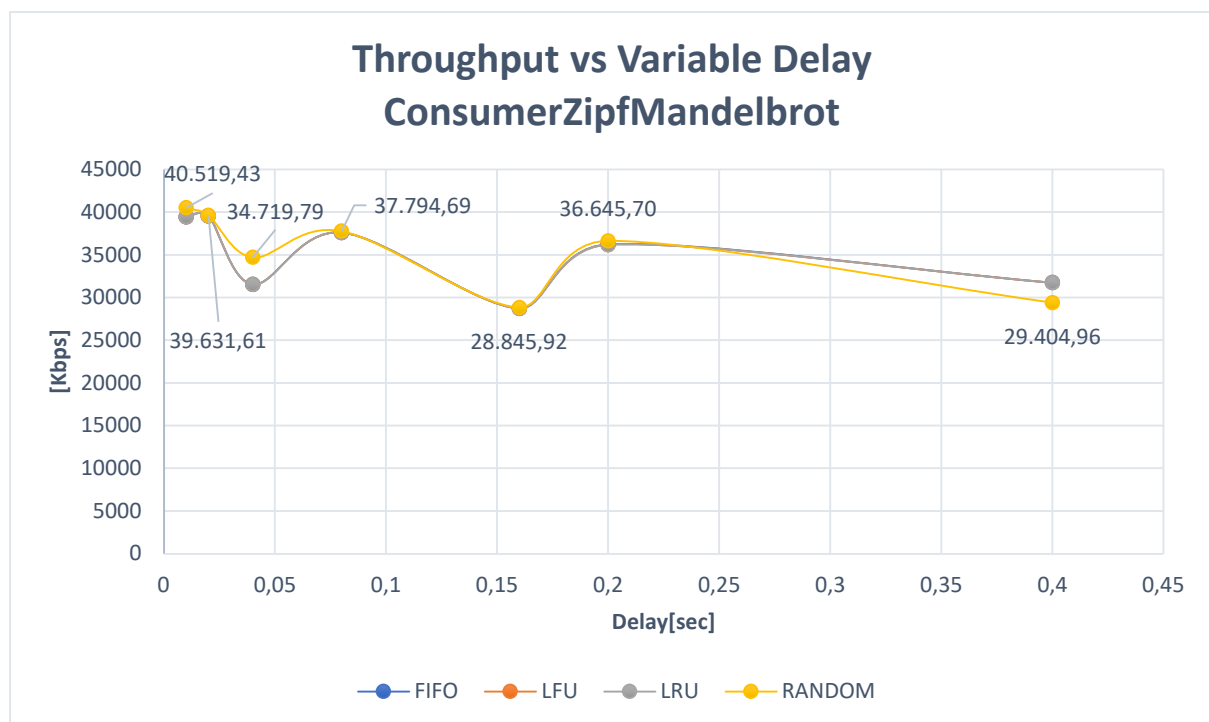
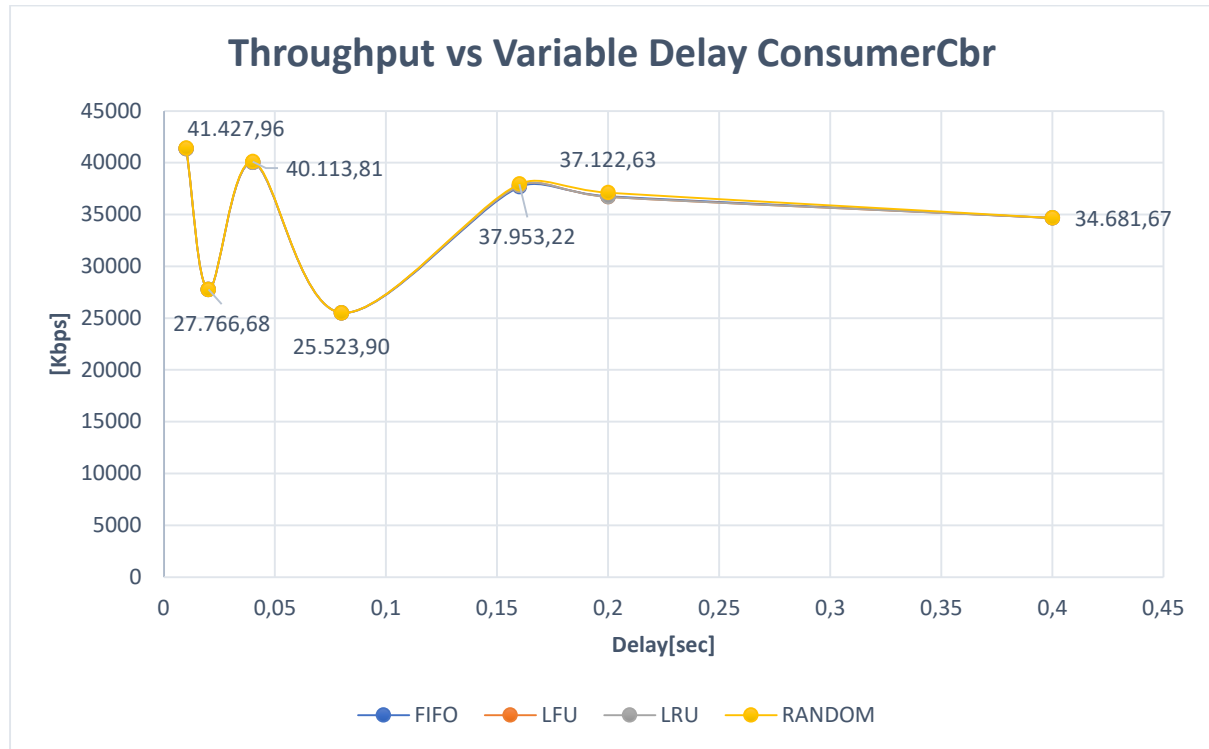


Βάσει των παραπάνω γραφικών αναπαραστάσεων, καθίσταται εμφανής η αύξηση του μέσου χρόνου παραμονής των πακέτων Δεδομένων, για έντονες συνθήκες συμφόρησης. Η πολιτική επεξεργασίας ουράς RANDOM διατηρεί ακόμη τις χαμηλότερες μετρικές, επηρεαζόμενη πλέον στον ανάλογο βαθμό από το μέγεθος πακέτου Δεδομένων.

5.3.2 Μεταβλητό μέγεθος καθυστέρησης διάδοσης

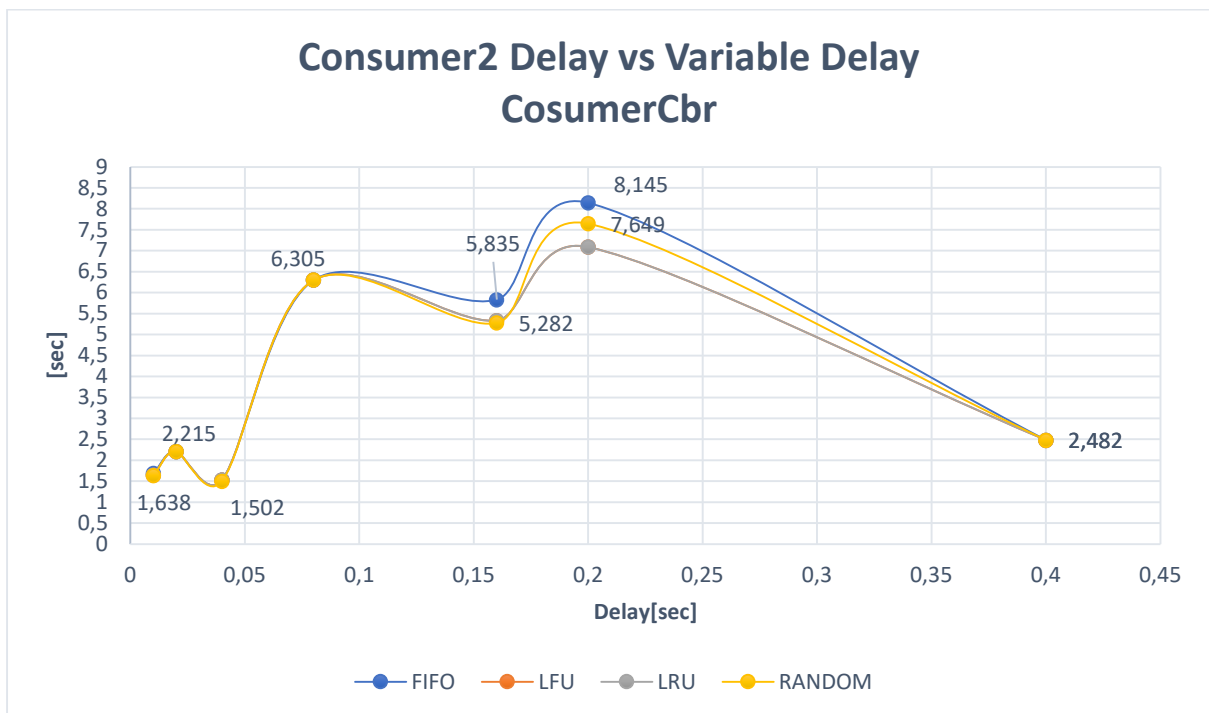
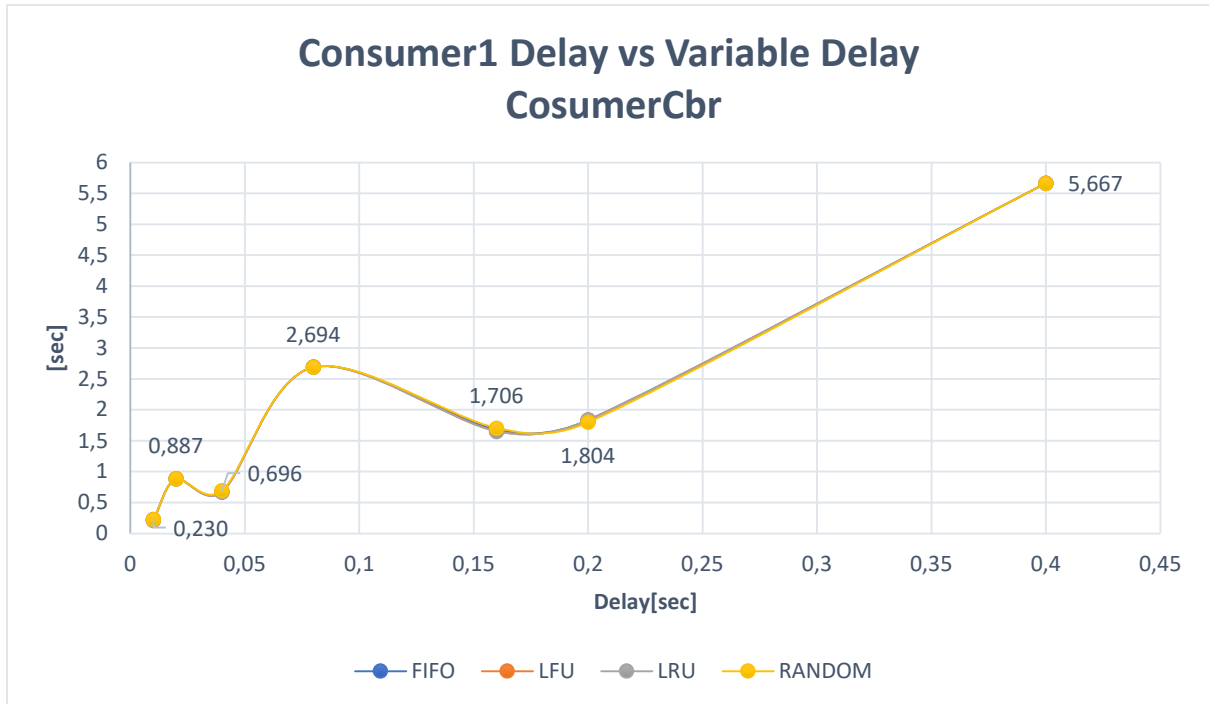
5.3.2.1 Throughput

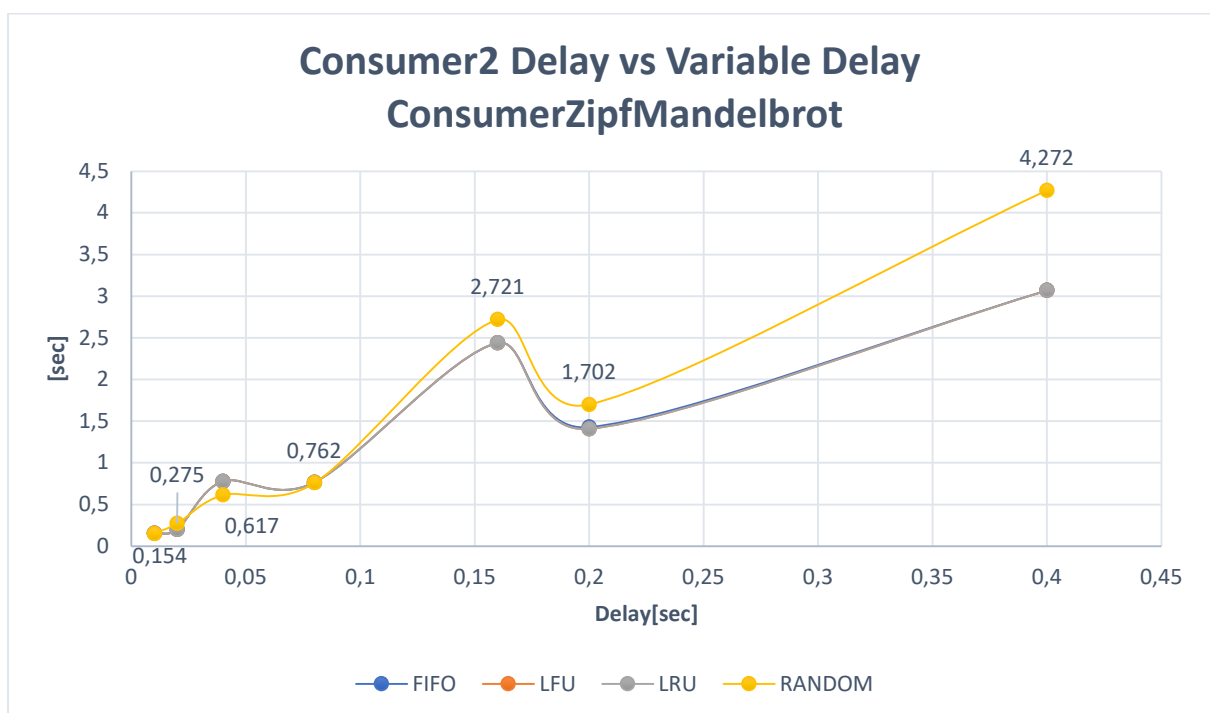
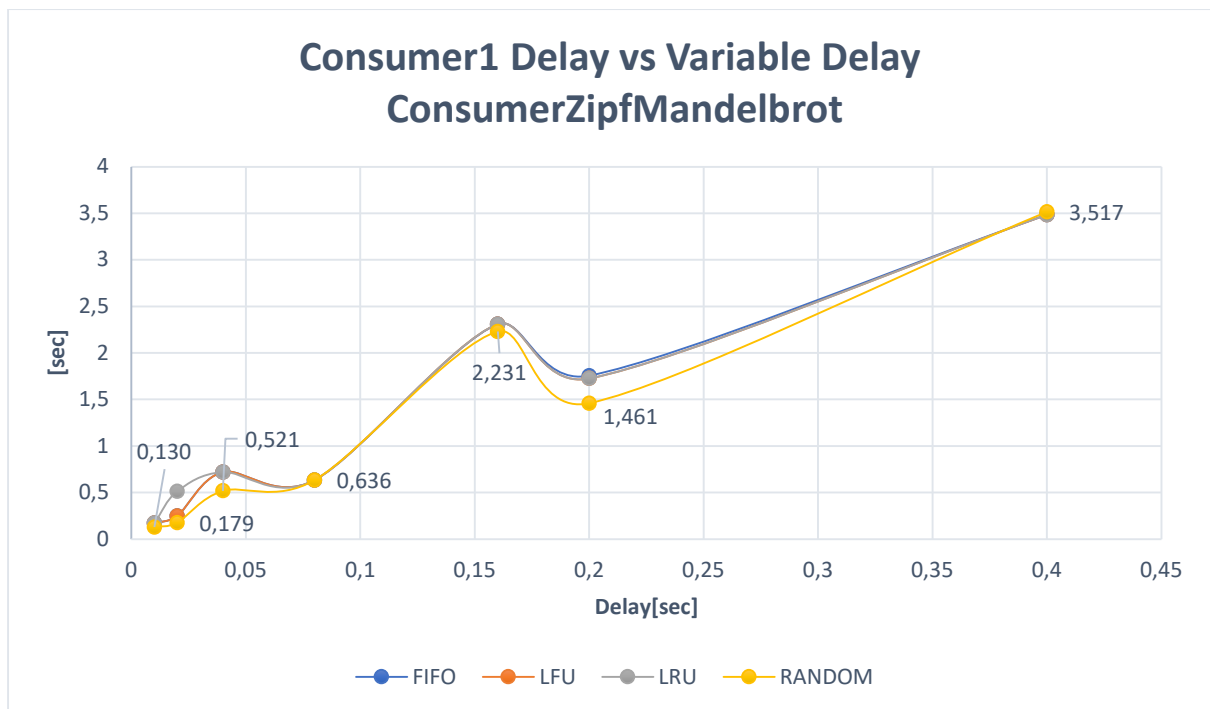
Οι ακόλουθες απεικονίσεις αποτελούν την συγκριτική απεικόνιση μεταξύ των δύο διαφορετικών εφαρμογών παραγωγής πακέτων Ενδιαφέροντος, για μεταβλητή τιμή της καθυστέρησης διάδοσης. Ως εκ τούτου, παρουσιάζεται η συνολική επίδραση της μεταβλητής τιμής, για το συνολικό Throughput.



Από τις προηγούμενες αναπαραστάσεις καθίσταται σαφής η επίδραση της καθυστέρησης διάδοσης στο συνολικό Throughput, σε συνθήκες περιορισμένου διαθέσιμου εύρους ζώνης. Η πολιτική επεξεργασίας ουράς δε διαδραματίζει καθοριστικό ρόλο στην τιμή του συνολικού throughput.

5.3.2.2 Συνολική καθυστέρηση – Delay

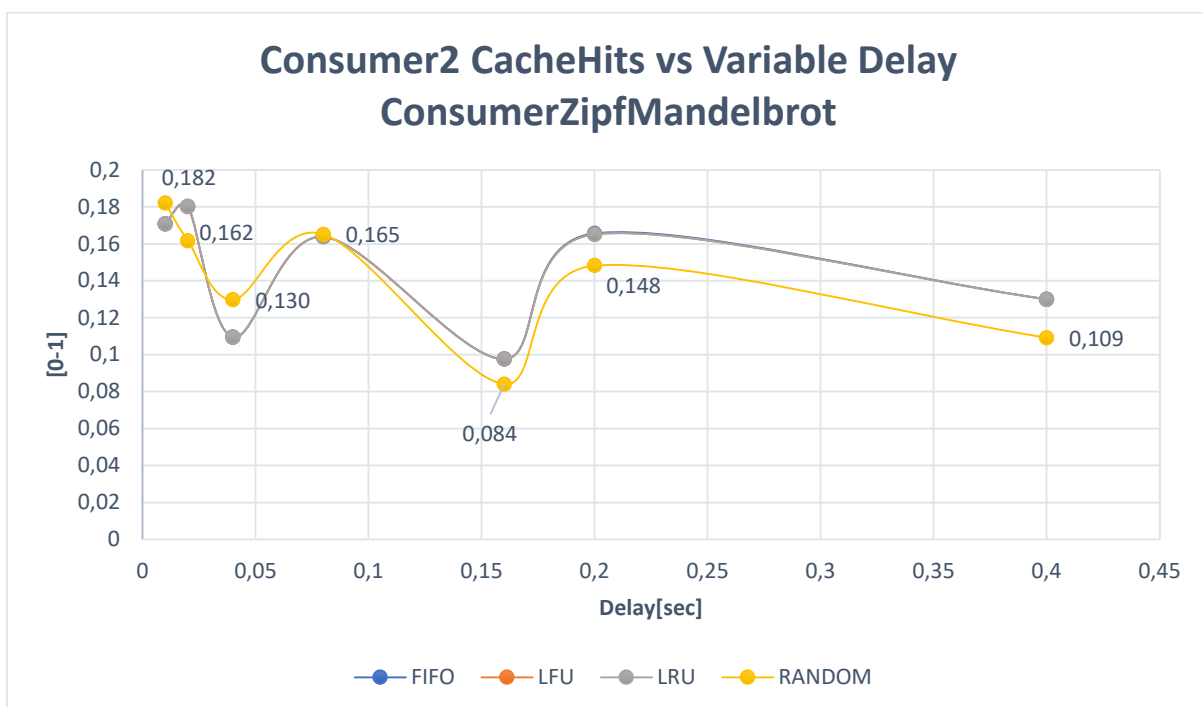
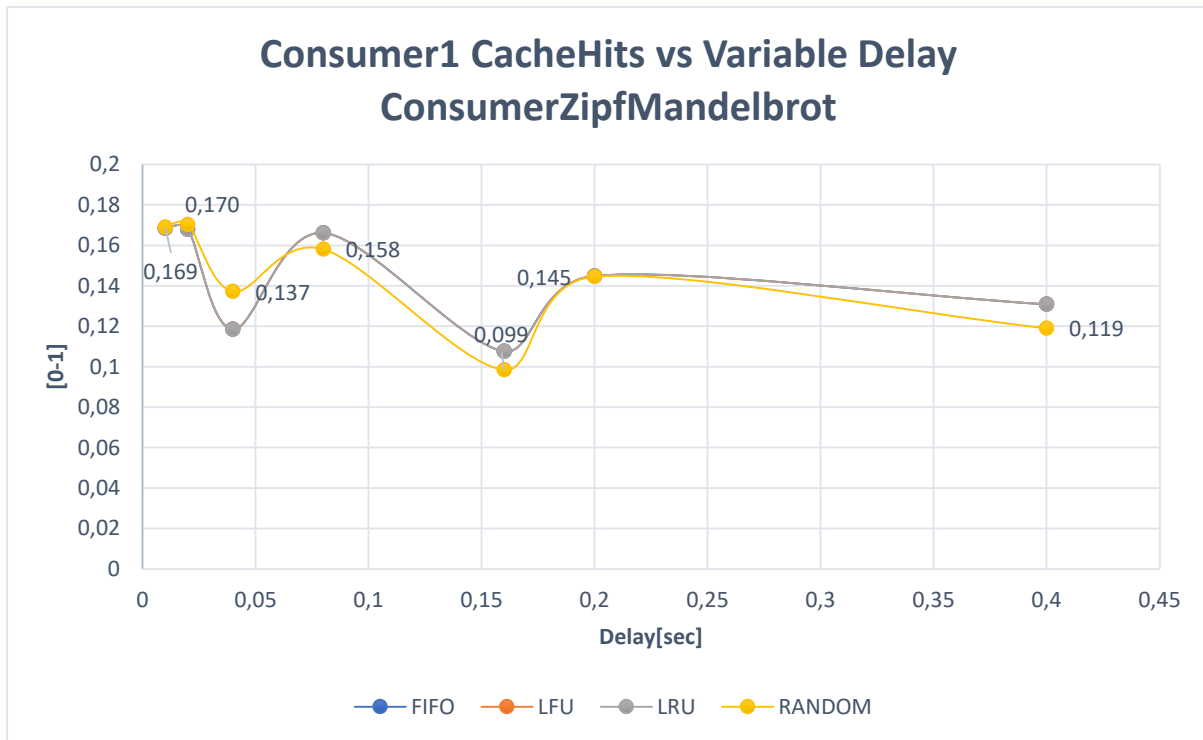


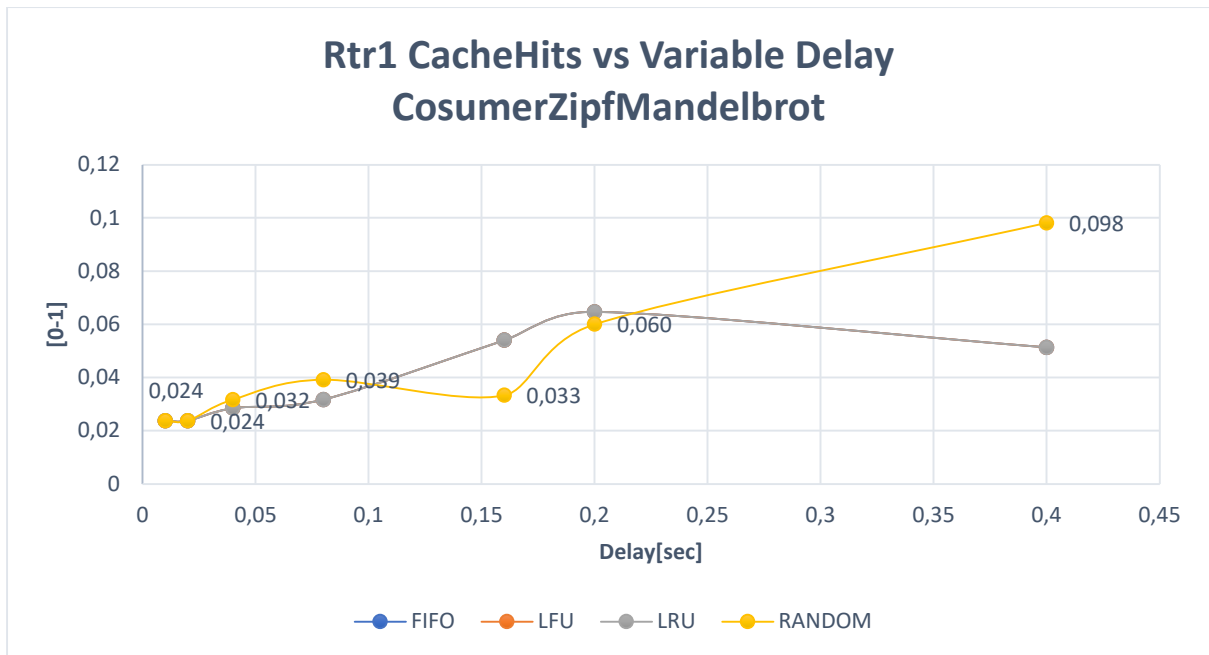


Οι προηγούμενες αναπαραστάσεις αποτέλεσαν την συγκριτική απεικόνιση για μεταβλητή τιμή καθυστέρησης διάδοσης, στους κόμβους Καταναλωτή. Ως εκ τούτου, παρατηρείται περαιτέρω αύξηση, σε σχέση με τις μετρικές των προηγούμενων υπό-ενοτήτων. Σημείο κλειδί των έντονων συνθηκών συμφόρησης αποτέλεσε η ιδιαίτερη και χαρακτηριστική αύξηση των τιμών καθυστέρησης στους δεύτερους Καταναλωτές.

5.3.2.3 Επιτυχής ανάκτηση – CacheHits

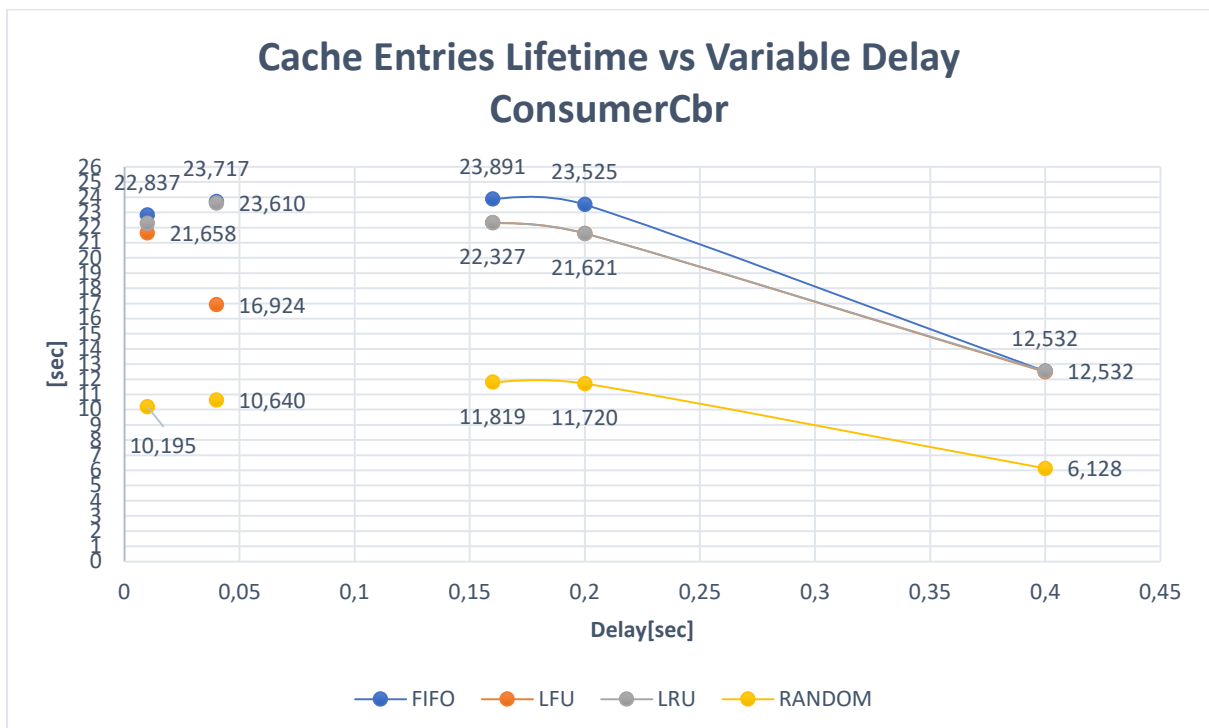
Οι μετρικές διαφοροποιούνται σε σημαντικό βαθμό, σε σχέση με αυτές των προηγούμενων υπό-ενοτήτων, καθώς τα ποσοστά ανάκτησης πακέτων Δεδομένων μειώνονται. Όλες οι πολιτικές επεξεργασίας ουράς παρουσιάζουν διακυμάνσεις στα ποσοστά ανάκτησης, κατά την αύξηση της καθυστέρησης διάδοσης.

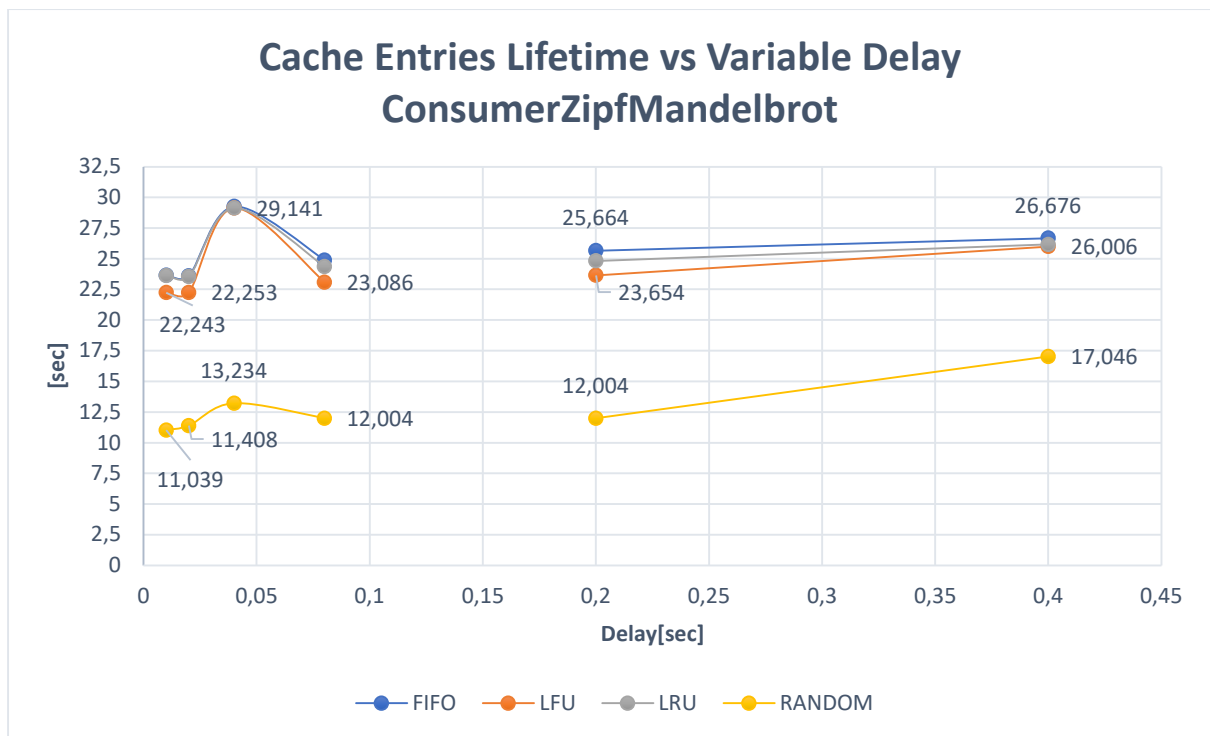




5.3.2.4 Χρόνος ζωής καταχωρίσεων - Cache Entries Lifetime

Οι κάτωθι γραφικές αναπαραστάσεις αποτελούν συγκριτική απεικόνιση μεταξύ των δύο διαφορετικών εφαρμογών παραγωγής πακέτων Ενδιαφέροντος, για μεταβλητή τιμή καθυστέρησης διάδοσης, στον κόμβο δρομολογητή-**Rtr1**-.





Η μερική ασυνέχεια των παραπάνω γραφημάτων, υποδηλώνει την ανυπαρξία πακέτων Δεδομένων στον πίνακα Περιεχόμενου για τα συγκεκριμένα μεγέθη τιμών καθυστέρησης διάδοσης, λόγω περαιτέρω συμφόρησης στη δικτυακή υποδομή. Σημειώνουμε ότι το εύρος ζώνης (BW) έχει μειωθεί δραστικά [60 Mbps] και η καθυστέρηση διάδοσης αυξάνεται σταδιακά από την αρχικοποιημένη τιμή των 10 χιλιοστών δευτερολέπτου [ms].

6 Μεθοδολογία-Συμπεράσματα

6.1 Μεθοδολογία

Η επίτευξη των τελικών αποτελεσμάτων αποτέλεσε μια διαδικασία συνεχούς αναζήτησης και πειραματισμού, καθότι στην ήδη υπάρχουσα βιβλιογραφία το σκέλος που συνδέεται με τα μεγάλα πακέτα Δεδομένων, χαρακτηρίζεται ως ανεπαρκές.

Η ερευνητική βάση της παρούσας διπλωματικής εργασίας αφορμάται από την ήδη προ-υπάρχουσα μελέτη[20,21], βάσει της οποίας προτάθηκαν υλοποιήσεις παροχής ad-hoc υπηρεσιών, οι οποίες βασίστηκαν στην ICN αρχιτεκτονικής. Οι υπηρεσίες αυτές βρίθουν χρησιμότητας, είτε σε σενάρια καταστροφών είτε σε απομακρυσμένες περιοχές, όπου η διαδικτυακή επικοινωνία καθίσταται δύσκολα επιτεύξιμη. Ως εκ τούτου, κρίνεται αναγκαία η μελέτη και επιπλέον επέκταση της έρευνας, για μεγάλα μεγέθη πακέτων Δεδομένων, τα οποία δύναται να μεταφέρονται μεταξύ των διάφορων ad-hoc υλοποιήσεων.

Θεμελιώδης γνώμονας στην όλη ερευνητική πορεία αποτέλεσαν οι βασικές δικτυακές αρχές, οι οποίες συνδυάστηκαν με συνεχή και αδιάλειπτη προσαρμογή βάσει των ανακτηθέντων δεδομένων. Η διαδικασία της ανατροφοδότησης οδήγησε στις αναγκαίες τροποποιήσεις της τοπολογίας, ώστε να επιτευχθεί ο αρχικός στόχος.

Ως εκ τούτου, οι διεργασίες υλοποίησης οδηγήθηκαν σταδιακά, με αργό βηματισμό, αφού στα πρώιμα στάδια ανέκυψε επιτακτική η ανάγκη μελέτης της επίδρασης και πακέτων Δεδομένων μικρού μεγέθους, στη συνολική απόδοση της δικτυακής αρχιτεκτονικής.

6.2 Συμπεράσματα

Τα αναλυθέντα αριθμητικά δεδομένα προσέδωσαν μια νέα πτυχή της δικτυακής υλοποίησης Named Data Networking, αφού απέδειξαν την αναντίρρητη ικανότητα διαχείρισης από πλευράς αρχιτεκτονικής, μεγάλων πακέτων Δεδομένων, συγκριτικά με το σύνηθες.

Η σταδιακή μεταβολή του μεγέθους και η μελέτη της συνολικής απόδοσης της υποδομής σε διαφορετικές συνθήκες διαθέσιμου εύρους ζώνης[BW], οδήγησε στα κάτωθι συμπεράσματα:

- Η τοπολογία αντιδρά επαρκώς και αποδοτικά, στα συνεχώς τροποποιημένα και αυξανόμενα, μεγέθη πακέτων Δεδομένων
- Για πακέτα Δεδομένων μεγέθους 5,10,20,40,80,120 [KB] η συνολική απόδοση της υποδομής δε διαφοροποιείται σε μεγάλο βαθμό
- Το σημείο κορεσμού της τοπολογίας εντοπίζεται καθ' υπέρβαση της τιμής πακέτου Δεδομένων 120 [KB]
- Για μέγεθος πακέτου Δεδομένων τιμής 160 [KB] παρατηρείται μείωση της συνολικής απόδοσης, συγκριτικά με τα προηγούμενα μεγέθη
- Η περαιτέρω αύξηση του μεγέθους πακέτου, οδήγησε σε ανεπαρκή αποτελέσματα, καθότι ο προσομοιωτής επέστρεφε μηδενικά δεδομένα. Ενδεικτικά πραγματοποιήθηκαν μετρήσεις για μεγέθη 200-320-640 [KB] και 1[MB], δίχως οιαδήποτε επιτυχία. Τα αποτελέσματα δεν παρουσιάστηκαν, καθότι δεν υπάρχει λόγος απεικόνισης μηδενικών μετρήσεων.

Η υλοποίηση των προγραφέντων πραγματοποιήθηκε για τέσσερις διαφορετικές πολιτικές διαχείρισης ουράς του πίνακα αποθήκευσης Πακέτων – Content Store. Τα δεδομένα που σχετίζονται με τις πολιτικές διαχείρισης FIFO,LFU,LRU δεν διαφοροποιούνται σε μεγάλο βαθμό, ιδιαίτερα για τις συνθήκες μη συμφόρησης.

Η πολιτική υλοποίησης RANDOM παρουσιάζει δυνητικά καλύτερη συμπεριφορά σε όλες τις συνθήκες συμφόρησης(ή μη), έχοντας είτε παρόμοια είτε ελάχιστα καλύτερη απόδοση, συγκριτικά με τις υπόλοιπες πολιτικές.

Εγγενές χαρακτηριστικό της RANDOM πολιτικής αποτέλεσε ο σχετικά μικρός χρόνος παραμονής του πακέτου Δεδομένων εντός του πίνακα Περιεχομένου. Το συγκεκριμένο χαρακτηριστικό μπορεί να θεωρηθεί

αξιοσημείωτο με όρους εξοικονόμησης αποθηκευτικού χώρου, αφού σε ορισμένες περιπτώσεις το χρονικό διάστημα ήταν μειωμένο κατά το ήμισυ.

Τα προγραφέντα δεδομένα επιβεβαιώνονται επιτυχώς και για τις δύο εφαρμογές παραγωγής πακέτων Ενδιαφέροντος.

Όπως ήδη προαναφέρθηκε στην ενότητα 5, η εγγενής υλοποίηση της εφαρμογής ConsumerCbr δεν επιτρέπει την σαφή ανάλυση δεδομένων που σχετίζονται με την επιτυχή ανάκτηση των πακέτων, από τον πίνακα Περιεχομένου(CS), αφού το πρόθεμα ονόματος διαφοροποιείται συνεχώς, οπότε και κανένα πακέτο Δεδομένων με ίδιο πρόθεμα δε ζητείται σε επόμενη χρονική στιγμή.

Από την άλλη η εφαρμογή ConsumerZipfMandelbrot, προσομοιώνει σε σημαντικό βαθμό την συνήθη συμπεριφορά των χρηστών, όπου ένα πακέτο Δεδομένων ζητείται πολλαπλές φορές. Ως εκ τούτου, μπορεί να ανακτηθεί κατευθείαν από τον κοντινότερο πίνακα Περιεχομένου, αντί να δρομολογηθεί στον “μη βέλτιστο” προορισμό αρχικής παραγωγής.

Το προηγούμενο συμπέρασμα μπορεί να επιβεβαιωθεί πλήρως από τη μειωμένη αξιοποίηση του δικτύου σε όρους αξιοποιήσιμου εύρους ζώνης, για κάθε υλοποιημένη προσομοίωση, αφού ο αριθμός των πακέτων Ενδιαφέροντος και του μεγέθους πακέτων Δεδομένων παρέμενε ίδιος.

Σε επόμενο στάδιο, χαρακτηρίστηκε ως “βέλτιστο” το μέγεθος πακέτου 120 [KB], οπότε και αξιοποιήθηκε για λόγους μελέτης απόδοσης στις περιπτώσεις μεταβλητής καθυστέρηση διάδοσης και εύρους ζώνης [BW].

Μέσω της συγκεκριμένης μεθοδολογίας επιβεβαιώθηκε το υπάρχον γνωσιακό μας υπόβαθρο, βάσει του οποίου η αυξανόμενη καθυστέρηση διάδοσης, διαδραματίζει καθοριστικό ρόλο στη μείωση της συνολικής αποδοτικότητας της δικτυακής υποδομής. Ειδικότερα, σε συνθήκες συμφόρησης εύρους ζώνης, η αυξανόμενη καθυστέρηση διάδοσης μπορεί να οδηγήσει στη συνολική κατάρρευση της υποδομής. Το χαρακτηριστικό αυτό πρέπει να αξιοποιείται εκ των προτέρων σε περιπτώσεις μετάδοσης δεδομένων, μεταξύ ευκαιριακών διασυνδέσεων.

Τέλος, πρέπει να αναφερθεί ότι οι εφαρμογές παραγωγής πακέτων Ενδιαφέροντος δε διαθέτουν κάποιον μηχανισμό ανάκτησης και αναμετάδοσης σε περιπτώσεις κατάρρευσης της τοπολογίας. Ως εκ τούτου και επί του πρακτέου, ενδεχόμενη κατάρρευση της τοπολογίας, οδηγεί είτε στη συνολική

αποτυχία της προσομοίωσης είτε στην άντληση ανεπαρκών και ανακριβών δεδομένων.

6.3 Προτάσεις

Σε επόμενο στάδιο μπορούν να ενσωματωθούν μηχανισμοί ανάκτησης και αναμετάδοσης πακέτων Ενδιαφέροντος που χρησιμοποιούνται από τις υπάρχουσες δικτυακές υποδομές, στις εφαρμογές παραγωγής μεγάλων πακέτων Ενδιαφέροντος και Δεδομένων, ώστε να μελετηθεί εκ νέου η συνολική αποδοτικότητα.

7 Βιβλιογραφία και ηλεκτρονικές αναφορές

- [1] V. Jacobson, *"Future Internet Summer School 2009 (FISS 09) - Special Invited Plenary Short Course: (CCN) Content Centric Networking,"* Podcast, July 2009, part 1 of 4. [Online]. Available: https://mlecture.uni-bremen.de/ml/index.php?option=com_mlplayer&template=ml2&mlid=1850
- [2] Statista, *"Number of IoT connected devices worldwide 2018,2025 and 2030"*, [Online], Available: <https://www.statista.com/statistics/802690/worldwide-connected-devices-by-access-technology/>
- [3] Wikipedia, *"Mobile IP"*, [Online]. Available: https://en.wikipedia.org/wiki/Mobile_IP
- [4] George Xylomenos, Christopher N. Ververidis, Vasilios A. Siris, Nikos Fotiou, Christos Tsilopoulos, Xenofon Vasilakos, Konstantinos V. Katsaros, and George C. Polyzos *"A Survey of Information-Centric Networking Research"*, in IEEE Communications Surveys & Tutorials, Vol.16, No. 2
- [5] Wikipedia, *"Internet Protocol Suite"*, [Online], Available: https://en.wikipedia.org/wiki/Internet_protocol_suite
- [6] D. E. Comer, *"Internetworking with TCP/IP : 1 : Principles, protocols, and architecture"*, 2nd ed., N. Englewood Cliffs, Ed. Prentice-Hall, 1991, vol. 1.
- [7] Internet Assigned Numbers Authority (IANA), *"Abuse Issues and IP Addresses"*, Webpage, Sep. 2016, [Online], Available: <https://www.iana.org/help/abuse-answers>
- [8] S. Deering and R. Hinden, *"Internet protocol, version 6 (ipv6) specification"*, IETF, Standard, Dec. 1998, [Online]
- [9] D. D. Clark, *"The design philosophy of the darpa internet protocols"*, Computer Communication Review, vol. 18, 114, August 1988
- [10] Spyridon Mastorakis, Alexander Afanasyev, Ilya Moiseenko and Lixia Zhang, *"ndnSIM: NDN simulator for NS-3, Technical Report NDN-0028"*, 2015
- [11] Van Jacobson and Diana K. Smetters, *"Networking Named Content"*, Palo Alto Research Center, CA, USA
- [12] Named Data Networking, *"Interest Packet"*, [Online], Available: <https://named-data.net/doc/NDN-packet-spec/current/interest.html>
- [13] Named Data Networking, *"Data Packet"*, [Online], Available: <https://named-data.net/doc/NDN-packet-spec/current/>
- [14] Named Data Networking, *"Names"*, [Online], Available: <https://named-data.net/doc/NDN-packet-spec/current/name.html>
- [15] Alexander Afanasyev, Junxiao Shi, Beichuan Zhang, Lixia Zhang, Ilya Moiseenko, Yingdi Yu, Wentao Shang, Yanbiao Li, Spyridon Mastorakis, Yi Huang, *"NFD Developer's Guide"*, NDN, Technical Report NDN-0021, [Online], Available: <http://named-data.net/publications/techreports/>
- [16] Afanasyev, S. Mastorakis, I. Moiseenko, and L. Zhang, *"ndnSIM applications"*, Web site, 2017, <http://ndnsim.net/current/applications.html>,
- [17] M. Newman, *"Power laws, pareto distributions and zipf's law"*, Department of Physics and Center for the Study of Complex Systems, University of Michigan, U.S.
- [18] Afanasyev, S. Mastorakis, I. Moiseenko, and L. Zhang, *"Obtaining metrics"*, 2017, [Online], Available: <https://ndnsim.net/current/metric.html>,
- [19] R project, *"Introduction to R"*, [Online], Available: <https://www.r-project.org/about.html>
- [20] Christos-Alexandros Sarros, Sotiris Diamantopoulos, Sergi Rene, Ioannis Psaras Adisorn Lertsinsrubtavee, Carlos Molina-Jimenez, Paulo Mendes, Rute Sofia, Arjuna Sathiaselalan, George Pavlou, Jon Crowcroft, and Vassilis Tsaoussidis, *"Connecting the Edges: A Universal, Mobile-Centric, and Opportunistic Communications Architecture"*, UMOBILE, 2018
- [21] Christos-Alexandros Sarros, Adisorn Lertsinsrubtavee, Konstantinos Prasopoulos, Sotiris Diamantopoulos, *"ICN-based Edge Service Deployment in Challenged Networks"*, Democritus University of Thrace, Greece