



**ΔΗΜΟΚΡΙΤΕΙΟ  
ΠΑΝΕΠΙΣΤΗΜΙΟ  
ΘΡΑΚΗΣ**  **DEMOCRITUS  
UNIVERSITY  
OF THRACE**

Δημοκρίτειο Πανεπιστήμιο Θράκης  
Τμήμα Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών  
Τομέας Λογισμικού και Ανάπτυξης Εφαρμογών

## **ΔΙΑΧΥΣΗ ΠΛΗΡΟΦΟΡΙΑΣ ΣΕ ΠΛΗΡΟΦΟΡΙΟΚΕΝΤΡΙΚΑ ΔΙΚΤΥΑ**

**Διπλωματική Εργασία  
Φλώρου Δήμητρα**

**Επιβλέπων Καθηγητής  
Τσαουσίδης Βασίλειος**

## Ευχαριστίες

Θα ήθελα να ευχαριστήσω θερμά τον κ. Βασίλειο Τσαουσίδα, Καθηγητή του Τμήματος Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών, για την εμπιστοσύνη που μου έδειξε αναθέτοντάς μου την παρούσα διπλωματική εργασία και την συμβολή του στην επιστημονική μου κατάρτιση. Επιπλέον, οφείλω να ευχαριστήσω τους υποψήφιους διδάκτορες Βασίλειο Δεμίρογλου και Αλέξανδρο Σάρρο για την πολύτιμη βοήθειά τους κατά τη διάρκεια της εκπόνησης της διπλωματικής εργασίας. Τέλος, θα ήθελα να ευχαριστήσω την οικογένειά μου και τους φίλους μου για τη στήριξη και την αγάπη τους καθ' όλη τη διάρκεια των σπουδών μου.

## Πίνακας περιεχομένων

|                                                                                                                               |    |
|-------------------------------------------------------------------------------------------------------------------------------|----|
| 1. Εισαγωγή .....                                                                                                             | 1  |
| 2. Βιβλιογραφική Ανασκόπηση .....                                                                                             | 3  |
| 2.1. Αρχιτεκτονικές Μελλοντικού Διαδικτύου .....                                                                              | 3  |
| 2.1.1. Delay Tolerant Networking .....                                                                                        | 3  |
| 2.1.2. Information Centric Networking .....                                                                                   | 11 |
| 2.1.3. Named Data Networking .....                                                                                            | 12 |
| 2.1.4. Universal, Mobile-Centric and Opportunistic Communications Architecture (UMOBILE) .....                                | 18 |
| 2.2. Σχετική Έρευνα .....                                                                                                     | 23 |
| 2.2.1. Συσκευές IoT με διαλείπουσες συνδέσεις: Πειράματα με μία αρχιτεκτονική NDN-DTN .....                                   | 23 |
| 2.2.2. Η επίδραση των ασύρματων κόμβων για Μοντέλα Κινητικότητας σε Πρωτόκολλα Δρομολόγησης Ανεκτικά στις Καθυστερήσεις ..... | 24 |
| 2.2.3. Εφαρμογή τεχνικών Μηχανικής Μάθησης σε δρομολόγηση δικτύων ανθεκτικών στις καθυστερήσεις .....                         | 25 |
| 3. Εργαλεία .....                                                                                                             | 26 |
| 3.1. Docker Container .....                                                                                                   | 26 |
| 3.2. Common Open Research Emulator .....                                                                                      | 31 |
| 3.3. IRB-DTN .....                                                                                                            | 31 |
| 3.4. Named Data Networking Forwarding Daemon (NFD) .....                                                                      | 32 |
| 3.5. BonnMotion .....                                                                                                         | 34 |
| 4. Πειραματική Διαδικασία .....                                                                                               | 35 |
| 4.1. Δημιουργία Docker Container .....                                                                                        | 35 |
| 4.2. Τοπολογία πειράματος .....                                                                                               | 36 |
| 4.3. Έναρξη πειράματος .....                                                                                                  | 36 |
| 4.4. Μεθοδολογία .....                                                                                                        | 37 |
| 4.4.1. Μοντέλα κινητικότητας .....                                                                                            | 38 |
| 4.4.2. Πειράματα .....                                                                                                        | 40 |
| 5. Αποτελέσματα πειραμάτων .....                                                                                              | 43 |
| 5.1. Πείραμα 1 .....                                                                                                          | 43 |
| 5.2. Πείραμα 2 .....                                                                                                          | 47 |
| 5.3. Πείραμα 3 .....                                                                                                          | 50 |
| 5.4. Πείραμα 4 .....                                                                                                          | 53 |
| 5.5. Πείραμα 5 .....                                                                                                          | 57 |

|                                                |    |
|------------------------------------------------|----|
| 5.6. Συμπεράσματα και Μελλοντική Εργασία ..... | 60 |
| Βιβλιογραφία .....                             | 61 |
| Παράρτημα.....                                 | 63 |
| Π.1. Κώδικες .....                             | 63 |
| Dockerfile .....                               | 63 |
| YML file .....                                 | 66 |
| Bash Script Producer .....                     | 66 |
| Bash Script Consumer.....                      | 67 |
| Π.2. Πίνακες Αποτελεσμάτων .....               | 69 |
| Πείραμα 1 .....                                | 69 |
| Πείραμα 2 .....                                | 72 |
| Πείραμα 3 .....                                | 74 |
| Πείραμα 4 .....                                | 77 |
| Πείραμα 5 .....                                | 87 |

# 1. Εισαγωγή

Στις μέρες μας, ζούμε σε μία κοινωνία όπου το Διαδίκτυο έχει περάσει τα όρια της αρχικής του χρήσης ως ένα δίκτυο υπολογιστών που θα ήταν συνδεδεμένοι μεταξύ τους με βασικό σκοπό την επικοινωνία. Η χρήση του Διαδικτύου έχει επεκταθεί σε άλλους τομείς όπως για παράδειγμα στο ηλεκτρονικό εμπόριο, στα μέσα κοινωνικής δικτύωσης, στις εφαρμογές στα smartphone αλλά και στα δίκτυα IoT και χρησιμοποιείται πλέον κατά βάση ως δίκτυο διανομής. Ο συνεχώς αυξανόμενος όγκος χρηστών αλλά και συσκευών συνδεδεμένων σε αυτό καθημερινά και η ανάγκη να εξυπηρετηθούν με το βέλτιστο τρόπο οδήγησε στην ανάπτυξη των μελλοντικών αρχιτεκτονικών του Διαδικτύου που εστιάζουν στα προβληματικά σημεία της ήδη υπάρχουσας αρχιτεκτονικής αλλά και στις μελλοντικές απαιτήσεις του Διαδικτύου.

Η πετυχημένη υπάρχουσα αρχιτεκτονική και τα πρωτόκολλα του Διαδικτύου, μπορεί να μην έχουν τη βέλτιστη απόδοση σε περιβάλλοντα με μεγάλες καθυστερήσεις, μη συνεχή συνδεσιμότητα στο δίκτυο και συχνές τμηματοποιήσεις του, σε μικρότερα υποδίκτυα, τα οποία μπορεί να έχουν τα δικά τους εξειδικευμένα πρωτόκολλα και να μην χρησιμοποιούν IP.

Για το λόγο αυτό αναπτύχθηκε η αρχιτεκτονική Delay Tolerant Networking(DTN) η οποία όντας σχεδιασμένη ειδικά για ετερογενή δίκτυα, έχει τη δυνατότητα να εφαρμοστεί στα περιβάλλοντα αυτά και να καλύψει τις ανάγκες τους. Χρησιμοποιώντας την τεχνική αποθήκευσης και προώθησης (store-and-forward) για τη δρομολόγηση του συνόλου των μηνυμάτων της, τα οποία ονομάζονται bundles[1], αντιμετωπίζει αποτελεσματικά τις διακοπτόμενες συνδέσεις. Επειδή η DTN αρχιτεκτονική είναι μια αρχιτεκτονική τύπου "overlay", είναι δηλαδή φτιαγμένη για να λειτουργεί πάνω από τις ήδη υπάρχουσες στοίβες πρωτοκόλλων, δεν είναι συμβατή μόνο με το σημερινό Διαδίκτυο το οποίο βασίζεται σε TCP/IP πρωτόκολλα, αλλά έχει την ικανότητα να συνδέσει μεταξύ τους εντελώς διαφορετικού τύπου δίκτυα, με ασυνήθιστες ιδιότητες συνδεσιμότητας.

Η αρχιτεκτονική Named Data Networking (NDN) [2] είναι μία προσέγγιση της αρχιτεκτονικής Information Centric Networking (ICN)[3]. Βασίζεται στη λογική ότι η επικοινωνία σ' ένα δίκτυο γίνεται με προσανατολισμό την ίδια την πληροφορία και όχι την τοποθεσία προορισμού της πληροφορίας όπως συμβαίνει σε TCP/IP πρωτόκολλα. Τα δίκτυα NDN βασίζονται στην αποθήκευση και την ανάκτηση της πληροφορίας. Πιο συγκεκριμένα, τα δεδομένα έχουν ένα καθορισμένο όνομα με το οποίο προσδιορίζονται και η ανάκτηση τους γίνεται με βάση αυτό. Έτσι όταν ένας συνδρομητής αιτηθεί για κάποιο περιεχόμενο, αποστέλλεται ένα πακέτο ενδιαφέροντος το οποίο πακέτο αυτό προωθείται στο δίκτυο και μπορεί να ικανοποιηθεί, είτε από αυτόν που παρήγαγε το περιεχόμενο αυτό, είτε από κάποιον άλλον κόμβο που έχει αποθηκευμένη την πληροφορία. Το αίτημα ικανοποιείται με την αποστολή ενός πακέτου δεδομένων προς τον αποστολέα του αιτήματος. Υπάρχει λοιπόν η πεποίθηση ότι αυτή η εννοιολογικά απλή μετατόπιση, από το χώρο στην ίδια την πληροφορία, θα έχει εκτεταμένες επιπτώσεις στον τρόπο που οι άνθρωποι σχεδιάζουν, αναπτύσσουν και χρησιμοποιούν δίκτυα και εφαρμογές.

Το UMOBILE[4] ενσωματώνει το μοντέλο της πληροφοριοκεντρικής επικοινωνίας (ICN) και τις ευκαιριακές επικοινωνίες (opportunistic communications) σε μία ενιαία αρχιτεκτονική που είναι ανεκτική στις καθυστερήσεις (delay-tolerant). Συγκεντρώνοντας τις υπηρεσίες κοντά στα άκρα του δικτύου, μια τέτοια αρχιτεκτονική μπορεί να λειτουργήσει σε οποιοδήποτε

περιβάλλον δικτύωσης και επιτρέπει την ανάπτυξη καινοτόμων εφαρμογών, παρέχοντας πρόσβαση στα δεδομένα, ανεξάρτητα από το επίπεδο διαθεσιμότητα σύνδεσης από άκρο σε άκρο (end-to-end). Με έμφαση στις τοπικές επικοινωνίες και υπηρεσίες στα άκρα του δικτύου, βελτιώνεται η ποιότητα της εμπειρίας του χρήστη (QoE) και μετριάζεται η συμφόρηση στο κέντρο του δικτύου.

Πρώτος στόχος της διπλωματικής αυτής εργασίας αποτέλεσε η δημιουργία ενός containerized περιβάλλοντος προσομοίωσης δικτύων που συνδυάζει τις τεχνολογίες NDN και DTN. Αυτό πραγματοποιήθηκε με τη χρήση της τεχνολογίας Docker και τη δημιουργία ενός container το οποίο περιλαμβάνει τον εξομοιωτή δικτύων CORE αλλά και όλα τα απαραίτητα εργαλεία για την προσομοίωση δικτύων NDN-DTN.

Στη συνέχεια έρχεται ο κύριος στόχος της παρούσας διπλωματικής που είναι η μελέτη της συμπεριφοράς δικτύων που συνδυάζουν τις τεχνολογίες NDN και DTN με τη χρήση διαφορετικών πρωτοκόλλων δρομολόγησης (Routing Protocols). Πιο συγκεκριμένα, θα μελετηθεί η επίδραση διαφόρων παραμέτρων, στην απόδοση ενός δικτύου με ασύρματους κινητούς κόμβους, που παρουσιάζουν διακοπόμενες συνδέσεις. Οι κόμβοι κινούνται βάσει γνωστών μοντέλων κινητικότητας που βασίζονται στην τυχαιότητα και τα πρωτόκολλα δρομολόγησης που εξετάζουμε είναι τα Flooding, Epidemic και ProPHET.

Το κύριο συμπέρασμα για τα πρωτόκολλα δρομολόγησης είναι ότι τα πρωτόκολλα Flooding και Epidemic παρουσιάζουν σε όλες τις περιπτώσεις κοινή συμπεριφορά, η οποία είναι καλύτερη της συμπεριφοράς του πρωτοκόλλου ProPHET. Πιο συγκεκριμένα, παρουσιάζουν σημαντικά μικρότερο μέσο χρόνο καθυστέρησης λήψης των Δεδομένων (Average Delay), ανεξάρτητα από το μοντέλο κινητικότητας και τον αριθμό των ενδιάμεσων κόμβων που χρησιμοποιείται στο πείραμα. Επίσης το ProPHET φαίνεται να παρουσιάζει καλύτερη συμπεριφορά σε δίκτυα με λιγότερους κόμβους απ' ότι σε δίκτυα με περισσότερους. Τέλος ακόμη και σχετικά μικρά ποσοστά απώλειας πακέτων, έχουν πολύ μεγάλη επίδραση στο μέσο χρόνο καθυστέρησης λήψης των Δεδομένων (Average Delay).

Η εργασία παρουσιάζει την ακόλουθη διάρθρωση:

Στο κεφάλαιο 2 θα γίνει η βιβλιογραφική ανασκόπηση μελλοντικών αρχιτεκτονικών Διαδικτύου (DTN, NDN και UMOBILE), οι οποίες χρησιμοποιήθηκαν στα πειράματά μας.

Στο κεφάλαιο 3 θα γίνει ανάλυση της λειτουργίας των Docker container, του εξομοιωτή CORE, αλλά και των εργαλείων που χρησιμοποιήθηκαν μέσα στο container που στήθηκε για την εκπόνηση των πειραμάτων.

Στο κεφάλαιο 4 θα γίνει περιγραφή της τοπολογίας και των παραμέτρων του δικτύου που χρησιμοποιήθηκαν στα πειράματα. Επίσης θα γίνει ανάλυση της μεθοδολογία που ακολουθήθηκε για την εκτέλεση τους.

Καταλήγοντας, στο κεφάλαιο 5 γίνεται παρουσίαση και σχολιασμός των αποτελεσμάτων, συνοψίζονται τα συμπεράσματα της εργασίας αυτής και αναφέρονται πιθανές μελλοντικές επεκτάσεις της.

## 2. Βιβλιογραφική Ανασκόπηση

### 2.1. Αρχιτεκτονικές Μελλοντικού Διαδικτύου

Η αρχιτεκτονική DTN (Delay Tolerant Networking)[5] έχει ως στόχο την παροχή διαλειτουργικής επικοινωνίας σ' ένα μεγάλο εύρος δικτύων που μπορεί να έχουν κακή απόδοση και ανομοιογενή χαρακτηριστικά. Ο σχεδιασμός της αρχιτεκτονικής DTN περιλαμβάνει τις έννοιες της ανταλλαγής μηνυμάτων με αποθήκευση εντός του δικτύου και αναμετάδοση, της δυναμικής δέσμευσης ονομάτων και της δρομολόγησης που είναι ανεκτική στις τμηματοποιήσεις του δικτύου σε μικρότερα υποδίκτυα. Ο σχεδιασμός αυτός υπάρχει ώστε να κατασκευαστεί ένα σύστημα το οποίο ταιριάζει σε διεργασίες που πραγματοποιούνται σε απαιτητικά περιβάλλοντα περισσότερο απ' ό,τι οι ήδη υπάρχουσες αρχιτεκτονικές δικτύου.

Στη συνέχεια έχουμε την αρχιτεκτονική NDN[2], η οποία ανήκει στην κατηγορία των αρχιτεκτονικών Πληροφοριοκεντρικών δικτύων (Information Centric Networking)[3] και η λογική που ακολουθεί, διαφέρει από του σημερινού Διαδικτύου. Λαμβάνοντας υπόψη τις εξελίξεις στη χρήση του Διαδικτύου τα τελευταία τριάντα χρόνια, και συγκεκριμένα την κυρίαρχη και αδιάκοπα αυξανόμενη έμφαση στο περιεχόμενο, η βασική αρχή λειτουργίας της είναι η ανάκτηση των δεδομένων τα οποία προσδιορίζονται από ένα δεδομένο όνομα και όχι η παράδοση του πακέτου σε μια συγκεκριμένη διεύθυνση προορισμού. Βασιζόμενη στις ήδη γνωστές αδυναμίες του TCP/IP ενσωματώνει έννοιες όπως τον ονοματοδοσία του περιεχομένου, προσαρμοστικές δυνατότητες δρομολόγησης και προώθησης (δρομολόγηση βάσει ονόματος και προσωρινή αποθήκευση εντός του δικτύου) και τον κρυπτογραφικό έλεγχο ταυτότητας κάθε πακέτου. Αυτή η δομή λοιπόν επιτρέπει στα δίκτυα NDN αρχιτεκτονικής να επιλύουν ένα μεγαλύτερο φάσμα προβλημάτων και ίσως με κάποιες τροποποιήσεις να εξυπηρετούν ιδανικά νεοεισαχθέντες κλάδους στο Διαδίκτυο, όπως αυτός του IoT.

Το χάσμα αυτό έρχεται να γεφυρώσει το UMOBILE[4] το οποίο αποτελεί μία συνδυασμένη προσέγγιση των δυο παραπάνω αρχιτεκτονικών, αξιοποιώντας τις λειτουργίες τους και βελτιώνοντας τις. Αντλώντας από ποικίλες εξελίξεις στο ICN, το DTN και την ευκαιριακή δικτύωση (opportunistic networking), έχουμε μια νέα αρχιτεκτονική που έχει σχεδιαστεί για τα άκρα του δικτύου και αξιοποιεί τους χρήστες και τις υπάρχουσες συνθήκες για να βελτιώσει την αποτελεσματικότητά του. Έχει ως σκοπό την αξιόπιστη παράδοση πληροφοριών αλλά και υπηρεσιών ακόμα και σε απομονωμένες περιοχές που παρουσιάζουν διακοπτόμενες συνδέσεις.

#### 2.1.1. Delay Tolerant Networking

Το Delay Tolerant Networking (DTN)[5] είναι μια αρχιτεκτονική δικτύου υπολογιστών, η οποία αναπτύχθηκε με στόχο να επιτευχθεί η διαλειτουργικότητα και ν' αντιμετωπιστούν τα τεχνικά ζητήματα σε ετερογενή δίκτυα, τα οποία ενδέχεται να μην έχουν συνεχή συνδεσιμότητα στο δίκτυο. Με τον όρο ετερογενή δίκτυα αναφερόμαστε σε δίκτυα που συνδέουν υπολογιστές και άλλες συσκευές με διαφορετικά μεταξύ τους λειτουργικά συστήματα ή/και πρωτόκολλα. Παραδείγματα τέτοιων δικτύων είναι αυτά που λειτουργούν

σε κινητά ή απομονωμένα χερσαία περιβάλλοντα, δίκτυα στο διάστημα ή δίκτυα αισθητήρων/ενεργοποιητών όπως τα δίκτυα IoT.

Η αρχιτεκτονική DTN είναι φτιαγμένη για να λειτουργεί πάνω από τις ήδη υπάρχουσες στοίβες πρωτοκόλλων, σε διάφορες αρχιτεκτονικές Διαδικτύου, αποτελεί δηλαδή μία αρχιτεκτονική τύπου “overlay”.

Παρέχει μια λειτουργία πυλών, αυτή της αποθήκευσης και προώθησης (store-and-forward) μεταξύ των ετερογενών δικτύων, όταν ένας κόμβος βρίσκεται, από φυσικής άποψης, σε δύο ή περισσότερα δίκτυα. Παραδείγματος χάριν, στο Διαδίκτυο μπορεί να λειτουργήσει πάνω από το TCP/IP, για συνδέσεις με το διάστημα μπορεί να παρέχει μια υπηρεσία πύλης στο CFDP[6] και σε Delay Tolerant δίκτυα αισθητήρων/ενεργοποιητών μπορεί να παρέχει διασύνδεση με κάποιο πρωτόκολλο μεταφοράς αισθητήρων, το οποίο δεν είναι ακόμα τυποποιημένο.

Η DTN αρχιτεκτονική βασίζεται στην αφαιρετικότητα των μηνυμάτων μεταγωγής. Το σύνολο των μηνυμάτων του DTN είναι γνωστό ως “bundle” και ο όρος υιοθετήθηκε από [1]. Το πρωτόκολλο bundle λειτουργεί προωθώντας τα πακέτα πάνω από το πρωτόκολλο IP, όμως τα εξερχόμενα πακέτα (bundles) αποθηκεύονται τοπικά στη μνήμη του εκάστοτε κόμβου, έως ότου εμφανιστούν διαθέσιμοι σύνδεσμοι προώθησης.

## Bundle Protocol

Το πρωτόκολλο Bundle[7] είναι ένα πρωτόκολλο “overlay” που μπορεί να συνδέσει ετερογενή δίκτυα. Βασίζεται στην αποθήκευση και προώθηση των πακέτων, δηλαδή τα πακέτα αποθηκεύονται όταν δεν υπάρχει σύνδεση στο δίκτυο και στη συνέχεια μεταδίδονται μόλις αποκατασταθεί η συνδεσιμότητα. Έτσι μετριάζονται οι καθυστερήσεις και οι διακοπές που παρατηρούνται, παραδείγματος χάριν σε ένα δίκτυο στο διάστημα, αφού τα πακέτα αποθηκεύονται σε μεγάλο χρονικό διάστημα έως ότου να είναι διαθέσιμη μια ευκαιρία επικοινωνίας με έναν άλλον κόμβο. Τα δεδομένα μπορούν να μεταδοθούν σε έναν προορισμό χωρίς να είναι εκ των προτέρων γνωστή η διαδρομή η οποία θα πρέπει να ακολουθηθεί από άκρο σε άκρο, με αναμετάδοση σε κόμβους που μπορεί τελικά να έρθουν σε επαφή με τον κόμβο προορισμού.

Επιπλέον, αυτή η έννοια του “overlay” παραβλέπει τις διαφορές στις λεπτομέρειες που υπάρχουν στα χαμηλότερα επίπεδα, έτσι ώστε οι κόμβοι να μπορούν να χρησιμοποιούν μια ποικιλία διαφορετικών πρωτοκόλλων από το επίπεδο της μεταφοράς έως και τα φυσικά επίπεδα, για όποιες ιδιαιτερότητες μπορεί να έχουν, ανάλογα με την περίπτωση.

Για παράδειγμα, κόμβοι που βρίσκονται σε κοντινή απόσταση μεταξύ τους, στην επιφάνεια ενός πλανήτη, μπορεί να χρησιμοποιούν ένα τυπικό δίκτυο που βασίζεται σε πρωτόκολλο TCP. Σύνδεσμοι μακρινών αποστάσεων όπως μεταξύ του Άρη και της Γης, μπορεί να χρησιμοποιούν πρωτόκολλα που έχουν σχεδιαστεί για μεγάλες αποστάσεις, όπως το πρωτόκολλο μετάδοσης Licklider (LTP)[8]. Το Bundle πρωτόκολλο, παραβλέπει αυτές τις διαφορές, οι οποίες θα αντιμετωπιστούν από μηχανισμούς σε χαμηλότερα επίπεδα. Το επίπεδο του Bundle, θα ασχοληθεί με την αποθήκευση των δεδομένων έως ότου αυτά μεταδοθούν, τη μεταφορά των δεδομένων και τη δρομολόγηση των δεδομένων.

Όπως αναφέρθηκε νωρίτερα, με τον όρο bundle δεν αναφερόμαστε μόνο στο πρωτόκολλο που χρησιμοποιεί το DTN αλλά bundle ονομάζεται και το κάθε πακέτο που αποστέλλεται με το πρωτόκολλο αυτό. Το κάθε πακέτο (bundle) αποτελείται από τουλάχιστον δύο μπλοκ, ένα



κύριο μπλοκ και ένα μπλοκ ωφέλιμου φορτίου(payload). Το πρώτο σε σειρά μπλοκ, είναι το κύριο μπλοκ το οποίο περιέχει πληροφορίες για την πηγή του bundle, τον τελικό προορισμό του, το πότε δημιουργήθηκε και άλλες πληροφορίες σχετικά με την επεξεργασία του . Το μπλοκ ωφέλιμου φορτίου περιέχει τα πραγματικά δεδομένα που αποστέλλονται μέσα στο bundle.

| Version                            | Proc. Flags          |
|------------------------------------|----------------------|
| Block Length                       |                      |
| Dest. Scheme Offset                | Dest. SSP offset     |
| Source Scheme Offset               | Source SSP offset    |
| Report-to Scheme Offset            | Report-to SSP offset |
| Custodian Scheme Offset            | Custodian SSP offset |
| Creation Timestamp Time            |                      |
| Creation Timestamp Sequence Number |                      |
| Lifetime                           |                      |
| Dictionary Length                  |                      |
| Dictionary Byte Array              |                      |
| Fragment Offset                    |                      |
| Total Application Data Unit Length |                      |

Εικόνα 1: Κύριο μπλοκ του πρωτοκόλλου Bundle[9].

| Block Type     | Proc. Flags | Block Length |
|----------------|-------------|--------------|
| Bundle Payload |             |              |

Εικόνα 2:2.Μπλοκ ωφέλιμου φορτίου του πρωτοκόλλου Bundle[9].

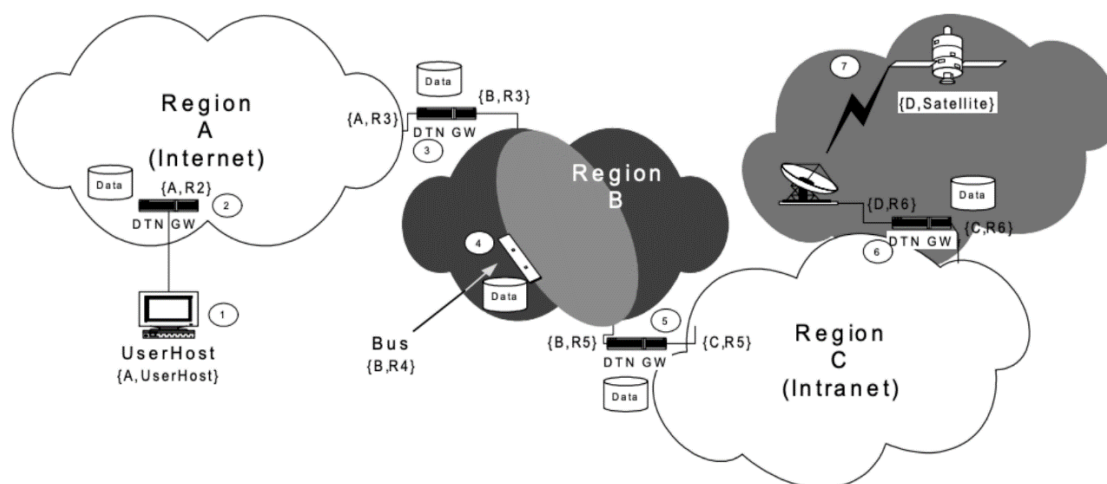
Οι προορισμοί των πακέτων εκφράζονται με ονόματα και όχι με διευθύνσεις, καθώς η θέση του κόμβου προορισμού μπορεί να αλλάξει ενώ τα δεδομένα βρίσκονται εν κινήσει. Οι δρομολογητές στο DTN ονομάζονται “bundle forwarders” ή πύλες DTN. Πύλες DTN βρίσκονται και στα σημεία διασύνδεσης μεταξύ ετερογενών δικτύων, όπως αυτά που αναφέρθηκαν παραπάνω, για τα οποία έχουν αναπτυχθεί εξειδικευμένες στοίβες πρωτοκόλλων και ειδική σημασιολογία ονομάτων που αντικατοπτρίζουν τον τομέα εφαρμογής τους. Μέσω των πυλών αυτών επιτυγχάνεται η διαλειτουργικότητα.

## Περιοχές και πύλες του DTN

Η αρχιτεκτονική DTN περιλαμβάνει τις έννοιες των περιοχών και των πυλών DTN, όπως αυτές φαίνονται στην Εικόνα 3. Σε αυτό το παράδειγμα απεικονίζονται τέσσερις περιοχές (A, B, C, D). Η Περιοχή B περιλαμβάνει μια πύλη DTN, η οποία έχει τη μορφή λεωφορείου το οποίο κινείται επαναλαμβανόμενα μεταξύ των πυλών DTN 3 και DTN 5 και μεταφέρει δεδομένα. Να σημειωθεί ότι η κίνηση του λεωφορείου αυτού δεν είναι απαραίτητα περιοδική διότι υπάρχει η πιθανότητα να υπόκειται σε κυκλοφοριακή συμφόρηση ή κάποιου άλλου είδους

καθυστερήσεις. Η Περιοχή D περιλαμβάνει ένα δορυφορικό σύνδεσμο low-earth-orbiting (LEO)[10] που παρέχει περιοδική σύνδεση με τη περιοχή C .

Τα όρια των περιοχών χρησιμοποιούνται ως σημεία διασύνδεσης μεταξύ των διαφορετικών πρωτοκόλλων που υπάρχουν σε κάθε περιοχή. Ουσιαστικά , δύο κόμβοι βρίσκονται στην ίδια περιοχή εάν μπορούν να επικοινωνήσουν μεταξύ τους χωρίς τη χρήση πυλών DTN, χρησιμοποιώντας απλά τα τοπικά πρωτόκολλα της περιοχής που ανήκουν. Γενικότερα αναμένεται ένας σχετικά μικρός αριθμός διαφορετικών περιοχών όπου την κάθε μία θα εφαρμόζεται μία όμοια στοίβα πρωτοκόλλων .



Εικόνα 3: Διασύνδεση ετερογενών περιοχών μέσω των πυλών DTN[5].

Οι DTN πύλες αντιστοιχούν τόσο στην έννοια "waypoint" του Metanet στο[11]. Η έννοια του "waypoint" περιγράφει ένα σημείο από το οποίο τα δεδομένα πρέπει να περάσουν ώστε ν' αποκτήσουν πρόσβαση σε μία περιοχή. Το σημείο αυτό ουσιαστικά εξυπηρετεί και για να μετατρέψει την κωδικοποίηση της μίας περιοχής ώστε να είναι κατανοητή από την άλλη αλλά και ως σημείο εφαρμογής ασφαλείας και ελέγχου.

Επίσης οι πύλες DTN είναι ανάλογες και της έννοιας των πυλών που περιγράφονται στην αρχική σχεδίαση ARPANET[12]. Μία πύλη DTN η οποία βρίσκεται μεταξύ δύο περιοχών αποτελείται από δύο "μισά", καθένα από τα οποία ανήκει σε μία από τις γειτονικές περιοχές και είναι πάνω από τα δικά τους τοπικά πρωτόκολλα μεταφοράς, αναλογικά με τις πύλες τύπου ARPANET. Ωστόσο, όταν λειτουργούν πάνω από το επίπεδο μεταφοράς, οι πύλες DTN διαφέρουν από τις πύλες ARPANET και είναι αντ' αυτού επικεντρωμένες στην αξιόπιστη δρομολόγηση μηνυμάτων και όχι στη best-effort[13] ανταλλαγή πακέτων, στην οποία δεν υπάρχει καμία εγγύηση για την ποιότητα των υπηρεσιών και την παράδοση των πακέτων.

Οι πύλες DTN είναι υπεύθυνες για την αποθήκευση μηνυμάτων σε μη πτητική μνήμη, όταν απαιτείται αξιόπιστη μεταφορά και αντιστοιχία καθολικών ονομάτων σε τοπικά ονόματα για τη μεταφορά μηνυμάτων που προορίζονται για γειτονικές περιοχές.

Τέλος έχουν τη δυνατότητα να πραγματοποιήσουν έλεγχο πρόσβασης και γνησιότητας στην εισερχόμενη ροή δεδομένων ώστε να εξασφαλίσουν ότι επιτρέπεται η προώθηση τους.

## Tuple Names

Για τη δρομολόγηση μηνυμάτων DTN, χρησιμοποιούνται αναγνωριστικά για τα αντικείμενα ή τις ομάδες αντικειμένων τα οποία ονομάζονται tuple names και αποτελούνται από δύο μεταβλητά μέρη. Στην Εικόνα 3, τα tuple names για κάθε τερματική συσκευή (end-point) και για κάθε "μισό" του δρομολογητή παρουσιάζονται μέσα σε αγκύλες με τη μορφή {Region

Name, Entity Name}. Το πρώτο μέρος περιγράφει ένα ιεραρχικά δομημένο όνομα μιας περιοχής. Χρησιμοποιείται από τις πύλες DTN για την εύρεση της διαδρομής προς μία ή περισσότερες πύλες DTN στα άκρα της εκάστοτε περιοχής. Τα ονόματα των περιοχών συμπληρώνονται στους πίνακες προώθησης είτε στατικά (από έναν διαχειριστή δικτύου), είτε από ένα ή περισσότερα δυναμικά πρωτόκολλα προώθησης στο επίπεδο του DTN .

Οι.

Το γεγονός ότι τα ονόματα των περιοχών δομούνται ιεραρχικά, παρέχει τη δυνατότητα μείωσης του μεγέθους των πινάκων προώθησης του DTN, με τρόπο παρόμοιο με τη συγχώνευση διαδρομών του Διαδικτύου όπως στο CIDR[14]. Είναι αξιοσημείωτο ότι παρόλο που τα ονόματα των περιοχών έχουν παρόμοια μορφή με τα ονόματα στο DNS [15], δεν είναι απαραίτητο να μεταφράζονται σε οποιοδήποτε μορφή διεύθυνσης ή να κατανέμονται ιεραρχικά όπως τα ονόματα στο DNS.

Το δεύτερο μεταβλητό μέρος των tuple names, προσδιορίζει ένα όνομα το οποίο είναι καθορισμένο εντός της συγκεκριμένης περιοχής , αλλά δε χρειάζεται να είναι μοναδικό εκτός της περιοχής αυτής. Το όνομα αυτό δεν είναι απαραίτητο να έχει συγκεκριμένη δομή και μπορεί να περιέχει ειδικά σύμβολα τα οποία είναι καθορισμένα μόνο στην περιοχή προέλευσης ή προορισμού. Για παράδειγμα στην περίπτωση του Διαδικτύου θα μπορούσαμε να έχουμε το ακόλουθο tuple:

{internet.icann.int, "http://www.ietf.org/oview.html"}

Το tuple αυτό αναφέρεται στην περιοχή του Διαδικτύου, μαζί μ' ένα τοπικό αναγνωριστικό, στην περίπτωση μας σε ένα Universal Resource Identifier(URI) [16]. Καθώς ένα μήνυμα μεταφέρεται δια μέσου περιοχών που πιθανώς παρουσιάζουν ετερογένεια, μόνο το πρώτο μέρος του αναγνωριστικού χρησιμοποιείται για τη δρομολόγηση του. Όταν τελικά φτάσει στα σύνορα της περιοχής προορισμού, το όνομα της οντότητας μεταφράζεται σε τοπικό επίπεδο, σε ένα όνομα ή διεύθυνση κατάλληλη για την συγκεκριμένη περιοχή. Αυτή η μέθοδος επιλογής ονομάτων οδηγεί σε μια μορφή late binding[17] για τα tuples, κατά την οποία, μόνο το πρώτο μέρος του tuple name χρησιμοποιείται από τις πύλες DTN

Η επιλογή υιοθέτησης ονομάτων και όχι διευθύνσεων ως βάση για τον χαρακτηρισμό των συμμετεχόντων στη δρομολόγηση από άκρη σε άκρη, προκύπτει από την παρατήρηση των σύγχρονων τάσεων στη λειτουργία του Διαδικτύου. Οι διευθύνσεις

χρησιμοποιούνται για τη δρομολόγηση και για την αναφορά σε έναν υπολογιστικό πόρο (π.χ. end-point server) και η ονοματοθεσία προστέθηκε έτσι ώστε να γίνει η διευθυνσιοδότηση ευκολότερη για τους ανθρώπους. Το Διαδίκτυο σήμερα περιέχει αντικείμενα, όπως μηχανές αναζήτησης τα οποία είναι ευρέως χρησιμοποιούμενα. Σε πολλές περιπτώσεις, ένα όνομα (με τη μορφή μιας διεύθυνσης URI ή μιας διεύθυνσης ιστού URL) αναφέρεται πιο αποτελεσματικά σε μία σειρά δεδομένων παρά ως αναγνωριστικό ενός συγκεκριμένου υπολογιστικού συστήματος που τα παρέχει.

## Επιλογή Διαδρομής και Προγραμματισμός

Η αρχιτεκτονική DTN στοχεύει σε δίκτυα όπου δεν μπορεί να θεωρηθεί ότι υπάρχει διαδρομή δρομολόγησης από άκρο σε άκρο. Περισσότερο θα λέγαμε ότι οι διαδρομές αποτελούνται από έναν σύνολο εξαρτώμενων από το χρόνο επαφών(ευκαιρίες επικοινωνίας) που χρησιμοποιούνται για τη μετακίνηση μηνυμάτων από τον κόμβο προέλευσης ως τον τελικό προορισμό τους. Οι επαφές αυτές παραμετροποιούνται από τους χρόνους έναρξης και λήξης τους (σε σχέση με την πηγή), τη χωρητικότητα, την καθυστέρηση, τις τερματικές συσκευές και την κατεύθυνση. Επιπλέον, ένα μέτρο της προβλεψιμότητας μιας επαφής μπορεί να

βοηθήσει στο να επιλεγούν οι επόμενοι κόμβοι για τη δρομολόγηση του μηνύματος, καθώς και να επιλεγεί το επόμενο μήνυμα προς αποστολή.

Η προβλεψιμότητα μίας διαδρομής κυμαίνεται σε ένα συνεχές φάσμα, από εντελώς προβλέψιμη π.χ. ενσύρματη σύνδεση ή περιοδική σύνδεση των οποίων η φάση και η συχνότητα είναι γνωστά έως πλήρως απρόβλεπτη μια "ευκαιριακή" επαφή στην οποία ένα κινητός δρομολογητής μηνυμάτων έχει έρθει στο εύρος επικοινωνίας ενός άλλου κόμβου DTN. Στην παρούσα διπλωματική, θα εξετάσουμε την περίπτωση των απρόβλεπτων επαφών των κόμβων.

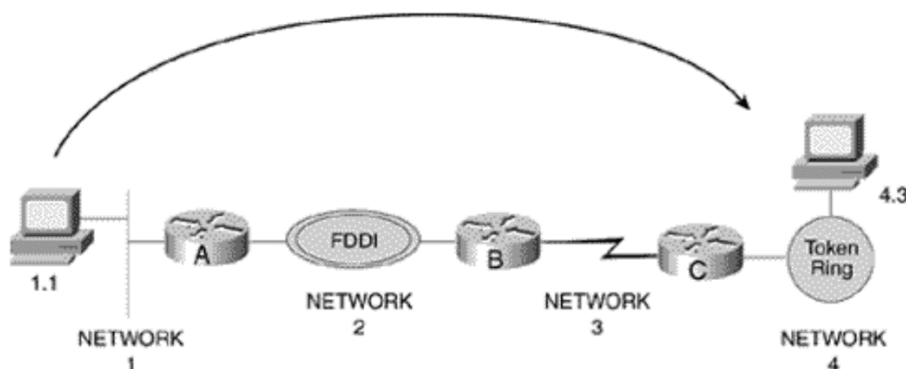
Οι συγκεκριμένες λεπτομέρειες για την επιλογή διαδρομής και τον προγραμματισμό μηνυμάτων αναμένεται να επηρεαστούν σε μεγάλο βαθμό από τα πρωτόκολλα δρομολόγησης και τους αλγόριθμους της κάθε περιοχής. Στο πρώιμο στάδιο της ανάπτυξης του DTN, υπήρξαν αρκετά προβλήματα που αποτέλεσαν πρόκληση όπως:

- ο προσδιορισμός της ύπαρξης, αλλά και της προβλεψιμότητας των επαφών
- η απόκτηση πληροφοριών για την κατάσταση των μηνυμάτων που βρίσκονται σε εκκρεμότητα, δεδομένης και μιας μεγάλης καθυστέρησης
- η αποτελεσματική ανάθεση των μηνυμάτων στις επαφές και ο προσδιορισμός της σειράς μετάδοσης τους.

Ενώ πολύ απλές μέθοδοι επίλυσης αυτών των προβλημάτων μπορούν να εφαρμοστούν χωρίς εκτεταμένα προβλήματα, κάθε θέμα αντιπροσωπεύει μία σημαντική πρόκληση και παραμένει μελλοντικό πεδίο έρευνας.

## Πρωτόκολλα Δρομολόγησης στην αρχιτεκτονική DTN

Η δρομολόγηση είναι μια θεμελιώδης ικανότητα των δικτύων υπολογιστών[18]. Με τον όρο πρωτόκολλο δρομολόγησης (routing protocol) αναφερόμαστε στη διαδικασία που χρησιμοποιείται από τους δρομολογητές (routers) ώστε να προσδιορίσουν και να επιλέξουν τη βέλτιστη διαδρομή, μέσω της οποίας θα στείλουν τα δεδομένα. Όπως ένας σύνδεσμος μετάδοσης δεδομένων μπορεί να συνδέσει απευθείας δύο συσκευές, ένας δρομολογητής δημιουργεί επίσης μια σύνδεση μεταξύ δύο συσκευών. Η διαφορά είναι ότι, όπως δείχνει η Εικόνα 4, ενώ η διαδρομή επικοινωνίας μεταξύ δύο συσκευών που μοιράζονται έναν κοινό σύνδεσμο δεδομένων είναι μια φυσική διαδρομή, η διαδρομή επικοινωνίας που παρέχεται από τους δρομολογητές μεταξύ δύο συσκευών σε διαφορετικά δίκτυα, είναι μια λογική διαδρομή υψηλότερου επιπέδου.



Εικόνα 4: Ένας δρομολογητής, δημιουργεί μία λογική διαδρομή μεταξύ των δικτύων[18].

Στην παρούσα διπλωματική τα πρωτόκολλα δρομολόγησης που χρησιμοποιήθηκαν για την εκπόνηση των πειραμάτων είναι τα ακόλουθα τρία.

### **Δρομολόγηση πλημμύρας (Flooding Routing)**

Στο πρωτόκολλο δρομολόγησης πλημμύρας[18], όταν ένα πακέτο φτάνει σ' έναν κόμβο, ο κόμβος αυτός το προωθεί σε όλους τους γειτονικούς του κόμβους, εκτός από τον κόμβο από τον οποίο το έλαβε. Μια δυσκολία στο πρωτόκολλο δρομολόγησης πλημμύρας, είναι πως όταν όλοι οι δρομολογητές έχουν λάβει το πακέτο, η προώθηση του πρέπει να σταματήσει γιατί εάν συνεχιστεί, δημιουργούνται άπειρα αντίγραφα του πακέτου και έχουμε πλημμύρα. Ένα πιθανό μέτρο για την ανακοπή της πλημμύρας είναι να περιέχεται ένας μετρητής αλμάτων στην κεφαλίδα κάθε πακέτου IP. Πρέπει όμως ο μετρητής αλμάτων να μην έχει τιμή μικρότερη από το πλήθος των αλμάτων που χρειάζονται για να φτάσει το πακέτο από την πηγή στον προορισμό. Για κάθε άλμα, ο μετρητής μειώνεται κατά ένα. Όταν μηδενιστεί, το πακέτο δεν θα αναμεταδοθεί, αλλά θα απορριφθεί.

### **Επιδημική δρομολόγηση(Epidemic Routing)**

Η επιδημική δρομολόγηση[19] βασίζεται στη δρομολόγηση πλημμύρας, καθώς ο κάθε κόμβος, αντιγράφει και προωθεί το πακέτο σε όποιον γειτονικό κόμβο ανακαλύψει και δεν έχει ήδη ένα αντίγραφο του πακέτου. Φυσικά στην πιο απλή περίπτωση, η δρομολόγηση επιδημίας πλημμυρίζει. Ωστόσο, πιο εξελιγμένες τεχνικές μπορούν να χρησιμοποιηθούν για τον περιορισμό του αριθμού των πακέτων που προωθούνται.

Η επιδημική δρομολόγηση έχει ως στόχο να μεγιστοποιήσει το ποσοστό παράδοσης πακέτων και να ελαχιστοποιήσει της καθυστέρησης τους, ελαχιστοποιώντας επίσης τους συνολικούς πόρους (π.χ. μνήμη και εύρος ζώνης δικτύου) που καταναλώνονται για την μεταφορά των πακέτων.

### **Πρωτόκολλο δρομολόγησης PRoPHET (Probabilistic Routing Protocol using History of Encounters and Transitivity)**

Το πρωτόκολλο δρομολόγησης PRoPHET (Πιθανολογικό Πρωτόκολλο Δρομολόγησης που χρησιμοποιεί το ιστορικό των Συναντήσεων και τη Μεταβατικότητα) προσπαθεί να μειώσει τον αριθμό των αντιγράφων των πακέτων στο δίκτυο, υπολογίζοντας την πιθανότητα επιτυχούς παράδοσης του πακέτου σε έναν δεδομένο προορισμό. Το PRoPHET βασίζεται στο μοντέλο της ανθρώπινης κινητικότητας και στην παρατήρηση ότι ένας μεγάλος αριθμός ευκαιριών επαφής μεταξύ δύο κόμβων, ακολουθούν ένα μη τυχαίο μοτίβο[20]. Τα πακέτα αντιγράφονται και αποστέλλονται σε γειτονικούς κόμβους που έχουν μεγάλη πιθανότητα να τα παραδώσουν στον προορισμό τους. Το PRoPHET καθορίζει αυτήν την πιθανότητα βασιζόμενο σε μια μετρική προβλεψιμότητας της παράδοσης του πακέτου. Κάθε κόμβος διατηρεί ένα διάνυσμα προβλεψιμότητας παράδοσης για όλους τους κόμβους που συναντήθηκαν και ανταλλάσσει αυτές τις πληροφορίες με άλλους κόμβους κατά τη διάρκεια μιας αρχικής φάσης επαφής. Η προβλεψιμότητα παράδοσης υπολογίζεται κάθε φορά που δύο κόμβοι έρχονται σε επαφή. Οι κόμβοι που βρίσκονται συχνότερα σε επαφή έχουν υψηλότερη προβλεψιμότητα παράδοσης και ως εκ τούτου ο αλγόριθμος θα επιλέξει αυτό το ζεύγος κόμβων ως προτιμώμενη διαδρομή.

### **Μεταφορά επιμέλειας και Αξιοπιστία**

Η μεταφορά επιμέλειας (Custody Transfer) αναφέρεται στην επιβεβαιωμένη παράδοση ενός μηνύματος από ένα DTN κόμβο στο επόμενο και μαζί με το μήνυμα αυτό, περνιέται και στον κόμβο αυτόν η ευθύνη της αξιόπιστης παράδοσης του στον τελικό προορισμό. Θα μπορούσε



να παρομοιαστεί με την εκχώρηση ευθύνης για την παράδοση ταχυδρομικής αλληλογραφίας σε ένα άτομο ή υπηρεσία που δεσμεύεται να το πράξει.

Η έννοια της μεταφοράς επιμέλειας είναι θεμελιώδης για την αρχιτεκτονική DTN προκειμένου να καταπολεμηθούν τα πιθανά υψηλά ποσοστά απώλειας και να ανακουφιστούν οι τελικοί κόμβοι που μπορεί να είναι ανεπαρκείς σε πόρους από ευθύνες που σχετίζονται με τη διατήρηση της συνεχούς σύνδεσης από άκρο σε άκρο.

Η αρχιτεκτονική DTN περιλαμβάνει δύο διακριτούς τύπους κόμβων δρομολόγησης, τους πτητικούς (persistent-P) και τους μη-πτητικούς (non-persistent-NP). Οι πτητικοί κόμβοι περιέχουν μη αμελητέες ποσότητες αποθηκευμένων μηνυμάτων ενώ οι μη-πτητικοί μπορεί να μην περιέχουν. Οι κόμβοι τύπου P εκτός από τις περιπτώσεις που είναι ανίκανοι ή απρόθυμοι να αποθηκεύσουν ένα συγκεκριμένο μήνυμα, γενικά συμμετέχουν στη μεταφορά επιμέλειας χρησιμοποιώντας τα κατάλληλα πρωτόκολλα μεταφοράς της περιοχής που βρίσκονται.

## Ασφάλεια

Οι απαιτήσεις ασφαλείας για την αρχιτεκτονική DTN διαφέρουν από τα παραδοσιακά μοντέλα ασφαλείας δικτύου λόγω του γεγονότος ότι το σύνολο των αρχών περιλαμβάνει τους ίδιους τους δρομολογητές (δηλαδή τις πύλες DTN) πέραν από τους κόμβους του δικτύου. Το DTN επικεντρώνεται στην επαληθευμένη πρόσβαση στο φορτίο της κίνησης μιας συγκεκριμένης κατηγορίας υπηρεσιών και στην αποφυγή μεταφοράς φορτίου κίνησης για ενδεχομένως μεγάλες αποστάσεις, στις οποίες μπορεί να διαπιστωθεί αργότερα ότι απαγορεύεται.

Για την εφαρμογή του μοντέλου ασφαλείας, κάθε μήνυμα περιλαμβάνει μία αμετάβλητη «ταχυδρομική σφραγίδα» η οποία περιέχει ένα εξακριβωμένο αναγνωριστικό του

αποστολέα, μία έγκριση από την αιτούσα κατηγορία υπηρεσιών (Class of Service - CoS) που σχετίζεται με το μήνυμα, και ένα ακόμα κρυπτογραφικό υλικό για την επιβεβαίωση της ορθότητας του περιεχομένου του μηνύματος. Οι δρομολογητές ελέγχουν τα διαπιστευτήρια σε κάθε DTN κόμβο, απορρίπτοντας την κίνηση όσο το δυνατόν γρηγορότερα, αν η επιβεβαίωση αποτύχει. Αυτή η προσέγγιση έχει το πλεονέκτημα ότι κάνει τις επιθέσεις DoS (denial-of-service attacks)[21] πολύ δυσκολότερες σε σχέση με τους συμβατικούς δρομολογητές του Διαδικτύου.

Η αντιμετώπιση των ζητημάτων ασφαλείας, αποτέλεσε βασικό στόχο του πρωτοκόλλου Bundle. Πιθανές επιθέσεις κάνουν έναν ή περισσότερους κόμβους του δικτύου να συμπεριφέρονται ως “μαύρη τρύπα” (black hole)[22] ή να δημιουργούν πλημμύρα (flooder)[23].

Οι ανησυχίες για την ασφάλεια των δικτύων DTN ποικίλλουν ανάλογα με το περιβάλλον και την εφαρμογή, αν και ο έλεγχος ταυτότητας και τα προσωπικά δεδομένα είναι συχνά κρίσιμης σημασίας. Αυτές οι εγγυήσεις ασφαλείας είναι δύσκολο να καθιερωθούν σε ένα δίκτυο χωρίς συνεχή συνδεσιμότητα επειδή το δίκτυο εμποδίζει περίπλοκα κρυπτογραφικά πρωτόκολλα και την ανταλλαγή κλειδιών κι επίσης κάθε συσκευή πρέπει να αναγνωρίζει άλλες συσκευές, οι οποίες είναι ορατές σε αυτήν κατά διαστήματα[24][25].

Οι επίσημες λύσεις από την ερευνητική κοινότητα του DTN περιλαμβάνουν

- τη χρήση κρυπτογράφησης που βασίζεται στην ταυτότητα (identity-based encryption), η οποία επιτρέπει στους κόμβους να λαμβάνουν πληροφορίες κρυπτογραφημένες με το δημόσιο αναγνωριστικό τους[26].
- τη χρήση πινάκων όπου φαίνεται αν κάτι έχει παραποιηθεί (tamper-evident tables) μαζί με ένα gossiping πρωτόκολλο[27].

## Συμφόρηση και Έλεγχος Ροής

Ως μια hop-by-hop αρχιτεκτονική, ο έλεγχος ροής και ο έλεγχος συμφόρησης για το DTN είναι στενά συνδεδεμένοι. Ο έλεγχος ροής αναφέρεται στη μείωση του ρυθμού αποστολής των μηνυμάτων από έναν DTN κόμβο στον επόμενο. Ο έλεγχος συμφόρησης αναφέρεται στη διαχείριση της διεκδίκησης αποθηκευτικού χώρου σε μια πύλη DTN.

Οι διαθέσιμοι μηχανισμοί για την αντιμετώπιση τέτοιων θεμάτων, κατηγοριοποιούνται ως προληπτικοί (proactive) ή αντιδραστικοί (reactive). Οι προληπτικές μέθοδοι περιλαμβάνουν κάποια μορφή ελέγχου της εισόδου για να αποφευχθεί η συμφόρηση εξ αρχής. Σε πολλές περιπτώσεις μια περιοχή μπορεί να βρίσκεται υπό το διαχειριστικό έλεγχο μιας οντότητας, και αυτή η προσέγγιση να είναι αποτελεσματική. Εάν οι προληπτικές μέθοδοι είναι ανεπαρκείς ή μη διαθέσιμες, πρέπει να χρησιμοποιηθούν αντιδραστικές μέθοδοι, που πιθανότατα περιλαμβάνουν άμεσο έλεγχο ροής και συνήθως οδηγούν σε μειωμένη απόδοση, όταν οι καθυστερήσεις είναι μεγάλες.

Στη συγκεκριμένη προσέγγιση, χρησιμοποιείται μια κοινόχρηστη ουρά προτεραιότητας για την κατανομή του αποθηκευτικού χώρου. Αρχικά, όλα τα μηνύματα που έχουν λήξει, διαγράφονται. Τα μηνύματα που καταφθάνουν και είναι πολύ μεγάλα, απορρίπτονται. Στη συνέχεια τα μηνύματα κατηγοριοποιούνται με βάση την προτεραιότητα και το χρόνο ζωής τους, ο οποίος καθορίζεται από τον αποστολέα και μεταφέρεται με κάθε μήνυμα.

Για την εφαρμογή του ελέγχου ροής, ένας δρομολογητής DTN θα προσπαθήσει να εκμεταλλευτεί οποιονδήποτε μηχανισμό ελέγχου ροής υπάρχει στα υποκείμενα πρωτόκολλα μεταφοράς κάθε περιοχής. Για τα περισσότερα δίκτυα, υπάρχουν ήδη κάποιοι τέτοιοι μηχανισμοί (π.χ. TCP, X.25, RTS / CTS, XON / XOFF, κ.λπ.). Για τα υπόλοιπα δίκτυα όμως, όπου εξακολουθούν να αναπτύσσονται τέτοιοι μηχανισμοί, μπορούν να κατασκευαστούν τοπικά ειδικοί μηχανισμοί, εξειδικευμένοι για την κάθε περιοχή στα επίπεδα convergence των δρομολογητών DTN. Σε κάθε περίπτωση, οι κύριες λειτουργίες ενός DTN δρομολογητή, βασίζονται στην υπόθεση της ύπαρξης ελέγχου ροής, συνεπώς κάποιος τέτοιος μηχανισμός θα πρέπει να υπάρχει για να διασφαλιστεί η αξιόπιστη παράδοση των μηνυμάτων.

### 2.1.2. Information Centric Networking

Το Διαδίκτυο, εξυπηρετεί καθημερινά δισεκατομμύρια χρήστες σε ολόκληρο τον κόσμο παρέχοντας τους τη δυνατότητα επικοινωνίας αλλά και πρόσβαση σε πληροφορίες και υπηρεσίες. Η συνεχώς αυξανόμενη ζήτηση για κλιμακούμενη και αποτελεσματική διανομή περιεχομένου στο Διαδίκτυο, έχει λειτουργήσει ως κίνητρο για την ανάπτυξη μελλοντικών αρχιτεκτονικών οι οποίες βασίζονται στην ονομασία των δεδομένων (Named Data Objects-NDOs) και όχι στις διευθύνσεις IP. Η βασική ιδέα είναι πως όταν υπάρχει ένα αίτημα για μία πληροφορία, αντί να χρησιμοποιηθεί η διεύθυνση στην οποία βρίσκεται αυτή η πληροφορία, θα γίνει αίτημα για το ίδιο το περιεχόμενο χρησιμοποιώντας μία περιγραφή του και το δίκτυο

θα είναι υπεύθυνο για την δρομολόγηση του αιτήματος, ώστε ο χρήστης να πάρει τελικά το αντίγραφο της πληροφορίας που επιθυμεί.

Η προσέγγιση αυτή τη αρχιτεκτονικής του Διαδικτύου ονομάζεται πληροφοριοκεντρικής δικτύωσης (Information Centric Networking-ICN)[3]. Η έννοια της πληροφοριοκεντρικής δικτύωσης, είναι μια αρκετά σημαντική κοινή προσέγγιση, πολλών μελλοντικών ερευνητικών δραστηριοτήτων, πάνω στο θέμα του Διαδικτύου. Ο στόχος είναι η παροχή μιας υπηρεσίας οργάνωσης δικτύου που ταιριάζει καλύτερα στη σημερινή χρήση του Διαδικτύου και πιο ανθεκτική σε διακοπές και σφάλματα.

Σ' ένα δίκτυο ICN το όνομα της πληροφορίας είναι ανεξάρτητο από την τοποθεσία στην οποία βρίσκεται, κι έτσι μπορεί να βρίσκεται οπουδήποτε στο δίκτυο. Επίσης όταν ικανοποιείται ένα αίτημα για μία πληροφορία, δικτυακές θέσεις προσωρινής αποθήκευσης κρατούν αντίγραφα είτε ολόκληρης είτε τμήματος της αιτηθείσας πληροφορίας τα οποία είναι πιθανό ν' αξιοποιηθούν για επόμενα αιτήματα. Η επικοινωνία μεταξύ του εκδότη μιας πληροφορίας και του παραλήπτη της, δεν είναι απαραίτητα συγχρονισμένη, καθιστώντας το μοντέλο αυτό ανεξάρτητο χώρου και χρόνου και διευκολύνοντας έτσι την κινητικότητα των κόμβων του. Παράλληλα ενισχύεται σ' ένα βαθμό η ασφάλεια διότι τα αιτήματα των ενδιαφερόμενων χρηστών μπορούν να αξιολογηθούν προτού φτάσουν στον τελικό τους προορισμό προσδίδοντας μάλιστα στον αιτούμενο ιδιωτικότητα καθώς ο πάροχος (εκδότης) της πληροφορίας δεν γνωρίζει την ταυτότητά του. Τα πλεονεκτήματα του ICN όσον αφορά την ταχεία και αποτελεσματική παράδοση δεδομένων και τη βελτιωμένη αξιοπιστία, το καθιστούν ένα πολλά υποσχόμενο μοντέλο δικτύωσης για εφαρμογές σε περιβάλλοντα IoT.

### 2.1.3. Named Data Networking

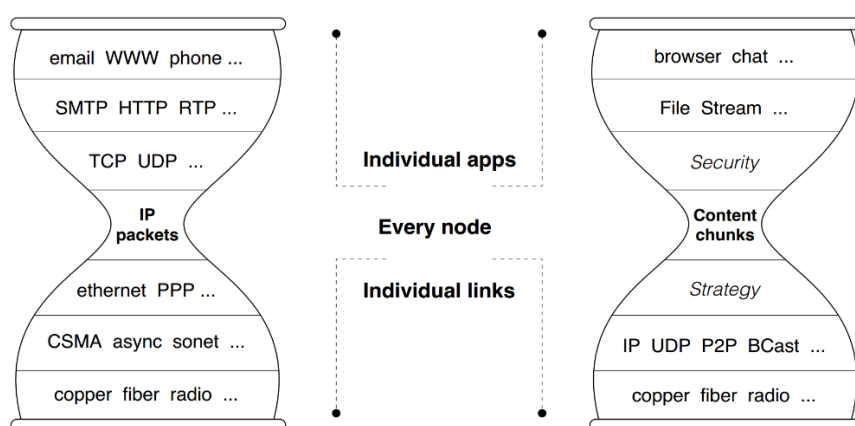
Ο σχεδιασμός μιας αρχιτεκτονικής δικτύων περιλαμβάνει δύο θεμελιώδεις αποφάσεις. Η πρώτη είναι ποιοι "χώροι ονομάτων" (namespaces) θα χρησιμοποιηθούν για τη μεταφορά των δεδομένων και η δεύτερη είναι ποιοι μηχανισμοί συγκεκριμένα θα χρησιμοποιηθούν για τη μεταφορά αυτή. Στην πρώτη απόφαση, οι επιλογές είναι δύο, είτε να ονομαστούν οι τοποθεσίες στις οποίες θ' αποσταλούν τα δεδομένα, είτε να ονομαστούν τα ίδια τα δεδομένα. Η δεύτερη απόφαση σχετικά με το σχεδιασμό μιας αρχιτεκτονικής, εξαρτάται από την πρώτη. Αν δοθούν ονόματα στις τοποθεσίες, τα bits των δεδομένων που θα σταλούν από τον κόμβο Α στον κόμβο Β μπορούν είτε να παραδοθούν από μία καθορισμένη διαδρομή μεταξύ των δύο σημείων επικοινωνίας (εικονικό κύκλωμα-virtual circuit), είτε να ταξιδέψουν μέσω του δικτύου ως ανεξάρτητα κομμάτια για να φτάσουν στον κόμβο Β (datagrams). Εάν κάποιος ονομάσει απευθείας τα ίδια τα bits δεδομένων αντί για την τοποθεσία προορισμού, τότε το δίκτυο θα μεταφέρει πακέτα δεδομένων που φέρουν ονόματα (named data packets) σ' αυτούς που τα ζήτησαν. Για την αρχιτεκτονική του πρωτοκόλλου TCP/IP που χρησιμοποιούμε σήμερα, έχει επιλεγεί η πρώτη επιλογή, δηλαδή η ονομασία της τοποθεσίας, ενώ αντίθετα για το Named Data Networking(NDN)[28] έχει γίνει η δεύτερη επιλογή, να ονομάζονται τα δεδομένα.

Υπάρχουν και άλλες θεμελιώδεις διαφορές μεταξύ των πρωτοκόλλων NDN και IP πέραν του προσανατολισμού στην πληροφορία και στην τοποθεσία αντίστοιχα. Παραδείγματος χάριν, το IP δεν κρατά την κατάσταση των εισερχόμενων δεδομένων στους δρομολογητές, το NDN περιλαμβάνει ένα πρωτόκολλο ασφαλείας απευθείας στη στενή μέση, όπου υπογράφεται κάθε πακέτο δεδομένων, χρησιμοποιούν διαφορετικό χώρο ονόματος (διεύθυνση IP και όνομα) και άλλα που θ' αναλυθούν στη συνέχεια. Βέβαια, υπάρχουν και ομοιότητες στο



σχεδιασμό και τη λειτουργία όπως ότι ακολουθούν το σχήμα κλεψύδρας με το επίπεδο IP / NDN να είναι η στενή μέση (Εικόνα 5), στέλνουν datagrams, ακολουθούν την αρχή από άκρο-σε-άκρο[27] και χρησιμοποιούν το δικό τους χώρο ονομάτων για την παράδοση των δεδομένων.

Το NDN αποτελεί λοιπόν μία αρχιτεκτονική που ανήκει στην κατηγορία ICN και προσπαθεί να εξελίξει τη σημερινή αρχιτεκτονική του ίντερνέτ που έχει ως βάση τη χρήση διευθύνσεων IP, σε μια αρχιτεκτονική δικτύου που έχει ως βάση το περιεχόμενο. Δε θα έπρεπε να υπάρξει η σύγχυση ότι NDN έχει εξάρτηση από το IP πρωτόκολλο, καθώς είναι μια εντελώς νέα αρχιτεκτονική, σχεδιασμένη από την αρχή και ο σχεδιασμός του αντικατοπτρίζει την κατανόηση των δυνατοτήτων και των περιορισμών της τρέχουσας αρχιτεκτονικής του Διαδικτύου.



Εικόνα 5: Η δομή της αρχιτεκτονικής IP (Αριστερά) σε αντιπαραβολή με τη δομή της αρχιτεκτονικής NDN (Δεξιά)[2].

## Ονόματα

Στην αρχιτεκτονική NDN, τα ονόματα είναι αδιαφανή για το δίκτυο κι αυτό συμβαίνει διότι ενώ οι δρομολογητές αναγνωρίζουν όρια μεταξύ των στοιχείων σε ένα όνομα, δεν αποδίδουν καμία σημασία στα ονόματα. Αυτή η απόφαση σχεδιασμού επιτρέπει σε κάθε εφαρμογή να επιλέξει το σχήμα ονοματοδοσίας που ταιριάζει στις ανάγκες της κι έτσι η ονομασία μπορεί να εξελιχθεί ανεξάρτητα από το δίκτυο.

Ο τρόπος με τον οποίο είναι σχεδιασμένο το NDN προϋποθέτει ιεραρχικά δομημένα ονόματα, π.χ. ένα βίντεο που παράγεται από το Δ.Π.Θ. μπορεί να έχει το όνομα /duth/videos/demo.mpg, όπου κάθε μέρος του ονόματος οριοθετείται από το χαρακτήρα '/', παρόμοια με τα URL. Αυτή η ιεραρχική δομή επιτρέπει στις εφαρμογές να αντιπροσωπεύουν το περιεχόμενο και τις σχέσεις μεταξύ των στοιχείων δεδομένων. Για παράδειγμα, το τμήμα 3 της πρώτης έκδοσης ενός demo βίντεο του Δ.Π.Θ. μπορεί να έχει το όνομα /ucla/videos/demo.mpg/version1/3.

Τα μη-ιεραρχικά ονόματα μπορούν να υπάρξουν σε ειδικές περιπτώσεις και πιθανώς να είναι χρήσιμα σε τοπικά περιβάλλοντα, ωστόσο, οι ιεραρχικοί χώροι ονομάτων είναι απαραίτητοι τόσο για την κλιμάκωση του συστήματος δρομολόγησης όσο και για την παροχή των απαραίτητων δεδομένων.

Για να ανακτηθούν δυναμικά παραγόμενα δεδομένα, οι καταναλωτές πρέπει να είναι σε θέση να καθορίσουν ντετερμινικά το όνομα για ένα επιθυμητό κομμάτι δεδομένων χωρίς να έχουν δει στο παρελθόν το όνομα ή τα δεδομένα. Αυτό μπορεί να συμβεί είτε μέσω ενός

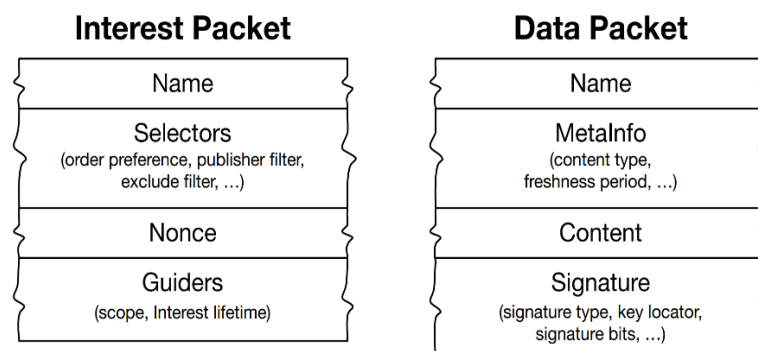
ντετερμινιστικού αλγορίθμου που επιτρέπει στον παραγωγός και τον καταναλωτή να καταλήξουν στο ίδιο όνομα με βάση πληροφορίες που είναι διαθέσιμες και στους δύο, είτε με Επιλογείς ενδιαφέροντος (Interest selectors), σε συνδυασμό με το μακρύτερο πρόθεμα που ταιριάζει στα επιθυμητά δεδομένα, με μια ή περισσότερες επαναλήψεις.

Η διαχείριση του χώρου των ονομάτων δεν αποτελεί μέρος της αρχιτεκτονικής NDN, όπως και η διαχείριση του χώρου διευθύνσεων δεν αποτελεί μέρος της IP αρχιτεκτονική. Ωστόσο, η ονομασία είναι το πιο σημαντικό μέρος στο σχεδιασμό εφαρμογών NDN. Η ονομασία δεδομένων αντί διευθύνσεων, επιτρέπει την υποστήριξη λειτουργικότητας όπως η διανομή περιεχομένου, η ομαδική επικοινωνία, η κινητικότητα και η δικτύωση ανεκτική στις καθυστερήσεις.

## Βασικές αρχές λειτουργίας του NDN

Στο NDN υπάρχουν δύο είδη πακέτων, τα Πακέτα Ενδιαφέροντος (Interest Packet) και τα Πακέτα Δεδομένων (Data Packet). Αυτοί οι δύο τύποι πακέτων μεταφέρουν το όνομα που προσδιορίζει ένα κομμάτι πληροφορίας το οποίο στην περίπτωση του Interest, είναι το τό όνομα του περιεχομένου το οποίο ζητάται, ενώ στην περίπτωση του Data Packet είναι το όνομα της δεδομένων που μεταφέρει το πακέτο αυτό. Το όνομα ενός NDN πακέτου μπορεί να αναφέρεται σε οτιδήποτε, όπως σε ένα τμήμα δεδομένων μιας ταινίας ή ενός βιβλίου, μία εντολή για την ενεργοποίηση φώτων, μία τιμή από έναν αισθητήρα κ.α.

Η επικοινωνία στο NDN είναι καθοδηγούμενη από τον παραλήπτη των δεδομένων ή αλλιώς καταναλωτή. Ο καταναλωτής βάζει το όνομα της επιθυμητής πληροφορίας σ' ένα Πακέτο Ενδιαφέροντος και το στέλνει στο δίκτυο. Οι δρομολογητές χρησιμοποιούν αυτό το όνομα για να προωθήσουν το πακέτο αυτό προς τους παραγωγούς δεδομένων (producers). Μόλις το Πακέτο Ενδιαφέροντος φτάσει σ' έναν κόμβο όπου υπάρχει διαθέσιμη η ζητούμενη πληροφορία, ο κόμβος θα του επιστρέψει ένα Πακέτο Δεδομένων το οποίο περιέχει τόσο το όνομα, όσο και το περιεχόμενο το οποίο ζητήθηκε, μαζί με μια υπογραφή από το κλειδί του παρόχου το οποίο δεσμεύει τα δύο παραπάνω και το πακέτο αυτό μεταφέρεται στον καταναλωτή μέσω των διαθέσιμων κόμβων.



Εικόνα 6: Τα πακέτα στην αρχιτεκτονική NDN. Δεξιά το Πακέτο Ενδιαφέροντος (Interest Packet) και αριστερά το Πακέτο Δεδομένων (Data Packet)[2].

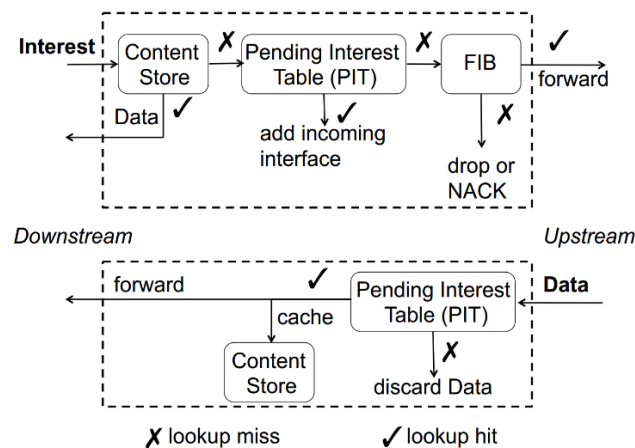
Για να μπορεί να επιτελέσει την επιτυχή μεταφορά των πακέτων Ενδιαφέροντος και Δεδομένων, ο κάθε δρομολογητής στο NDN (Content Router-CR) διατηρεί τις ακόλουθες τρεις δομές δεδομένων:

1. Pending Interest Table (PIT): Θα μπορούσαμε να τον θεωρήσουμε ως έναν πίνακα που διατηρεί τα αιτήματα προς αναμονή. Το PIT αποθηκεύει όλα τα Πακέτα

Ενδιαφέροντος που ένας δρομολογητής έχει προωθήσει αλλά δεν έχουν ικανοποιηθεί ακόμη. Κάθε καταχώρηση στο PIT καταγράφει το όνομα του περιεχομένου του πακέτου, μαζί με τις διεπαφές εισόδου και εξόδου του Πακέτου Ενδιαφέροντος.

2. Forwarding Information Base(FIB): Η FIB είναι μία βάση όπου αποθηκεύονται οι πληροφορίες προώθησης. Περιέχει προθέματα ονομάτων βασισμένα στο πρωτόκολλο δρομολόγησης και μπορεί να έχει πολλαπλές διεπαφές εξόδου για κάθε πρόθεμα που διατηρεί.
3. Content Store(CS): Η CS λειτουργεί ως η βραχυπρόθεσμη μνήμη των Πακέτων Δεδομένων που λαμβάνει ο κάθε δρομολογητής. Ένα Πακέτο Δεδομένων, ανεξάρτητα από που έρχεται ή που προωθείται, μπορεί να αποθηκευτεί στην κρυφή μνήμη για να ικανοποιήσει κάποιο μελλοντικό Πακέτο Ενδιαφέροντος.

Οι δομές αυτές παρουσιάζονται στην εικόνα 7 και η λειτουργικότητα τους θ' αναλυθεί στη συνέχεια σε συνδυασμό με τη Στρατηγική Προώθησης (Forwarding Strategy) και είναι αυτή που καθορίζει εάν, πότε και που θα προωθηθεί το κάθε Πακέτο Ενδιαφέροντος.



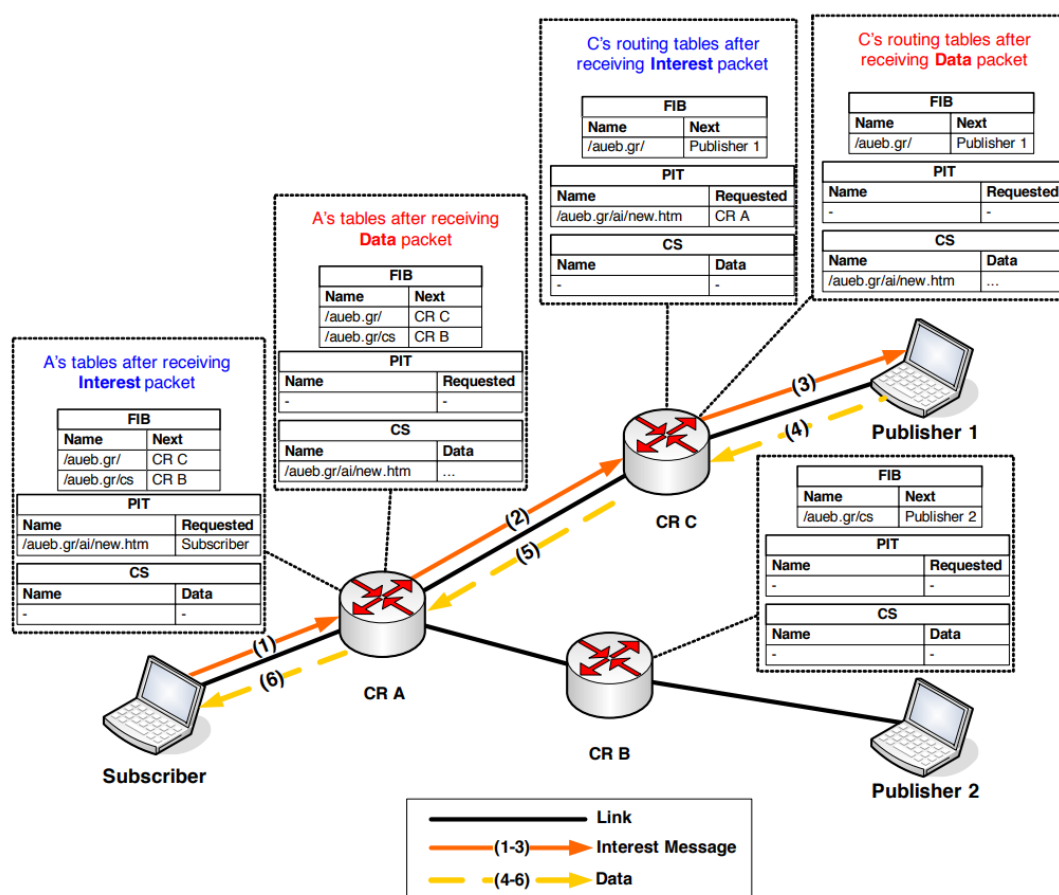
Εικόνα 7: Η διαδικασία προώθησης σ' έναν NDN κόμβο[2].

Όπως προαναφέρθηκε, η επικοινωνία καθοδηγείται από τον καταναλωτή, συνεπώς όλα ξεκινούν όταν ένα Πακέτο Ενδιαφέροντος αποσταλεί από αυτόν προς το δίκτυο. Όταν ένα Πακέτο Ενδιαφέροντος φτάνει σ' έναν δρομολογητή CR, ο δρομολογητής πρώτα ελέγχει το Content Store για δεδομένα που ταιριάζουν με αυτό που ζητείται και αν αυτά είναι διαθέσιμα, επιστρέφει το αντίστοιχο Πακέτο Δεδομένων στη διεπαφή από την οποία κατέφτασε το Πακέτο Ενδιαφέροντος. Ειδικά, ο δρομολογητής στον πίνακα PIT εάν υπάρχει το όνομα και αν υπάρχει ήδη μια καταχώρηση που του ταιριάζει, απλά καταγράφει στην καταχώρηση αυτήν την διεπαφή από την οποία ήρθε το Πακέτο Ενδιαφέροντος. Σε περίπτωση που δεν υπάρχει αντίστοιχη καταχώρηση στον PIT ο δρομολογητής θα προωθήσει το Πακέτο Ενδιαφέροντος προς τον παραγωγό των δεδομένων, βάσει των πληροφοριών που υπάρχουν στην FIB καθώς και τη Στρατηγική Προώθησης του δρομολογητή(η οποία προσαρμόζεται όποτε χρειάζεται).

Όταν ένας δρομολογητής λαμβάνει από πολλούς κόμβους Πακέτα Ενδιαφέροντος για το ίδιο όνομα, προωθεί μόνο το πρώτο προς τον παραγωγό των δεδομένων. Η FIB συμπληρώνεται από ένα πρωτόκολλο δρομολόγησης που βασίζεται στο πρόθεμα του ονόματος και μπορεί να έχει πολλαπλές διεπαφές εξόδου για κάθε πρόθεμα. Για κάθε Πακέτο Ενδιαφέροντος, η Στρατηγική Προώθησης ανακτά από την FIB την καταχώρηση που το πρόθεμα της έχει το

μεγαλύτερο ποσοστό αντιστοιχίας. Η Στρατηγική Προώθησης μπορεί ν' αποφασίσει ν' απορρίψει ένα Πακέτο Ενδιαφέροντος υπό συγκεκριμένες συνθήκες, όπως στην περίπτωση που όλοι οι σύνδεσμοι προς τον παραγωγό είναι υπό συμφόρηση ή εάν υπάρχει υποψία ότι το Πακέτο Ενδιαφέροντος αποτελεί μέρος μιας επίθεσης Denial of Service (DoS)[21].

Όταν καταφθάνει ένα Πακέτο Δεδομένων, ο δρομολογητής βρίσκει την αντίστοιχη PIT καταχώρηση και προωθεί τα δεδομένα σε όλες τις διεπαφές που του αντιστοιχούν στην καταχωρημένη PIT λίστα. Έπειτα, αφαιρεί αυτήν την καταχώρηση από τον PIT και αποθηκεύει τα δεδομένα Content Store. Τα Πακέτα Δεδομένων πάντα ακολουθούν την αντίθετη διαδρομή από αυτή του Πακέτου Ενδιαφέροντος και εφόσον δεν υπάρχουν απώλειες πακέτων, ένα Πακέτο Ενδιαφέροντος που ικανοποιείται, αντιστοιχεί σ' ένα Πακέτο Δεδομένων στο δίκτυο κι έτσι υπάρχει ισορροπία ροής.



Εικόνα 8: Ο καταναλωτής στέλνει ένα πακέτο Ενδιαφέροντος (Interest Packet) (βέλη 1-3) και ο παραγωγός απαντάει μ' ένα πακέτο Δεδομένων (Data Packet)(βέλη 4-6)[29].

## Αποθήκευση εντός Δικτύου

Επειδή κάθε Πακέτο Δεδομένων φέρει ένα όνομα και μια υπογραφή, είναι σημαντικό ανεξάρτητα από το ποιος το ζήτησε ή από όπου ανασύρεται. Έτσι, ένας δρομολογητής μπορεί να αποθηκεύσει προσωρινά στο Content Store Πακέτα Δεδομένων και να τα χρησιμοποιήσει τα για να ικανοποιήσει μελλοντικά αιτήματα. Το Content Store είναι ανάλογο με την προσωρινή μνήμη (buffer) σε δρομολογητές IP, με τη διαφορά ότι οι δρομολογητές IP δεν μπορούν να επαναχρησιμοποιήσουν ένα πακέτο αφού το προωθήσουν στον προορισμό του, ενώ οι δρομολογητές NDN μπορούν.

Το NDN υποστηρίζει την προσωρινή αποθήκευση εντός διαδρομής (on-path) και αντιμετωπίζει τα κανάλια αποθήκευσης και τα κανάλια δικτύου με τον ίδιο τρόπο όσον αφορά την ανάκτηση δεδομένων. Είναι αρκετά αποτελεσματικό για το χειρισμό ενός κύματος χρηστών, όπου πολλοί χρήστες ζητούν τα ίδια δεδομένα διαδοχικά. Για στατικά αρχεία, το NDN επιτυγχάνει σχεδόν βέλτιστη παράδοση δεδομένων. Ακόμα και για δυναμικό περιεχόμενο μπορεί να επωφεληθεί από την προσωρινή αποθήκευση σε περίπτωση πολλαπλής εκπομπής (π.χ. μία τηλεδιάσκεψη σε πραγματικό χρόνο) ή σε περίπτωση αναμετάδοσης μετά από την απώλεια κάποιου πακέτου.

Η προσωρινή αποθήκευση ονομαστικών δεδομένων εγείρει διαφορετικές ανησυχίες απορρήτου από αυτές του πρωτοκόλλου IP. Στην περίπτωση του IP, μπορεί κανείς να εξετάσει επικεφαλίδες πακέτων, και ενδεχομένως ωφέλιμο φορτίο, για να μάθει ποιος καταναλώνει ποια δεδομένα. Η ονομασία και η αποθήκευση δεδομένων σε δίκτυα NDN μπορεί να διευκολύνει την παρακολούθηση των δεδομένων που ζητούνται, αλλά χωρίς διευθύνσεις προορισμού, είναι πιο δύσκολο να προσδιοριστεί ποιος το ζητά. Επομένως, το NDN προσφέρει ένα εντελώς διαφορετικό είδος προστασίας της ιδιωτικότητας από τα τρέχοντα δίκτυα IP.

## Ασφάλεια

Σε αντίθεση με το TCP/IP το οποίο μεταθέτει την ευθύνη για την ασφάλεια στα τερματικά σημεία, το NDN διαφυλάσσει τα ίδια τα δεδομένα απαιτώντας από τους παραγωγούς δεδομένων να κρυπτογραφούν κάθε ένα Πακέτο Δεδομένων, ενώνοντας έτσι το την ονομασία του περιεχομένου με τον εκδότη του και το ίδιο το περιεχόμενο. Αυτό επιτρέπει σε οποιονδήποτε κόμβο, συμπεριλαμβανομένων των δρομολογητών CR, να εξακριβώνει τη σύνδεση μεταξύ του ονόματος του πακέτου και της πληροφορίας που το συνοδεύει.

Η υπογραφή του παραγωγού εξασφαλίζει την ακεραιότητα και επιτρέπει τον προσδιορισμό της προέλευσης των δεδομένων, επιτρέποντας στον καταναλωτή να εμπιστευτεί την προέλευση των δεδομένων. Επίσης αυτή η άρρηκτη σύνδεση μεταξύ εκδότη και περιεχομένου, δίνει στον καταναλωτή τη δυνατότητα να αναλογιστεί αν το κλειδί του ιδιοκτήτη είναι αποδεκτό για να θεωρηθεί εκδότης για ένα συγκεκριμένο κομμάτι πληροφορίας σε ένα συγκεκριμένο περιεχόμενο. Μπορεί κάποιος επίσης να κρυπτογραφήσει τα δεδομένα όταν απαιτείται εμπιστευτικότητα δεδομένων [17]. Αυτές οι ιδιότητες ασφαλείας (αυθεντικότητα και εμπιστευτικότητα) παραμένουν με τα ίδια τα δεδομένα, ανεξαρτήτως ενθυλάκωσης των δεδομένων ή των διαύλων επικοινωνίας που χρησιμοποιούνται για τη μεταφορά τους.

Η απαίτηση υπογραφών στη δρομολόγηση του δικτύου και ο έλεγχος των NDN δεδομένων, παρέχουν σταθερά θεμέλια για την ασφάλεια των πρωτόκολλων δρομολόγησης ενάντια σε πλαστογράφηση και αλλοίωση. Η χρήση πολλών διαδρομών δρομολόγησης μαζί με το γεγονός ότι η στρατηγική προώθησης προσαρμόζεται, μετριάζουν την πιθανότητα hijacking του προθέματος του ονόματος, διότι οι δρομολογητές έχουν τη δυνατότητα να εντοπίσουν ανωμαλίες που δημιουργούνται από hijacking και ν' ανακτούν τα δεδομένα μέσω εναλλακτικών διαδρομών [30]. Εφόσον τα NDN πακέτα αναφέρουν περιεχόμενο και όχι συσκευές, είναι δυσκολότερο να στοχεύσει κάποιος κακόβουλα μια συγκεκριμένη συσκευή. Παρόλα αυτά που θα χρειαστούν μηχανισμοί εξομάλυνσης-καταπολέμησης εναντίον άλλων ειδικών επιθέσεων NDN όπως το Interest flooding DoS[31].

## 2.1.4. Universal, Mobile-Centric and Opportunistic Communications Architecture (UMOBILE)

Βάσει των εξελίξεων που ήδη αναφέρθηκαν στον τομέα της αρχιτεκτονικής δικτύων, αναπτύχθηκε το UMOBILE[4], μια αρχιτεκτονική που μετατοπίζει την παρούσα εστίαση του αρχιτεκτονικού σχεδιασμού. Στο σχεδιασμό αλλά και τη λειτουργία του UMOBILE, η περιφέρεια του δικτύου αποτελεί το επίκεντρο της προσοχής, καθώς το μεγαλύτερο μέρος της κίνησης δεδομένων προέρχεται από τα άκρα του δικτύου. Σκοπός είναι η ενσωμάτωση ευκαιριακών επικοινωνιών (opportunistic communications) και επικοινωνιών που είναι ανεκτικές στις καθυστερήσεις(delay-tolerant communications) με τον πυρήνα του δικτύου.

Ο κύριος στόχος του UMOBILE είναι να αναπτύξει μια αρχιτεκτονική με επίκεντρο την κινητικότητα και τις υπηρεσίες, η οποία παρέχει αποτελεσματικά περιεχόμενο και υπηρεσίες στους τελικούς χρήστες. Με τον όρο αποτελεσματικά εννοούμε το περιεχόμενο ή οι υπηρεσίες να έχουν την αναμενόμενη ποιότητα και να είναι αξιόπιστα διαθέσιμες, παρά τις όποιες βλάβες ή έλλειψη διαθεσιμότητας μπορεί να υπάρχουν σε κάποιο μέρος της υποδομής του δικτύου.

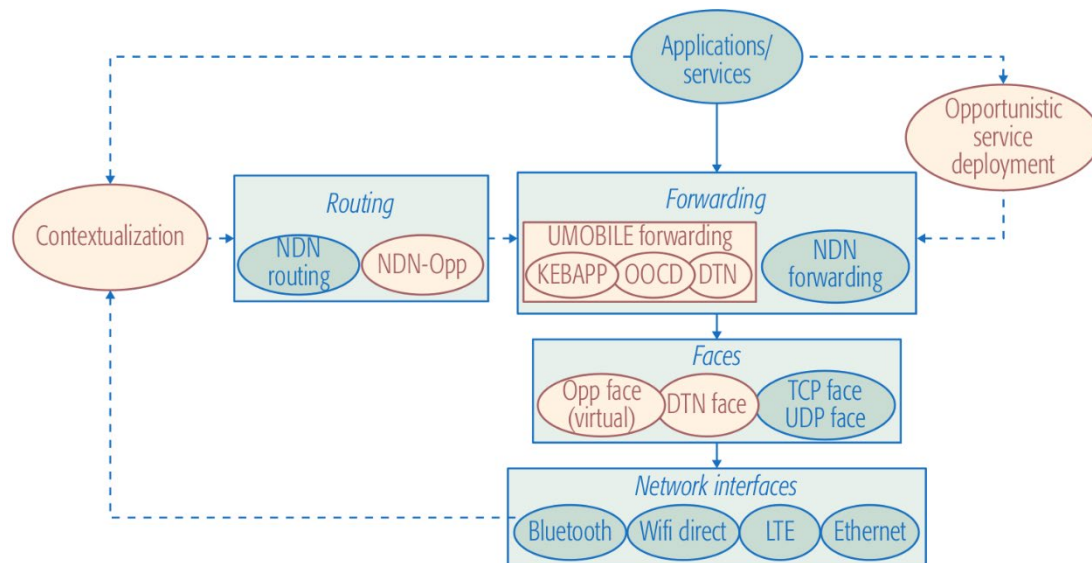
Το UMOBILE αποσυνδέει τις υπηρεσίες από τις τοποθεσίες προέλευσης τους, μετατοπίζοντας τον προσανατολισμό από το κέντρο του δικτύου στα άκρα, ενσωματώνοντας πτυχές τόσο της πληροφοριοκεντρικής όσο και της ευκαιριακής δικτύωσης, με απώτερο σκοπό για την παροχή μιας αρχιτεκτονικής που εστιάζει στα ακόλουθα:

1. Τη βελτίωση πτυχών της υπάρχουσας υποδομής π.χ. διατήρηση της τοπικής κυκλοφορίας σε χαμηλότερες καθυστερήσεις και χαμηλότερα λειτουργικά έξοδα(OPEX).
2. Τη βελτίωση της κοινωνικής ρουτίνας των χρηστών του Διαδικτύου μέσω τεχνολογικών προσεγγίσεων.
3. Την επέκταση της εμβέλειας των υπηρεσιών σε περιοχές με μικρή ή καμία υποδομή (π.χ. απομακρυσμένες περιοχές, καταστάσεις έκτακτης ανάγκης).

Το UMOBILE στοχεύει στην προώθηση υπηρεσιών δικτύου όπως η διαχείριση κινητικότητας και η υποστήριξη διαλείπουσας συνδεσιμότητας και υπηρεσίες χρηστών όπως η διάχυτη διαχείριση περιεχομένου όσο το δυνατόν πιο κοντά στους τελικούς χρήστες. Προωθώντας αυτές τις υπηρεσίες πιο κοντά στους χρήστες, μπορούν να βελτιστοποιηθούν με κλιμακούμενο τρόπο, πτυχές όπως η χρήση του εύρους ζώνης και η διαχείριση πόρων. Επίσης βελτιώνεται η διαθεσιμότητα των υπηρεσιών σε απαιτητικά περιβάλλοντα.

Η αρχιτεκτονική UMOBILE βασίζεται στην πλατφόρμα NDN. Προκειμένου να καλυφθούν οι ελλείψεις της NDN αρχιτεκτονικής γίνανε τροποποιήσεις και βελτιώσεις όπως νέες στρατηγικές προώθησης προσαρμοσμένες στις συγκεκριμένες ανάγκες, ένας ευκαιριακός (opportunistic) μηχανισμός δρομολόγησης για NDN και άλλα. Το UMOBILE εφαρμόζεται αυτή τη στιγμή σε σταθερά συστήματα τεχνολογίας Linux και σε κινητές συσκευές Android. Παρακάτω θα παρουσιαστούν οι κύριες τεχνικές αναβαθμίσεις του μοντέλου και πιο αναλυτικά η αρχιτεκτονική μπορεί να βρεθεί στο[32]. Στην εικόνα 9, παρουσιάζεται η αρχιτεκτονική του UMOBILE. Οι κόκκινες μονάδες αναπαριστούν τα νέα στοιχεία που αναπτύχθηκαν στην αρχιτεκτονική UMOBILE, ενώ οι μπλε αναπαριστούν τις μονάδες που υπήρχαν ήδη στο πλαίσιο του NDN.





Εικόνα 9: Η αρχιτεκτονική του UMOBILE[4].

## Πρώθηση

Ο πυρήνας του προτεινόμενου πληροφοριοκεντρικού μοντέλου επικοινωνίας έγκειται στη χρήση των Πακέτων Ενδιαφέροντος που εκδίδονται από πελάτες (pull-based model) ή παρόχους περιεχομένου (push-based model). Για να μεταφερθεί μ' επιτυχία, πρέπει να ληφθεί μία απόφαση για το αν, τότε και σε ποιον θα προωθηθεί το Πακέτο Ενδιαφέροντος. Δεδομένων των διαφορετικών απαιτήσεων εφαρμογής, καθώς και των συνθηκών δικτύωσης που ενδέχεται να εκτείνονται από συνεχή συνδεσιμότητα υψηλής ταχύτητας έως και διαλείπουσες επικοινωνίες με διαταραχές, το UMOBILE χρειάζεται ένα σύνολο από διαφορετικές στρατηγικές πρώτωσης. Στις ακόλουθες υποενότητες, θα παρουσιαστούν οι δύο επιλογές πρώτωσης που αναπτύχθηκαν για το UMOBILE.

### DTN Tunneling

Το DTN tunneling αποτελεί τη μια επιλογή πρώτωσης που αναπτύχθηκε στο πλαίσιο του UMOBILE. Χρησιμοποιεί την υλοποίηση IBR-DTN[33] (θ' αναλυθεί σε επόμενη ενότητα) του πρωτοκόλλου Bundle και ενισχύει το NDN δημιουργώντας ένα DTN face. Ουσιαστικά ενθυλακώνει πακέτα NDN βάζοντας τους ως περίβλημα το καινοτόμο DTN face, επεκτείνοντας έτσι τη λειτουργία του NDN και προσφέροντας πρόσβαση σε απομακρυσμένες περιοχές όπου δεν υπάρχει τυπική σύνδεση στο Διαδίκτυο, καθώς και αξιοπιστία στις υπηρεσίες σε απαιτητικά περιβάλλοντα.

Επίσης το tunneling μέσω συσκευών που χρησιμοποιούν DTN, επιλύει και το πρόβλημα που δημιουργείται στο NDN από το γεγονός ότι κάθε Πακέτο Δεδομένων πρέπει να επιστρέφει από την ίδια διαδρομή που ήρθε το αντίστοιχο Πακέτο Ενδιαφέροντος, μία απαίτηση που αναχαιτίζει σημαντικά την απόδοση σε δίκτυα όπου οι κόμβοι δεν είναι σταθεροί.

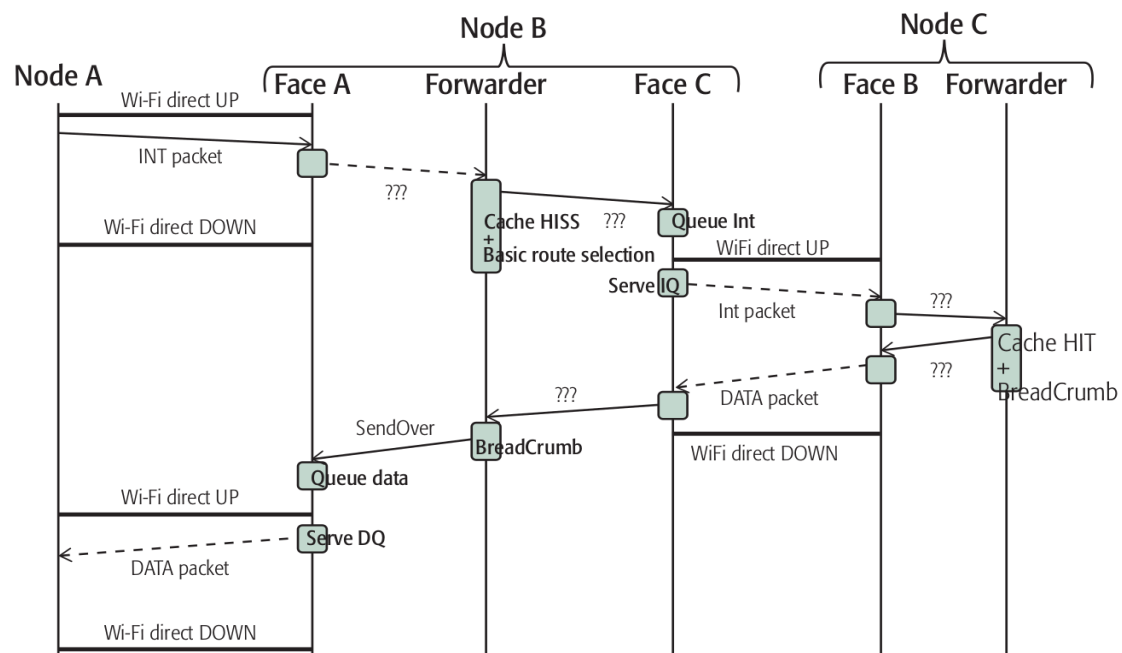
Κατά την προσέγγισή αυτή, η πρώτωση των πακέτων γίνεται στο επίπεδο DTN μέσω των νησίδων DTN που υπάρχουν μεταξύ των κόμβων NDN. Ένας κόμβος UMOBILE (π.χ. ένα σημείο πρόσβασης) διατηρεί μία καταχώρηση στα FIB/PIT προς τον επόμενο κόμβο, ενώ τα πακέτα Ενδιαφέροντος/Δεδομένων ενθυλακώνονται και προωθούνται με διαφάνεια από τους ενδιαμέσους κινητούς κόμβους DTN. Αυτό επιτρέπει την πρώτωση των δεδομένων μεταξύ δύο απομακρυσμένων UMOBILE κόμβων που δεν έχουν σταθερή σύνδεση μεταξύ



τους, χρησιμοποιώντας διαφορετική διαδρομή για τα πακέτα Ενδιαφέροντος/Δεδομένων, χωρίς ν' αλλάξει η σημασιολογία του NDN.

### Opportunistic Off-Path Content Discovery (OOC):

Όλες οι προτεινόμενες αρχιτεκτονικές, συμπεριλαμβανομένου και του NDN, επικεντρώνονται στη δρομολόγηση των αιτημάτων προς το κέντρο του δικτύου. Αυτή η μέθοδος αντιτίθεται στη δημιουργία ενός εγγενούς δικτύου διανομής περιεχομένου, αποτρέπει την ανακάλυψη δεδομένων που μπορεί να βρίσκονται σε γειτονικούς κόμβους, όταν οι κόμβοι αυτοί δε βρίσκονται πάνω στην πιο σύντομη διαδρομή προς το κέντρο του δικτύου. Με σκοπό να βελτιωθεί η δυνατότητα εύρεσης περιεχομένου, αναπτύχθηκε μια βελτίωση της υπάρχουσας δρομολόγησης του NDN. Δημιουργήθηκε ένας καινούργιος πίνακας D-FIB που αποθηκεύει όλα τα αιτήματα που έχουν ικανοποιηθεί επιτυχώς. Ο πίνακας D-FIB λειτουργεί παρόμοια με τον FIB, μόνο που αυτός καταχωρεί τους κόμβους που έχουν πρόσφατα λάβει πακέτα Δεδομένων, άρα θα τα έχουν και αποθηκευμένα τοπικά στη μνήμη τους. Οι πίνακες D-FIB χρησιμοποιούνται αποτελεσματικά για να ανιχνεύσει περιεχόμενο το οποίο είναι αποθηκευμένο εκτός διαδρομής (off-path). Έτσι αυξάνεται ο ρυθμός Cache Hit και μειώνεται το overhead[34].



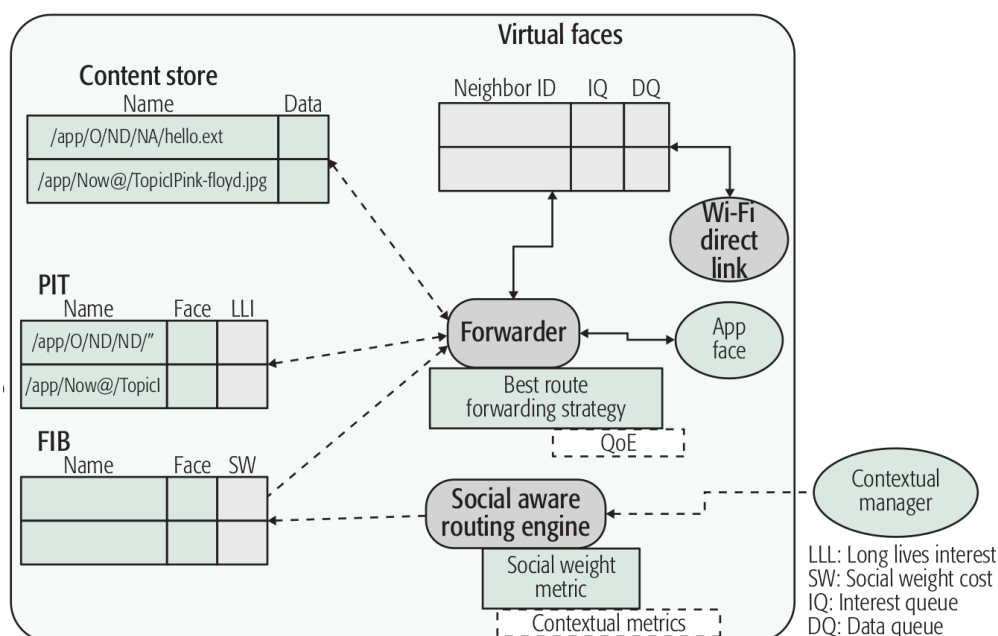
Εικόνα 10: Λειτουργία NDN-Opp[4].

### Δρομολόγηση

Η δρομολόγηση στο UMOBILE στοχεύει πρωτίστως στην ασύρματη επικοινωνία σε ευκαιρικά (opportunistic) δίκτυα. Έτσι στην αρχιτεκτονική αυτή αναπτύχθηκε ένα μοντέλο δρομολόγησης, το οποίο εκμεταλλεύεται τις δυνατότητες της ασύρματης επικοινωνίας, το framework NDN για ευκαιρικά δίκτυα (NDN-Opp). Το NDN-Opp προωθεί πακέτα Ενδιαφέροντος σε γειτονικούς κόμβους δίνοντας τους μεγάλη πιθανότητα να συναντήσουν κάποιον κόμβο ο οποίος θα διαθέτει τα επιθυμητά δεδομένα. Κάθε Opp face μπορεί να βρίσκεται στην κατάσταση ON ή OFF και διαθέτει δύο ουρές, την ουρά των Interest (IQ) όπου αποθηκεύει τα πακέτα Ενδιαφέροντος που πρόκειται να στείλει και την ουρά των Data (DQ) όπου αποθηκεύει δείκτες προς το Content Store για τα Δεδομένα που πρόκειται να στείλει.

Η λειτουργία της επαφής Orr είναι παρόμοια με τη λειτουργία του NDN, όπως φαίνεται στην Εικόνα 10, καθώς η κατάσταση του κόμβου αλλάζει κατά την άφιξη ενός πακέτου Ενδιαφέροντος ή Δεδομένων. Η βασική διαφορά τους είναι το πότε τερματίζει η λειτουργία του καθενός. Η λειτουργία του NDN τερματίζει με την αποστολή του πακέτου, ενώ η λειτουργία των Orr Faces τερματίζει με την αποθήκευση των πακέτων τοπικά στη μνήμη τους.

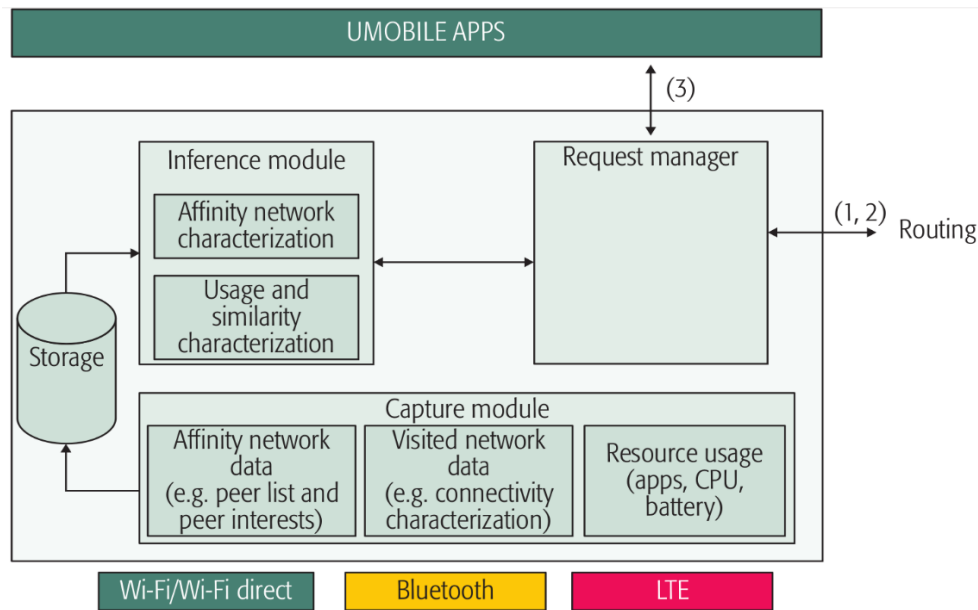
Η Εικόνα 11, δείχνει το σχέδιο NDN-Orr, βασισμένο σε νέο μοντέλο προώθησης και δρομολόγησης. Όταν λαμβάνεται ένα πακέτο Ενδιαφέροντος, το PIT αποθηκεύει επίσης πληροφορίες που σχετίζονται με τη διάρκεια ζωής του (LLI). Τα πακέτα ενδιαφέροντος προωθούνται με βάση τη στρατηγική προώθησης βέλτιστης διαδρομής του NDN, ενώ το κόστος των προθεμάτων των ονομάτων (αποθηκευμένα στην FIB) υπολογίζεται από μία μονάδα δρομολόγησης (social-aware routing engine) που έχει επίγνωση της κοινωνίας (π.χ. κοινωνικές επαφές ή ομοιότητες ζήτησης δεδομένων). Στην παρούσα έκδοση, το NDN-Orr χρησιμοποιεί τον Time-Evolving Contact Duration algorithm (TECD)[35]. Τα πακέτα Δεδομένων (Data Packet) ακολουθούν τους γνωστούς κανόνες προώθησης του NDN.



Εικόνα 11: Κόμβος NDN-Orr[4].

## Contextualization

Το UMOBILE λαμβάνει υπόψιν του την κοινωνία όπως προείπαμε και αυτό εφαρμόζεται και στο Context Plane με σκοπό τη βελτίωση της διάδοσης των δεδομένων. Το UMOBILE Context Plane ορίζεται από τον διαχειριστή του, τον contextual manager(CM) που φαίνεται στην Εικόνα 12, μία υπηρεσία που τρέχει στο παρασκήνιο σε μία τερματική συσκευή ή σε ένα τελικό σημείο πρόσβασης. Το CM λαμβάνει πληροφορίες σχετικά με το δίκτυο συσχέτισης των συσκευών, τις συνήθειες χρήσης και τα ενδιαφέροντα (εσωτερικές πληροφορίες συσκευής). Τα κόστη που προκύπτουν απ' το contextualization περνούν στη συνέχεια, κατόπιν αιτήματος ή περιοδικά, σε άλλα λειτουργικά τμήματα του δικτύου UMOBILE με στόχο της επίτευξης πιο αποτελεσματικής διάδοσης δεδομένων. Το CM διαθέτει τρεις διαφορετικές διεπαφές προς τα τμήματα αυτά: δύο διεπαφές για το τμήμα της δρομολόγησης και μία για τις εφαρμογές, επιτρέποντας έτσι σε άλλα τμήματα να λαμβάνουν πληροφορίες από το CM.



Εικόνα 12: Contextual manager[4].

## Northbound APIs

Στο σημείο αυτό παρουσιάζεται το πλαίσιο λογισμικού το οποίο παρέχει το UMOBILE, για να καλύψει τις διάφορες ανάγκες αλληλεπίδρασης των εφαρμογών με το δίκτυο.

### Υπηρεσίες Push

Το NDN υποστηρίζει εγγενώς υπηρεσίες pull, όπου η μεταφορά του περιεχομένου ξεκινά έπειτα από το αίτημα του καταναλωτή. Αυτό το μοντέλο δεν ταιριάζει όμως σε εφαρμογές που αφορούν τη μετάδοση περιοδικά παραγόμενων εφήμερων δεδομένων[36]. Για το λόγο, αυτό αναπτύχθηκε το μοντέλο push υπηρεσιών, με το οποίο οι παραγωγοί παίρνουν πρωτοβουλία για τη μεταφορά δεδομένων στους καταναλωτές[32].

### Polling Interest:

Ιδανικό για εφαρμογές όπου ο συνδρομητής δεν γνωρίζει πότε το περιεχόμενο θα γίνει διαθέσιμο από τον παραγωγό. Βασίζεται σε ένα μηχανισμό διερεύνησης ο οποίος επιτρέπει στο καταναλωτή να εκδίδει περιοδικά αιτήματα προς τον παραγωγό. Ο παραγωγός βάζει το περιεχόμενο σ' ένα Πακέτο Δεδομένων και το αποστέλλει στον καταναλωτή. Το μειονέκτημα αυτής της υπηρεσίας είναι ότι αυξάνει το λειτουργικό κόστος (overhead) του δικτύου και ότι το πόσο πρόσφατα θεωρούνται τα ληφθέντα δεδομένα, εξαρτάται από τη συχνότητα της διερεύνησης.

### Ειδοποιήσεις Interest:

Χρήσιμο όταν το περιεχόμενο είναι ένα κομμάτι κειμένου που είναι αρκετά μικρό ώστε να επισυνάπτεται από το παραγωγό στο όνομα ενός πακέτου Ενδιαφέροντος. Κατά τη λήψη του Πακέτου Ενδιαφέροντος, ο παραλήπτης έχει τη δυνατότητα εάν θέλει, να στείλει ένα "ack Data" για να επιβεβαιώσει τη λήψη του.

### Κοινοποίηση Διάδοσης Δεδομένων:

Ενδείκνυται για τη μεταφορά περιεχομένου κάθε είδους και μεγέθους. Όταν ο παραγωγός κάνει διαθέσιμο το περιεχόμενο, στέλνει μία ειδοποίηση στο συνδρομητή ο οποίος απαντά με ένα Πακέτο Ενδιαφέροντος με το οποίο ζητά το περιεχόμενο. Το περιεχόμενο χωρίζεται σε τμήματα το πρώτο κομμάτι του περιεχομένου αποστέλλεται μαζί με πληροφορίες σχετικά με τα υπόλοιπα τμήματα, τα οποία μεταφέρονται σε επόμενες ανταλλαγές.

#### **KEBAPP:**

Το πλαίσιο αυτό επιτρέπει την πρόσβαση στην επιθυμητή επεξεργασμένη και μη εξατομικευμένη πληροφορία μέσω της κοινής χρήσης εφαρμογών, αξιοποιώντας αποτελεσματικά ένα σύνολο πόρων. Εισάγει την έννοια των λέξεων κλειδιών (keywords), επεκτείνοντας έτσι τον χώρο ονομάτων, για να επιτρέψει την κλήση εφαρμογών που μπορεί να βρίσκονται σε κινητές συσκευές με κοινή τοποθεσία.

## **2.2. Σχετική Έρευνα**

Στο σημείο αυτό θα γίνει μία σύντομη αναφορά σε εργασίες οι οποίες βοήθησαν τη σκέψη μας να καθοδηγηθεί στο στήσιμο της συγκεκριμένης διπλωματικής εργασίας είτε από πλευράς θεωρητικού υποβάθρου, είτε από πλευράς εργαλείων. Ο αρχικός σχεδιασμός των πειραμάτων της παρούσας διπλωματικής ενισχύθηκε από τη μελέτη των ακόλουθων δημοσιεύσεων και της διατριβής, των οποίων θα γίνει μία σύντομη παρουσίαση στη συνέχεια. Για το σχεδιασμό των πειραμάτων τα οποία θέλαμε να πραγματοποιήσουμε, συνδυάστηκαν με τη βασική ιδέα, διαφορετικά στοιχεία από το καθένα, τα οποία συνέβαλλαν άλλο σε μικρότερο και άλλο σε μεγαλύτερο βαθμό στο επιθυμητό αποτέλεσμα. Με τον τρόπο αυτό, καταφέραμε να οδηγηθούμε σε μια λειτουργική υλοποίηση και μία σειρά από ενδιαφέροντα πειράματα στον τομέα των NDN και DTN τεχνολογιών.

### **2.2.1. Συσκευές IoT με διαλείπουσες συνδέσεις: Πειράματα με μία αρχιτεκτονική NDN-DTN**

Στη δημοσίευση με τίτλο “Συσκευές IoT με διαλείπουσες συνδέσεις: Πειράματα με μία αρχιτεκτονική NDN-DTN”(Intermittently-connected IoT devices: Experiments with an NDN-DTN architecture)[37], υποστηρίζεται ότι οι τεχνολογίες NDN και DTN μπορούν να συμπληρώσουν η μία την άλλη για τη διαμόρφωση ενός κοινού σχήματος NDN-DTN που μπορεί να βελτιώσει την ανάκτηση δεδομένων από συσκευές με περιοδική σύνδεση, όπως π.χ. Δίκτυα IoT και αισθητήρων σε απομακρυσμένες περιοχές. Για να ποσοτικοποιηθούν οι βελτιώσεις αυτές, χρησιμοποιούνται τόσο μετρικές που σχετίζονται με DTN όσο και NDN. Συνδυάζονται τες σε ένα κοινό αναλυτικό πλαίσιο, παρουσιάζεται ότι το NDN-DTN μπορεί να βελτιώσει σημαντικά το Δείκτη Ικανοποίησης Ενδιαφέροντος (Interest Satisfaction Ratio-ISR) και τη Μέση Καθυστέρηση (Average Delay) σε σύγκριση με την προσέγγιση μόνο με DTN, ιδιαίτερα σε περιβάλλοντα με χαμηλό Δείκτη Παράδοσης (Delivery Ratio).

Τα αποτελέσματά αυτά επαληθεύτηκαν μέσω πειραμάτων με Συσκευές IoT με διαλείπουσες συνδέσεις και διερευνήθηκε το αντίκτυπο παραμέτρων όπως π.χ. ο χρόνος ζωής ενός Πακέτου Ενδιαφέροντος (Interest Lifetime), ο το χρονικό διάστημα κατά το οποίο ένα Πακέτου Δεδομένων θεωρείται "φρέσκο" (Data Freshness Period) και η διάρκεια επαφής δύο κόμβων (Contact Duration). Το τελικό συμπέρασμα ήταν ότι μια κοινή προσέγγιση NDN-DTN υπερτερεί της προσέγγισης μόνο με DTN.

Ορμώμενοι λοιπόν από το συμπέρασμα ότι η κοινή προσέγγιση NDN-DTN παρουσιάζει βελτιωμένη συμπεριφορά, αποφασίσαμε να δοκιμάσουμε να εξερευνήσουμε σε ένα λίγο διαφορετικό πλαίσιο την τεχνολογία αυτή. Σκεφτήκαμε ότι θα παρουσίαζε ιδιαίτερο ενδιαφέρον να δούμε τι συμβαίνει σε περιπτώσεις όπου έχουμε περισσότερους ενδιάμεσους κόμβους οι οποίοι κινούνται, Έτσι μετά από προσεκτικό σχεδιασμό καταλήξαμε το αντικείμενο της μελέτη μας να είναι η συμπεριφοράς δικτύων που συνδυάζουν τις τεχνολογίες NDN και DTN, με κινούμενους ασύρματους κόμβους και με τη χρήση διαφορετικών πρωτοκόλλων δρομολόγησης (Routing Protocols) και συγκεκριμένα των πρωτοκόλλων Flooding, Epidemic και ProPHET. Τέλος, επιλέξαμε να εξεταστούν κι εμείς τ' αποτελέσματα της μεταβολής διαφόρων παραμέτρων του δικτύου, στην απόδοση του.

### **2.2.2. Η επίδραση των ασύρματων κόμβων για Μοντέλα Κινητικότητας σε Πρωτόκολλα Δρομολόγησης Ανεκτικά στις Καθυστερήσεις**

Η δημοσίευση με τίτλο “ Η επίδραση των ασύρματων κόμβων για Μοντέλα Κινητικότητας σε Πρωτόκολλα Δρομολόγησης Ανεκτικά στις Καθυστερήσεις” (Intermittently-connected IoT devices: Experiments with an NDN-DTN architecture) ερευνά την απόδοση πρωτοκόλλων δρομολόγησης DTN, τα οποία βασίζονται στην αντιγραφή του μηνύματος που έχουν να προωθήσουν και την προώθηση του αντιγράφου αυτού τον επόμενο κόμβο. Πιο συγκεκριμένα μελετώνται τα ακόλουθα πρωτόκολλα του DTN

- Epidemic, Probabilistic Routing Protocol using History of Encounters and Transitivity (PRoPHET)
- MaxProp
- Resource Allocation Protocol for Intentional DTN (RAPID)
- Binary Sprayand-Wait (B-SNW)
- Spray-and-Focus (SNF)

Εξετάζονται για διαφορετικό αριθμό κινητών κόμβων και για τρία μοντέλα κινητικότητας, συγκεκριμένα για τα Random Walk (RW), Random Direction (RD) και Shortest Path Map Based (SPMB) . Η απόδοση τους έχει αξιολογηθεί και αναλυθεί χρησιμοποιώντας τον προσομοιωτή Opportunistic Network Environment (ONE), λαμβάνοντας υπόψιν τις ακόλουθες τρεις μετρικές:

1. Πιθανότητα Παράδοσης(Delivery Probability)
2. Μέση Καθυστέρηση (Average Latency)
3. Overhead Ratio

Τα αποτελέσματα της προσομοίωσης δείχνουν ότι τα πρωτόκολλα δρομολόγησης DTN, γενικά, παρουσιάζουν καλύτερη απόδοση στο μοντέλο κίνησης SPMB από άλλα μοντέλα κίνησης (RW και RD) επειδή αποδίδουν μέγιστη πιθανότητα παράδοσης μηνυμάτων, ελάχιστο overhead ratio (για B-SNW και SNF), και ελάχιστη μέση καθυστέρηση.

Όπως και στο προαναφερθέν paper, έτσι και στη δική μας υλοποίηση, γίνεται προσομοίωση διαφορετικών τοπολογιών και σεναρίων για ασύρματους κινητούς κόμβους. Στο πρώτο μας πείραμα, εξετάζεται ο διαφορετικός αριθμός κινητών κόμβων για διαφορετικά πρωτόκολλα δρομολόγησης DTN. Στη συνέχεια, σ' ένα μέρος των πειραμάτων μας, εξετάζονται διαφορετικά μοντέλα κινητικότητας σε συνδυασμό με τα τρία πρωτόκολλα δρομολόγησης που χρησιμοποιούμε. Έτσι καταλήγουμε κι εμείς σε συμπεράσματα σχετικά με μετρικές όπως η Μέση Καθυστέρηση για διαφορετικά σενάρια κινητικότητας.

### 2.2.3. Εφαρμογή τεχνικών Μηχανικής Μάθησης σε δρομολόγηση δικτύων ανθεκτικών στις καθυστερήσεις

Η διατριβή με τίτλο “Εφαρμογή τεχνικών Μηχανικής Μάθησης σε δρομολόγηση δικτύων ανθεκτικών στις καθυστερήσεις” (Application of machine learning techniques to Delay Tolerant Network Routing)[9] πραγματεύεται διάφορες τεχνικές μηχανικής μάθησης για τη βελτίωση της δρομολόγησης σε δίκτυα με ανοχή στις καθυστερήσεις (DTN). Χρησιμοποιούνται ιστορικά στατιστικά στοιχεία δικτύου για την εκπαίδευση ενός ταξινομητή (classifier) για την πρόβλεψη αξιόπιστων διαδρομών μέσω του δικτύου. Επιπλέον χρησιμοποιείται μία τεχνική ομαδοποίησης για την πρόβλεψη των μελλοντικών τοποθεσιών των κινητών κόμβων. Αυτές οι τεχνικές χρησιμοποιούνται για να μειώσουν την κατανάλωση πόρων όπως το εύρος ζώνης, τη μνήμη και την αποθήκευση δεδομένων που απαιτείται από τις μεθόδους δρομολόγησης με αντιγραφή (π.χ. ProPHET, epidemic) που χρησιμοποιούνται συχνά σε ευκαιρικά περιβάλλοντα DTN.

Η συγκεκριμένη προσέγγιση μπορεί να μην έχει κοινό στόχο μελέτης με τη δική μας αλλά υπάρχουν κοινά χαρακτηριστικά στα δίκτυα τα οποία προσομοιώνουμε. Χρησιμοποιεί τον εξομοιωτή CORE για μέρος των πειραμάτων με DTN ασύρματα δίκτυα με κινητούς κόμβους. Για το λόγο αυτό συνέβαλε στην επιλογή του CORE ως το περιβάλλον πραγματοποίησης των πειραμάτων αλλά και στην επιλογή κάποιων εκ των σεναρίων κινητικότητας που δοκιμάσαμε κι εμείς.



## 3. Εργαλεία

Στην ενότητα αυτή θ' αναλυθούν τα εργαλεία τα οποία χρησιμοποιήθηκαν για την εκπόνηση της παρούσας διπλωματικής. Θα ξεκινήσουμε από το βασικότερο, το οποίο είναι τα Docker Container, το οποίο αποτελεί τη βασική τεχνολογία η οποία λειτούργησε σα βάση για ολόκληρη τη υλοποίηση. Θα συνεχίσουμε με τον εξομοιωτή που χρησιμοποιήσαμε για τα πειράματα, τον CORE, ο οποίος εξομοιώνει δίκτυα και παρέχει πολλές δυνατότητες παραμετροποίησης. Τέλος θ' αναφερθούμε στον NDN daemon που χρησιμοποιούμε, τον NFD, στην υλοποίηση IBR\_DTN αλλά και στο εργαλείο που χρησιμοποιήθηκε για την παραγωγή των σεναρίων κινητικότητας, το BonnMotion.

### 3.1. Docker Container

Τα Container είναι μια παλιά έννοια. Για δεκαετίες, τα συστήματα UNIX είχαν την εντολή chroot που παρέχει μια απλή μορφή απομόνωσης συστήματος αρχείων. Έπειτα από πολλές εξελίξεις στον τομέα αυτό, το 2013, το Docker έφερε τα τελευταία κομμάτια στο παζλ του containerization και η τεχνολογία αυτή άρχισε να μπαίνει στις επικρατούσες τάσεις. Το Docker πήρε την υπάρχουσα τεχνολογία των Linux Container (LXC)[38] και την επέκτεινε με διάφορους τρόπους, κυρίως μέσω φορητών εικόνων και της φιλικής προς το χρήστη διεπαφής, ώστε να δημιουργήσει μια ολοκληρωμένη λύση για τη δημιουργία και τη διανομή container.

Η πλατφόρμα του Docker έχει δύο ξεχωριστά στοιχεία: το Docker Engine, το οποίο είναι υπεύθυνο για δημιουργία και λειτουργία των container και το Docker Hub, μια υπηρεσία cloud για τη διανομή των container.

Το Docker Engine παρέχει μια γρήγορη και βολική διεπαφή για τη λειτουργία container. Πριν από αυτό, η λειτουργία ενός container χρησιμοποιώντας τεχνολογία όπως το LXC απαιτούσε αρκετά εξειδικευμένες γνώσεις και χειροκίνητες ρυθμίσεις. Το Docker Hub προσφέρει έναν τεράστιο αριθμό δημόσιων εικόνων από container για λήψη, επιτρέποντας έτσι στους χρήστες να ξεκινήσουν γρήγορα και ν' αποφύγουν την επανάληψη εργασιών που έχουν ήδη γίνει από άλλους.

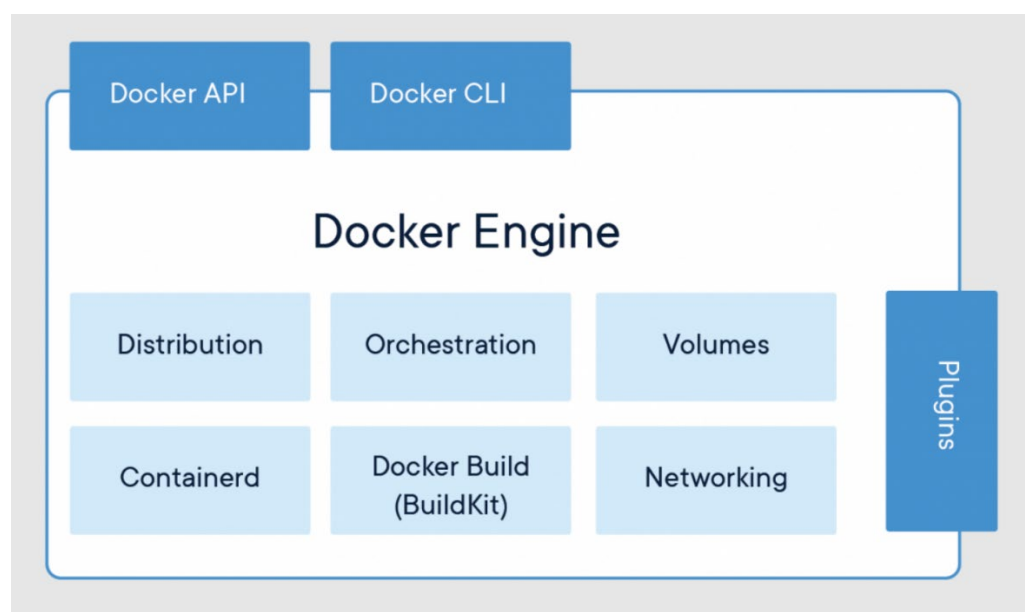
Η χρήση των container καθοδηγήθηκε σε μεγάλο βαθμό από προγραμματιστές, στους οποίους για πρώτη φορά δόθηκαν τα εργαλεία για την αποτελεσματική χρήση των container. Ο μικρός χρόνος εκκίνησης των Docker container είναι απαραίτητος για προγραμματιστές που επιθυμούν γρήγορους και επαναλαμβανόμενους κύκλους ανάπτυξης προγραμμάτων, όπου μπορούν να δουν άμεσα τα αποτελέσματα των αλλαγών στον κώδικα. Η φορητότητα και απομόνωση εγγυώνται ότι τα container θα διευκολύνουν πολλές λειτουργίες και τη συνεργασία με άλλους προγραμματιστές. Οι προγραμματιστές μπορούν να είναι σίγουροι ότι ο κώδικας τους θα λειτουργεί σε διαφορετικά περιβάλλοντα και οι λειτουργίες μπορούν να εστιάσουν στη οργάνωση και ρύθμιση των container αντί του κώδικα που τρέχει μέσα σε αυτά.

Οι αλλαγές που επιφέρει το Docker αλλάζουν σημαντικά τον τρόπο που αναπτύσσουμε λογισμικό. Χωρίς το Docker, τα container θα είχαν παραμείνει στη σκιά του IT για πολύ καιρό ακόμη.

## Docker Engine

Το Docker Engine[39] είναι μια τεχνολογία containerization ανοιχτού κώδικα για τη δημιουργία και τη μεταφορά εφαρμογών. Το Docker Engine λειτουργεί ως εφαρμογή πελάτη-διακομιστή με:

- Έναν διακομιστή με μία διεργασία μεγάλης διάρκειας που διαχειρίζεται τα container και ονομάζεται docker daemon(dockerd).
- Διεπαφές προγραμματισμού εφαρμογών(APIs) οι οποίες καθορίζουν τις διεπαφές που μπορούν να χρησιμοποιήσουν τα προγράμματα για να επικοινωνήσουν και να καθοδηγήσουν το docker daemon.
- Μία διεπαφή γραμμής εντολών (command line interface-CLI) πελάτη



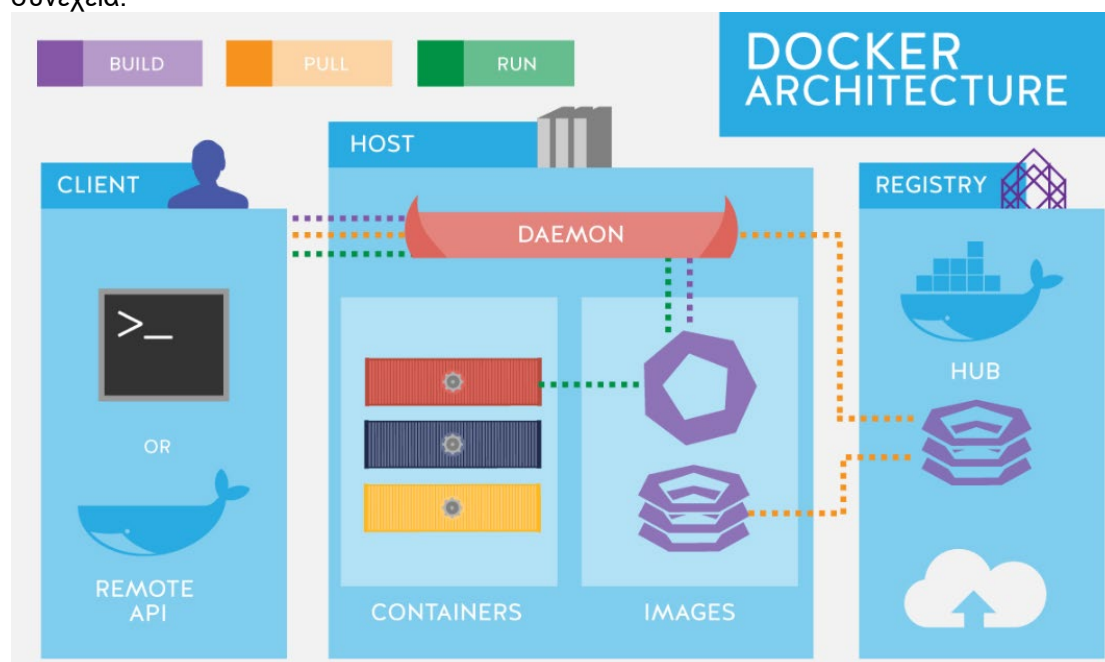
Εικόνα 13: Docker Engine[40].

Το CLI χρησιμοποιεί API Docker για τον έλεγχο ή την αλληλεπίδραση με τον Docker daemon μέσω σεναρίων ή απευθείας εντολών CLI. Πολλές άλλες εφαρμογές Docker χρησιμοποιούν το υποκείμενο API και το CLI. Ο daemon δημιουργεί και διαχειρίζεται αντικείμενα Docker, όπως εικόνες, container, δίκτυα και volumes.

## Αρχιτεκτονική του Docker

Το Docker χρησιμοποιεί μια αρχιτεκτονική πελάτη-διακομιστή[38]. Το πιο σημαντικό μέρος το οποίο είναι το κομμάτι της κατασκευής, λειτουργίας και διανομής των container γίνεται από το Docker daemon, με τον οποίο επικοινωνεί ο πελάτης Docker. Ο πελάτης και ο daemon μπορούν να εκτελούνται στο ίδιο σύστημα ή ένας πελάτης μπορεί να συνδεθεί με έναν απομακρυσμένο daemon. Ο πελάτης και ο Docker daemon επικοινωνούν χρησιμοποιώντας ένα REST API, μέσω υποδοχών UNIX ή διεπαφής δικτύου. Ένας άλλος πελάτης Docker είναι το Docker Compose, το οποίο θα αναλυθεί στη

συνέχεια.



Εικόνα 14: Αρχιτεκτονική του Docker[41].

## Αντικείμενα Docker(Docker objects)

Αντικείμενα στο Docker ονομάζονται οι οντότητες οι οποίες δημιουργούνται κατά τη χρήση του Docker και χρησιμοποιούνται για τη σύνθεση των εφαρμογών με Docker. Παρακάτω θ' αναλύσουμε τα βασικά αντικείμενα.

### Εικόνες(Images)

Μια εικόνα ενός docker container είναι ένα ελαφρύ, αυτόνομο, εκτελέσιμο πακέτο λογισμικού που περιλαμβάνει όλα όσα χρειάζονται για την εκτέλεση μιας εφαρμογής: τον κώδικα, το χρόνο εκτέλεσης, τα εργαλεία του συστήματος, τις βιβλιοθήκες συστήματος και τις ρυθμίσεις.

Μια εικόνα όμως είναι ένα πρότυπο το οποίο είναι μόνο προς ανάγνωση κι έχει ουσιαστικά οδηγίες για τη δημιουργία ενός Docker container. Συχνά, μια εικόνα βασίζεται σε άλλη εικόνα, με κάποια πρόσθετη προσαρμογή. Για παράδειγμα, μπορεί κάποιος να δημιουργήσει μια εικόνα που βασίζεται στην εικόνα του λειτουργικού ubuntu, αλλά εγκαθιστά πάνω σε αυτό την εφαρμογή του, καθώς και τις λεπτομέρειες διαμόρφωσης που απαιτούνται για να εκτελεστεί η εφαρμογή αυτή.

Υπάρχει η δυνατότητα δημιουργίας νέων εξατομικευμένων εικόνων και η δυνατότητα χρήσης αυτών που δημιουργήθηκαν από άλλους και δημοσιεύτηκαν σε μητρώο. Για να δημιουργηθεί μία νέα εικόνα, προϋποθέτει τη δημιουργία ενός Dockerfile με μια απλή σύνταξη για τον καθορισμό των βημάτων που απαιτούνται για τη δημιουργία της εικόνας και την εκτέλεσή της. Κάθε εντολή σε ένα Dockerfile δημιουργεί ένα στρώμα στην εικόνα. Όταν γίνονται αλλαγές στο Dockerfile και ξαναχτίζεται η εικόνα, μόνο αυτά τα επίπεδα που έχουν αλλάξει ξαναχτίζονται. Αυτό είναι ένα από τα χαρακτηριστικά που κάνουν τις εικόνες τόσο ελαφριές, μικρές και γρήγορες, σε σύγκριση με άλλες τεχνολογίες εικονοποίησης.

## Containers

Ένα container είναι μια τυπική μονάδα λογισμικού που “συσκευάζει” τον κώδικα και όλες τις εξαρτήσεις του, ώστε η εφαρμογή να τρέχει γρήγορα και αξιόπιστα από το ένα υπολογιστικό περιβάλλον στο άλλο.

Το container είναι ουσιαστικά το αποτέλεσμα της εκτέλεσης μιας εικόνας. Υπάρχουν οι δυνατότητες δημιουργίας, εκκίνησης, τερματισμού, μετακίνησης και διαγραφής ενός container χρησιμοποιώντας το Docker API ή το CLI. Ένα container μπορεί να συνδεθεί σε ένα ή περισσότερα δίκτυα, να συνδεθεί ένας χώρος αποθήκευσης σε αυτό ή ακόμη και να δημιουργηθεί μια νέα εικόνα με βάση την τρέχουσα κατάστασή του.

Από προεπιλογή, ένα container είναι αρκετά απομονωμένο από άλλα container και τον κεντρικό υπολογιστή που το φιλοξενεί. Το πόσο απομονωμένο είναι το δίκτυο ενός container, ο χώρος αποθήκευσης του αλλά και τα υπόλοιπα υποσυστήματα ενός container, από άλλα container ή από τον κεντρικό υπολογιστή, είναι ελέγξιμο από το δημιουργό του.

Ένα container ορίζεται από την εικόνα του καθώς και από τις επιλογές παραμετροποίησης που του παρέχονται κατά τη δημιουργία ή την εκκίνησή του. Όταν αφαιρείται ένα container, τυχόν αλλαγές στην κατάστασή του που δεν είναι αποθηκευμένες σε μόνιμο χώρο αποθήκευσης, εξαφανίζονται.

## Docker Volume

Όταν σταματήσει ένα container, τα αρχεία που δημιουργήθηκαν ή αντιγράφηκαν όσο αυτό έτρεχε, διαγράφονται. Το Docker Volume είναι μια λύση για την περίπτωση όπου θέλουμε να σώσουμε στον υπολογιστή μας αρχεία μέσα από ένα container. Ουσιαστικά είναι σα να δημιουργούμε με αυτόν τον τρόπο μία σύνδεση ανάμεσα στο φάκελο που είναι αποθηκευμένος τοπικά στον υπολογιστή μας, με το εσωτερικό του container. Όποια αλλαγή συμβαίνει στα δεδομένα αυτού του φακέλου από τη μία πλευρά, συμβαίνει και στην άλλη. Τα volume αποθηκεύονται ανεξάρτητα από το χρόνο ζωής του container. Αυτό επιτρέπει στους χρήστες να δημιουργούν αντίγραφα ασφαλείας των δεδομένων που υπάρχουν εντός του container και να μοιράζονται συστήματα αρχείων με ευκολία μεταξύ των container.

## Docker Daemon

Ο δαίμονας Docker (dockerd) ακούει σε αιτήματα Docker API και διαχειρίζεται αντικείμενα Docker όπως εικόνες (images), container, δίκτυα και volumes. Ένας daemon μπορεί επίσης να επικοινωνήσει με άλλους daemon για τη διαχείριση των υπηρεσιών Docker.

## Docker client

Ο Docker πελάτης (client) είναι ο κύριος τρόπος αλληλεπίδρασης πολλών χρηστών του Docker με το ίδιο το Docker. Όταν χρησιμοποιούνται εντολές όπως docker run, ο client στέλνει αυτές τις εντολές στο daemon ο οποίος τις εκτελεί. Η εντολή docker χρησιμοποιεί το Docker API. Ο Docker client μπορεί να επικοινωνήσει με περισσότερους από έναν daemon.

## Μητρώα Docker (Docker Registries)

Το μητρώο Docker αποθηκεύει τις εικόνες Docker. Το Docker Hub είναι ένα δημόσιο μητρώο που μπορεί να χρησιμοποιήσει ο καθένας και η αναζήτηση εικόνων στο Docker Hub αποτελεί για το Docker προεπιλεγμένη ρύθμιση. Όμως, δίνεται η δυνατότητα να έχει κάποιος και το δικό του ιδιωτικό μητρώο. Όταν χρησιμοποιούνται οι εντολές docker pull ή docker run, οι

ζητούμενες εικόνες λαμβάνονται από το διαμορφωμένο μητρώο. Όταν χρησιμοποιείται η εντολή `docker push` η εικόνα μεταφέρεται στο διαμορφωμένο μητρώο.

## Docker Compose

Το Docker Compose [42] είναι ένα εργαλείο για τον καθορισμό και την εκτέλεση εφαρμογών Docker που αποτελούνται από πολλαπλά container. Χρησιμοποιεί αρχεία YAML για τη διαμόρφωση των υπηρεσιών της εφαρμογής και εκτελεί τη διαδικασία δημιουργίας και εκκίνησης όλων των container με μία μόνο εντολή. Το βοηθητικό πρόγραμμα `docker-compose CLI` επιτρέπει στους χρήστες να εκτελούν εντολές ταυτόχρονα σε πολλά container, για παράδειγμα, τη δημιουργία εικόνων, την κλιμάκωση των container, εκκίνηση container που έχουν σταματήσει και άλλα. Το αρχείο `docker-compose.yml` χρησιμοποιείται για τον καθορισμό των υπηρεσιών μιας εφαρμογής και περιλαμβάνει διάφορες επιλογές παραμετροποίησης.

Η χρήση του Docker Compose είναι μία διαδικασία τριών βημάτων.

1. Ορισμός του περιβάλλοντος της εφαρμογής με ένα `Dockerfile`, ώστε να μπορεί να αναπαραχθεί οπουδήποτε.
2. Ορισμός των υπηρεσιών που αποτελούν την εφαρμογή στο `docker-compose.yml`, ώστε να μπορούν να εκτελούνται μαζί σε ένα απομονωμένο περιβάλλον.
3. Εκτέλεση της εντολής `docker-compose up` ώστε να ξεκινήσει να εκτελείται ολόκληρη η εφαρμογή.

## Επιλογή Χρήσης Docker Container

Στη δική μας περίπτωση, επιλέχθηκε να χρησιμοποιηθεί η τεχνολογία των Docker container λόγω των πλεονεκτημάτων που παρουσιάζει σε σχέση με μία αντίστοιχη υλοποίηση η οποία θα μπορούσε να γίνει τοπικά σ' έναν υπολογιστή. Πρώτο και κύριο κριτήριο αποτέλεσε η φορητότητα η οποία προσφέρεται από τα container, η οποία μας δίνει τη δυνατότητα να αναπαράγουμε την υλοποίηση αυτή σε οποιοδήποτε σύστημα που εκτελεί το Docker, έχοντας την ίδια ακριβώς απόδοση. Η δυνατότητα του Docker, ν' αναπτύξει κάποιος μία ήδη υπάρχουσα εικόνα, είναι επίσης πολύ σημαντική για τη δική μας υλοποίηση. Παρέχεται λοιπόν με αυτόν τον τρόπο ένα χρήσιμο εργαλείο, το οποίο μπορεί να συμπληρωθεί από νέες δυνατότητες και να εξελιχθεί ακόμη περισσότερο. Τα δύο αυτά στοιχεία, κάνουν πιο εύκολη την χρησιμοποίηση του container που φτιάξαμε, όχι μόνο για την αναπαραγωγή των ίδιων πειραμάτων και για την εξέλιξή τους, αλλά και για την πιθανή υλοποίηση μίας μεγάλης γκάμας πειραμάτων που βασίζονται στις τεχνολογίες NDN,DTN και UMOBILE μέσα στο περιβάλλον του CORE.

Βεβαίως δε θα μπορούσε να παραβλεφθεί το γεγονός ότι τα container είναι γρήγορα στη δημιουργία και την εκκίνηση τους και έχουν χαμηλές απαιτήσεις σε υπολογιστικούς πόρους, γεγονός που τα κάνει ιδιαίτερα εύχρηστα. Η εγκατάσταση ενός λογισμικού σ' ένα container, αντί του υπολογιστή μας, βοηθά επίσης να αποφευχθεί τυχόν έλλειψη συμβατότητας με άλλα πακέτα στο περιβάλλον όπου εγκαθίσταται το πρόγραμμα, αφού όπως προαναφέρθηκε, το container είναι απομονωμένο και μέσα σ' αυτό υπάρχουν μόνοι οι απαραίτητες βιβλιοθήκες και ρυθμίσεις για την εφαρμογή μας.

Συνεπώς λόγω της διαθεσιμότητας του τόσο για Linux, macOS ,όσο και για Windows, το containerized λογισμικό που αναπτύχθηκε θα λειτουργεί πάντα το ίδιο, ανεξάρτητα από την υποδομή υλικού.

## 3.2. Common Open Research Emulator

Ο Common Open Research Emulator, ή CORE[43], είναι ένα εργαλείο για τη δημιουργία εικονικών δικτύων. Ως εξομοιωτής, το CORE δημιουργεί μια αναπαράσταση ενός πραγματικού δικτύου υπολογιστών που λειτουργεί σε πραγματικό χρόνο, σε αντίθεση με την προσομοίωση, όπου χρησιμοποιούνται αφηρημένα μοντέλα. Εξομοιώνει δρομολογητές, προσωπικούς υπολογιστές και άλλους κεντρικούς υπολογιστές και προσομοιώνει τους συνδέσμους δικτύου μεταξύ τους. Επειδή είναι ένας εξομοιωτής που λειτουργεί σε πραγματικό χρόνο, αυτά τα δίκτυα μπορούν να συνδεθούν σε πραγματικό χρόνο με φυσικά δίκτυα και δρομολογητές. Παρέχει ένα περιβάλλον για την εκτέλεση πραγματικών εφαρμογών και πρωτοκόλλων, αξιοποιώντας τα εργαλεία που παρέχονται από το λειτουργικό σύστημα Linux.

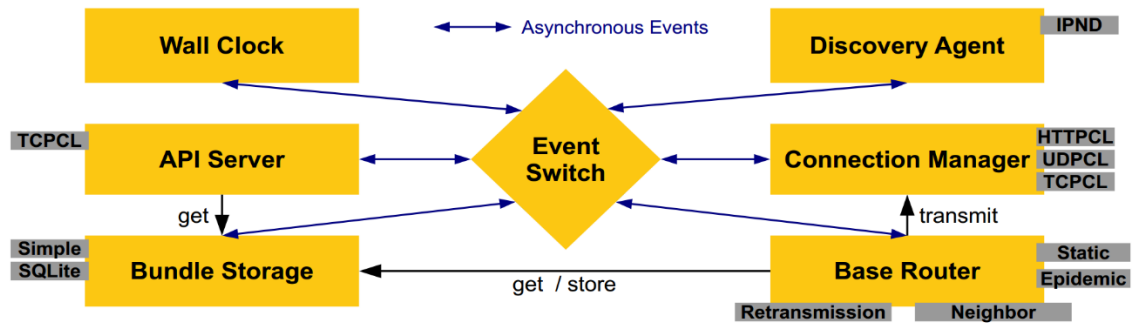
Το αρκτικόλεξο πηγάζει από την αρχική χρήση αυτού του εξομοιωτή για μελέτη πρωτοκόλλων δρομολόγησης ανοικτού κώδικα, αλλά οι δυνατότητες του CORE έχουν επεκταθεί για υποστήριξη ασύρματων δικτύων. Χρησιμοποιείται συνήθως για έρευνα δικτύου και πρωτοκόλλων, επιδείξεις, δοκιμές εφαρμογών και πλατφορμών, αξιολόγηση σεναρίων δικτύωσης, μελέτες ασφαλείας και δοκιμαστική αύξηση του μεγέθους φυσικών δικτύων.

## 3.3. IBR-DTN

Το IBR-DTN[33], μια ελαφριά και φορητή υλοποίηση του πρωτοκόλλου Bundle και του DTN δαίμονα (DTN daemon). Το IBR-DTN έχει ένα μικρό αποτύπωμα μνήμης και μπορεί να χρησιμοποιηθεί σχεδόν σε κάθε πλατφόρμα που υποστηρίζει Linux, ειδικά σε ενσωματωμένες πλατφόρμες. Αυτό επιτρέπει την ανάπτυξη μιας πλήρους έκδοσης του DTN προτεινόμενη για ασύρματα δίκτυα αισθητήρων.

Καθώς το IBR-DTN προορίζεται για ενσωματωμένες πλατφόρμες, είναι πλήρως συμβατό με το uClibc, μια βιβλιοθήκη συστήματος ειδικά προσαρμοσμένη για ενσωματωμένα συστήματα. Ακολουθήθηκαν δύο γενικές κατευθυντήριες γραμμές για την ανάπτυξη του IBR-DTN: Ένας στόχος είναι να διατηρηθούν οι εξωτερικές απαιτήσεις (π.χ. βιβλιοθήκες), στο ελάχιστο. Συχνά αυτές δεν είναι διαθέσιμες σε συγκεκριμένες πλατφόρμες, επειδή δεν έχουν μεταφερθεί ή επειδή ο χώρος που απαιτείται για όλα αυτά τα εξαρτήματα, θα ήταν τόσο μεγάλος που καθίσταται απαγορευτικό. Ένα μικρό σύνολο εξωτερικών εξαρτήσεων διασφαλίζει επίσης ότι η μεταφορά του IBR-DTN σε νέες πλατφόρμες είναι σχετικά εύκολη. Ένας άλλος στόχος είναι να διατηρηθεί το λογισμικό σε όσο το δυνατόν πιο σπονδυλωτή μορφή γίνεται. Η λειτουργικότητα στο IBR-DTN εφαρμόζεται σε χαλαρά συνδεδεμένες μονάδες που επικοινωνούν χρησιμοποιώντας ένα μηχανισμός συμβάντων. Έτσι, κατά τη μεταγλώττιση, το IBR-DTN μπορεί να προσαρμοστεί ειδικά στις δυνατότητες της χρησιμοποιούμενης πλατφόρμας. Επίσης, εάν κάποια λειτουργικότητα εξαρτάται από εξωτερικές βιβλιοθήκες, η εξάρτηση εφαρμόζεται μόνο για το προαιρετικό τμήμα.





Εικόνα 15: Αρχιτεκτονική του IBR-DTN[44].

Το IBR-DTN παρέχει τις υλοποιήσεις δρομολόγησης που αναλύσαμε νωρίτερα, δηλαδή τη δρομολόγηση πλημμύρας (Flooding), τη δρομολόγηση επιδημίας (Epidemic) και τη δρομολόγηση PROPHET.

Συγκεκριμένα στην υλοποίηση της Επιδημικής δρομολόγησης που χρησιμοποιείται από τον IBR-DTN, στόχος του είναι να μεταφέρει όλα τα πακέτα σε όλους τους κόμβους που συμμετέχουν και έτσι αν δεν υπάρχει περιορισμός στους πόρους, μπορεί να εγγυηθεί την τελική παράδοση ενός bundle. Αντί να χρησιμοποιεί διανύσματα αθροισμάτων (summary vectors), όπως σε άλλες υλοποιήσεις, χρησιμοποιεί ένα πιο αποτελεσματικό μηχανισμό που ονομάζεται φίλτρο Bloom [45], το οποίο είναι μία δομή δεδομένων πιθανολογικά αποδοτική, που χρησιμοποιείται για να ελέγξει εάν ένα στοιχείο είναι μέρος ενός συνόλου και έχει προταθεί ως βελτιστοποίηση στο [19]. Ειδικότερα, τα φίλτρα Bloom χρησιμεύουν στην ανταλλαγή πληροφοριών σχετικά με το περιεχόμενο των buffer πριν από την αποστολή πληροφοριών, προκειμένου να αποφευχθούν περιττές αναμεταδόσεις. Ως προέκταση της επιδημικής δρομολόγησης, το IBR-DTN διαχειρίζεται ένα διάνυσμα καθαρισμού (purge vector) που διανέμεται σε όλους τους γειτονικούς κόμβους, εξασφαλίζοντας ότι τα παραδοθέντα πακέτα διαγράφονται σε όλο το δίκτυο.

Στο επόμενο κεφάλαιο θα γίνει εκτενής αναφορά της υλοποίησης των πειραμάτων μας, όμως σε αυτό το σημείο, αξίζει να σημειωθεί πως ο IBR-DTN τρέχει σε κάθε κόμβο της τοπολογίας μας. Χρησιμοποιείται δηλαδή αυτή η υλοποίηση και στους ενδιάμεσους κινητούς κόμβους αλλά και στους ακριανούς που είναι σταθεροί. Περισσότερες λεπτομέρειες σχετικά με την τοπολογία αλλά και τη χρήση του στα πειράματα θα υπάρξουν στη συνέχεια.

### 3.4. Named Data Networking Forwarding Daemon (NFD)

Ο Named Data Networking Forwarding Daemon (NFD) [46] είναι ένας προωθητής δικτύου που εφαρμόζεται και εξελίσσεται μαζί με το πρωτόκολλο Named Data Networking (NDN) και αποτελεί ένα δομικό συστατικό της πλατφόρμας NDN. Αν και η πρώτη κυκλοφορία του έγινε από τα μέλη της ομάδας έργου NDN που χρηματοδοτήθηκε από το NSF, ο NFD αναπτύσσεται μέσω της κοινότητας του NDN και περιέχει ήδη σημαντικές συνεισφορές από άτομα εκτός της ομάδας έργου. Ο κύριος σχεδιαστικός στόχος του NFD είναι να υποστηρίξει ποικίλους πειραματισμούς της τεχνολογίας NDN.



Ο σχεδιασμός δίνει έμφαση στην τμηματοποίηση και την επεκτασιμότητα ώστε να επιτρέπονται εύκολα πειράματα με νέες δυνατότητες πρωτοκόλλου, αλγόριθμους και εφαρμογές. Ο κώδικας δεν είναι πλήρως βελτιστοποιημένος από πλευράς απόδοσης διότι η βασική λογική είναι ότι οι βελτιστοποιήσεις απόδοσης είναι κάτι που μπορεί να πραγματοποιηθεί από τους προγραμματιστές δοκιμάζοντας διαφορετικές δομές δεδομένων και διαφορετικούς αλγόριθμους και με την πάροδο του χρόνου, μπορεί να προκύψουν καλύτερες υλοποιήσεις στο ίδιο πλαίσιο σχεδιασμού.

Ο NFD διαθέτει τα ακόλουθα δομικά στοιχεία:

### **Πυρήνας**

Παρέχει διάφορες κοινές υπηρεσίες που μοιράζονται μεταξύ διαφορετικών ενότητων NFD. Αυτές περιλαμβάνουν ρουτίνες υπολογισμού κατακερματισμού, αρχεία παραμετροποίησης, παρακολούθηση των faces και άλλα.

### **Faces**

Εφαρμόζεται η αφαιρετικότητα των NDN faces πάνω από διαφορετικούς μηχανισμούς μεταφοράς χαμηλότερου επιπέδου.

### **Πίνακες**

Υπάρχουν οι δομές δεδομένων των Content Store (CS), Pending Interest Table (PIT), Forwarding Information Base (FIB) και άλλες δομές δεδομένων για την υποστήριξη της προώθησης Πακέτων Δεδομένων και Ενδιαφέροντος.

### **Πρωώθηση**

Εφαρμόζει βασικές διαδρομές επεξεργασίας πακέτων, οι οποίες αλληλεπιδρούν με τα faces, τους πίνακες και τις στρατηγικές. Ένα σημαντικό μέρος της πρωώθησης αποτελεί η Υποστήριξη Στρατηγικής (Strategy Support) η οποία εφαρμόζει ένα πλαίσιο για την υποστήριξη διαφορετικών στρατηγικών πρωώθησης.

### **Διαχείριση NDF**

Εφαρμόζεται το Πρωτόκολλο Διαχείρισης NFD, το οποίο επιτρέπει στις εφαρμογές να παραμετροποιούν τον NFD και να ορίζουν/αιτούνται εσωτερικές καταστάσεις του NFD. Η αλληλεπίδραση πρωτοκόλλου γίνεται μέσω ανταλλαγής Ενδιαφέροντος/Δεδομένων του NDN μεταξύ εφαρμογών και NFD.

### **Διαχείριση της Βάσης Πληροφοριών Δρομολόγησης**

Η Βάση Πληροφοριών Δρομολόγησης (Routing Information Base-RIB) μπορεί να ενημερώνεται από διαφορετικά μέρη με διαφορετικούς τρόπους, συμπεριλαμβανομένων διαφόρων πρωτοκόλλων δρομολόγησης, καταχωρίσεων προθέματος εφαρμογής και χειρισμού γραμμής εντολών από τους διαχειριστές του συστήματος. Η μονάδα διαχείρισης RIB επεξεργάζεται όλα αυτά τα αιτήματα για να δημιουργήσει έναν πίνακα προώθησης και στη συνέχεια το συγχρονίζει με την FIB του NFD, η οποία περιέχει μόνο τις ελάχιστες πληροφορίες που απαιτούνται για τις αποφάσεις προώθησης. Στην πραγματικότητα, η διαχείριση RIB είναι μέρος της ενότητας διαχείρισης NFD. Ωστόσο, λόγω της σπουδαιότητάς της όσον αφορά τις συνολικές λειτουργίες και την πιο περίπλοκη επεξεργασία τους, την αναφέρουμε ως ξεχωριστή ενότητα.

### 3.5. BonnMotion

Το BonnMotion[47] είναι ένα λογισμικό Java ανοιχτού κώδικα που δημιουργεί και αναλύει σενάρια κινητικότητας. Έχει αναπτυχθεί στο Πανεπιστήμιο της Βόννης στη Γερμανία, όπου λειτουργεί ως εργαλείο για τη διερεύνηση των χαρακτηριστικών σεναρίων δικτύων με πολλαπλούς κινητούς κόμβους. Τα σενάρια κινητικότητας που δημιουργεί το BonnMotion, μπορούν επίσης να εξαχθούν για προσομοιωτές δικτύου ns-2, GloMoSim/QualNet, COOJA και MiXiM. Στην παρούσα διπλωματική χρησιμοποιήθηκαν σενάρια κινητικότητας ns-2 διότι είναι συμβατά με τον εξομοιωτή δικτύων που χρησιμοποιήσαμε.

## 4. Πειραματική Διαδικασία

Στο κεφάλαιο αυτό θα γίνει παρουσίαση της διαδικασίας που ακολουθήθηκε για την εκπόνηση των πειραμάτων. Πιο συγκεκριμένα θα πούμε λίγα πράγματα για το Docker container που δημιουργήσαμε, θ' αναφερθούμε στη μεθοδολογία που ακολουθήθηκε για το στήσιμο των πειραμάτων, την τοπολογία δικτύου που χρησιμοποιήσαμε και κάποιες λεπτομέρειες σχετικά με τα μοντέλα κινητικότητας που χρησιμοποιήθηκαν.

Σκοπός των πειραμάτων ήταν να μελετήσουμε την επίδραση διαφορετικών DTN routing σε ασύρματα δίκτυα με κινητούς κόμβους, τα οποία χρησιμοποιούν την τεχνολογία NDN-over-DTN, στο περιβάλλον εξομοίωσης του CORE. Επίσης, εξετάζουμε την επίδραση της αλλαγής των εξής παραμέτρων στο δίκτυο.

- Ο χρόνος για τον οποίον ένα πακέτο Δεδομένων θεωρείται "πρόσφατο" (Freshness Period)
- Ο χρόνος ζωής ενός πακέτου ενδιαφέροντος (Lifetime)
- Το ποσοστό απώλειας πακέτων στο δίκτυο (% of Loss)

Το σενάριο που υποθέτουμε ότι προσομοιώνουμε είναι ένα σενάριο IoT. Πιο συγκεκριμένα, θεωρούμε πως έχουμε κάποιους αισθητήρες τους οποίους προσομοιώνουμε με έναν κόμβο-παραγωγό που βρίσκεται στο ένα άκρο του δικτύου σταθερός, έναν μεταβλητό αριθμό ενδιάμεσων κινητών έναν κόμβο-καταναλωτή σταθερό στην άλλη άκρη το δικτύου. Ο καταναλωτής αιτείται μία συγκεκριμένη μέτρηση και αυτή μεταφέρεται από τον παραγωγό σε αυτόν μέσω των ενδιάμεσων κόμβων και η διαδικασία αυτή επαναλαμβάνεται για έναν πεπερασμένο αριθμό μετρήσεων.

### 4.1. Δημιουργία Docker Container

Για την εκπόνηση των πειραμάτων δημιουργήθηκε ένα Docker container το οποίο προέκυψε από συνδυασμό μίας containerized υλοποίησης του CORE από το framework MACI[48] και ενός Dockerfile που προσομοιώνει NDN-DTN δίκτυο με 3 container που συνδέονται μεταξύ τους και το καθένα απ' αυτά αποτελεί εάν κόμβο του δικτύου. Εκμεταλλευόμενοι λοιπόν τη δυνατότητα που προσφέρει το Docker ν' αξιοποιήσεις μία ήδη υπάρχουσα εικόνα, συνδυάσαμε τις δύο αυτές υλοποιήσεις, προσθέσαμε κάποια εργαλεία και απαραίτητες βιβλιοθήκες και καταλήξαμε στο Dockerfile που χρησιμοποιήσαμε. Στο σημείο αυτό θα πρέπει να σημειωθεί πως το Dockerfile αλλά και οι υπόλοιποι κώδικες στους οποίους θ' αναφερθούμε από εδώ και στο εξής, παρατίθεται στο Παράρτημα. Στη συνέχεια, δημιουργήσαμε yml αρχείο, μέσω του οποίου χρησιμοποιούμε το εργαλείο Docker-Compose το οποίο μας εξυπηρετεί για τη διαμόρφωση υπηρεσιών αλλά και για την ευκολότερη εκκίνηση του container μας. Είναι σημαντικό να σημειωθεί πως μέσα από το compose, δημιουργήσαμε ένα volume το οποίο μας επέτρεπε να πάρουμε τ' αποτελέσματα των πειραμάτων μας από το container και να τα αποθηκεύσουμε τοπικά στον υπολογιστή μας ώστε να μπορέσουμε να τα επεξεργαστούμε στη συνέχεια. Κατά την εκκίνηση του container από το terminal, ανοίγει αυτόματα το περιβάλλον της εφαρμογής CORE, το οποίο θα δούμε πως μοιάζει στη συνέχεια.

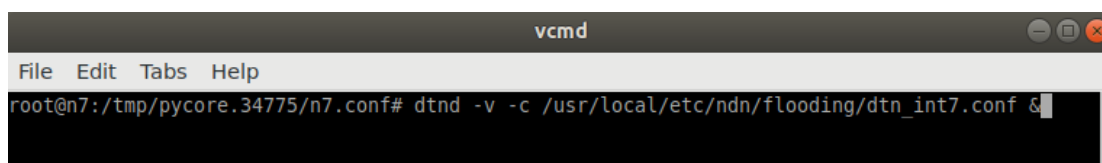
## 4.2. Τοπολογία πειράματος

Όπως έχουμε αναφέρει ξανά, το σενάριο μας, προσομοιώνει ένα σενάριο IoT. Για το λόγο αυτό, θεωρήσαμε δύο απομακρυσμένες περιοχές σε δύο άκρα της περιοχής προσομοίωσης. Η μια περιοχή είναι η περιοχή του consumer-καταναλωτή και η άλλη του producer-παραγωγού. Αναφερόμαστε σε περιοχές και όχι σε σημεία διότι όλοι οι κόμβοι μας είναι ασύρματοι, με συγκεκριμένη ακτίνα την οποία θ' αναφέρουμε στη συνέχεια. Οι περιοχές λοιπόν αυτές, συνδέονται μεταξύ τους μέσω ασύρματων κινούμενων κόμβων. Το σύστημα αυτό μπορεί παραδείγματος χάριν να προσομοιώνει κάποιες απομακρυσμένες περιοχές στις οποίες δεν έχουμε πρόσβαση με άλλον τρόπο και χρειαζόμαστε τέτοιου τύπου ενδιάμεσους κόμβους για να παραλάβουν/παραδώσουν τα δεδομένα.

Για το στήσιμο της τοπολογίας του πειράματος λοιπόν, χρειαζόμαστε αρχικά δύο σταθερούς κόμβους στα άκρα, έναν αριθμό ενδιάμεσων κόμβων, ο οποίος ποικίλει ανάλογα με το πείραμα και ένα wireless LAN δίκτυο, στο οποίο πρέπει να συνδέσουμε όλους τους κόμβους μας για να μπορούν να επικοινωνούν μεταξύ τους. Αφού τοποθετήσουμε αυτά, θα πρέπει να ρυθμίσουμε και κάποιες παραμέτρους του δικτύου, όπως το μοντέλο κινητικότητας το οποίο θέλουμε να χρησιμοποιήσουμε, την ακτίνα των κόμβων του δικτύου κ.α. Μόλις ρυθμιστούν αυτές οι παράμετροι, μπορεί να γίνει εκκίνηση της λειτουργίας του CORE.

## 4.3. Έναρξη πειράματος

Εκκίνηση της προσομοίωσης, δε συνεπάγεται και εκκίνηση του πειράματος μας. Για να εκκινήσουμε το εκάστοτε πείραμά μας, πρέπει να ξεκινήσουμε τον DTN daemon σε όλους τους κόμβους, το οποίο γίνεται με μία εντολή που καθορίζει το πρωτόκολλο δρομολόγησης που θα χρησιμοποιεί ο κάθε κόμβος. Επίσης πρέπει να εκκινήσουμε τον NFD στους producer και consumer και αφότου καταχωρηθούν οι υποθετικές τιμές αισθητήρων σ' αυτούς ν' αρχίσουν να ζητάνε και να διαθέτουν δεδομένα αντίστοιχα. Αυτή η διαδικασία γίνεται μέσα από scripts τα οποία ρυθμίζουν επίσης και το πρωτόκολλο δρομολόγησης DTN που θα χρησιμοποιούν ο καταναλωτής και ο παραγωγός των δεδομένων. Για να γίνει αυτό, πρέπει ενώ έχουμε εκκινήσει τη διαδικασία προσομοίωσης αλλά όχι το σενάριο κινητικότητας, ν' ανοίξουμε τον κάθε κόμβο ξεχωριστά και να τρέξουμε την εντολή ή το script, όπως φαίνεται στην εικόνα 16.

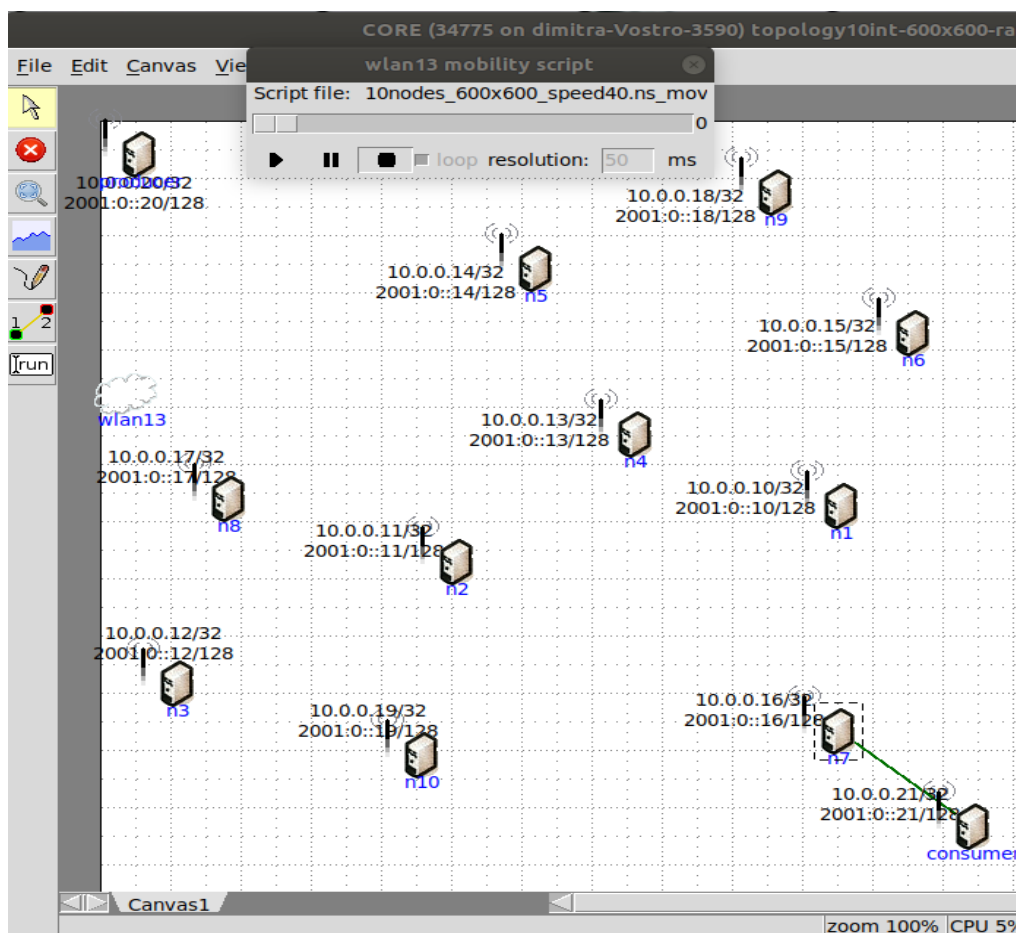


Εικόνα 16: Εντολή που τρέχουμε στους ενδιάμεσους κόμβους για την εκκίνηση του DTN daemon.

Όταν θα γίνει αυτό, θα πρέπει οι κόμβοι μας να μη βρίσκονται ο ένας στην εμβέλεια του άλλου και κυρίως σε σύνδεση με τον consumer, διότι αυτό σημαίνει πως θ' αρχίσουν ν' ανταλλάζουν πακέτα μεταξύ τους.

Για παράδειγμα στην Εικόνα 17, βλέπουμε πριν την εκκίνηση, πάνω αριστερά τον producer, κάτω δεξιά τον consumer και στη μέση τους 10 κινητούς ενδιάμεσους κόμβους από τους οποίους όλοι εκτός από τον n7, βρίσκονται σωστή θέση για εκκίνηση, αφού όπως βλέπουμε υπάρχει σύνδεση μεταξύ του n7 και του consumer. Αφότου εκκινήσουμε λοιπόν τις

απαραίτητες διεργασίες στους υπόλοιπους κόμβους, ξεκινάμε τελευταίο τον consumer, ο οποίος σηματοδοτεί και την έναρξη του πειράματος και αμέσως ξεκινάμε και το μοντέλο κινητικότητας.



Εικόνα 17: Παράδειγμα λάθος τοπολογίας στον CORE, πριν την εκκίνηση του πειράματος.

## 4.4. Μεθοδολογία

Ο producer έχει καταχωρημένα δεδομένα απο αισθητήρες τα οποία στέλνει στον consumer μέσω της στοίβας NDN-DTN κάθε φορά που λαμβάνει ένα πακέτο Ενδιαφέροντος (Interest Packet) για το συγκεκριμένο δεδομένο. Μόλις ο consumer λάβει το αντίστοιχο Πακέτο Δεδομένων (Data Packet) τότε στέλνει ένα πακέτο Ενδιαφέροντος (Interest Packet) στον producer το οποίο αναφέρεται σ' ένα δεδομένο που επιλέγεται τυχαία μεταξύ των διαθέσιμων καταχωρήσεων.

Το δεδομένο αυτό, μπορεί να τύχει να ζητηθεί ξανά μετά απο κάποια ώρα. Εάν κάποιο δεδομένο ζητηθεί ξανά μέσα στο χρονικό διάστημα κατά το οποίο το πακέτο Δεδομένων θεωρείται “πρόσφατο”(Data Freshness Period), το δεδομένο αυτό θα είναι αποθηκευμένο στην cache του consumer και θα ληφθεί από εκεί. Αυτό είναι και το λεγόμενο cache hit. Ειδάλλως, με την πάροδο του χρόνου που καθορίστηκε ως Freshness, το δεδομένο αυτό διαγράφεται απο την cache του consumer και θα πρέπει να σταλεί ξανά αίτημα στον producer.

Μία μετρική που χρησιμοποιούμε για ν' αξιολογήσουμε τα αποτελέσματα των πειραμάτων μας είναι η παράμετρος Cache Hit Ratio (h), η οποία υπολογίζεται από την εξίσωση (1), όπου η μεταβλητή Cache Hit είναι ο συνολικός αριθμός των αιτημάτων που ικανοποιούνται από τη

κρυφή μνήμη του consumer και η μεταβλητή Received Interests είναι ο συνολικός αριθμός των Πακέτων Ενδιαφέροντος για το οποία έχουμε λάβει αντίστοιχα πακέτα Δεδομένων (είτε από τον producer, είτε από τη κρυφή μνήμη του consumer).

$$h = \frac{\text{Cache Hit}}{\text{ReceivedInterests}} \quad (1)$$

Το πιο σημαντικό στοιχείο όμως για ν' αξιολογήσουμε την απόδοση του δικτύου είναι ο χρόνος καθυστέρησης λήψης των Πακέτων Δεδομένων (Delay). Το Delay, αναφέρεται στο χρόνο που περνάει από τη στιγμή που ένα Πακέτο Ενδιαφέροντος ξεκινάει από τον consumer, έως τη στιγμή που το αντίστοιχο Πακέτο δεδομένων φτάνει πίσω στον consumer. Επίσης στα πειράματα 4 και 5 θα χρησιμοποιηθεί ως μετρική αξιολόγησης και ο ρυθμός με τον οποίο ικανοποιούνται τα Πακέτα Ενδιαφέροντος (Interest Satisfaction Ratio-ISR). Ο ρυθμός αυτός υπολογίζεται από την εξίσωση (2), όπου η μεταβλητή Satisfied Interest είναι ο συνολικός αριθμός των Πακέτων Ενδιαφέροντος τα οποία έχουν ικανοποιηθεί, λαμβάνοντας τα αντίστοιχα Πακέτα Δεδομένων και η μεταβλητή Sent Interest είναι ο συνολικός αριθμός των απεσταλμένων Πακέτων Ενδιαφέροντος.

$$\text{ISR} = \frac{\text{SatisfiedInterests}}{\text{SentInterests}} \quad (2)$$

Ο CORE μας δίνει πάρα πολλές δυνατότητες παραμετροποίησης στο ασύρματο δίκτυο, όμως εμείς στην υλοποίηση μας χρησιμοποιούμε το μοντέλο κινητικότητας (mobility script), την ακτίνα των κόμβων του δικτύου (Range), την καθυστέρηση του δικτύου (Delay) και την απώλεια πακέτων (Loss). Το μοντέλο κινητικότητας όπως θα δούμε στη συνέχεια διαφοροποιείται σε κάποια πειράματα και από το μοντέλο αυτό ρυθμίζουμε το μέγεθος της επιφάνειας προσομοίωσης. Σε όλα τα πειράματα έχουμε καθορίσει μια τετράγωνη επιφάνεια προσομοίωσης, με πλευρά 600 μέτρα. Με βάση την επιφάνεια αυτή και μετά από δοκιμές που έγιναν, επιλέχθηκε η ακτίνα του κάθε κόμβου να είναι 100 μέτρα. Βάζοντας μεγαλύτερες τιμές ακτίνας, δημιουργούταν σε κάποιες περιπτώσεις συνεχής διαδρομή ανάμεσα στον consumer και τον producer και δοκιμάζοντας μικρότερες, είδαμε πως τα πειράματα έπαιρναν πολλές ώρες για να ολοκληρωθούν. Το delay του δικτύου είναι ρυθμισμένο σε όλα τα πειράματα στα 20ms. Με το ποσοστό απώλειας πακέτων, ασχολούμαστε μόνο στο πείραμα 5 και μέχρι τότε το διατηρούμε μηδενικό.

Ένα άλλο δεδομένο το οποίο παραμένει σταθερό σε όλα τα πειράματα αλλά δε σχετίζεται με τον CORE, είναι ο αριθμός των επαναλήψεων ο οποίος αντικατοπτρίζει τον αριθμό των Πακέτων Δεδομένων που πρέπει να λάβει ο consumer ως απάντηση στα Πακέτα Ενδιαφέροντος που στέλνει. Ο αριθμός των επαναλήψεων λοιπόν σε όλες τις περιπτώσεις είναι 20.

#### 4.4.1. Μοντέλα κινητικότητας

Για την υλοποίηση των πειραμάτων, έπρεπε να ορίσουμε με ποιον τρόπο θα κινούνται οι κόμβοι μας. Επιλέξαμε για τη χρήση αυτή, να παράγουμε μοντέλα κινητικότητας στη μορφή ns-2 mobility scripts, μέσω του BonnMotion, τα οποία είναι συμβατά με τον CORE. Από όλα τα πιθανά μοντέλα τα οποία μπορεί το εργαλείο αυτό να δημιουργήσει, εμείς επιλέξαμε τα



ακόλουθα τρία τα οποία βασίζονται σε μεγάλο βαθμό στην τυχαιότητα κι επίσης δεν ανήκουν στην κατηγορία των μοντέλων που δεν προσεγγίζουν ιδιαίτερα τα άκρα του δικτύου (όπως παραδείγματος χάριν Random Waypoint), κάτι το οποίο δε θα εξυπηρετούσε για τη δική μας περίπτωση.

## Random Walk

Το Random Walk Mobility Model (Μοντέλο Κινητικότητας Τυχαίου Περιπάτου[49]) είναι ένα μοτίβο κινητικότητας χωρίς μνήμη, διότι δε διατηρεί καμία γνώση σχετικά με τις προηγούμενες τοποθεσίες και τιμές ταχύτητας και περιγράφηκε για πρώτη φορά μαθηματικά από τον Αϊνστάιν το 1926. Δεδομένου ότι πολλές οντότητες κινούνται με εξαιρετικά απρόβλεπτους τρόπους, το μοντέλο Random Walk αναπτύχθηκε για να μιμηθεί αυτήν την ακανόνιστη κίνηση. Σε αυτό το μοντέλο κινητικότητας, ένας κόμβος μετακινείται από την τρέχουσα τοποθεσία του σε μια νέα τοποθεσία, επιλέγοντας τυχαία μια κατεύθυνση και μία ταχύτητα με την οποία θα ταξιδέψει. Εάν ένας κόμβος που κινείται σύμφωνα με αυτό το μοντέλο φτάσει σε ένα όριο της επιφάνειας της προσομοίωσης, "αναπηδά" από το όριο με την γωνία του να καθορίζεται από την εισερχόμενη κατεύθυνση και συνεχίζει σε αυτή τη νέα πορεία.

Η ταχύτητα και η κατεύθυνση επιλέγονται και τα δύο από προκαθορισμένα εύρη, [speedmin, speedmax] και  $[0, 2\pi]$ , αντίστοιχα. Κάθε κίνηση στο μοντέλο αυτό συμβαίνει είτε σε ένα σταθερό χρονικό διάστημα το οποίο καθορίζεται από το όρισμα -t, είτε μετά από μια σταθερή απόσταση που διανύθηκε και καθορίζεται από το όρισμα -d. Μόλις παρέλθει αυτός ο ορισμένος χρόνος ή απόσταση, υπολογίζεται μια νέα κατεύθυνση και ταχύτητα.

## Random Direction

Το Random Direction Mobility Model (Μοντέλο Κινητικότητας Τυχαίας Κατεύθυνσης)[49] δημιουργήθηκε για να καταπολεμήσει την ομαδοποίηση των κόμβων σε ένα μέρος της περιοχής προσομοίωσης που προέκυπτε από άλλα μοντέλα. Σε αυτό το μοντέλο, οι κόμβοι επιλέγουν μια τυχαία κατεύθυνση προς την οποία ταξιδεύουν με τρόπο παρόμοιο με το Random Walk Mobility Model. Στη συνέχεια, ένας κόμβος ταξιδεύει μέχρι τα όρια της περιοχής προσομοίωσης προς αυτήν την κατεύθυνση. Μόλις φτάσει στο όριο της περιοχής προσομοίωσης, ο κόμβος κάνει παύση για συγκεκριμένο χρόνο και επιλέγει άλλη κατεύθυνση με τυχαία γωνία μεταξύ 0 και 180 μοιρών και συνεχίζει να κινούμενος προς τα εκεί, μέχρι να φτάσει πάλι στα όρια της περιοχής και να επαναλάβει τη διαδικασία.

Αφού οι κόμβοι αυτοί ταξιδεύουν και συνήθως κάνουν παύση στα όρια της περιοχής προσομοίωσης, ο μέσος αριθμός αναπηδήσεων(hop) για Πακέτα Δεδομένων που χρησιμοποιούν το Random Direction Mobility Model θα είναι πολύ υψηλότερος από το μέσο αριθμό αναπηδήσεων των περισσότερων άλλων μοντέλων κινητικότητας. Επιπλέον, η τμηματοποίηση του δικτύου σε μικρότερα υποδίκτυα(network partition) είναι πιο πιθανή με αυτό το μοντέλο σε σύγκριση με άλλα μοντέλα κινητικότητας.

## Manhattan

Το μοντέλο του Manhattan Grid(πλέγμα του Μανχάταν)[47] ανήκει στην κατηγορία μοντέλων κινητικότητας με γεωγραφικούς περιορισμούς. Στο μοντέλο αυτό η περιοχή της προσομοίωσης χωρίζεται σε ίσα τετράγωνα μέρη, τα οποία θα θεωρήσουμε ως πλατείες. Οι γραμμές που οριοθετούν τις πλατείες αυτές, θεωρούνται δρόμοι. Οι κόμβοι του δικτύου λειτουργούν ως πεζοί που κινούνται στις ακμές των τετραγώνων-πλατειών, δηλαδή στους

δρόμους. Αρχικά οι κόμβοι κατανέμονται τυχαία στους δρόμους και κάθε κόμβος επιλέγει μια κατεύθυνση και μια ταχύτητα. Όταν ένας κόμβος φτάνει σε μια γωνία, αλλάζει κατεύθυνση με μια ορισμένη πιθανότητα. Η ταχύτητα του αλλάζει με την πάροδο του χρόνου. Σε αυτό το μοντέλο, οι κόμβοι κινούνται μόνο οριζόντια και κάθετα, πάνω στις προκαθορισμένες διαδρομές που ονομάσαμε δρόμους.

Στο BonnMotion τα ορίσματα  $-u$  και  $-v$  καθορίζουν τον αριθμό των τετραγώνων που θα υπάρχουν μεταξύ των δρόμων. Το  $-u$  καθορίζει τον αριθμό για το άξονα  $x$  και το  $-v$  για τον άξονα  $y$ . Η υλοποίηση αυτή περιέχει ορισμένες τροποποιήσεις του μοντέλου πλέγματος Manhattan. Μια πρόσθετη παράμετρος που εισάγεται είναι η ελάχιστη ταχύτητα ενός κινητού κόμβου η οποία καθορίζεται με το όρισμα  $-m$ . Αυτό είναι χρήσιμο επειδή η ταχύτητα ενός κινητού κόμβου μπορεί να είναι αυθαίρετα κοντά στο 0 και αφού το συγκεκριμένο μοντέλο ορίζει ότι η ταχύτητα πρέπει να ενημερώνεται ανά κάποια διαστήματα απόστασης, χωρίς αυτήν την παράμετρο, μπορεί να υπάρχουν πολύ μεγάλες περιόδους πολύ αργής κίνησης των κόμβων. Επίσης υπάρχει η δυνατότητα παύσης των κόμβων σε περίπτωση που ο κόμβος δεν αλλάζει την ταχύτητα του, αλλά η δυνατότητα αυτή δε χρησιμοποιήθηκε στη δική μας υλοποίηση.

## 4.4.2. Πειράματα

### Πείραμα 1

Στο πείραμα 1 θα εξετάσουμε την επίδραση της μεταβολής του αριθμού των ενδιάμεσων κινητών κόμβων στο δίκτυο, για τα πρωτόκολλα δρομολόγησης Flooding, Epidemic και PРоPHET. Πιο συγκεκριμένα θα παρατηρήσουμε την απόδοση του συστήματος έχοντας ως μέτρο τη μέση τιμή του χρόνου καθυστέρησης λήψης των Πακέτων Δεδομένων (Average Delay) σε συνδυασμό με το ρυθμό λήψης δεδομένων από την κρυφή μνήμη του consumer (Cache Hit Ratio) αλλά και τις αναλυτικές μετρήσεις του χρόνου καθυστέρησης λήψης των Πακέτων Δεδομένων (Delay). Θ' αυξάνουμε κάθε φορά τον αριθμό των κόμβων κατά δύο, ξεκινώντας από 2 και φτάνοντας έως τους 10. Στην προκειμένη περίπτωση, έχουμε καταχωρήσει 180 διαφορετικές τιμές μετρήσεων στο σύστημα και ο consumer θα κάνει 20 επαναλήψεις κατά τις οποίες θα στέλνει ένα τυχαίο Πακέτο Ενδιαφέροντος. Η διάρκεια ζωής (Lifetime) του Πακέτου Ενδιαφέροντος έχει τεθεί στη 1 ώρα και ο χρόνος για τον οποίον ένα Πακέτο Δεδομένων θεωρείται "πρόσφατο" (Data Freshness Period) στις 2 ώρες. Οι επιλογές αυτές γίνανε ώστε να υπερκαλύπτουν ακόμη κι έναν πολύ μεγάλο χρόνο ταξιδιού ενός Πακέτου Ενδιαφέροντος και του αντίστοιχου Πακέτου Δεδομένων, ώστε να μας δοθεί η δυνατότητα να παρατηρήσουμε τη συμπεριφορά του συστήματος, όσο πιο απλοποιημένο γίνεται, χωρίς επιπλέον απώλεια πακέτων. Η επίδραση των παραμέτρων αυτών, θα εξεταστεί σε επόμενα πειράματα. Στο πείραμα αυτό χρησιμοποιούμε κάθε φορά το ίδιο σενάριο κινητικότητας Random Walk, το οποίο δημιουργήσαμε για 10 ενδιάμεσους κόμβους αλλά κάθε φορά χρησιμοποιούσαμε μόνο όσους χρειαζόμαστε. Οπότε ουσιαστικά έχουμε στα άκρα του δικτύου σταθερούς τους consumer-καταναλωτή και producer-παραγωγό και ενδιάμεσα έχουμε αρχικά 2 κόμβους που κινούνται και προσθέτουμε άλλους 2 σε κάθε επόμενη δοκιμή, χωρίς να μεταβάλλεται ο τρόπος που κινούνται οι ήδη υπάρχοντες κόμβοι. Σ' αυτό το σενάριο κινητικότητας έχουμε θέσει ως μέγιστη ταχύτητα κίνησης των κόμβων τα 40 m/s.

### Πείραμα 2

Στο πείραμα 2 θα μελετήσουμε πως ανταποκρίνονται τα πρωτόκολλα δρομολόγησης Flooding, Epidemic και PРоPHET, σε διαφορετικά μοντέλα κινητικότητας. Θα παρατηρήσουμε

τις μεταβολές του χρόνου καθυστέρησης λήψης των Πακέτων Δεδομένων (Delay) χρησιμοποιώντας και τη μέση τιμή του χρόνου καθυστέρησης λήψης των Πακέτων Δεδομένων (Average Delay) αλλά και το ρυθμό λήψης δεδομένων από την κρυφή μνήμη του consumer (Cache Hit Ratio) για να εξαγάγουμε συμπεράσματα. Το πείραμα θα επαναληφθεί για τα μοντέλα Random Walk, Random Direction και Manhattan όπου στο τελευταίο, χωρίσαμε την επιφάνεια προσομοίωσης σε 36 ίσα τετράγωνα μέρη, με πλευρά 100m το καθένα. Σε αυτό το πείραμα, έχουμε και πάλι 180 διαφορετικές καταχωρήσεις τιμών στο σύστημα και θα γίνουν 20 επαναλήψεις αποστολής ενός τυχαίου Πακέτου Ενδιαφέροντος από τον consumer. Η διάρκεια ζωής (Lifetime) του Πακέτου Ενδιαφέροντος έχει ρυθμιστεί στη 1 ώρα και ο χρόνος για τον οποίον ένα Πακέτο Δεδομένων θεωρείται "πρόσφατο" (Data Freshness Period) στις 2 ώρες. Οι ενδιαμέσοι κόμβοι αριθμούνται στους 10 και στα τρία αυτά σενάρια έχουμε επιλέξει η μέγιστη ταχύτητα των κόμβων να είναι και πάλι 40 m/s, όμως έχουμε θέσει και ελάχιστη ταχύτητα κίνησης τα 15 m/s.

### Πείραμα 3

Στο πείραμα 3 θα δούμε τι αντίκτυπο έχει η μεταβολή του χρόνου για τον οποίον ένα Πακέτο Δεδομένων θεωρείται "πρόσφατο" (Data Freshness Period) στο δίκτυο για τα πρωτόκολλα δρομολόγησης Flooding, Epidemic και PRoPHET. Στην περίπτωση αυτή θα έχουμε ως βασικό άξονα παρατήρησης το ρυθμό λήψης δεδομένων από την κρυφή μνήμη του consumer (Cache Hit Ratio) διότι αυτός περιμένουμε να είναι άρρηκτα συνδεδεμένος με το Freshness, καθώς το μεγαλύτερο Freshness, δίνει και περισσότερο χρόνο για να γίνει cache hit για ένα πακέτο που θα ξαναζητηθεί. Θα μελετήσουμε επίσης και την επίδραση του στη μέση τιμή του χρόνου καθυστέρησης λήψης των Πακέτων Δεδομένων (Average Delay) που πηγάζει από τις αναλυτικές μετρήσεις του χρόνου καθυστέρησης λήψης των Πακέτων Δεδομένων (Delay). Το Freshness θ' αυξάνεται κάθε φορά κατά 15 λεπτά, ξεκινώντας από τα 15 λεπτά και φτάνοντας έως τα 60 λεπτά. Στο πείραμα αυτό, επιλέξαμε να μειώσουμε τον αριθμό των καταχωρήσεων διαφορετικών τιμών μετρήσεων από 180 στις 60. Η επιλογή αυτή έγινε έτσι ώστε να υπάρχει μία πιο ισορροπημένη αναλογία ανάμεσα στον αριθμό των καταχωρήσεων και στον αριθμό των επαναλήψεων, ο οποίος ανέρχεται πάλι στις 20. Έτσι δίνεται η δυνατότητα να εξετάσουμε καλύτερα την επίδραση του Freshness, διότι ο μεγάλος αριθμός καταχωρήσεων αυξάνει κατά πολύ την τυχαιότητα. Η διάρκεια ζωής (Lifetime) του Πακέτου Ενδιαφέροντος έχει τεθεί και πάλι στη 1 ώρα. Στο πείραμα αυτό θα έχουμε 10 κινούμενους ενδιάμεσους κόμβους που χρησιμοποιούμε κάθε φορά το ίδιο σενάριο κινητικότητας Random Walk, με μέγιστη ταχύτητα κίνησης των κόμβων τα 40 m/s .

### Πείραμα 4

Στο πείραμα 4, θα μελετήσουμε την επίδραση της αλλαγής της διάρκειας ζωής (Lifetime) του Πακέτου Ενδιαφέροντος στο δίκτυο για τα πρωτόκολλα δρομολόγησης Flooding, Epidemic και PRoPHET για τρία διαφορετικά μοντέλα κινητικότητας. Θα εισάγουμε και μία ακόμη μετρική αξιολόγησης, το ρυθμό με τον οποίο ικανοποιούνται τα πακέτα Ενδιαφέροντος (Interest Satisfaction Ratio-ISR) και θα τη χρησιμοποιήσουμε μαζί με τη μέση τιμή του χρόνου καθυστέρησης λήψης των Πακέτων Δεδομένων (Average Delay), το χρόνο καθυστέρησης λήψης των Πακέτων Δεδομένων (Delay) και το ρυθμό λήψης δεδομένων από την κρυφή μνήμη του consumer (Cache Hit Ratio). Η διάρκεια ζωής (Lifetime) του Πακέτου Ενδιαφέροντος θα ξεκινάει από το 1 λεπτό και ανάλογα με τις ανάγκες του εκάστοτε σεναρίου κινητικότητας, θα φτάνει έως και τα 7 λεπτά, αυξάνοντας κατά 1 λεπτό κάθε φορά. Θα έχουμε 180 διαφορετικές καταχωρήσεις τιμών στο σύστημα και θα γίνουν 20 επαναλήψεις αποστολής ενός τυχαίου Πακέτου Ενδιαφέροντος από τον consumer. Ο χρόνος για τον οποίον ένα

Πακέτο Δεδομένων θεωρείται "πρόσφατο" (Data Freshness Period) ανέρχεται στις 2 ώρες ώστε να υπερκαλύπτει το πείραμα. Θα υπάρχουν 10 κινητοί κόμβοι και θα χρησιμοποιούμε σενάρια κινητικότητας Random Walk, Random Direction και Manhattan όπου στο τελευταίο, χωρίσαμε την επιφάνεια προσομοίωσης σε 36 ίσα τετράγωνα μέρη, με πλευρά 100m. Επιλέξαμε ως μέγιστη ταχύτητα κίνησης των κόμβων τα 40 m/s και ελάχιστη ταχύτητα κίνησης τα 15 m/s.

## Πείραμα 5

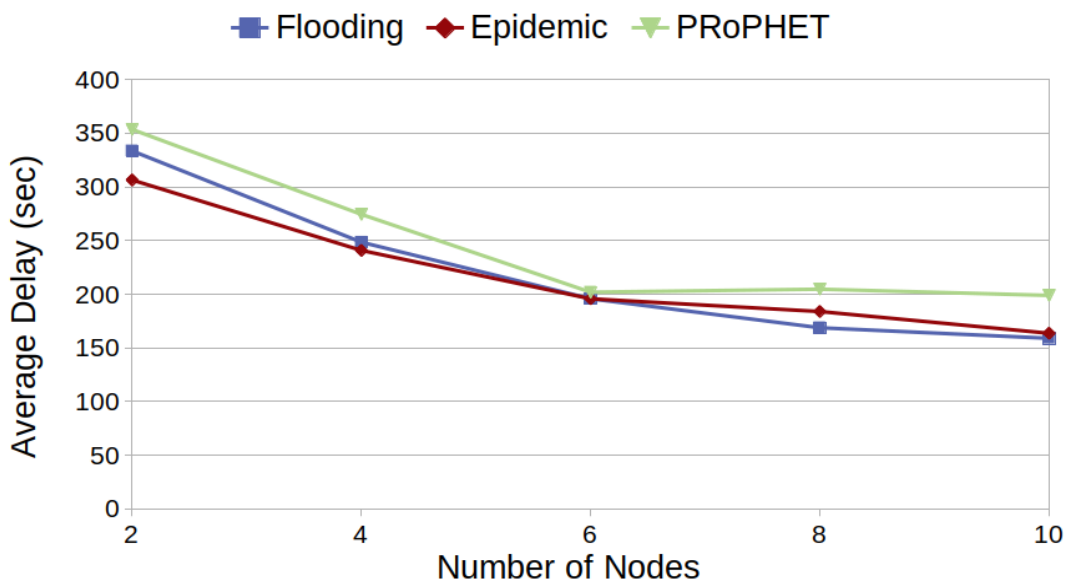
Στο πέμπτο και τελευταίο πείραμα, επιλέξαμε να χρησιμοποιήσουμε το πρωτόκολλο δρομολόγησης Flooding, για να μελετήσουμε την επίδραση της απώλειας ενός ποσοστού των συνολικών πακέτων (% of Loss) στην απόδοση του δικτύου. Πρέπει να τονιστεί σε αυτό το σημείο, ότι το ποσοστό αυτό της απώλειας αφορά τόσο τα Πακέτα Ενδιαφέροντος, όσο και τα Πακέτα Δεδομένων αλλά και όλους τους κόμβους (producer, consumer και ενδιάμεσους). Δηλαδή σε κάθε επαφή που γίνεται μεταξύ δύο κόμβων, υπάρχει ένα ποσοστό της τάξεως του 0-50%, το πακέτο να χαθεί, ακόμη κι αν αυτό σημαίνει την απώλεια ενός Πακέτου Δεδομένων, ακριβώς πριν παραδοθεί πίσω στον consumer.

Σε αυτή την περίπτωση θα χρησιμοποιήσουμε και πάλι για να εξάγουμε τα συμπεράσματα μας, τη μέση τιμή του χρόνου καθυστέρησης λήψης των Πακέτων Δεδομένων (Average Delay), το ρυθμό λήψης δεδομένων από την κρυφή μνήμη του consumer (Cache Hit Ratio), τις αναλυτικές μετρήσεις του χρόνου καθυστέρησης λήψης των Πακέτων Δεδομένων (Delay) και το ρυθμό με τον οποίο ικανοποιούνται τα πακέτα Ενδιαφέροντος (Interest Satisfaction Ratio-ISR). Το ποσοστό της απώλειας θα ξεκινάει από 0% και θα αυξάνει κατά 10% έως ότου φτάσει στο 50%. Στο πείραμα αυτό, θα έχουμε και πάλι 180 διαφορετικές καταχωρήσεις τιμών στο σύστημα και θα γίνουν 20 επαναλήψεις αποστολής ενός τυχαίου Πακέτου Ενδιαφέροντος από τον consumer. Η διάρκεια ζωής (Lifetime) του Πακέτου Ενδιαφέροντος θα είναι 1 ώρα και ο χρόνος για τον οποίον ένα Πακέτο Δεδομένων θεωρείται "πρόσφατο" (Data Freshness Period) θα είναι 2 ώρες. Θα έχουμε 10 κινητούς κόμβους και θα χρησιμοποιούμε κάθε φορά το ίδιο σενάριο κινητικότητας Random Walk, με μέγιστη ταχύτητα κίνησης των κόμβων τα 40 m/s .

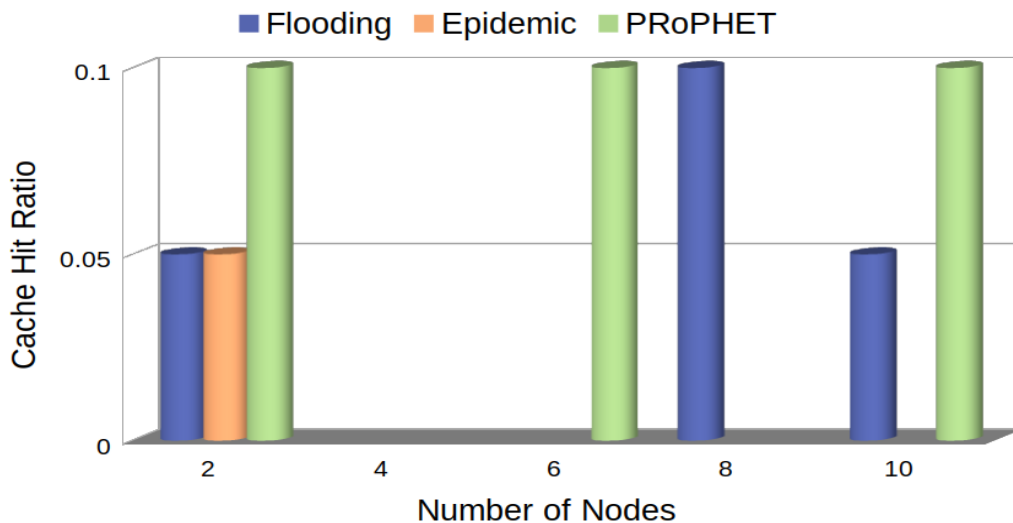
## 5. Αποτελέσματα πειραμάτων

### 5.1. Πείραμα 1

Στο πρώτο πείραμα, εξετάζουμε την επίδραση της αλλαγής του αριθμού των ενδιαμέσων κινητών κόμβων στο δίκτυο για τα πρωτόκολλα δρομολόγησης Flooding, Epidemic και PRoPHET. Ως βασική μετρική αξιολόγησης του πειράματος αυτού, χρησιμοποιήσαμε τις αναλυτικές μετρήσεις του χρόνου καθυστέρησης λήψης των Πακέτων Δεδομένων (Delay) και για κάποια επιπλέον συμπεράσματα τη μέση τιμή του χρόνου καθυστέρησης λήψης των Πακέτων Δεδομένων (Average Delay) σε συνδυασμό με το ρυθμό λήψης δεδομένων από την κρυφή μνήμη του consumer (Cache Hit Ratio). Οι ενδιαμέσοι κόμβοι θα κυμαίνονται από 2 έως 10, με βήμα 2.



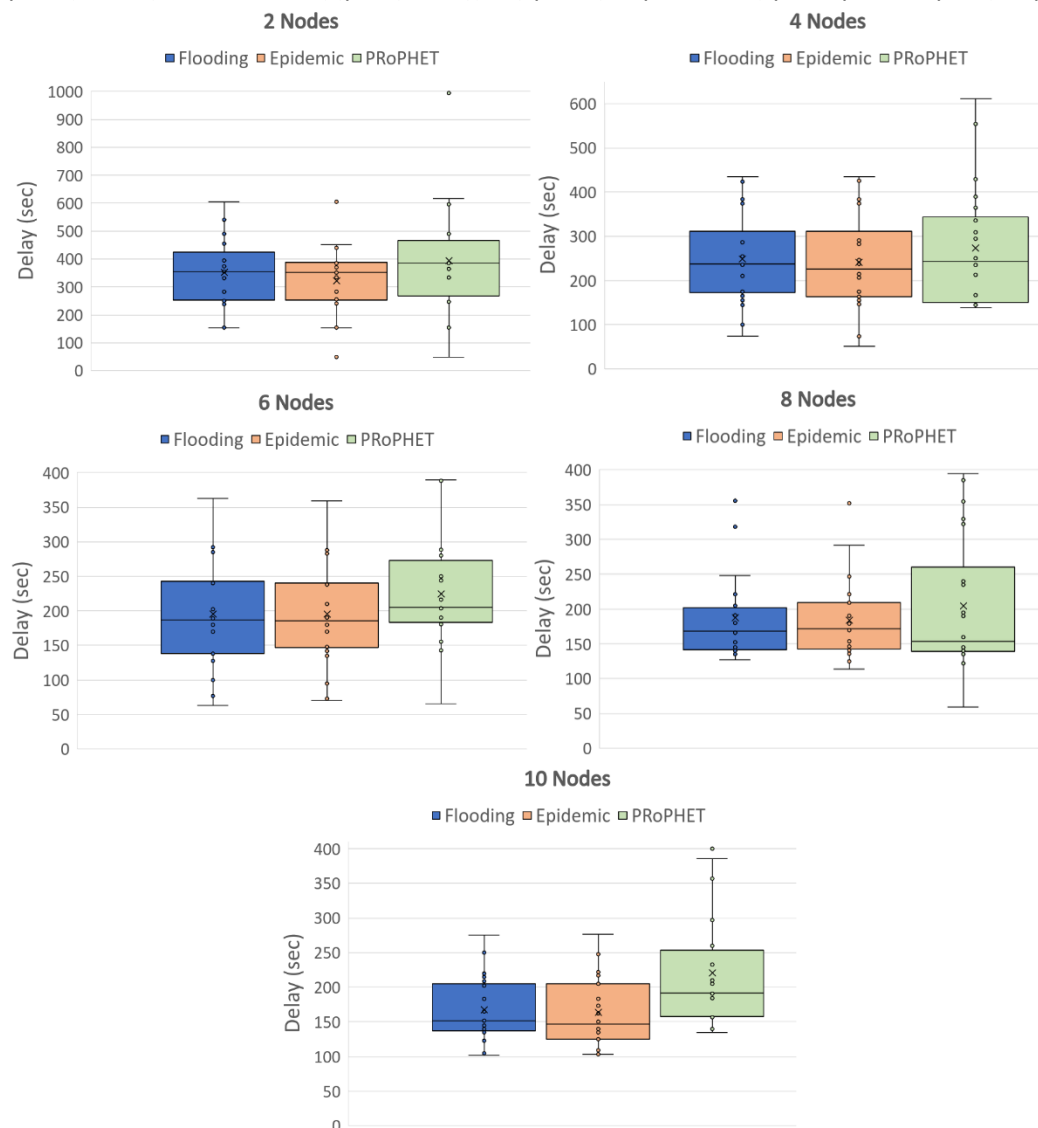
Εικόνα 18: Μέσος χρόνος καθυστέρησης λήψης δεδομένων (Average Delay) για τα πρωτόκολλα δρομολόγησης Flooding, Epidemic και PRoPHET, για τις διάφορες τιμές ενδιαμέσων κόμβων (Number of nodes).



Εικόνα 19: Ρυθμός λήψης δεδομένων από την κρυφή μνήμη του consumer (Cache Hit Ratio) για τα πρωτόκολλα δρομολόγησης Flooding, Epidemic και PRoPHET, για τις διάφορες τιμές ενδιαμέσων κόμβων (Number of nodes).

Βλέπουμε λοιπόν από τ' αποτελέσματα ότι συνολικά όσο αυξάνεται ο αριθμός των ενδιάμεσων κόμβων, τόσο μειώνεται η μέση τιμή του χρόνου καθυστέρησης λήψης των πακέτων Δεδομένων (Average Delay). Έχοντας περισσότερους κινούμενους κόμβους ανάμεσα στον producer και τον consumer, δημιουργούνται περισσότερες πιθανές διαδρομές για να φτάσουν τα Πακέτα Ενδιαφέροντος και Δεδομένων από το ένα άκρο του δικτύου στο άλλο. Αυτό συμβαίνει λόγω των περισσότερων επαφών που δημιουργούνται όταν έχουμε ένα μεγαλύτερο πλήθος κόμβων σε ίδιου μεγέθους επιφάνεια.

Πιο συγκεκριμένα, παρατηρούμε ότι τα πρωτόκολλα Flooding και Epidemic παρουσιάζουν παρόμοια συμπεριφορά η οποία είναι καλύτερη του πρωτοκόλλου PRoPHET. Βέβαια, παρατηρούμε ότι για 2 κόμβους είχαμε κοντινά αποτελέσματα μεταξύ των τριών πρωτοκόλλων αλλά καθώς αυξάνεται ο αριθμός των κόμβων, παρατηρούμε ότι η απόδοση του PRoPHET χειροτερεύει συγκριτικά με των άλλων. Αυτό θα μπορούσε να εξηγηθεί απο το γεγονός ότι στο πρωτόκολλο PRoPHET έχει σημασία το πόσες φορές θα έρθει σε επαφή ένας κόμβος με τους υπόλοιπους, για την αύξηση της πιθανότητας να μεταδοθεί ένα πακέτο σ' έναν άλλον κόμβο. Συνεπώς με την ύπαρξη λιγότερων κόμβων, η πιθανότητα μετάδοσης μεταξύ τους, θα είναι αυξημένη σε σχέση με τις περιπτώσεις με περισσότερους κόμβους.

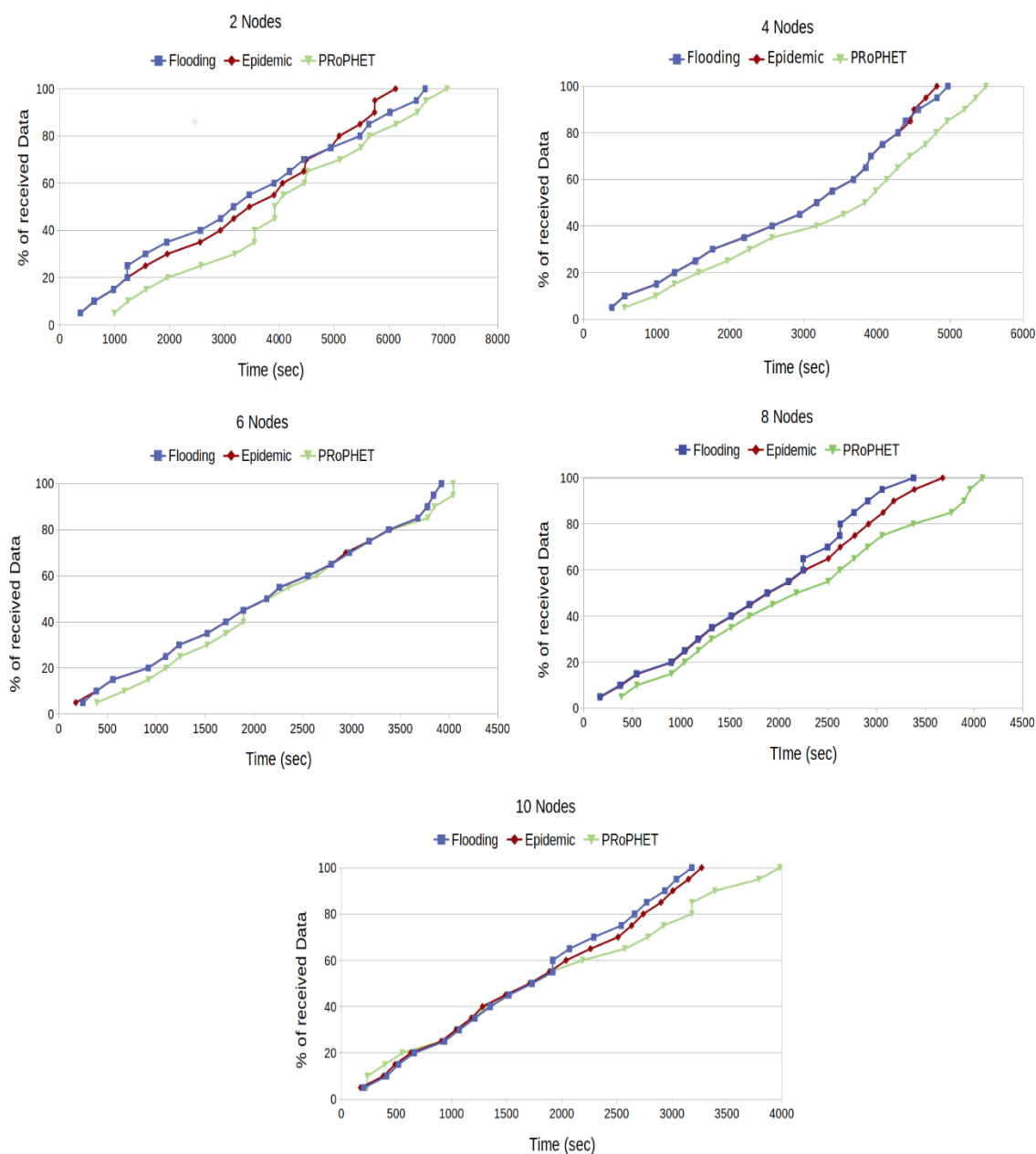


Εικόνα 20: Αναλυτικές τιμές του Delay και της διασποράς των τιμών για τα πρωτόκολλα δρομολόγησης Flooding, Epidemic και PRoPHET, για διάφορες τιμές ενδιάμεσων κόμβων (Number of nodes).



Αξίζει να σημειωθεί ότι η περίπτωση των 6 κόμβων όπου στην Εικόνα 1 φαίνεται σχεδόν να συμπίπτουν και τα τρία πρωτόκολλα, αν μελετηθεί συνδυαστικά με την Εικόνα 2, αντιλαμβανόμαστε ότι αυτό συμβαίνει διότι υπάρχει υψηλός ρυθμός λήψης δεδομένων από την κρυφή μνήμη του consumer (Cache Hit Ratio) στο πρωτόκολλο PRoPHET, ο οποίος αποτελεί το 10% των συνολικών πακέτων που μεταφέρονται. Ο ρυθμός αυτός είναι μηδενικός στα άλλα δύο πρωτόκολλα και γι' αυτό η τιμή του PRoPHET προσεγγίζει των υπολοίπων.

Η Εικόνα 3, βοηθάει στην αναλυτικότερη απεικόνιση των δεδομένων μας. Ουσιαστικά παρουσιάζει το μέγεθος της διασποράς των τιμών που είχε σε κάθε πείραμα ο χρόνος καθυστέρησης λήψης των δεδομένων (Delay), μη συμπεριλαμβανομένων των περιπτώσεων που είχαμε cache hit και παρουσιάζεται και η ασυμμετρία των δεδομένων.



Εικόνα 21: Ποσοστό των παραδοθέντων Πακέτων Δεδομένων ως προς το χρόνο εκτέλεσης του πειράματος για τα πρωτόκολλα δρομολόγησης Flooding, Epidemic και PRoPHET, για τις διάφορες τιμές ενδιαμέσων κόμβων (Number of nodes).

Μία παρατήρηση που αξίζει να γίνει, είναι το γεγονός ότι σε γενικές γραμμές, το πρωτόκολλο PRoPHET, παρουσιάζει μεγαλύτερο εύρος τιμών από τα άλλα δύο. Αυτό είναι συμβαίνει διότι, λόγω της πιθανότητας που συμπεριλαμβάνεται στη δρομολόγηση του PRoPHET, ένα πακέτο μπορεί να μη μεταφέρεται παρά τις πολλές επαφές των κόμβων. Αυτό έχει ως αποτέλεσμα να συναντάμε πολύ μεγαλύτερες τιμές στην περίπτωση του PRoPHET.

Στην Εικόνα 4, μπορούμε να δούμε ότι στην περίπτωση των 10 κόμβων, το PRoPHET παρουσιάζει σημαντικά μεγαλύτερη καθυστέρηση ως προς το χρόνο ολοκλήρωσης του πειράματος.

Η παρόμοια συμπεριφορά των πρωτοκόλλων flooding και epidemic επιβεβαιώνεται και από η Εικόνα 4 στο οποίο φαίνεται ότι υπάρχει ταύτιση των τιμών τους σε πολύ μεγάλο μέρος των πειραμάτων.

Αν τώρα θέλουμε να παρατηρήσουμε λεπτομερώς τι συμβαίνει για 4-10 ενδιάμεσους κόμβους, έχοντας ως σημείο αναφοράς τους 2 κόμβους, αρκεί να δούμε τα παρακάτω στατιστικά τα οποία προκύπτουν από την ανάλυση των επιμέρους μετρήσεων του χρόνου καθυστέρησης λήψης των Πακέτων Δεδομένων (Delay) για κάθε πρωτόκολλο δρομολόγησης, αφαιρώντας τις περιπτώσεις που έχουμε cache hit.

Για το πρωτόκολλο Flooding

- Για 4 ενδιάμεσους κόμβους, το Average Delay μειώνεται κατά 29.22%
- Για 6 ενδιάμεσους κόμβους, το Average Delay μειώνεται κατά 44.22%
- Για 8 ενδιάμεσους κόμβους, το Average Delay μειώνεται κατά 46.58%
- Για 10 ενδιάμεσους κόμβους, το Average Delay μειώνεται κατά 52.57%

Για το πρωτόκολλο Epidemic

- Για 4 ενδιάμεσους κόμβους, το Average Delay μειώνεται κατά 25.36%
- Για 6 ενδιάμεσους κόμβους, το Average Delay μειώνεται κατά 39.3%
- Για 8 ενδιάμεσους κόμβους, το Average Delay μειώνεται κατά 42.96%
- Για 10 ενδιάμεσους κόμβους, το Average Delay μειώνεται κατά 49.17%

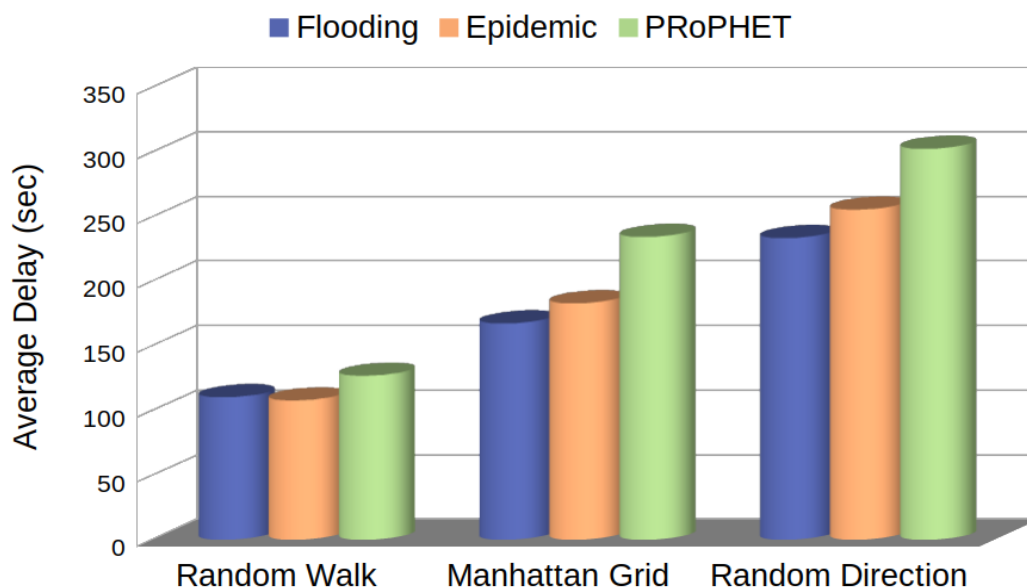
Για το πρωτόκολλο PRoPHET

- Για 4 ενδιάμεσους κόμβους, το Average Delay μειώνεται κατά 30.13%
- Για 6 ενδιάμεσους κόμβους, το Average Delay μειώνεται κατά 42.87%
- Για 8 ενδιάμεσους κόμβους, το Average Delay μειώνεται κατά 47.87%
- Για 10 ενδιάμεσους κόμβους, το Average Delay μειώνεται κατά 42.94%

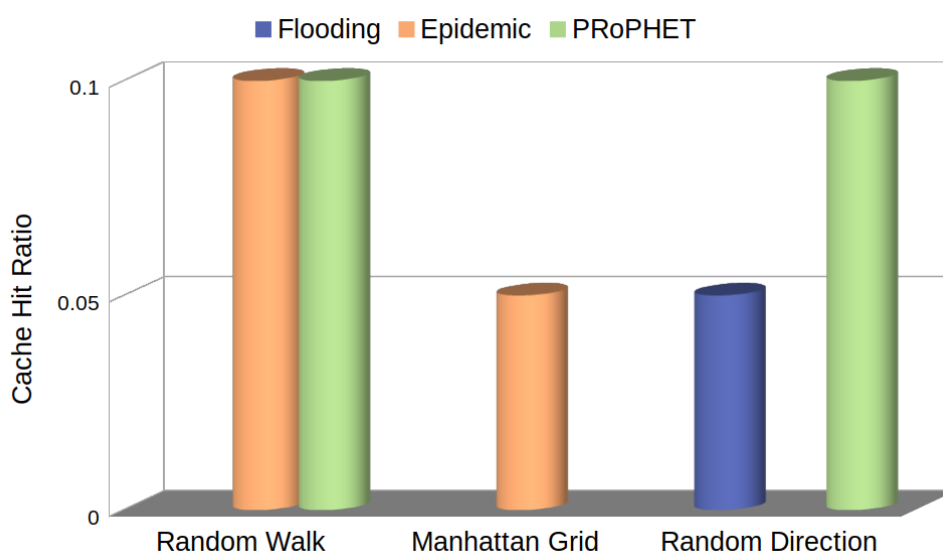
Τέλος, αν παρατηρήσουμε τη μέση τιμή του χρόνου καθυστέρησης λήψης των Πακέτων Δεδομένων (Average Delay), χωρίς να συμπεριλάβουμε τα cache hit, φαίνεται ότι το Epidemic παρουσιάζει ελαφρώς καλύτερη συμπεριφορά από το Flooding αλλά όσο αυξάνονται οι κόμβοι η διαφορά αυτή μειώνεται, πηγαίνοντας από τα 31s στα 2.5s, φτάνοντας να έχουν σχεδόν ίδια απόδοση.

## 5.2. Πείραμα 2

Στο δεύτερο πείραμα, θα μελετήσουμε εάν οι συμπεριφορές των πρωτοκόλλων δρομολόγησης Flooding, Epidemic και PRoPHET, παρουσιάζουν ομοιότητες ή διαφορές για διαφορετικά μοντέλα κινητικότητας. Σε αυτό το πείραμα θ' αξιολογήσουμε τα πρωτόκολλα με βάση τη μέση τιμή του χρόνου καθυστέρησης λήψης των Πακέτων Δεδομένων (Average Delay) και το ρυθμό λήψης δεδομένων από την κρυφή μνήμη του consumer (Cache Hit Ratio) αλλά θα χρησιμοποιήσουμε και τις αναλυτικές μετρήσεις του χρόνου καθυστέρησης λήψης των Πακέτων Δεδομένων (Delay). Τα μοντέλα που θα δοκιμάσουμε θα είναι τα Random Walk, Manhattan και Random Direction.



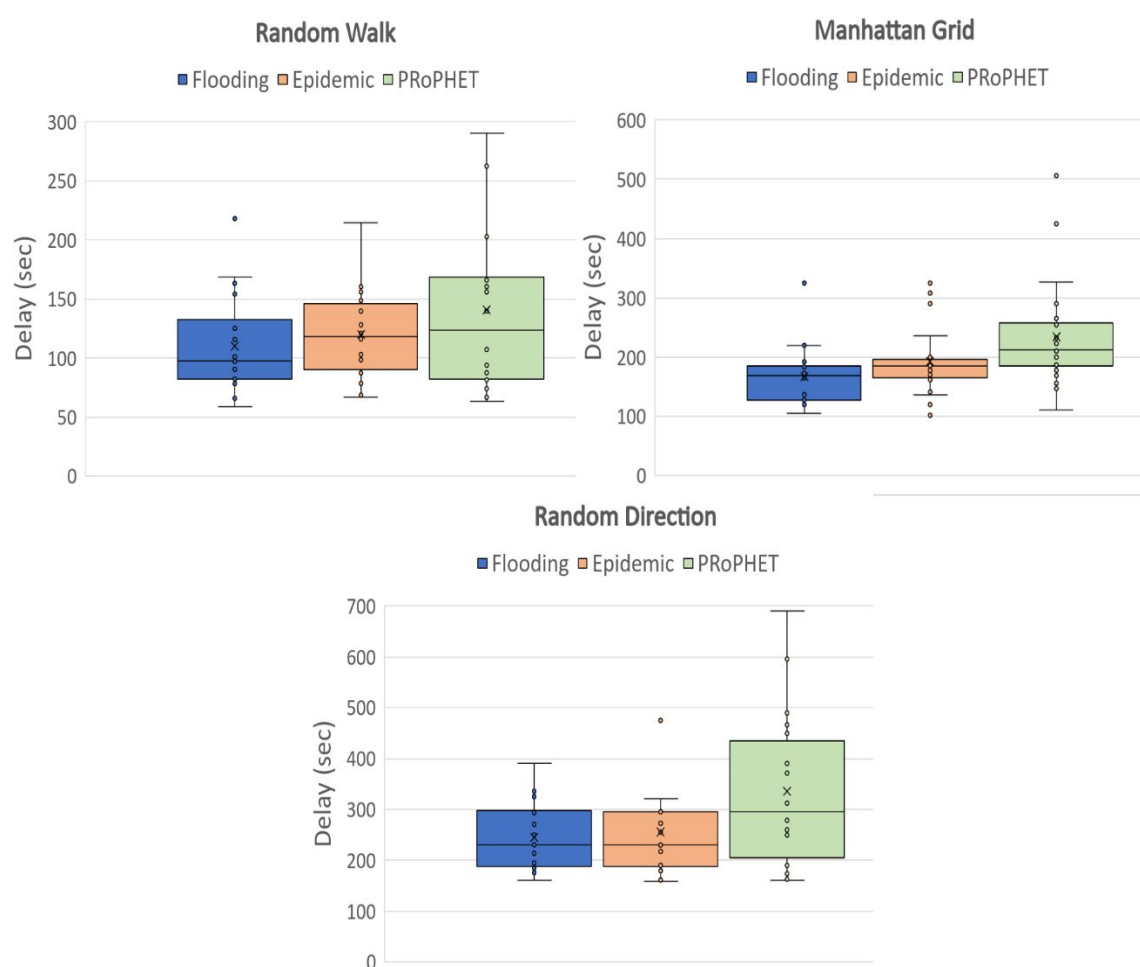
Εικόνα 22: Μέσος χρόνος καθυστέρησης λήψης δεδομένων (Average Delay) για τα πρωτόκολλα δρομολόγησης Flooding, Epidemic και PRoPHET, για τα μοντέλα κινητικότητας Random Walk, Manhattan και Random Direction.



Εικόνα 23: Ρυθμός λήψης δεδομένων από την κρυφή μνήμη του consumer (Cache Hit Ratio) για τα πρωτόκολλα δρομολόγησης Flooding, Epidemic και PRoPHET, για τα μοντέλα κινητικότητας Random Walk, Manhattan και Random Direction.

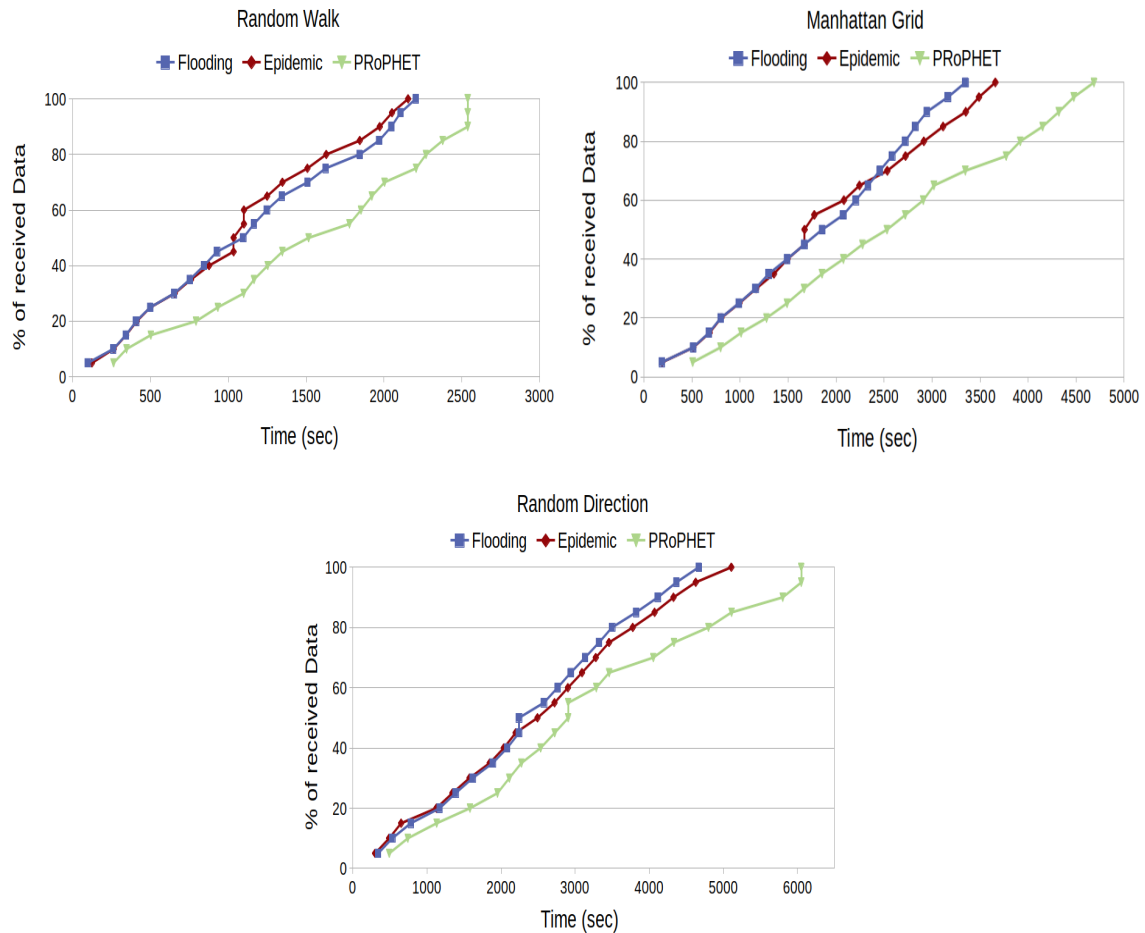
Από το μέσο χρόνο καθυστέρησης λήψης δεδομένων (Average Delay) που παρουσιάζεται στο Εικόνα 22 φαίνεται να επαληθεύεται και πάλι το συμπέρασμα πως το πρωτόκολλο δρομολόγησης PRoPHET, παρουσιάζει ξεκάθαρα τη χειρότερη απόδοση από άποψη χρόνου, ανεξαρτήτως μοντέλου κινητικότητας.

Συνδυάζοντας τα δεδομένα από τις Εικόνες 22 και 23, φαίνεται ότι το Flooding, έχει την καλύτερη απόδοση και για τα τρία πρωτόκολλα με το Epidemic να ακολουθεί με μικρή διαφορά. Το συμπέρασμα αυτό επιβεβαιώνεται και απο τη μελέτη των επιμέρους μετρήσεων του χρόνου καθυστέρησης λήψης των Πακέτων Δεδομένων (Delay) για κάθε πρωτόκολλο δρομολόγησης, αφαιρώντας τις περιπτώσεις που έχουμε cache hit. Και για τα τρία μοντέλα κινητικότητας, το Flooding παρουσιάζει ελαφρώς καλύτερη απόδοση από το Epidemic.



Εικόνα 24: Αναλυτικές τιμές του Delay και της διασποράς των τιμών για τα πρωτόκολλα δρομολόγησης Flooding, Epidemic και PRoPHET, για τα μοντέλα κινητικότητας Random Walk, Manhattan και Random Direction.

Στην Εικόνα 24, στο οποίο παρουσιάζεται η διασπορά των τιμών που είχε σε κάθε πείραμα ο χρόνος καθυστέρησης λήψης των δεδομένων (Delay), χωρίς τις περιπτώσεις που είχαμε cache hit, σε κάποια πειράματα φαίνεται και η ασυμμετρία των δεδομένων, επαληθεύεται και πάλι πως πρωτόκολλο PRoPHET, παρουσιάζει μεγαλύτερο εύρος τιμών από τα Flooding και Epidemic, ανεξάρτητα από το σενάριο κινητικότητας που χρησιμοποιήθηκε για το εκάστοτε πείραμα.

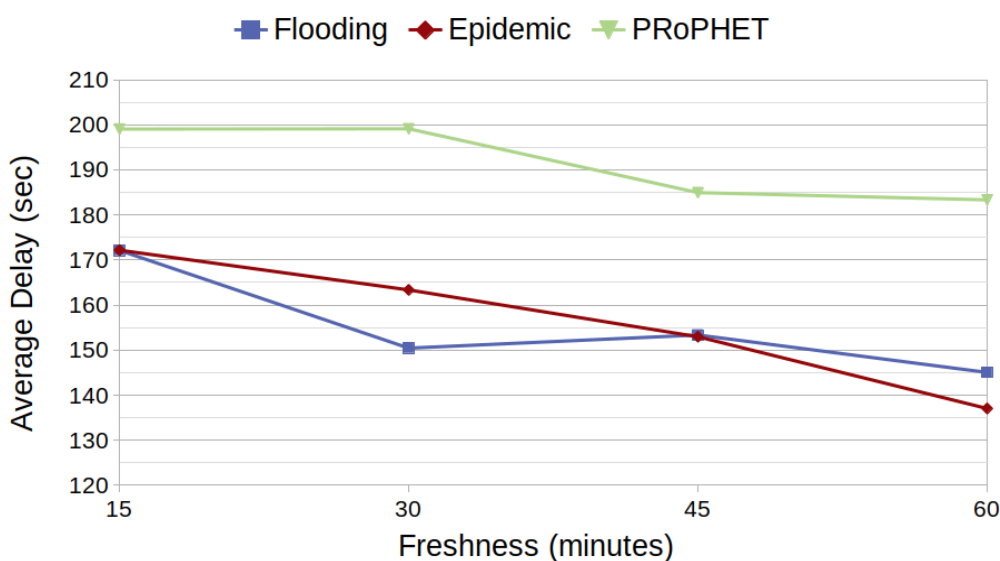


Εικόνα 25: Ποσοστό των παραδοθέντων Πακέτων Δεδομένων ως προς το χρόνο εκτέλεσης του πειράματος για τα πρωτόκολλα δρομολόγησης Flooding, Epidemic και PRoPHET, για τα μοντέλα κινητικότητας Random Walk, Manhattan και Random Direction.

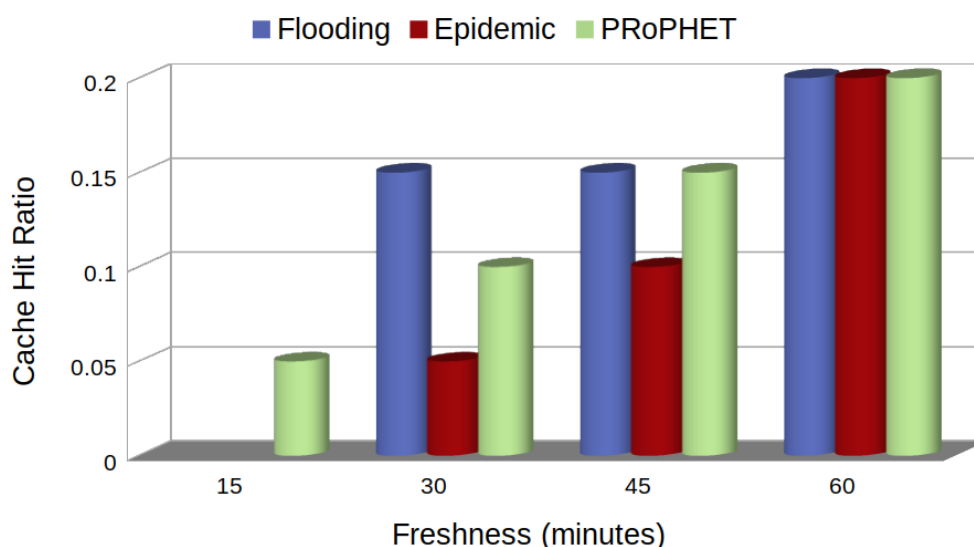
Παρατηρούμε στην Εικόνα 25 ότι και πάλι υπάρχει ταύτιση μέχρι ενός σημείου και πέραν αυτού του σημείου, παρόμοια πορεία των καμπυλών των πρωτόκολλων Flooding και Epidemic. Στις δύο εκ των τριών περιπτώσεων μάλιστα (Manhattan και Random Direction) το σημείο αυτό βρίσκεται εκεί που έχουμε το πρώτο cache hit. Το γεγονός αυτό μαζί με την εικόνα της παράλληλης πορείας τους μετά από αυτό το σημείο, μας επιτρέπει να υποθέσουμε πως είναι πολύ πιθανό να ταυτίζονταν οι δύο καμπύλες για πολύ μεγαλύτερο μέρος του πειράματος. Και στα τρία σενάρια κινητικότητας, ο χρόνος διεκπεραίωσης των πειραμάτων του πρωτοκόλλου PRoPHET, είναι σημαντικά μεγαλύτερος από των άλλων δύο πρωτοκόλλων.

### 5.3. Πείραμα 3

Στο τρίτο πείραμα, θα μελετήσουμε την επίδραση που θα έχει στο δίκτυο η μεταβολή του χρόνου για τον οποίον ένα Πακέτο Δεδομένων θεωρείται "πρόσφατο" (Data Freshness Period) για τα πρωτόκολλα δρομολόγησης Flooding, Epidemic και PRoPHET. Οι παράμετροι αξιολόγησης που θα χρησιμοποιήσουμε σε αυτό το πείραμα θα είναι οι αναλυτικές μετρήσεις του χρόνου καθυστέρησης λήψης των Πακέτων Δεδομένων (Delay) και η μέση τιμή του χρόνου καθυστέρησης λήψης των Πακέτων Δεδομένων (Average Delay) σε συνδυασμό με το ρυθμό λήψης δεδομένων από την κρυφή μνήμη του consumer (Cache Hit Ratio), με ιδιαίτερη έμφαση στο τελευταίο διότι αναμένεται να παρουσιάζει ισχυρή σύνδεση με το Freshness. Οι τιμές για τις οποίες θα εξετάσουμε το Freshness θα είναι τα 15,30,45,και 60 λεπτά.



Εικόνα 26: Μέσος χρόνος καθυστέρησης λήψης δεδομένων (Average Delay) για τα πρωτόκολλα δρομολόγησης Flooding, Epidemic και PRoPHET, για τις διάφορες τιμές του χρόνου για τον οποίον ένα Πακέτο Δεδομένων θεωρείται "πρόσφατο" (Data Freshness Period).

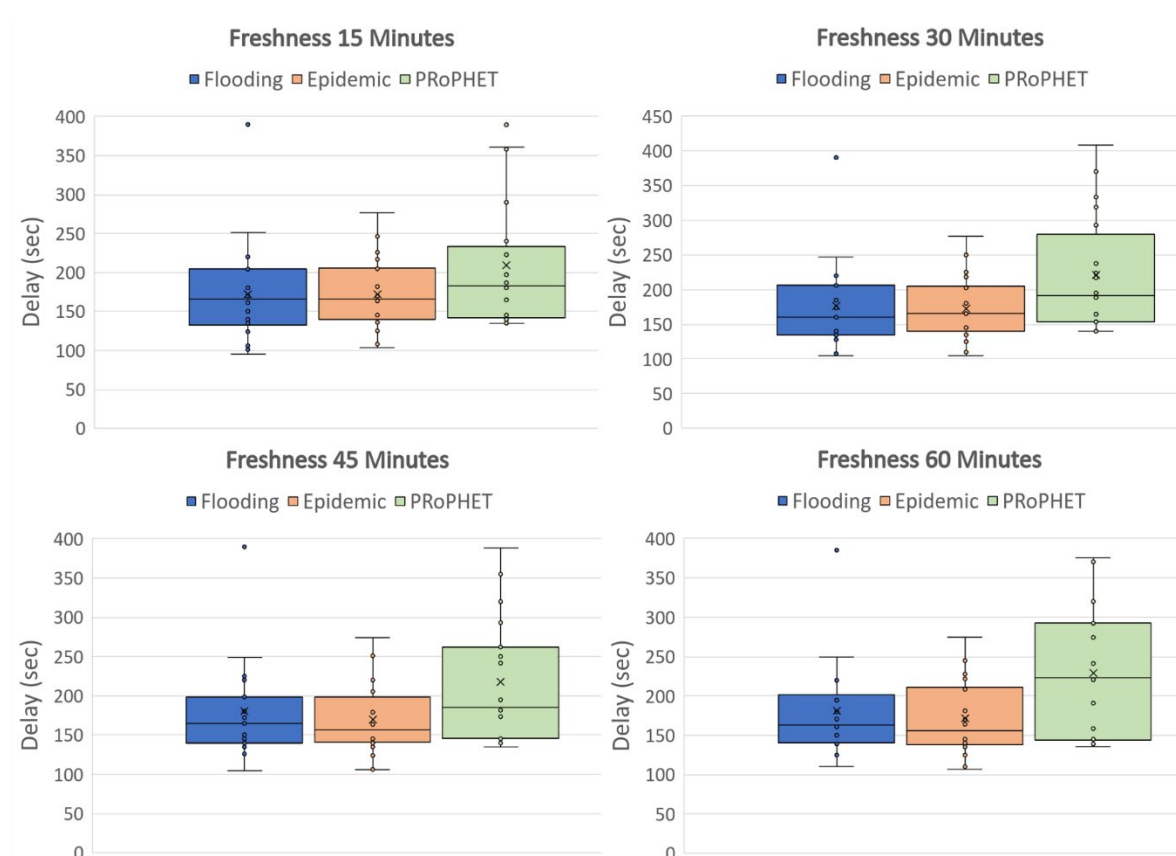


Εικόνα 27: Ρυθμός λήψης δεδομένων από την κρυφή μνήμη του consumer (Cache Hit Ratio) για τα πρωτόκολλα δρομολόγησης Flooding, Epidemic και PRoPHET, για τις διάφορες τιμές του χρόνου για τον οποίον ένα Πακέτο Δεδομένων θεωρείται "πρόσφατο" (Data Freshness Period).



Βλέπουμε λοιπόν στην Εικόνα 27, όπως ήταν αναμενόμενο, ότι όσο αυξάνεται ο χρόνος για τον οποίον ένα Πακέτο Δεδομένων θεωρείται "πρόσφατο" (Data Freshness Period), τόσο αυξάνεται και ο ρυθμός λήψης δεδομένων από την κρυφή μνήμη του consumer (Cache Hit Ratio). Και στις τρεις περιπτώσεις πρωτοκόλλων, όταν αυξάνουμε το Freshness στη 1 ώρα, οι περιπτώσεις στις οποίες έχουμε cache hit, αγγίζουν το ποσοστό του 20% των συνολικών Πακέτων Ενδιαφέροντος.

Στην Εικόνα 26, φαίνεται ότι η μέση τιμή του χρόνου καθυστέρησης λήψης των πακέτων Δεδομένων (Average Delay) παρουσιάζει μείωση, όσο αυξάνεται το Freshness και αυτό συμβαίνει λόγω του αυξημένου Cache Hit Ratio. Για κάθε φορά που δε χρειάστηκε να σταλεί ένα Πακέτο Ενδιαφέροντος στο δίκτυο, γιατί ικανοποιήθηκε από την cache του consumer, ο χρόνος που θα έκανε το πακέτο αυτό να φτάσει στον producer και να γυρίσει το αντίστοιχο Πακέτο Δεδομένων, δεν αθροίζεται στο συνολικό χρόνο του πειράματος, άρα η μείωση του Average Delay είναι απολύτως λογική.

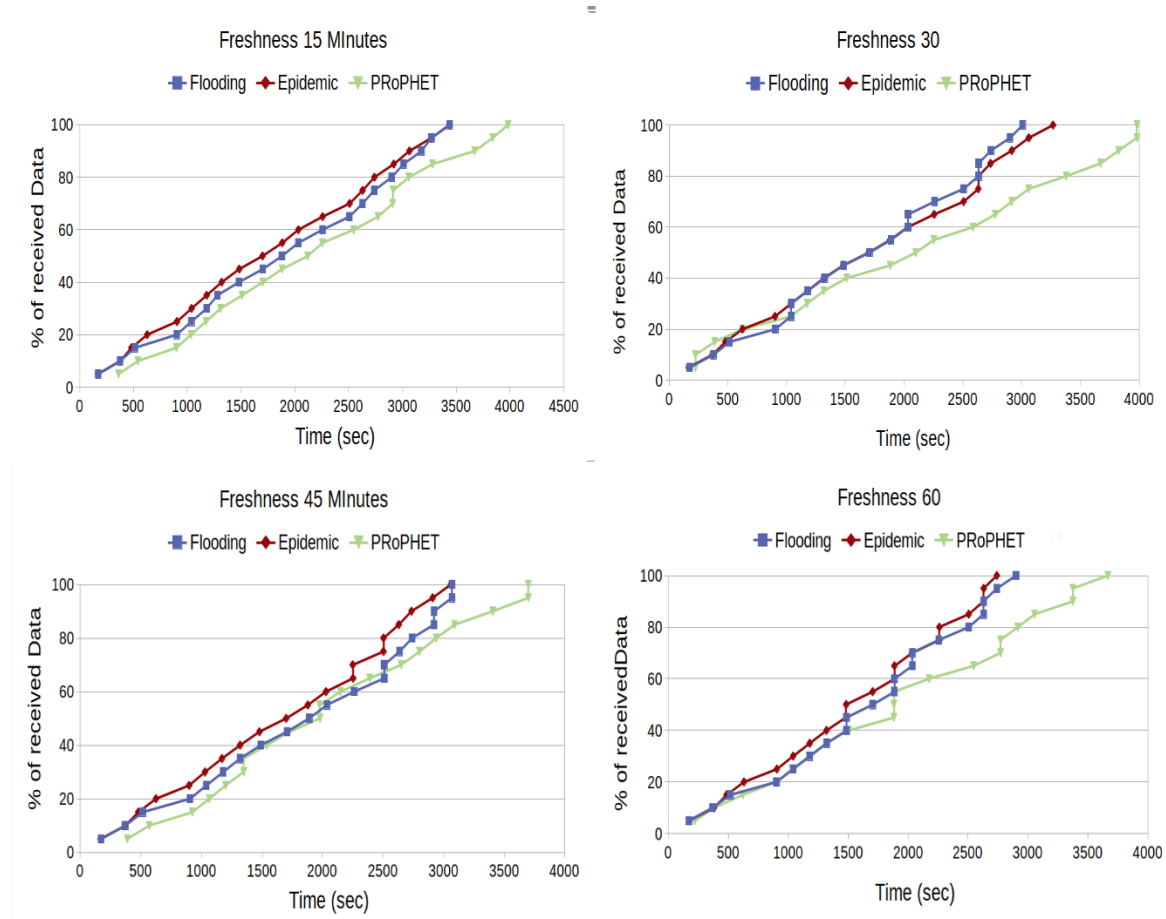


Εικόνα 28: Αναλυτικές τιμές του Delay και της διασποράς των τιμών για τα πρωτόκολλα δρομολόγησης Flooding, Epidemic και PRoPHET, για τις διάφορες τιμές του χρόνου για τον οποίον ένα Πακέτο Δεδομένων θεωρείται "πρόσφατο" (Data Freshness Period).

Στην Εικόνα 28, όπου παρουσιάζεται η διασπορά των τιμών που είχε σε κάθε πείραμα ο χρόνος καθυστέρησης λήψης των δεδομένων (Delay), μη συμπεριλαμβανομένων των περιπτώσεων που είχαμε Cache Hit και η ασυμμετρία των δεδομένων, φαίνεται και πάλι πως το πρωτόκολλο PRoPHET, παρουσιάζει μεγαλύτερο εύρος τιμών από τα άλλα δύο, για τους λόγους που εξηγήσαμε στο Πείραμα 1.

Επίσης, μια συμπεριφορά που παρουσιάζει ενδιαφέρον και αποτυπώνεται στην Εικόνα 28 είναι ότι ενώ στο πρωτόκολλο Flooding υπάρχει μεγάλη συγκέντρωση των τιμών του χρόνου

καθυστερήσης λήψης δεδομένων σε τιμές κάτω των 200 δευτερολέπτων σε όλες τις περιπτώσεις, υπάρχει μια τιμή η οποία σταθερά προσεγγίζει το 400 και ανεβάζει έτσι και το χρόνο εκτέλεσης του πειράματος που φαίνεται στην Εικόνα 29 αλλά και το μέσο χρόνο καθυστέρησης λήψης δεδομένων (Average Delay).



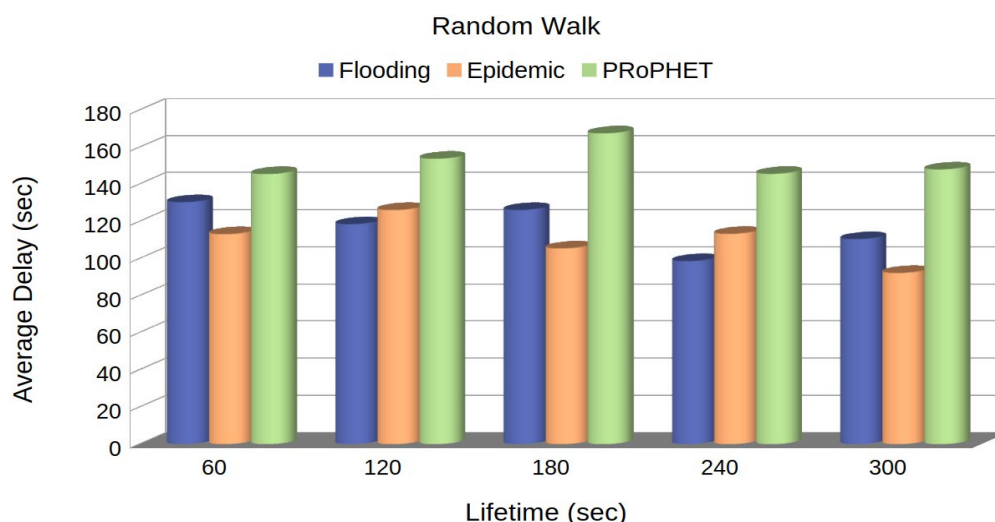
Εικόνα 29: Ποσοστό των παραδοθέντων Πακέτων Δεδομένων ως προς το χρόνο εκτέλεσης του πειράματος για τα πρωτόκολλα δρομολόγησης Flooding, Epidemic και PRoPHET, για τις διάφορες τιμές του χρόνου για τον οποίον ένα Πακέτο Δεδομένων θεωρείται "πρόσφατο" (Data Freshness Period)

Παρατηρούμε στην Εικόνα 29 ότι στο παρόν πείραμα ενώ υπάρχει ομοιότητα στις καμπύλες των πρωτόκολλων Flooding και Epidemic, δεν υπάρχει τόσο μεγάλη ταύτιση. Όμως και πάλι, σε όλες της περιπτώσεις, η συμπεριφορά τους είναι βελτιωμένη σε σχέση με τη συμπεριφορά του πρωτοκόλλου PRoPHET, το οποίο παρουσιάζει σημαντική καθυστέρηση στην ολοκλήρωση των πειραμάτων.

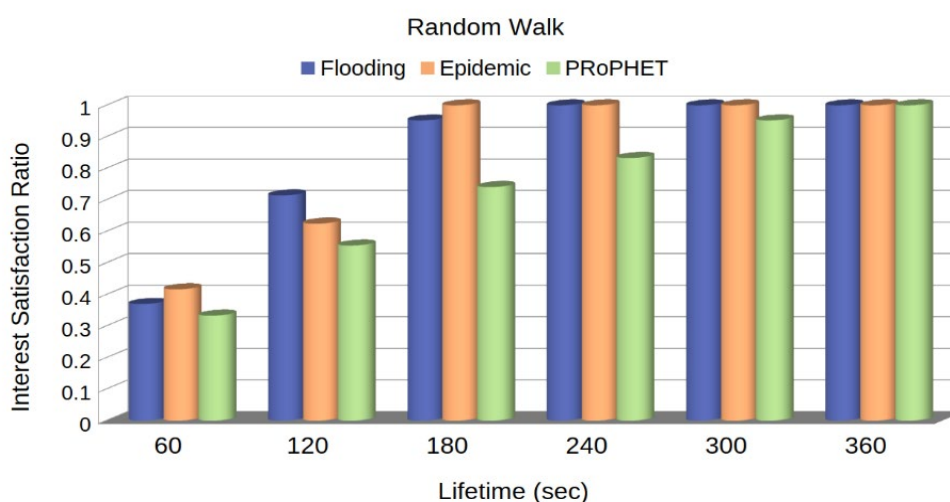
Είναι άξιο παρατήρησης πως εάν οι Εικόνες 27 και 29 μελετηθούν συνδυαστικά, προκύπτει ότι στις περιπτώσεις που έχουν το ίδιο Cache Hit Ratio, η στρατηγική δρομολόγησης Epidemic, παρουσιάζει ίδια ή και καλύτερη απόδοση από τη στρατηγική Flooding.

## 5.4. Πείραμα 4

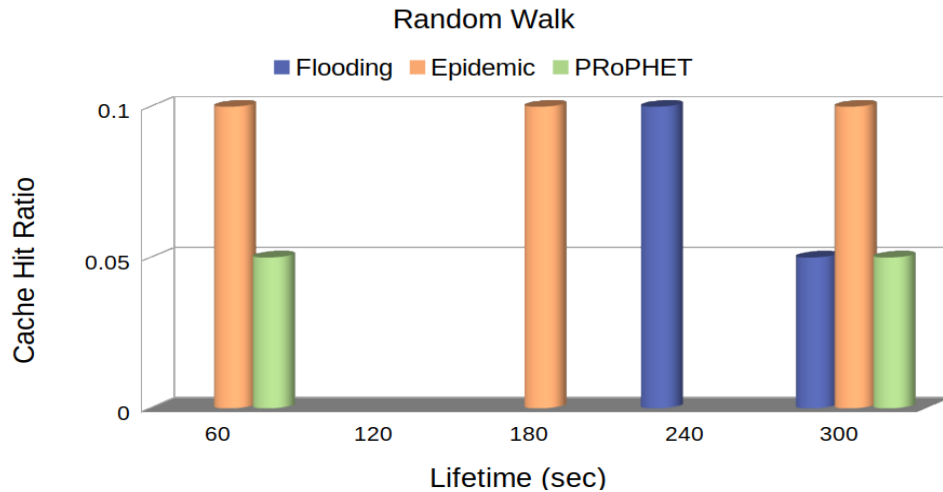
Στο πείραμα 4 θα μελετήσουμε την επίδραση της μεταβολής της διάρκειας ζωής (Lifetime) του Πακέτου Ενδιαφέροντος στην απόδοση του δικτύου για τα πρωτόκολλα δρομολόγησης Flooding, Epidemic και PRoPHET, για τρία διαφορετικά μοντέλα κινητικότητας. Για να εξάγουμε συμπεράσματα, θα χρησιμοποιήσουμε τις μετρήσεις της τιμής του χρόνου καθυστέρησης λήψης των Πακέτων Δεδομένων (Delay), της μέσης τιμής του χρόνου καθυστέρησης λήψης των Πακέτων Δεδομένων (Average Delay), του ρυθμού λήψης δεδομένων από την κρυφή μνήμη του consumer (Cache Hit Ratio) και του ρυθμού με τον οποίο ικανοποιούνται τα πακέτα Ενδιαφέροντος (Interest Satisfaction Ratio-ISR). Οι τιμές της διάρκειας ζωής (Lifetime) που θα εξετάσουμε θα ξεκινάνε από 1 λεπτό και θα φτάσουν σε κάποιες περιπτώσεις έως 7 λεπτά.



Εικόνα 30: Μέσος χρόνος καθυστέρησης λήψης δεδομένων (Average Delay), για τις διάφορες τιμές της διάρκειας ζωής (Lifetime) του Πακέτου Ενδιαφέροντος, για τα πρωτόκολλα δρομολόγησης Flooding, Epidemic και PRoPHET, για το σενάριο κινητικότητας Random Walk.



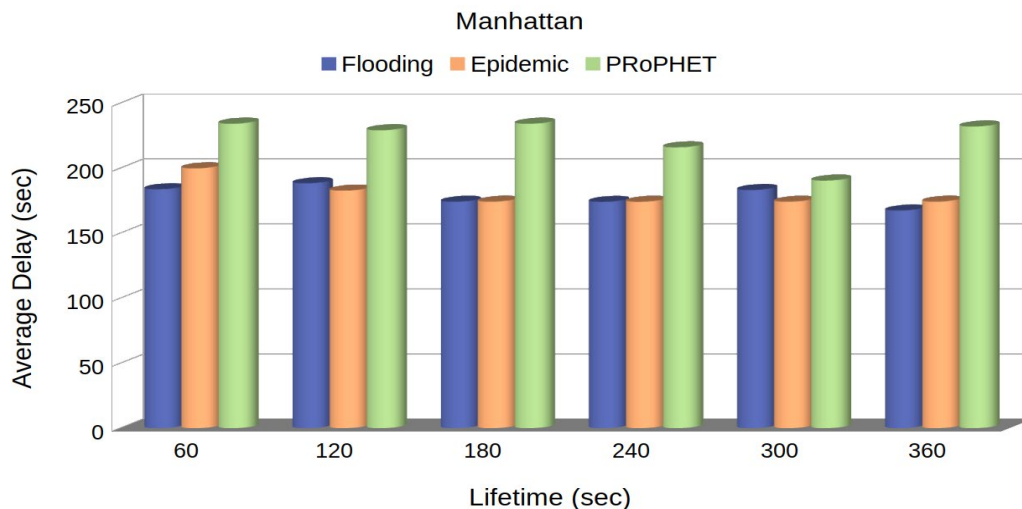
Εικόνα 31: Ρυθμός με τον οποίο ικανοποιούνται τα πακέτα Ενδιαφέροντος (Interest Satisfaction Ratio-ISR), για τις διάφορες τιμές της διάρκειας ζωής (Lifetime) του Πακέτου Ενδιαφέροντος, για τα πρωτόκολλα δρομολόγησης Flooding, Epidemic και PRoPHET, για το σενάριο κινητικότητας Random Walk.



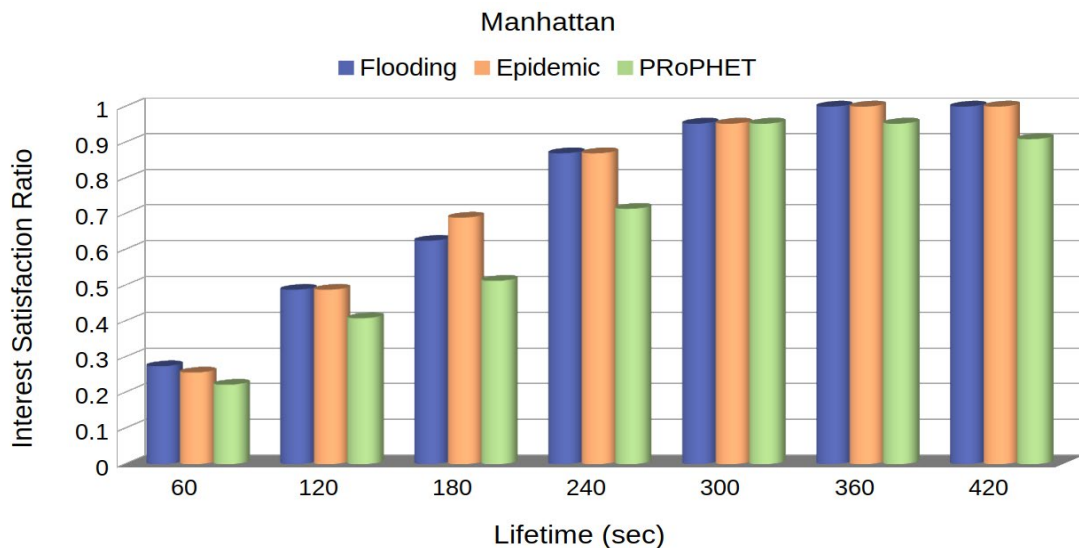
Εικόνα 32: Ρυθμός λήψης δεδομένων από την κρυφή μνήμη του consumer (Cache Hit Ratio), για τις διάφορες τιμές της διάρκειας ζωής (Lifetime) του Πακέτου Ενδιαφέροντος, για τα πρωτόκολλα δρομολόγησης Flooding, Epidemic και PRoPHET, για το σενάριο κινητικότητας Random Walk.

Συνολικά από τις Εικόνες 30, 33 και 36, φαίνεται πως η μεταβολή του Lifetime, δεν επηρεάζει ιδιαίτερα το μέσο χρόνο καθυστέρησης λήψης δεδομένων (Average Delay), για όλα τα μοντέλα κινητικότητας και τα πρωτόκολλα δρομολόγησης, με τις μεγαλύτερες διακυμάνσεις να παρουσιάζονται με τη μορφή χαμηλότερων Average Delay, στις περιπτώσεις οποίου έχουμε υψηλό Cache Hit Ratio. Η σχετική σταθερότητα βεβαίως του μέσου χρόνου καθυστέρησης λήψης δεδομένων (Average Delay), ήταν αναμενόμενη μιας και το Timeout που προκαλεί η πάροδος του χρόνου ζωής του πακέτου ενδιαφέροντος, προκαλεί επαναποστολή του πακέτου, χωρίς να επηρεάζει τα ήδη υπάρχοντα Πακέτα Ενδιαφέροντος και τυχόν ανάλογα Πακέτα Δεδομένων που βρίσκονται στο δίκτυο.

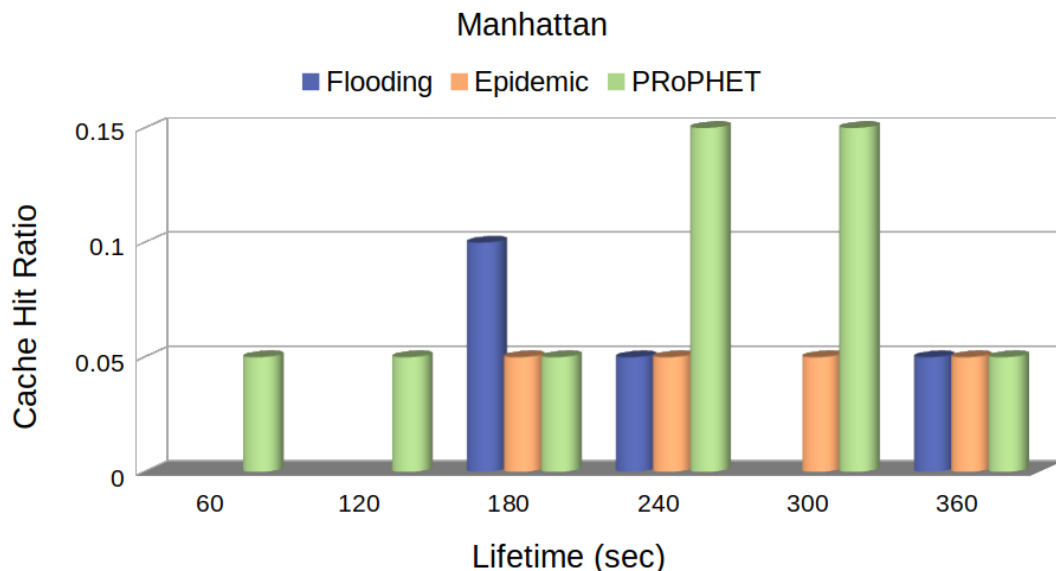
Επίσης, επιβεβαιώνεται και πάλι το συμπέρασμα ότι τα πρωτόκολλα Flooding και Epidemic, παρουσιάζουν μικρότερο μέσο χρόνο καθυστέρησης λήψης δεδομένων (Average Delay), από το πρωτόκολλο PRoPHET, ανεξάρτητα από το μοντέλο κινητικότητας που ακολουθούν οι κόμβοι.



Εικόνα 33: Μέσος χρόνος καθυστέρησης λήψης δεδομένων (Average Delay), για τις διάφορες τιμές της διάρκειας ζωής (Lifetime) του Πακέτου Ενδιαφέροντος, για τα πρωτόκολλα δρομολόγησης Flooding, Epidemic και PRoPHET, για το σενάριο κινητικότητας Manhattan.



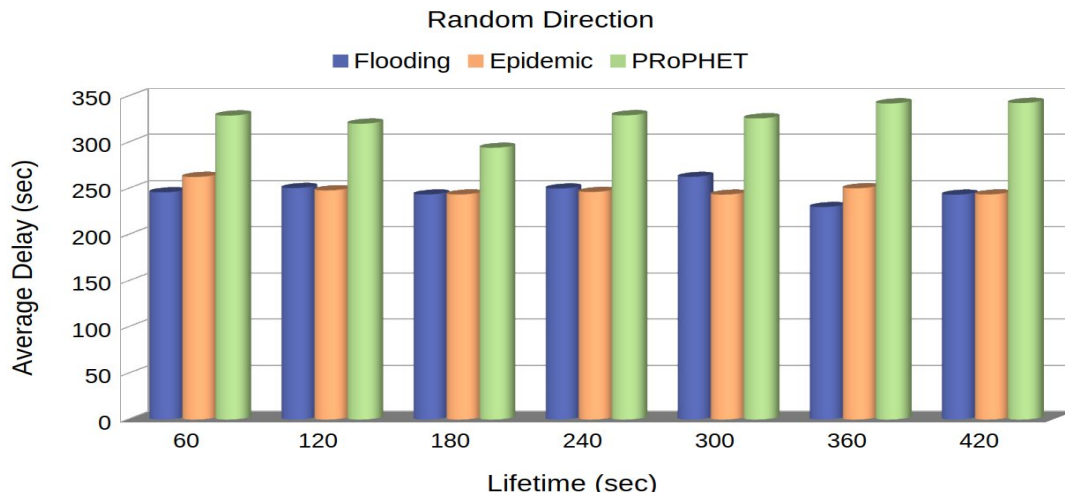
Εικόνα 34: Ρυθμός με τον οποίο ικανοποιούνται τα πακέτα Ενδιαφέροντος (Interest Satisfaction Ratio-ISR), για τις διάφορες τιμές της διάρκειας ζωής (Lifetime) του Πακέτου Ενδιαφέροντος, για τα πρωτόκολλα δρομολόγησης Flooding, Epidemic και PRoPHET, για το σενάριο κινητικότητας Manhattan.



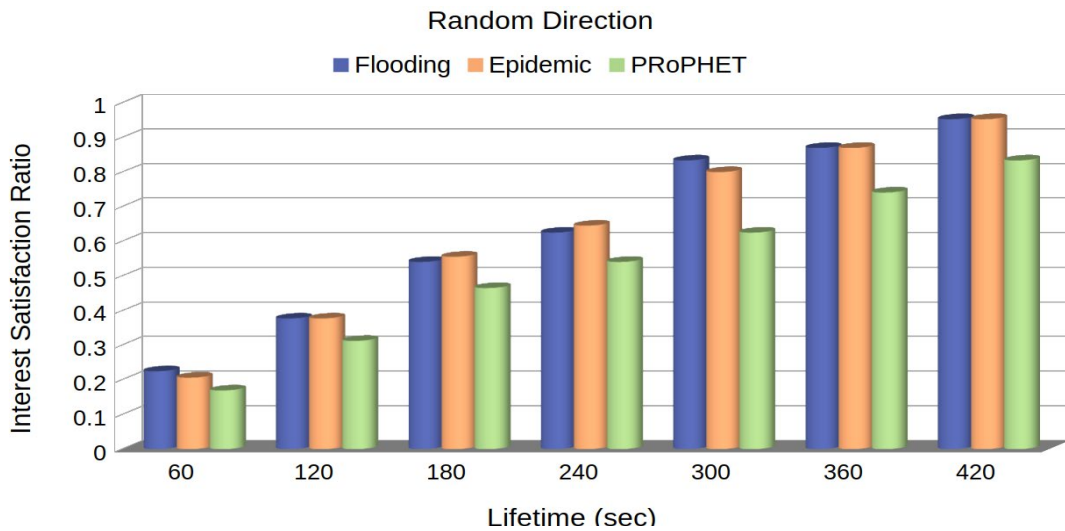
Εικόνα 35: Ρυθμός λήψης δεδομένων από την κρυφή μνήμη του consumer (Cache Hit Ratio), για τις διάφορες τιμές της διάρκειας ζωής (Lifetime) του Πακέτου Ενδιαφέροντος, για τα πρωτόκολλα δρομολόγησης Flooding, Epidemic και PRoPHET, για το σενάριο κινητικότητας Manhattan.

Παρατηρώντας συνδυαστικά τις Εικόνες 31, 34 και 37, προκύπτει το συμπέρασμα ότι ο ρυθμός με τον οποίο ικανοποιούνται τα πακέτα Ενδιαφέροντος (ISR) αυξάνεται όσο μεγαλώνει και η διάρκεια ζωής (Lifetime). Αυτό είναι και η λογική συμπεριφορά του δικτύου, διότι όσο περισσότερος χρόνος ζωής δίνεται σε ένα πακέτο, τόσο αυξάνονται και οι πιθανότητες το πακέτο αυτό να ικανοποιηθεί.

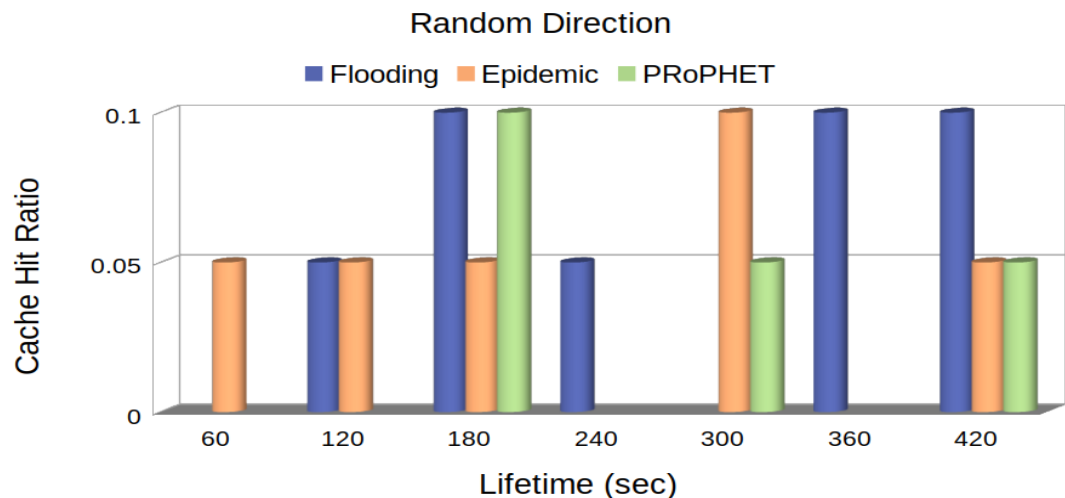
Βέβαια, σε μεμονωμένες περιπτώσεις, όπως στα 420 s στο Σχήμα 5, το ISR για το πρωτόκολλο PRoPHET μειώνεται ελαφρώς σε σχέση με τα 360 s. Αυτό είναι κάτι που μπορεί να παρατηρηθεί για το πρωτόκολλο PRoPHET, διότι λόγω της πιθανότητας μετάδοσης του πακέτου στην οποία βασίζεται η λογική του, έχει ένα βαθμό τυχαιότητα κι έτσι μπορεί να δημιουργήσει μεγαλύτερες καθυστερήσεις και να παρουσιάσει τέτοιες μικρές ανομοιομορφίες.



Εικόνα 36: Μέσος χρόνος καθυστέρησης λήψης δεδομένων (Average Delay), για τις διάφορες τιμές της διάρκειας ζωής (Lifetime) του Πακέτου Ενδιαφέροντος, για τα πρωτόκολλα δρομολόγησης Flooding, Epidemic και PRoPHET, για το σενάριο κινητικότητας Random Direction.



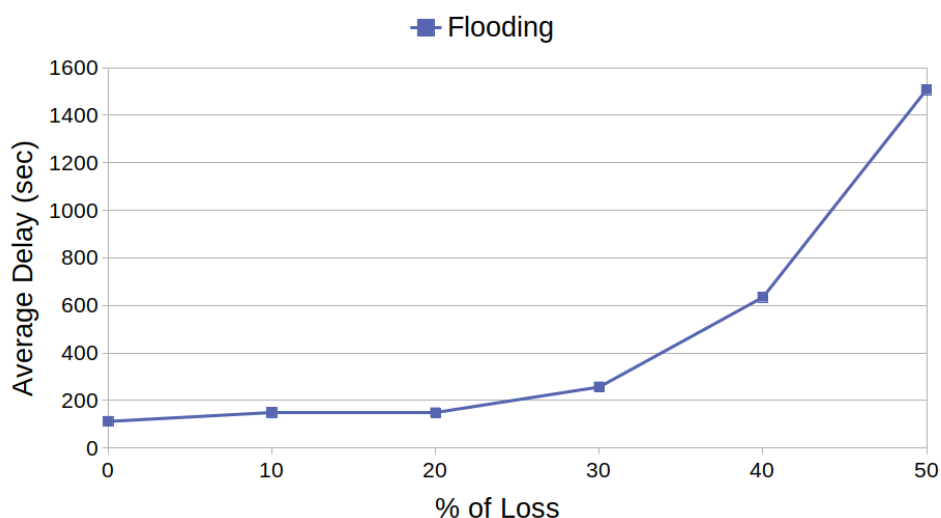
Εικόνα 37: Ρυθμός με τον οποίο ικανοποιούνται τα πακέτα Ενδιαφέροντος (Interest Satisfaction Ratio-ISR), για τις διάφορες τιμές της διάρκειας ζωής (Lifetime) του Πακέτου Ενδιαφέροντος, για τα πρωτόκολλα δρομολόγησης Flooding, Epidemic και PRoPHET, για το σενάριο κινητικότητας Random Direction.



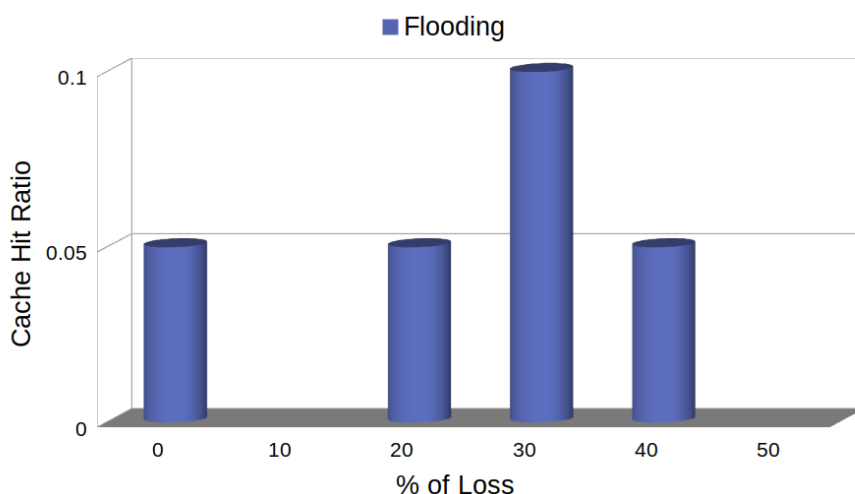
Εικόνα 38: Ρυθμός λήψης δεδομένων από την κρυφή μνήμη του consumer (Cache Hit Ratio), για τις διάφορες τιμές της διάρκειας ζωής (Lifetime) του Πακέτου Ενδιαφέροντος, για τα πρωτόκολλα δρομολόγησης Flooding, Epidemic και PRoPHET, για το σενάριο κινητικότητας Random Direction.

## 5.5. Πείραμα 5

Στο πείραμα 5 θα μελετήσουμε πως επηρεάζει η απώλεια (Loss) ενός ποσοστού των συνολικών πακέτων, την απόδοση του δικτύου για το πρωτόκολλο δρομολόγησης Flooding. Για να εξάγουμε συμπεράσματα θα χρησιμοποιήσουμε τις μετρήσεις της τιμής του χρόνου καθυστέρησης λήψης των Πακέτων Δεδομένων (Delay), τη μέση τιμή του χρόνου καθυστέρησης λήψης των Πακέτων Δεδομένων (Average Delay), το ρυθμό λήψης δεδομένων από την κρυφή μνήμη του consumer (Cache Hit Ratio) αλλά και το ρυθμό με τον οποίο ικανοποιούνται τα πακέτα Ενδιαφέροντος (Interest Satisfaction Ratio-ISR) στο σημείο όπου αυτός παρουσιάζει κάποιο ενδιαφέρον. Οι τιμές του ποσοστού απώλεια δεδομένων που θα εξετάσουμε στο δίκτυο θα είναι 0%, 10%, 20%, 30%, 40% και 50%.



Εικόνα 39: Μέσος χρόνος καθυστέρησης λήψης δεδομένων (Average Delay) για το πρωτόκολλο δρομολόγησης Flooding, για τις διάφορες τιμές του ποσοστού της απώλειας (% of Loss).

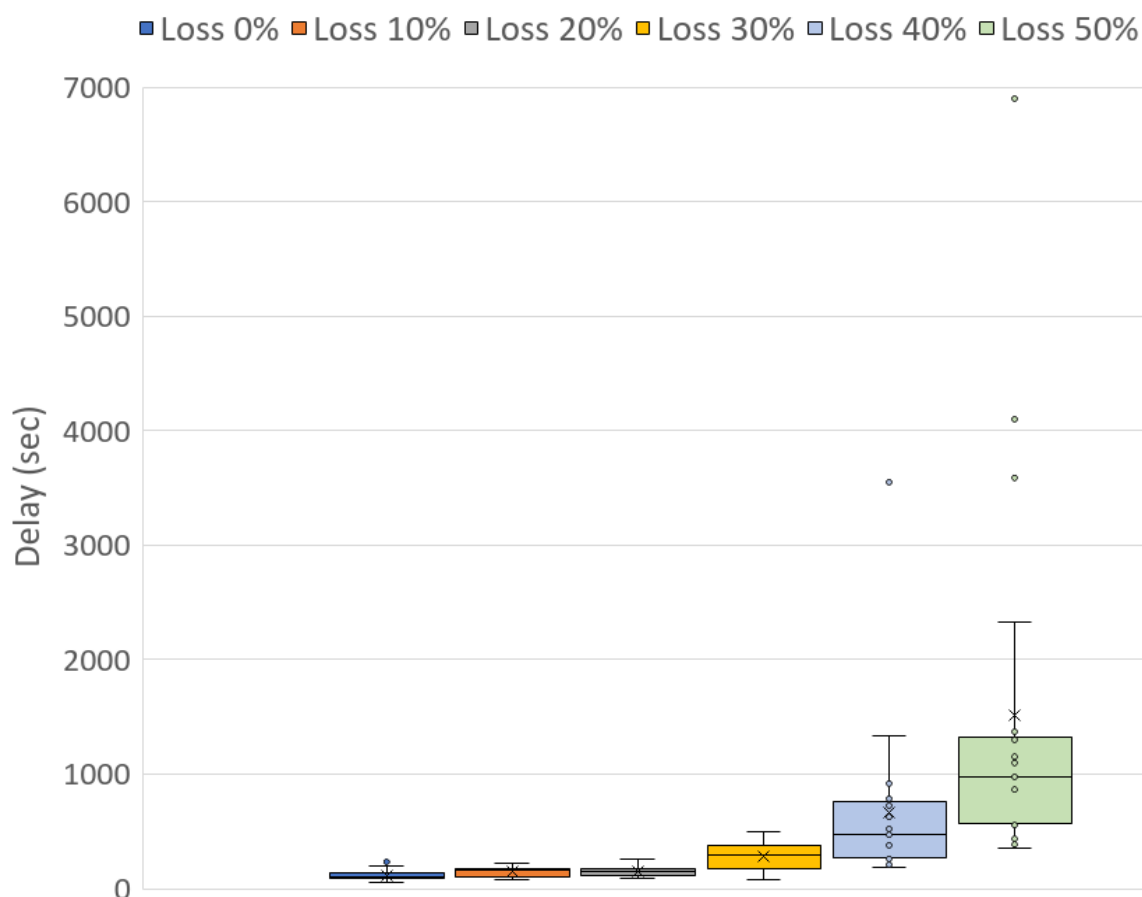


Εικόνα 40: Ρυθμός λήψης δεδομένων από την κρυφή μνήμη του consumer (Cache Hit Ratio) για το πρωτόκολλο δρομολόγησης Flooding, για τις διάφορες τιμές του ποσοστού της απώλειας (% of Loss).

Παρατηρούμε από την Εικόνα 39 πως από το 20% κι έπειτα, όσο αυξάνεται η τιμή του ποσοστού της απώλειας (% of Loss), τόσο αυξάνεται και η μέση τιμή του χρόνου καθυστέρησης λήψης των πακέτων Δεδομένων (Average Delay), με το βήμα της αύξησης να μεγαλώνει συνεχώς. Αν παρατηρήσουμε συνδυαστικά τα Σχήματα 1 και 2 θα δούμε πως στα



μικρότερα ποσοστά 0-20% όπου φαίνεται να υπάρχει μία σχετική σταθερότητα στο Average Delay, αυτή οφείλεται στο αυξημένο Cache Hit Ratio στο 20% και στην έλλειψη cache hit στο 10%. Συνεπώς τ' αποτελέσματα αυτά επιβεβαιώνουν αυτό που περιμέναμε λογικά, δηλαδή την αύξηση του Average Delay, όσο αυξάνεται το ποσοστό της απώλειας των πακέτων.

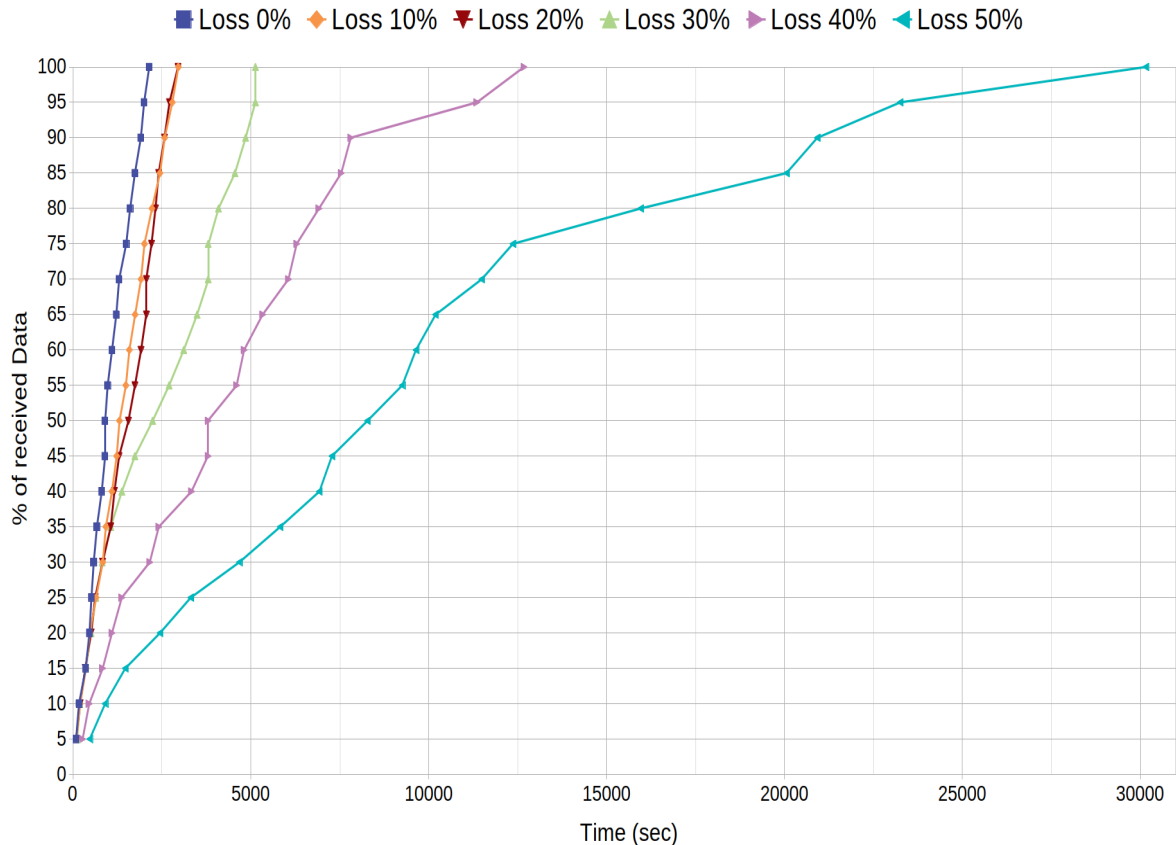


Εικόνα 41: Αναλυτικές τιμές του Delay και της διασποράς των τιμών για το πρωτόκολλο δρομολόγησης Flooding, για τις διάφορες τιμές του ποσοστού της απώλειας (% of Loss).

Στο Σχήμα 3, στο οποίο παρουσιάζεται η διασπορά των τιμών που είχε στο πείραμα ο χρόνος καθυστέρησης λήψης των δεδομένων (Delay), μη συμπεριλαμβανομένων των περιπτώσεων που είχαμε Cache Hit και παρουσιάζεται και η ασυμμετρία των δεδομένων, φαίνεται και πάλι πως ο χρόνος καθυστέρησης λήψης των Πακέτων Δεδομένων (Delay) αυξάνεται όσο αυξάνεται το ποσοστό της απώλειας.

Αξίζει να σημειωθεί πως πέραν του προηγούμενου πειράματος που εξετάζαμε το Lifetime, ήταν η πρώτη φορά που ο χρόνος καθυστέρησης λήψης των Πακέτων Δεδομένων (Delay) ξεπέρασε το ορισμένο από εμάς Lifetime. Αυτό συνέβη για δύο πακέτα ενδιαφέροντος, στο πείραμα με ποσοστό απώλειας πακέτων 50%, το οποίο διήρκεσε συνολικά 8 ώρες και 15 λεπτά και είχε ως αποτέλεσμα τη μείωση του ρυθμού με τον οποίο ικανοποιούνται τα πακέτα Ενδιαφέροντος (Interest Satisfaction Ratio-ISR) στο 0.909.

Επίσης, κάτι που παρουσιάζει ιδιαίτερο ενδιαφέρον και αποτυπώνεται στο Σχήμα 3 είναι ότι στην περίπτωση που έχουμε ποσοστό απώλειας 50%, οι μεγαλύτερες τιμές και κυρίως αυτή που προσεγγίζει τις 2 ώρες, είναι τόσο υψηλές σε σχέση με τις υπόλοιπες που ανεβάζουν τόσο πολύ το μέσο όρο (πάνω από το 75ο εκατοστημόριο-75th percentile)



Εικόνα 42: Ποσοστό των παραδοθέντων Πακέτων Δεδομένων ως προς το χρόνο εκτέλεσης του πειράματος για το πρωτόκολλο δρομολόγησης Flooding, για τις διάφορες τιμές του ποσοστού της απώλειας (% of Loss).

Το Σχήμα 4, έρχεται να επιβεβαιώσει τον ισχυρισμό ότι χωρίς cache hit, το πείραμα με το 20% Loss θα είχε μεγαλύτερο συνολικό χρόνο ολοκλήρωσης από το 10% και αυτό φαίνεται από το γεγονός ότι από το 45% των ληφθέντων δεδομένων, αρχίζει η καμπύλη του να αποκλίνει προς τα δεξιά και τελικά συγκλίνει με την καμπύλη του 10% Loss μετά από 2 cache hit. Βέβαια η διαφορά τους και πάλι δεν είναι μεγάλη κι αυτό είναι λογικό γιατί τα ποσοστά αυτά απώλειας είναι σχετικά μικρά και αρκετά κοντινά μεταξύ τους.

Τέλος, θα μπορούσαμε να γίνουμε πιο συγκεκριμένοι υπολογίζοντας το ποσοστό της αύξησης του μέσου χρόνου καθυστέρησης λήψης δεδομένων (Average Delay) για τις διάφορες τιμές του ποσοστού της απώλειας (% of Loss). Ο υπολογισμός αυτός έγινε έχοντας αφαιρέσει τις περιπτώσεις που είχαμε cache hit, έτσι ώστε να έχουμε μια πιο ξεκάθαρη εικόνα της επίδρασης της απώλειας πακέτων στην απόδοση του δικτύου.

Για 10% Loss το Average Delay αυξήθηκε κατά 26.045%

Για 20% Loss το Average Delay αυξήθηκε κατά 32.241%

Για 30% Loss το Average Delay αυξήθηκε κατά 142.366%

Για 40% Loss το Average Delay αυξήθηκε κατά 467.975%

Για 50% Loss το Average Delay αυξήθηκε κατά 1182.732%

## 5.6. Συμπεράσματα και Μελλοντική Εργασία

Στη διπλωματική εργασία αυτή δημιουργήθηκε ένα Docker container που περιλαμβάνει τον εξομοιωτή δικτύων CORE και αφορά τις τεχνολογίες NDN και DTN. Το container αυτό χρησιμοποιήθηκε για τη μελέτη της απόδοσης των πρωτόκολλων δρομολόγησης Flooding, Epidemic και ProPHET σε ασύρματα δίκτυα που προσομοιώνουν ένα δίκτυο IoT και αποτελούνται από σταθερούς αλλά και κινούμενους κόμβους δικτύων και παρουσίαζαν διακοπτόμενες συνδέσεις. Εξετάστηκε η επίδραση διαφόρων παραμέτρων για διαφορετικά μοντέλα κινητικότητας και τ' αποτελέσματα στα οποία οδηγηθήκαμε είναι τα εξής:

A. Από τη μελέτη των αποτελεσμάτων όλων των πειραμάτων, προκύπτει ότι τα πρωτόκολλα δρομολόγησης Flooding και Epidemic παρουσιάζουν παρόμοια συμπεριφορά μεταξύ τους, η οποία είναι καλύτερη του πρωτοκόλλου PROPHET. Η απόδοση του πρωτοκόλλου PROPHET συγκριτικά με τα άλλα δύο πρωτόκολλα, χειροτερεύει όσο αυξάνονται οι κόμβοι του δικτύου.

B. Η αύξηση των ενδιάμεσων κόμβων του δικτύου επιφέρει σημαντική μείωση της μέσης τιμής του χρόνου καθυστέρησης λήψης δεδομένων (Average Delay). Το μεγαλύτερο πλήθος κόμβων, δημιουργεί περισσότερες πιθανές διαδρομές για τα Πακέτα Ενδιαφέροντος και συνεπώς καλύτερη απόδοση από ένα δίκτυο με λιγότερους κόμβους.

Γ. Η επιλογή ενός μεγάλου χρονικού διαστήματος για το οποίο ένα Πακέτο Δεδομένων θεωρείται πρόσφατο (Data Freshness Period) ιδανικά θα πρέπει να διατηρείται σε όσο μεγαλύτερα επίπεδα γίνεται διότι επηρεάζει το ρυθμό με τον οποίο λαμβάνονται δεδομένα από την κρυφή μνήμη (Cache Hit Ratio) και η αύξηση αυτού, επιφέρει μείωση του μέσου χρόνου καθυστέρησης (Average Delay).

Δ. Η διάρκεια ζωής των πακέτων Ενδιαφέροντος (Interest Lifetime) καθορίζει το ρυθμό με τον οποίο ικανοποιούνται τα πακέτα Ενδιαφέροντος (Interest Satisfaction Ratio), όμως δεν επηρεάζει τη μέση τιμή του χρόνου καθυστέρησης λήψης δεδομένων (Average Delay). Η επιλογή της βέλτιστης τιμής της διάρκειας ζωής των πακέτων Ενδιαφέροντος (Interest Lifetime) προκύπτει από τους διαθέσιμους πόρους του συστήματος (εύρος ζώνης, διαθέσιμος αποθηκευτικός χώρος ).

E. Ακόμη και μικρά ποσοστά απώλειας, επιφέρουν σημαντική αύξηση στο μέσο χρόνο καθυστέρησης λήψης των δεδομένων (Average Delay).

Z. Τα μοντέλα κινητικότητας δεν επηρέασαν τη συγκριτική εικόνα των διαφορετικών πρωτοκόλλων δρομολόγησης και την επίδραση των παραμέτρων στην απόδοση του δικτύου.

Χρησιμοποιώντας το container το οποίο φτιάξαμε, θα μπορούσαν να γίνουν πειράματα για να διερευνηθούν και άλλες πτυχές της απόδοσης του δικτύου NDN-DTN. Θα μπορούσε να δοθεί έμφαση στη σκοπιά του DTN και χρησιμοποιώντας εργαλεία που είναι ήδη ενσωματωμένα στην υλοποίηση μας ή προσθέτοντας κι άλλα, να μελετηθεί η κίνηση των bundles στο δίκτυο. Έτσι, διαφοροποιώντας τις ίδιες παραμέτρους με τα δικά μας πειράματα, θα δινόταν η δυνατότητα να εξεταστεί η επίδραση τους στο επίπεδο DTN του δικτύου. Ενδιαφέρον θα είχε επίσης να παρατηρηθεί η επίδραση κι άλλων παραμέτρων στην απόδοση του δικτύου, μιας και ο CORE μας δίνει τη δυνατότητα να τις ρυθμίσουμε, όπως το Εύρος Ζώνης (Bandwidth) ή η Καθυστέρηση των πακέτων στο δίκτυο (Delay).

## Βιβλιογραφία

- [1] V. Cerf *et al.*, "Interplanetary internet (ipn): architectural definition," *undefined*, 2001, Available: [https://www.semanticscholar.org/paper/Interplanetary-internet-\(ipn\)%3A-architectural-Cerf-Burleigh/fc29fa1160d4af4ba18a93a5afc92922af416b21](https://www.semanticscholar.org/paper/Interplanetary-internet-(ipn)%3A-architectural-Cerf-Burleigh/fc29fa1160d4af4ba18a93a5afc92922af416b21)
- [2] L. Zhang *et al.*, "Named data networking," *SIGCOMM Comput. Commun. Rev.*, vol. 44, no. 3, pp. 66–73, Jul. 2014, doi: 10.1145/2656877.2656887.
- [3] B. Ahlgren, C. Dannewitz, C. Imbrenda, D. Kutscher, and B. Ohlman, "A survey of information-centric networking," *IEEE Communications Magazine*, vol. 50, no. 7, pp. 26–36, Jul. 2012, doi: 10.1109/MCOM.2012.6231276.
- [4] C.-A. Sarros *et al.*, "Connecting the Edges: A Universal, Mobile-Centric, and Opportunistic Communications Architecture," *IEEE Communications Magazine*, vol. 56, pp. 136–143, Feb. 2018, doi: 10.1109/MCOM.2018.1700325.
- [5] K. Fall, "A Delay-Tolerant Network Architecture for Challenged Internets," p. 8.
- [6] "CFDP Protocol Specification, CCSDS 727.0-B-1, Jan 2002, <http://www.ccsds.org>."
- [7] S. Burleigh *et al.*, "Delay-tolerant networking: an approach to interplanetary Internet," *IEEE Communications Magazine*, vol. 41, no. 6, pp. 128–136, Jun. 2003, doi: 10.1109/MCOM.2003.1204759.
- [8] S. Burleigh, V. G. Cerf, J. Crowcroft, and V. Tsaoussidis, "Space for Internet and Internet for space," *Ad Hoc Networks*, vol. 23, pp. 80–86, Dec. 2014, doi: 10.1016/j.adhoc.2014.06.005.
- [9] R. Dudukovich, "APPLICATION OF MACHINE LEARNING TECHNIQUES TO DELAY TOLERANT NETWORK ROUTING," p. 184.
- [10] "Low Earth orbit." [https://www.esa.int/ESA\\_Multimedia/Images/2020/03/Low\\_Earth\\_orbit](https://www.esa.int/ESA_Multimedia/Images/2020/03/Low_Earth_orbit)
- [11] J. Wroclawski, "The Metanet: White Pape," Vienna, May 1997.
- [12] V. Cerf and R. Kahn, "A Protocol for Packet Network Intercommunication," *IEEE Transactions on Communications*, vol. 22, no. 5, pp. 637–648, May 1974, doi: 10.1109/TCOM.1974.1092259.
- [13] S. Floyd and M. Allman, "Comments on the Usefulness of Simple Best-Effort Traffic," RFC Editor, RFC5290, Jul. 2008. doi: 10.17487/rfc5290.
- [14] E. Chen and J. Stewart, "A Framework for Inter-Domain Route Aggregation," RFC Editor, RFC2519, Feb. 1999. doi: 10.17487/rfc2519.
- [15] P. V. Mockapetris, *RFC1034: Domain names - concepts and facilities*. USA: RFC Editor, 1987.
- [16] M. H. Mealling and R. Denenberg, "Report from the Joint W3C/IETF URI Planning Interest Group: Uniform Resource Identifiers (URIs), URLs, and Uniform Resource Names (URNs): Clarifications and Recommendations," Internet Engineering Task Force, Request for Comments RFC 3305, Aug. 2002. doi: 10.17487/RFC3305.
- [17] W. Adjie-Winoto, E. Schwartz, H. Balakrishnan, and J. Lilley, "The design and implementation of an intentional naming system," p. 16.
- [18] "Doyle, Jeff & Carroll, Jennifer (2005). Routing TCP/IP, Volume I, Second Ed. Cisco Press. ISBN 978-1-58705-202-6. Ciscopress ISBN 1-58705-202-4."
- [19] A. Vahdat and D. Becker, "Epidemic Routing for Partially-Connected Ad Hoc Networks," p. 14.
- [20] A. Lindgren, A. Doria, and O. Schelen, "Probabilistic Routing in Intermittently Connected Networks," p. 16.
- [21] "Understanding Denial-of-Service Attacks | CISA." <https://us-cert.cisa.gov/ncas/tips/ST04-015>
- [22] D. Bucur, G. Iacca, G. Squillero, and A. Tonda, "Black Holes and Revelations: Using Evolutionary Algorithms to Uncover Vulnerabilities in Disruption-Tolerant Networks," *Applications of Evolutionary Computation: 18th European Conference, EvoApplications 2015, Copenhagen, Denmark, April 8-10, 2015, Proceedings*, pp. 29–41, 2015, doi: 10.1007/978-3-319-16549-3\_3.
- [23] D. Bucur and G. Iacca, "Improved search methods for assessing Delay-Tolerant Networks vulnerability to colluding strong heterogeneous attacks," *Expert Systems with Applications*, vol. 80, pp. 311–322, Sep. 2017, doi: 10.1016/j.eswa.2017.03.035.
- [24] A. Kate, G. M. Zaverucha, and U. Hengartner, "Anonymity and security in delay tolerant networks," in *2007 Third International Conference on Security and Privacy in Communications Networks and the Workshops - SecureComm 2007*, Nice, France, 2007, pp. 504–513. doi: 10.1109/SECCOM.2007.4550373.
- [25] V. C. Stephen Farrell, "Security Considerations in Space and Delay Tolerant Networks," in *2nd IEEE International Conference on Space Mission Challenges for Information Technology (SMC-IT'06)*, Pasadena, CA, USA, 2006, pp. 29–38. doi: 10.1109/SMC-IT.2006.66.

- [26] A. Seth and S. Keshav, "Practical security for disconnected nodes," in *1st IEEE ICNP Workshop on Secure Network Protocols, 2005. (NPSEC)*, Boston, Massachusetts, USA, 2005, pp. 31–36. doi: 10.1109/NPSEC.2005.1532050.
- [27] D. Quercia, S. Hailes, and L. Capra, "MobiRate: making mobile raters stick to their word," Jan. 2008, pp. 212–221. doi: 10.1145/1409635.1409664.
- [28] A. Afanasyev, J. Burke, T. Refaei, L. Wang, B. Zhang, and L. Zhang, "A Brief Introduction to Named Data Networking," in *MILCOM 2018 - 2018 IEEE Military Communications Conference (MILCOM)*, Oct. 2018, pp. 1–6. doi: 10.1109/MILCOM.2018.8599682.
- [29] M.-D. Robl, "Routing in Information Centric Networks," 2019, doi: 10.2313/NET-2019-10-1\_03.
- [30] C. Yi, A. Afanasyev, I. Moiseenko, L. Wang, B. Zhang, and L. Zhang, "A case for stateful forwarding plane," *Computer Communications*, vol. 36, no. 7, pp. 779–791, Apr. 2013, doi: 10.1016/j.comcom.2013.01.005.
- [31] A. Afanasyev, P. Mahadevan, I. Moiseenko, E. Uzun, and L. Zhang, "Interest Flooding Attack and Countermeasures in Named Data Networking," p. 9.
- [32] "D3.2\_-\_UMOBILE\_architecture\_report\_(2).pdf." Available: [http://www.umobile-project.eu/phocadownload/deliverables/D3.2\\_-\\_UMOBILE\\_architecture\\_report\\_\(2\).pdf](http://www.umobile-project.eu/phocadownload/deliverables/D3.2_-_UMOBILE_architecture_report_(2).pdf)
- [33] S. Schildt, J. Morgenroth, W.-B. Pöttner, and L. Wolf, "IBR-DTN: A lightweight, modular and highly portable Bundle Protocol implementation," *ECEASST*, vol. 37, Jan. 2011, doi: 10.14279/tuj.eceasst.37.512.
- [34] O. Ascigil, V. Sourlas, I. Psaras, and G. Pavlou, "Opportunistic off-path content discovery in information-centric networks," in *2016 IEEE International Symposium on Local and Metropolitan Area Networks (LANMAN)*, Rome, Italy, Jun. 2016, pp. 1–7. doi: 10.1109/LANMAN.2016.7548860.
- [35] W. Moreira and P. Mendes, "Impact of human behavior on social opportunistic forwarding," *Ad Hoc Networks*, vol. 25, pp. 293–302, Feb. 2015, doi: 10.1016/j.adhoc.2014.07.001.
- [36] A. Carzaniga, M. Papalini, and A. L. Wolf, "Content-based publish/subscribe networking and information-centric networking," in *Proceedings of the ACM SIGCOMM workshop on Information-centric networking - ICN '11*, Toronto, Ontario, Canada, 2011, p. 56. doi: 10.1145/2018584.2018599.
- [37] C.-A. Sarros, V. Demiroglou, and V. Tsaoussidis, "Intermittently-connected IoT devices: Experiments with an NDN-DTN architecture," in *2021 IEEE 18th Annual Consumer Communications & Networking Conference (CCNC)*, Las Vegas, NV, USA, Jan. 2021, pp. 1–9. doi: 10.1109/CCNC49032.2021.9369578.
- [38] A. Mouat, *Using Docker*, First edition. Beijing; Boston: O'Reilly, 2015.
- [39] "Docker Engine overview," *Docker Documentation*, Oct. 13, 2021. <https://docs.docker.com/engine/>
- [40] "Container Runtime with Docker Engine | Docker." <https://www.docker.com/products/container-runtime>
- [41] "Docker Architecture," *Aqua*. <https://www.aquasec.com/cloud-native-academy/docker-container/docker-architecture/>
- [42] "Overview of Docker Compose," *Docker Documentation*, Oct. 13, 2021. <https://docs.docker.com/compose/>
- [43] J. Ahrenholz, C. Danilov, T. R. Henderson, and J. H. Kim, "CORE: A real-time network emulator," in *MILCOM 2008 - 2008 IEEE Military Communications Conference*, Nov. 2008, pp. 1–7. doi: 10.1109/MILCOM.2008.4753614.
- [44] P. Lonkar, S. Chaudhari, and P. Chouskey, "A REVIEW ON DTN IMPLEMENTATION ON ANDROID PLATFORM FOR SOCIAL NEEDS," Feb. 2018. doi: 10.13140/RG.2.2.24070.42566.
- [45] A. Broder and M. Mitzenmacher, "Network Applications of Bloom Filters: A Survey," *Internet Mathematics*, vol. 1, no. 4, pp. 485–509, Jan. 2004, doi: 10.1080/15427951.2004.10129096.
- [46] "NFD: Named Data Networking Forwarding Daemon — Named Data Networking Forwarding Daemon (NFD) 0.7.1 documentation." <https://named-data.net/doc/NFD/current/>
- [47] N. Aschenbruck, R. Ernst, E. Gerhards-Padilla, and M. Schwamborn, "BonnMotion: a mobility scenario generation and analysis tool," presented at the 3rd International ICST Conference on Simulation Tools and Techniques, Malaga, Spain, 2010. doi: 10.4108/ICST.SIMUTOOLS2010.8684.
- [48] A. Frömmgen, D. Stohr, B. Koldehofe, and A. Rizk, "Don't Repeat Yourself: Seamless Execution and Analysis of Extensive Network Experiments," *arXiv:1802.03455 [cs]*, Feb. 2018, Available: <http://arxiv.org/abs/1802.03455>
- [49] T. Camp, J. Boleng, and V. Davies, "A survey of mobility models for ad hoc network research," *Wirel. Commun. Mob. Comput.*, vol. 2, no. 5, pp. 483–502, Aug. 2002, doi: 10.1002/wcm.72.

## Π.1. Κώδικες

### Dockerfile

from ubuntu:14.04 AS builder

#install the dependencies of build process

RUN apt-get update && apt-get install -y devscripts build-essential \

pkg-config libnl-3-dev libnl-genl-3-dev libnl-route-3-dev \

libnl-nf-3-dev libnl-cli-3-dev libssl-dev libsqlite3-dev openssl libdaemon-dev

libvmime-dev libarchive-dev \

automake autoconf pkg-config libtool libcppunit-dev libcrypto++-dev libboost-all-dev pkg-config libpcap-dev curl \

&& apt-get clean \

&& rm -rf /var/lib/apt/lists/\*

# copy all files from tar folder to /root

COPY ./tars /root/

COPY ./my\_confs /usr/local/etc/ndn

#untar the files and install the components

RUN cd /root && \

tar -xvf ibrccommon-1.0.1.tar.gz && \

cd ibrccommon-1.0.1 && \

./configure --with-openssl && \

make && \

make install && \

ldconfig

RUN cd /root && \

tar -xvf ibrdtn-1.0.1.tar.gz && \

cd ibrdtn-1.0.1 && \

./configure && \

make && \

make install && \

ldconfig

RUN cd /root && \

tar -xvf ibrdtnd-1.0.1.tar.gz && \

cd ibrdtnd-1.0.1 && \

./configure --with-curl && \

make && \

make install && \

ldconfig

RUN cd /root && \

tar -xvf ibrdtn-tools-1.0.1.tar.gz && \

```
cd ibrdtn-tools-1.0.1 && \  
./configure && \  
make && \  
make install && \  
ldconfig
```

```
RUN cd /root && \  
tar -xvf ndn-cxx_umobile.tar.gz && \  
cd ndn-cxx_umobile && \  
./waf configure && \  
./waf && \  
./waf install && \  
ldconfig
```

```
RUN cd /root && \  
tar -xvf ndn-dtn.tar.gz && \  
cd ./ndn-dtn && \  
mkdir websocketpp && \  
curl -L https://github.com/zaphoyd/websocketpp/archive/0.5.1.tar.gz >  
websocket.tar.gz && \  
tar xzf websocket.tar.gz -C websocketpp/ --strip 1 && \  
./waf configure && \  
./waf && \  
./waf install && \  
ldconfig && \  
rm -rf /root/ndn-dtn/*.tar.gz
```

```
RUN cd /root && \  
tar -xvf ndn-tools.tar.gz && \  
cd ndn-tools-ndn-tools-0.3 && \  
./waf configure && \  
./waf && \  
./waf install
```

```
RUN rm -rf /root/*.tar.gz
```

```
##### The software used in this evaluation is built in individual  
##### containers in a multi-step build and then copied to the  
##### experiment container in the end.
```

```
##### Compose the actual worker container  
FROM maciresearch/core_worker:0.4.2
```

```
RUN apt-get update && apt-get install -y sudo build-essential libcrypto++-dev  
libsqlite3-dev libboost-all-dev  
RUN sudo ldconfig
```

```
#Install wireshark
```



```

RUN apt-get update \
&& apt-get install --no-install-recommends -yq \
    libtk-img \
    lxterminal \
    tmux \
    wireshark \
&& apt-get clean

```

# Install further measuring tools

```

RUN apt-get update \
&& apt-get install --no-install-recommends -yq \
    bwm-ng \
    sysstat \
    tcpdump \
    patch \
    python-nacl \
    python-ipcalc \
    libdaemon-dev \
    libnl-3-dev \
    libnl-cli-3-dev \
    libnl-genl-3-dev \
    libnl-nf-3-dev \
    libnl-route-3-dev \
    libarchive-dev \
    psmisc \
&& apt-get clean

```

#copy binaries

```

COPY --from=builder /root/ndn-cxx_umobile /ndn-cxx_umobile
COPY --from=builder /usr/local/bin /usr/local/bin
COPY --from=builder /usr/local/sbin /usr/local/sbin
RUN sudo ldconfig

```

#copy libs ibrdtn

```

COPY --from=builder /usr/local/lib /usr/local/lib
RUN sudo ldconfig

```

#other libs for dtn and ndn

```

COPY --from=builder /usr/lib/libdaemon.so.0.5.0 /usr/lib/libvmime.so.0.0.0
/usr/lib/libgsasl.so.7.9.6 /usr/lib/
RUN sudo ldconfig

```

```

COPY --from=builder /usr/lib/x86_64-linux-gnu/libcrypt* /usr/lib/x86_64-linux-
gnu/libboost_random.so.1.54.0 /usr/lib/x86_64-linux-gnu/libboost_chrono.so.1.54.0
/usr/**/libcrypto++.so.9 /usr/lib/x86_64-linux-gnu/libboost_regex.so.1.54.0
/usr/lib/x86_64-linux-gnu/libboost_filesystem.so.1.54.0 /usr/lib/x86_64-linux-
gnu/libboost_program_options.so.1.54.0 /usr/lib/x86_64-linux-
gnu/libboost_thread.so.1.54.0 /usr/lib/x86_64-linux-gnu/libboost_system.so.1.54.0
/usr/lib/x86_64-linux-gnu/libnl-route-3.so.200.16.1 /usr/lib/x86_64-linux-
gnu/libntlm.so.0.0.19 /usr/lib/x86_64-linux-gnu/libgnutls.so.26.22.6 /usr/lib/x86_64-

```

```
linux-gnu/libgssapi_krb5.so.2.2 /usr/lib/x86_64-linux-gnu/libidn.so.11.6.11
/usr/lib/x86_64-linux-gnu/libk5crypto.so.3.1 /usr/lib/x86_64-linux-gnu/libkrb5.so.3.3
/usr/lib/x86_64-linux-gnu/libarchive.so.13.1.2 /usr/lib/x86_64-linux-
gnu/libnettle.so.4.7 /usr/lib/x86_64-linux-gnu/libkrb5support.so.0.1 /usr/lib/x86_64-
linux-gnu/libxml2.so.2.9.1 /usr/lib/x86_64-linux-gnu/libicuuc.so.52.1
/usr/lib/x86_64-linux-gnu/libicui18n.so.52.1 /usr/lib/x86_64-linux-
gnu/libpcap.so.1.5.3 /usr/lib/x86_64-linux-gnu/libicudata.so.52.1 /usr/lib/x86_64-
linux-gnu/
RUN sudo ldconfig
```

```
COPY --from=builder /lib/x86_64-linux-gnu/libnl-3.so.200.16.1 /lib/x86_64-linux-
gnu/libkeyutils.so.1.4 /lib/x86_64-linux-gnu/libssl.so.1.0.0 /lib/x86_64-linux-
gnu/liblzo2.so.2.0.0 /lib/x86_64-linux-gnu/
RUN sudo ldconfig
```

```
#copy nfd_conf for nfd
COPY --from=builder /usr/local/etc/ndn/ /usr/local/etc/ndn/
RUN sudo ldconfig
```

## YML file

```
version: "3"
services:
  core_ndn-dtn:
    build:
      context: .
      dockerfile: Dockerfile
    volumes:
      - /tmp/.X11-unix:/tmp/.X11-unix
      - $PWD/shared:/root/shared
    privileged: true
    network_mode: host
    environment:
      - DISPLAY=${DISPLAY}
```

## Bash Script Producer

```
#!/bin/bash

echo "Starting DTN daemon"
dtnd -v -c /usr/local/etc/ndn/flooding/dtn_producer3.conf &

sleep 5

echo "Starting NDN daemon"
cp /usr/local/etc/ndn/my_nfd_producer.conf /usr/local/etc/ndn/nfd.conf
nfd-start &

sleep 5
```

```

cd /usr/local/etc/ndn

#Register
sh ./nfdc_register_producer.sh;

DB_SENSORS="hum temp sound light movement hall"

#Start ndnpoke for all register values
for k in $DB_SENSORS
do
    for i in $(seq 0 29);
    do
        echo 'RANDOM' | ndnpoke -x 3600000 -w 7200000
ndn:/example/$k$i &
    done
    sleep 3
done

#When ndnpoke satisfied,it has to restart
while true
do
    for k in $DB_SENSORS
    do
        for i in $(seq 0 29);
        do
            if ! pgrep -f "/example/$k$i" > /dev/null
            then
                echo 'RANDOM' | ndnpoke -x 3600000 -w 7200000
ndn:/example/$k$i &
            fi
        done
    done
done

```

## Bash Script Consumer

```

#!/bin/bash

echo "Starting DTN daemon"
dtnd -v -c /usr/local/etc/ndn/prophet/dtn_consumer3.conf &

sleep 5

echo "Starting NDN daemon"
cp /usr/local/etc/ndn/my_nfd_consumer.conf /usr/local/etc/ndn/nfd.conf

```

```
nfd-start &

sleep 5

cd /usr/local/etc/ndn
#Register
sh ./nfdc_register_consumer.sh;

cp /usr/local/etc/ndn/my_consumer13.cpp /ndn-
cxx_umobile/examples/my_consumer13.cpp
cd /ndn-cxx_umobile/
./waf configure --with-examples
sleep 1
./waf
sleep 1
./build/examples/my_consumer13
```

## Π.2. Πίνακες Αποτελεσμάτων

### Πείραμα 1

| Average Delay |          |          |         |
|---------------|----------|----------|---------|
|               | Flooding | Epidemic | PRoPHET |
| 2 Κόμβοι      | 333.489  | 306.423  | 353.491 |
| 4 Κόμβοι      | 248.474  | 240.823  | 274.434 |
| 6 Κόμβοι      | 195.837  | 195.798  | 201.93  |
| 8 Κόμβοι      | 168.802  | 183.938  | 204.738 |
| 10 Κόμβοι     | 158.991  | 163.648  | 198.972 |

| Cache Hit Ratio |          |          |         |
|-----------------|----------|----------|---------|
|                 | Flooding | Epidemic | PRoPHET |
| 2 Κόμβοι        | 0.05     | 0.05     | 0.1     |
| 4 Κόμβοι        | 0        | 0        | 0       |
| 6 Κόμβοι        | 0        | 0        | 0.1     |
| 8 Κόμβοι        | 0.1      | 0        | 0       |
| 10 Κόμβοι       | 0.05     | 0        | 0.1     |

| 2 Κόμβοι                  |            |             |             |
|---------------------------|------------|-------------|-------------|
|                           | Flooding   | Epidemic    | PRoPHET     |
| 1 <sup>st</sup> Interest  | 374.413    | 373.234     | 993.559     |
| 2 <sup>nd</sup> Interest  | 250.053    | 255.054     | 246.865     |
| 3 <sup>rd</sup> Interest  | 355.079    | 350.747     | 333.255     |
| 4 <sup>th</sup> Interest  | 248.344    | 248.736     | 395.082     |
| 5 <sup>th</sup> Interest  | 0.00017847 | 334.992     | 605.13      |
| 6 <sup>th</sup> Interest  | 331.781    | 395.828     | 615.13      |
| 7 <sup>th</sup> Interest  | 395.091    | 605.126     | 365.072     |
| 8 <sup>th</sup> Interest  | 605.133    | 370.077     | 0.000118652 |
| 9 <sup>th</sup> Interest  | 374.839    | 240.742     | 367.41      |
| 10 <sup>th</sup> Interest | 237.581    | 282.654     | 0.000158654 |
| 11 <sup>th</sup> Interest | 282.53     | 452.505     | 155.036     |
| 12 <sup>th</sup> Interest | 455.335    | 155.038     | 387.75      |
| 13 <sup>th</sup> Interest | 284.832    | 388.379     | 49.3946     |
| 14 <sup>th</sup> Interest | 254.282    | 49.5624     | 595.703     |
| 15 <sup>th</sup> Interest | 495.117    | 440.538     | 387.47      |
| 16 <sup>th</sup> Interest | 540.129    | 155.033     | 155.038     |
| 17 <sup>th</sup> Interest | 155.045    | 385.081     | 490.114     |
| 18 <sup>th</sup> Interest | 385.078    | 263.208     | 387.715     |
| 19 <sup>th</sup> Interest | 490.1      | 0.000147428 | 155.032     |
| 20 <sup>th</sup> Interest | 155.035    | 381.948     | 385.081     |

| 4 Κόμβοι                  |          |          |         |
|---------------------------|----------|----------|---------|
|                           | Flooding | Epidemic | PRoPHET |
| 1 <sup>st</sup> Interest  | 384.397  | 379.223  | 554.082 |
| 2 <sup>nd</sup> Interest  | 175.034  | 175.03   | 429.354 |
| 3 <sup>rd</sup> Interest  | 435.085  | 435.08   | 250.128 |
| 4 <sup>th</sup> Interest  | 243.401  | 246.161  | 335.92  |
| 5 <sup>th</sup> Interest  | 286.708  | 282.939  | 390.102 |
| 6 <sup>th</sup> Interest  | 240.087  | 240.091  | 299.853 |
| 7 <sup>th</sup> Interest  | 424.042  | 425.65   | 309.436 |
| 8 <sup>th</sup> Interest  | 385.086  | 384.466  | 611.022 |
| 9 <sup>th</sup> Interest  | 374.392  | 374.595  | 365.093 |
| 10 <sup>th</sup> Interest | 235.681  | 237.598  | 294.896 |
| 11 <sup>th</sup> Interest | 210.079  | 206.338  | 145.222 |
| 12 <sup>th</sup> Interest | 290.054  | 291.301  | 149.866 |
| 13 <sup>th</sup> Interest | 165.031  | 163.797  | 150.047 |
| 14 <sup>th</sup> Interest | 74.9862  | 73.7132  | 166.765 |
| 15 <sup>th</sup> Interest | 155.056  | 155.038  | 213.344 |
| 16 <sup>th</sup> Interest | 215.04   | 215.347  | 149.204 |
| 17 <sup>th</sup> Interest | 100.02   | 169.29   | 150.04  |
| 18 <sup>th</sup> Interest | 175.036  | 52.22    | 235.061 |
| 19 <sup>th</sup> Interest | 255.255  | 161.181  | 149.903 |
| 20 <sup>th</sup> Interest | 145.031  | 147.428  | 139.36  |

| 4 Κόμβοι                  |          |          |         |
|---------------------------|----------|----------|---------|
|                           | Flooding | Epidemic | PRoPHET |
| 1 <sup>st</sup> Interest  | 384.397  | 379.223  | 554.082 |
| 2 <sup>nd</sup> Interest  | 175.034  | 175.03   | 429.354 |
| 3 <sup>rd</sup> Interest  | 435.085  | 435.08   | 250.128 |
| 4 <sup>th</sup> Interest  | 243.401  | 246.161  | 335.92  |
| 5 <sup>th</sup> Interest  | 286.708  | 282.939  | 390.102 |
| 6 <sup>th</sup> Interest  | 240.087  | 240.091  | 299.853 |
| 7 <sup>th</sup> Interest  | 424.042  | 425.65   | 309.436 |
| 8 <sup>th</sup> Interest  | 385.086  | 384.466  | 611.022 |
| 9 <sup>th</sup> Interest  | 374.392  | 374.595  | 365.093 |
| 10 <sup>th</sup> Interest | 235.681  | 237.598  | 294.896 |
| 11 <sup>th</sup> Interest | 210.079  | 206.338  | 145.222 |
| 12 <sup>th</sup> Interest | 290.054  | 291.301  | 149.866 |
| 13 <sup>th</sup> Interest | 165.031  | 163.797  | 150.047 |
| 14 <sup>th</sup> Interest | 74.9862  | 73.7132  | 166.765 |
| 15 <sup>th</sup> Interest | 155.056  | 155.038  | 213.344 |
| 16 <sup>th</sup> Interest | 215.04   | 215.347  | 149.204 |
| 17 <sup>th</sup> Interest | 100.02   | 169.29   | 150.04  |
| 18 <sup>th</sup> Interest | 175.036  | 52.22    | 235.061 |
| 19 <sup>th</sup> Interest | 255.255  | 161.181  | 149.903 |
| 20 <sup>th</sup> Interest | 145.031  | 147.428  | 139.36  |

| 6 Κόμβοι                  |          |          |             |
|---------------------------|----------|----------|-------------|
|                           | Flooding | Epidemic | PRoPHET     |
| 1 <sup>st</sup> Interest  | 244.636  | 173.152  | 388.296     |
| 2 <sup>nd</sup> Interest  | 137.897  | 210.092  | 279.974     |
| 3 <sup>rd</sup> Interest  | 169.995  | 169.986  | 249.998     |
| 4 <sup>th</sup> Interest  | 362.242  | 359.11   | 180.525     |
| 5 <sup>th</sup> Interest  | 180.035  | 180.042  | 142.759     |
| 6 <sup>th</sup> Interest  | 137.503  | 141.87   | 280.949     |
| 7 <sup>th</sup> Interest  | 285.432  | 283.231  | 190         |
| 8 <sup>th</sup> Interest  | 188.589  | 190.043  | 181.501     |
| 9 <sup>th</sup> Interest  | 182.89   | 180.981  | 0.000262414 |
| 10 <sup>th</sup> Interest | 242.199  | 240.05   | 243.649     |
| 11 <sup>th</sup> Interest | 127.881  | 135.031  | 216.491     |
| 12 <sup>th</sup> Interest | 292.205  | 288.135  | 288.227     |
| 13 <sup>th</sup> Interest | 240.049  | 240.055  | 155.499     |
| 14 <sup>th</sup> Interest | 184.383  | 147.786  | 185.059     |
| 15 <sup>th</sup> Interest | 202.179  | 238.211  | 203.435     |
| 16 <sup>th</sup> Interest | 206.425  | 210.041  | 206.73      |
| 17 <sup>th</sup> Interest | 292.222  | 290.081  | 389.123     |
| 18 <sup>th</sup> Interest | 100.018  | 95.0403  | 66.0187     |
| 19 <sup>th</sup> Interest | 62.8539  | 69.9805  | 190.374     |
| 20 <sup>th</sup> Interest | 77.1341  | 73.0514  | 0.000311431 |

| 8 Κόμβοι                  |             |          |         |
|---------------------------|-------------|----------|---------|
|                           | Flooding    | Epidemic | PRoPHET |
| 1 <sup>st</sup> Interest  | 166.181     | 172.733  | 385.127 |
| 2 <sup>nd</sup> Interest  | 205.112     | 210.127  | 159.622 |
| 3 <sup>rd</sup> Interest  | 170.707     | 169.958  | 354.31  |
| 4 <sup>th</sup> Interest  | 355.047     | 351.691  | 135.007 |
| 5 <sup>th</sup> Interest  | 135.033     | 138.942  | 140.085 |
| 6 <sup>th</sup> Interest  | 140.076     | 136.164  | 140.824 |
| 7 <sup>th</sup> Interest  | 140.034     | 143.896  | 195.054 |
| 8 <sup>th</sup> Interest  | 194.389     | 194.533  | 190.057 |
| 9 <sup>th</sup> Interest  | 190.704     | 190.543  | 235.07  |
| 10 <sup>th</sup> Interest | 180.048     | 178.542  | 240.069 |
| 11 <sup>th</sup> Interest | 221.078     | 221.333  | 329.376 |
| 12 <sup>th</sup> Interest | 152.368     | 153.752  | 122.071 |
| 13 <sup>th</sup> Interest | 0.000163835 | 246.58   | 147.022 |
| 14 <sup>th</sup> Interest | 247.833     | 124.987  | 140.048 |
| 15 <sup>th</sup> Interest | 127.26      | 146.214  | 145.05  |
| 16 <sup>th</sup> Interest | 0.000193695 | 143.848  | 321.779 |
| 17 <sup>th</sup> Interest | 146.668     | 141.212  | 394.107 |
| 18 <sup>th</sup> Interest | 140.03      | 113.835  | 124.395 |
| 19 <sup>th</sup> Interest | 145.035     | 208.576  | 59.4271 |
| 20 <sup>th</sup> Interest | 318.459     | 291.315  | 136.281 |



| 10 Κόμβοι                 |             |          |             |
|---------------------------|-------------|----------|-------------|
|                           | Flooding    | Epidemic | PRoPHET     |
| 1 <sup>st</sup> Interest  | 202.09      | 173.178  | 232.728     |
| 2 <sup>nd</sup> Interest  | 208.814     | 206.901  | 0.000207239 |
| 3 <sup>rd</sup> Interest  | 104.859     | 105.542  | 161.561     |
| 4 <sup>th</sup> Interest  | 143.412     | 143.543  | 156.786     |
| 5 <sup>th</sup> Interest  | 275.072     | 276.217  | 357.09      |
| 6 <sup>th</sup> Interest  | 139.994     | 135.04   | 135.039     |
| 7 <sup>th</sup> Interest  | 135.072     | 140.038  | 140.037     |
| 8 <sup>th</sup> Interest  | 145.038     | 102.995  | 141.876     |
| 9 <sup>th</sup> Interest  | 165.04      | 205.047  | 193.218     |
| 10 <sup>th</sup> Interest | 215.055     | 216.684  | 188.851     |
| 11 <sup>th</sup> Interest | 183.157     | 183.419  | 184.243     |
| 12 <sup>th</sup> Interest | 0.000309303 | 150.044  | 297.06      |
| 13 <sup>th</sup> Interest | 151.93      | 221.891  | 385.101     |
| 14 <sup>th</sup> Interest | 220.073     | 248.282  | 210.061     |
| 15 <sup>th</sup> Interest | 250.099     | 124.994  | 140.04      |
| 16 <sup>th</sup> Interest | 123.104     | 109.271  | 260.071     |
| 17 <sup>th</sup> Interest | 108.362     | 161.503  | 0.00017512  |
| 18 <sup>th</sup> Interest | 165.192     | 103.329  | 205.057     |
| 19 <sup>th</sup> Interest | 101.565     | 140.038  | 400.106     |
| 20 <sup>th</sup> Interest | 141.908     | 125.032  | 190.54      |

## Πείραμα 2

| Average Delay    |          |          |         |
|------------------|----------|----------|---------|
|                  | Flooding | Epidemic | PRoPHET |
| Random Walk      | 110.149  | 107.701  | 126.837 |
| Manhattan Grid   | 167.35   | 182.921  | 234.295 |
| Random Direction | 233.368  | 255.339  | 302.515 |

| Cache Hit Ratio  |          |          |         |
|------------------|----------|----------|---------|
|                  | Flooding | Epidemic | PRoPHET |
| Random Walk      | 0        | 0.1      | 0.1     |
| Manhattan Grid   | 0        | 0.05     | 0       |
| Random Direction | 0.05     | 0        | 0.1     |

| Random Direction          |             |          |             |
|---------------------------|-------------|----------|-------------|
|                           | Flooding    | Epidemic | PRoPHET     |
| 1 <sup>st</sup> Interest  | 339.917     | 298.721  | 489.666     |
| 2 <sup>nd</sup> Interest  | 190.01      | 189.501  | 250.087     |
| 3 <sup>rd</sup> Interest  | 250.106     | 161.603  | 390.098     |
| 4 <sup>th</sup> Interest  | 390.11      | 475.12   | 449.239     |
| 5 <sup>th</sup> Interest  | 213.989     | 217.038  | 371.059     |
| 6 <sup>th</sup> Interest  | 231.144     | 2.31E+02 | 1.60E+02    |
| 7 <sup>th</sup> Interest  | 270.78      | 2.73E+02 | 1.63E+02    |
| 8 <sup>th</sup> Interest  | 195.105     | 1.90E+02 | 2.60E+02    |
| 9 <sup>th</sup> Interest  | 159.69      | 159.112  | 189.542     |
| 10 <sup>th</sup> Interest | 0.000205294 | 2.95E+02 | 1.82E+02    |
| 11 <sup>th</sup> Interest | 335.464     | 230.054  | 0.000213021 |
| 12 <sup>th</sup> Interest | 184.777     | 181.064  | 379.316     |
| 13 <sup>th</sup> Interest | 178.991     | 190.364  | 173.921     |
| 14 <sup>th</sup> Interest | 193.126     | 1.86E+02 | 5.96E+02    |
| 15 <sup>th</sup> Interest | 187.466     | 178.945  | 278.168     |
| 16 <sup>th</sup> Interest | 175.644     | 3.20E+02 | 4.66E+02    |
| 17 <sup>th</sup> Interest | 324.562     | 2.95E+02 | 3.12E+02    |
| 18 <sup>th</sup> Interest | 293.98      | 254.979  | 691.146     |
| 19 <sup>th</sup> Interest | 250.843     | 3.00E+02 | 2.50E+02    |
| 20 <sup>th</sup> Interest | 301.681     | 4.81E+02 | 2.03E-04    |

| Random Walk               |          |            |             |
|---------------------------|----------|------------|-------------|
|                           | Flooding | Epidemic   | PRoPHET     |
| 1 <sup>st</sup> Interest  | 97.9196  | 124.359    | 263.089     |
| 2 <sup>nd</sup> Interest  | 163.272  | 139.668    | 83.1338     |
| 3 <sup>rd</sup> Interest  | 80.7875  | 79.6255    | 156.014     |
| 4 <sup>th</sup> Interest  | 65.7972  | 68.6039    | 290.074     |
| 5 <sup>th</sup> Interest  | 90.679   | 87.2157    | 139.993     |
| 6 <sup>th</sup> Interest  | 1.54E+02 | 1.56E+02   | 165.815     |
| 7 <sup>th</sup> Interest  | 1.01E+02 | 1.04E+02   | 66.8119     |
| 8 <sup>th</sup> Interest  | 9.10E+01 | 1.16E+02   | 87.4938     |
| 9 <sup>th</sup> Interest  | 82.9933  | 157.776    | 94.0998     |
| 10 <sup>th</sup> Interest | 1.69E+02 | 4.61E-04   | 168.574     |
| 11 <sup>th</sup> Interest | 69.117   | 66.4408    | 262.464     |
| 12 <sup>th</sup> Interest | 82.4208  | 0.00025155 | 74.1131     |
| 13 <sup>th</sup> Interest | 97.0459  | 148.622    | 69.2233     |
| 14 <sup>th</sup> Interest | 1.65E+02 | 9.86E+01   | 81.7212     |
| 15 <sup>th</sup> Interest | 115.856  | 160.372    | 202.655     |
| 16 <sup>th</sup> Interest | 2.18E+02 | 1.21E+02   | 63.6114     |
| 17 <sup>th</sup> Interest | 1.25E+02 | 2.15E+02   | 107.395     |
| 18 <sup>th</sup> Interest | 78.2635  | 128.116    | 160.484     |
| 19 <sup>th</sup> Interest | 5.91E+01 | 7.86E+01   | 0.000199806 |
| 20 <sup>th</sup> Interest | 9.77E+01 | 1.03E+02   | 0.000202158 |

| Manhattan Grid           |          |          |          |
|--------------------------|----------|----------|----------|
|                          | Flooding | Epidemic | PRoPHET  |
| 1 <sup>st</sup> Interest | 184.557  | 187.612  | 505.728  |
| 2 <sup>nd</sup> Interest | 325.171  | 325.08   | 290.077  |
| 3 <sup>rd</sup> Interest | 165.039  | 168.815  | 213.692  |
| 4 <sup>th</sup> Interest | 122.619  | 119.779  | 265.225  |
| 5th Interest             | 188.689  | 189.825  | 213.537  |
| 6th Interest             | 1.73E+02 | 1.75E+02 | 1.78E+02 |
| 7th Interest             | 1.38E+02 | 1.86E+02 | 1.87E+02 |
| 8th Interest             | 1.92E+02 | 1.41E+02 | 2.23E+02 |
| 9th Interest             | 180.045  | 177.815  | 200.056  |
| 10th Interest            | 1.85E+02 | 2.59E-04 | 2.55E+02 |
| 11th Interest            | 220.047  | 102.261  | 190.193  |
| 12th Interest            | 129.634  | 307.843  | 187.876  |
| 13th Interest            | 127.693  | 162.271  | 110.564  |
| 14th Interest            | 1.22E+02 | 2.90E+02 | 3.27E+02 |
| 15th Interest            | 128.091  | 189.963  | 425.402  |
| 16th Interest            | 1.37E+02 | 1.91E+02 | 1.47E+02 |
| 17th Interest            | 1.05E+02 | 2.00E+02 | 2.33E+02 |
| 18th Interest            | 120.025  | 236.568  | 168.893  |
| 19th Interest            | 2.20E+02 | 1.37E+02 | 1.56E+02 |
| 20th Interest            | 1.83E+02 | 1.70E+02 | 2.10E+02 |

### Πείραμα 3

| Average Delay |          |          |         |
|---------------|----------|----------|---------|
|               | Flooding | Epidemic | PRoPHET |
| Freshness 15  | 172.111  | 172.176  | 199.058 |
| Freshness 30  | 150.433  | 163.354  | 199.107 |
| Freshness 45  | 153.345  | 152.954  | 184.949 |
| Freshness 60  | 145.037  | 137.026  | 183.342 |

| Cache Hit Ratio |          |          |         |
|-----------------|----------|----------|---------|
|                 | Flooding | Epidemic | PRoPHET |
| Freshness 15    | 0        | 0        | 0.05    |
| Freshness 30    | 0.15     | 0.05     | 0.1     |
| Freshness 45    | 0.15     | 0.1      | 0.15    |
| Freshness 60    | 0.2      | 0.2      | 0.2     |

| Freshness 15 minutes      |          |          |             |
|---------------------------|----------|----------|-------------|
|                           | Flooding | Epidemic | PRoPHET     |
| 1 <sup>st</sup> Interest  | 171.335  | 169.751  | 361.257     |
| 2 <sup>nd</sup> Interest  | 205.096  | 207.747  | 180.24      |
| 3 <sup>rd</sup> Interest  | 136.122  | 103.827  | 357.93      |
| 4 <sup>th</sup> Interest  | 390.091  | 146.61   | 135.083     |
| 5 <sup>th</sup> Interest  | 134.993  | 276.785  | 140.043     |
| 6 <sup>th</sup> Interest  | 140.074  | 135.808  | 140.043     |
| 7 <sup>th</sup> Interest  | 100.818  | 140.04   | 196.956     |
| 8 <sup>th</sup> Interest  | 204.254  | 140.046  | 187.482     |
| 9 <sup>th</sup> Interest  | 220.053  | 163.316  | 182.377     |
| 10 <sup>th</sup> Interest | 180.043  | 216.968  | 240.298     |
| 11 <sup>th</sup> Interest | 150.035  | 182.728  | 135.032     |
| 12 <sup>th</sup> Interest | 223.983  | 149.467  | 290.066     |
| 13 <sup>th</sup> Interest | 251.169  | 225.639  | 227.209     |
| 14 <sup>th</sup> Interest | 123.929  | 246.411  | 140.043     |
| 15 <sup>th</sup> Interest | 106.091  | 124.991  | 0.000206762 |
| 16 <sup>th</sup> Interest | 161.441  | 108.175  | 145.047     |
| 17 <sup>th</sup> Interest | 107.569  | 181.901  | 222.775     |
| 18 <sup>th</sup> Interest | 169.927  | 145.038  | 389.215     |
| 19 <sup>th</sup> Interest | 95.2281  | 205.054  | 165.039     |
| 20 <sup>th</sup> Interest | 169.999  | 173.23   | 145.036     |

| Freshness 30              |             |             |             |
|---------------------------|-------------|-------------|-------------|
|                           | Flooding    | Epidemic    | PRoPHET     |
| 1 <sup>st</sup> Interest  | 173.251     | 166.151     | 223.659     |
| 2 <sup>nd</sup> Interest  | 205.772     | 202.606     | 0.000206123 |
| 3 <sup>rd</sup> Interest  | 134.307     | 109.903     | 164.56      |
| 4 <sup>th</sup> Interest  | 390.095     | 145.708     | 237.816     |
| 5 <sup>th</sup> Interest  | 134.991     | 276.92      | 408.343     |
| 6 <sup>th</sup> Interest  | 0.000166394 | 135.073     | 140.039     |
| 7 <sup>th</sup> Interest  | 140.074     | 139.996     | 141.835     |
| 8 <sup>th</sup> Interest  | 145.033     | 140.08      | 195.08      |
| 9 <sup>th</sup> Interest  | 160.035     | 165.055     | 370.104     |
| 10 <sup>th</sup> Interest | 220.049     | 217.573     | 217.44      |
| 11 <sup>th</sup> Interest | 184.777     | 181.558     | 154.626     |
| 12 <sup>th</sup> Interest | 145.299     | 150.051     | 333.136     |
| 13 <sup>th</sup> Interest | 0.000176061 | 225.062     | 188.383     |
| 14 <sup>th</sup> Interest | 224.797     | 250.113     | 140.047     |
| 15 <sup>th</sup> Interest | 247.187     | 124.996     | 145.071     |
| 16 <sup>th</sup> Interest | 127.906     | 0.000325015 | 318.836     |
| 17 <sup>th</sup> Interest | 0.000174616 | 105.029     | 293.067     |
| 18 <sup>th</sup> Interest | 104.686     | 180.05      | 153.681     |
| 19 <sup>th</sup> Interest | 162.566     | 145.051     | 156.432     |
| 20 <sup>th</sup> Interest | 107.848     | 206.116     | 0.000579817 |

| Freshness 45              |             |             |             |
|---------------------------|-------------|-------------|-------------|
|                           | Flooding    | Epidemic    | PRoPHET     |
| 1 <sup>st</sup> Interest  | 172.331     | 168.402     | 388.204     |
| 2 <sup>nd</sup> Interest  | 198.913     | 205.145     | 181.907     |
| 3 <sup>rd</sup> Interest  | 145.04      | 105.57      | 355.183     |
| 4 <sup>th</sup> Interest  | 390.057     | 145.969     | 140.049     |
| 5 <sup>th</sup> Interest  | 135.031     | 274.267     | 135.172     |
| 6 <sup>th</sup> Interest  | 140.087     | 135.049     | 145         |
| 7 <sup>th</sup> Interest  | 140.04      | 140.044     | 0.000241544 |
| 8 <sup>th</sup> Interest  | 165.039     | 141.172     | 195.093     |
| 9 <sup>th</sup> Interest  | 220.056     | 163.62      | 185.048     |
| 10 <sup>th</sup> Interest | 180.044     | 220.297     | 249.887     |
| 11 <sup>th</sup> Interest | 150.037     | 180.253     | 0.000487894 |
| 12 <sup>th</sup> Interest | 225.056     | 149.636     | 173.772     |
| 13 <sup>th</sup> Interest | 249.309     | 223.651     | 241.527     |
| 14 <sup>th</sup> Interest | 0.000168081 | 0.000207268 | 262.114     |
| 15 <sup>th</sup> Interest | 125.785     | 250.88      | 147.998     |
| 16 <sup>th</sup> Interest | 105.028     | 0.000270432 | 140.036     |
| 17 <sup>th</sup> Interest | 180.041     | 124.232     | 145.04      |
| 18 <sup>th</sup> Interest | 0.000206003 | 106.468     | 320.082     |
| 19 <sup>th</sup> Interest | 145.035     | 179.394     | 292.882     |
| 20 <sup>th</sup> Interest | 0.000182628 | 145.05      | 0.000223966 |

| Freshness 60              |             |             |             |
|---------------------------|-------------|-------------|-------------|
|                           | Flooding    | Epidemic    | PRoPHET     |
| 1 <sup>st</sup> Interest  | 170.822     | 169.794     | 220.486     |
| 2 <sup>nd</sup> Interest  | 194.852     | 208.132     | 158.817     |
| 3 <sup>rd</sup> Interest  | 150.083     | 106.953     | 241.271     |
| 4 <sup>th</sup> Interest  | 385.209     | 145.079     | 274.061     |
| 5 <sup>th</sup> Interest  | 139.989     | 274.655     | 136.043     |
| 6 <sup>th</sup> Interest  | 139.071     | 135.077     | 139.078     |
| 7 <sup>th</sup> Interest  | 140.032     | 139.051     | 145.045     |
| 8 <sup>th</sup> Interest  | 165.035     | 140.039     | 190.996     |
| 9 <sup>th</sup> Interest  | 0.000133254 | 163.746     | 375.096     |
| 10 <sup>th</sup> Interest | 220.007     | 0.000184272 | 0.000196545 |
| 11 <sup>th</sup> Interest | 179.902     | 221.683     | 0.000128186 |
| 12 <sup>th</sup> Interest | 0.0001898   | 181.077     | 295.076     |
| 13 <sup>th</sup> Interest | 150.218     | 0.000161997 | 370.094     |
| 14 <sup>th</sup> Interest | 0.000191129 | 147.42      | 225.062     |
| 15 <sup>th</sup> Interest | 220.065     | 227.688     | 0.000164475 |
| 16 <sup>th</sup> Interest | 250.095     | 0.00021157  | 143.141     |
| 17 <sup>th</sup> Interest | 124.987     | 245.109     | 141.176     |
| 18 <sup>th</sup> Interest | 0.000097996 | 124.992     | 319.836     |
| 19 <sup>th</sup> Interest | 109.856     | 0.000212335 | 0.000179138 |
| 20 <sup>th</sup> Interest | 160.538     | 110.035     | 291.57      |

## Πείραμα 4

### 1. Random Walk

| Average Delay           |          |          |         |
|-------------------------|----------|----------|---------|
|                         | Flooding | Epidemic | PRoPHET |
| <b>Lifetime 60 sec</b>  | 130.368  | 113.141  | 145.542 |
| <b>Lifetime 120 sec</b> | 118.513  | 126.131  | 153.766 |
| <b>Lifetime 180 sec</b> | 126.169  | 105.431  | 167.425 |
| <b>Lifetime 240 sec</b> | 98.627   | 113.301  | 145.621 |
| <b>Lifetime 300 sec</b> | 110.462  | 92.228   | 147.989 |

| Cache Hit Ratio         |          |          |         |
|-------------------------|----------|----------|---------|
|                         | Flooding | Epidemic | PRoPHET |
| <b>Lifetime 60 sec</b>  | 0        | 0.1      | 0.05    |
| <b>Lifetime 120 sec</b> | 0        | 0        | 0       |
| <b>Lifetime 180 sec</b> | 0        | 0.1      | 0       |
| <b>Lifetime 240 sec</b> | 0.1      | 0        | 0       |
| <b>Lifetime 300 sec</b> | 0.05     | 0.1      | 0.05    |

| Success Ratio           |          |          |          |
|-------------------------|----------|----------|----------|
|                         | Flooding | Epidemic | PRoPHET  |
| <b>Lifetime 60 sec</b>  | 0.37037  | 0.416667 | 0.333333 |
| <b>Lifetime 120 sec</b> | 0.714286 | 0.625    | 0.555556 |
| <b>Lifetime 180 sec</b> | 0.952381 | 1        | 0.740741 |
| <b>Lifetime 240 sec</b> | 1        | 1        | 0.833333 |
| <b>Lifetime 300 sec</b> | 1        | 1        | 0.952381 |

| Lifetime 60 sec                 |          |             |             |
|---------------------------------|----------|-------------|-------------|
|                                 | Flooding | Epidemic    | PRoPHET     |
| <b>1<sup>st</sup> Interest</b>  | 119.451  | 98.2497     | 261.652     |
| <b>2<sup>nd</sup> Interest</b>  | 158.75   | 164.678     | 81.366      |
| <b>3<sup>rd</sup> Interest</b>  | 128.088  | 79.8822     | 63.4839     |
| <b>4<sup>th</sup> Interest</b>  | 133.456  | 153.878     | 200.922     |
| <b>5<sup>th</sup> Interest</b>  | 146.722  | 156.708     | 143.694     |
| <b>6<sup>th</sup> Interest</b>  | 156.907  | 97.2215     | 171.207     |
| <b>7<sup>th</sup> Interest</b>  | 84.8442  | 92.8327     | 148.888     |
| <b>8<sup>th</sup> Interest</b>  | 170.294  | 83.3534     | 136.494     |
| <b>9<sup>th</sup> Interest</b>  | 149.784  | 171.759     | 241.423     |
| <b>10<sup>th</sup> Interest</b> | 193.48   | 148.299     | 216.8       |
| <b>11<sup>th</sup> Interest</b> | 70.4321  | 0.000268599 | 128.289     |
| <b>12<sup>th</sup> Interest</b> | 190.439  | 99.9244     | 135.035     |
| <b>13<sup>th</sup> Interest</b> | 88.3145  | 220.184     | 160.041     |
| <b>14<sup>th</sup> Interest</b> | 175.07   | 0.000289664 | 0.000206896 |
| <b>15<sup>th</sup> Interest</b> | 138.81   | 135.81      | 105.026     |
| <b>16<sup>th</sup> Interest</b> | 95.4937  | 144.166     | 161.26      |
| <b>17<sup>th</sup> Interest</b> | 65.537   | 74.9118     | 82.0068     |
| <b>18<sup>th</sup> Interest</b> | 107.979  | 80.3618     | 139.78      |
| <b>19<sup>th</sup> Interest</b> | 149.034  | 149.811     | 157.096     |
| <b>20<sup>th</sup> Interest</b> | 84.4932  | 110.811     | 0.0678802   |

| Lifetime 120 sec         |          |          |         |
|--------------------------|----------|----------|---------|
|                          | Flooding | Epidemic | PRoPHET |
| 1 <sup>st</sup> Interest | 119.145  | 122.733  | 264.68  |
| 2 <sup>nd</sup> Interest | 142.497  | 137.7    | 83.1673 |
| 3 <sup>rd</sup> Interest | 79.8908  | 81.5875  | 152.662 |
| 4 <sup>th</sup> Interest | 66.3029  | 155.083  | 188.527 |
| 5th Interest             | 87.9725  | 155.795  | 156.696 |
| 6th Interest             | 158.532  | 94.8048  | 192.422 |
| 7th Interest             | 96.2795  | 92.1605  | 61.0531 |
| 8th Interest             | 93.2411  | 86.0302  | 156.603 |
| 9th Interest             | 81.7499  | 172.745  | 190.093 |
| 10th Interest            | 173.209  | 147.307  | 185.014 |
| 11th Interest            | 146.877  | 100.397  | 215.102 |
| 12th Interest            | 99.3089  | 158.097  | 125.036 |
| 13th Interest            | 163.374  | 122.764  | 140.038 |
| 14th Interest            | 116.238  | 213.091  | 210.018 |
| 15th Interest            | 215.51   | 126.669  | 54.0557 |
| 16th Interest            | 125.932  | 138.409  | 150.038 |
| 17th Interest            | 139.142  | 95.348   | 118.31  |
| 18th Interest            | 93.3944  | 65.5793  | 114.357 |
| 19th Interest            | 69.5904  | 106.3    | 157.415 |
| 20th Interest            | 102.094  | 150.049  | 160.044 |

| Lifetime 180 sec         |          |             |         |
|--------------------------|----------|-------------|---------|
|                          | Flooding | Epidemic    | PRoPHET |
| 1 <sup>st</sup> Interest | 121.009  | 120.712     | 259.799 |
| 2 <sup>nd</sup> Interest | 140.274  | 142.867     | 82.6515 |
| 3 <sup>rd</sup> Interest | 83.3309  | 80.011      | 155.026 |
| 4 <sup>th</sup> Interest | 151.556  | 156.078     | 188.144 |
| 5th Interest             | 158.02   | 154.03      | 70.7824 |
| 6th Interest             | 100.02   | 100.041     | 171.056 |
| 7th Interest             | 90.0618  | 0.000275071 | 236.28  |
| 8th Interest             | 83.4122  | 87.5958     | 180.123 |
| 9th Interest             | 172.638  | 88.475      | 161.124 |
| 10th Interest            | 147.538  | 165.985     | 342.577 |
| 11th Interest            | 95.7018  | 67.5838     | 121.662 |
| 12th Interest            | 162.27   | 82.5377     | 139.635 |
| 13th Interest            | 119.957  | 0.000114398 | 89.1171 |
| 14th Interest            | 217.987  | 103.028     | 175.944 |
| 15th Interest            | 76.5845  | 157.217     | 160.417 |
| 16th Interest            | 80.5981  | 120.371     | 77.335  |
| 17th Interest            | 150.513  | 154.949     | 145.372 |
| 18th Interest            | 112.442  | 62.0785     | 149.54  |
| 19th Interest            | 110.027  | 125.566     | 246.288 |
| 20th Interest            | 149.473  | 139.518     | 195.652 |



| Lifetime 240 sec          |             |             |         |
|---------------------------|-------------|-------------|---------|
|                           | Flooding    | Epidemic    | PRoPHET |
| 1 <sup>st</sup> Interest  | 100.062     | 123.936     | 264.119 |
| 2 <sup>nd</sup> Interest  | 165.087     | 139.667     | 80.4706 |
| 3 <sup>rd</sup> Interest  | 78.663      | 81.2261     | 67.0719 |
| 4 <sup>th</sup> Interest  | 68.3515     | 67.8401     | 87.8093 |
| 5 <sup>th</sup> Interest  | 87.8242     | 86.389      | 259.075 |
| 6 <sup>th</sup> Interest  | 157.244     | 157.409     | 118.335 |
| 7 <sup>th</sup> Interest  | 96.9472     | 100.01      | 227.773 |
| 8 <sup>th</sup> Interest  | 84.9919     | 91.7473     | 147.314 |
| 9 <sup>th</sup> Interest  | 93.1272     | 80.9725     | 262.29  |
| 10 <sup>th</sup> Interest | 167.521     | 102.416     | 139.26  |
| 11 <sup>th</sup> Interest | 66.8161     | 69.1838     | 197.531 |
| 12 <sup>th</sup> Interest | 82.1382     | 64.5525     | 122.709 |
| 13 <sup>th</sup> Interest | 98.6236     | 83.9644     | 137.668 |
| 14 <sup>th</sup> Interest | 221.467     | 99.1754     | 96.4658 |
| 15 <sup>th</sup> Interest | 138.621     | 0.000241676 | 255.994 |
| 16 <sup>th</sup> Interest | 0.000157008 | 220.966     | 73.4276 |
| 17 <sup>th</sup> Interest | 76.8686     | 60.032      | 106.628 |
| 18 <sup>th</sup> Interest | 0.000168541 | 148.553     | 116.012 |
| 19 <sup>th</sup> Interest | 68.1679     | 66.5443     | 84.044  |
| 20 <sup>th</sup> Interest | 120.03      | 0.000344071 | 68.4322 |

| Lifetime 300 sec          |             |             |             |
|---------------------------|-------------|-------------|-------------|
|                           | Flooding    | Epidemic    | PRoPHET     |
| 1 <sup>st</sup> Interest  | 106.568     | 119.256     | 264.612     |
| 2 <sup>nd</sup> Interest  | 163.063     | 141.96      | 79.4072     |
| 3 <sup>rd</sup> Interest  | 80.0392     | 83.4963     | 155.026     |
| 4 <sup>th</sup> Interest  | 67.8398     | 65.0301     | 295.154     |
| 5 <sup>th</sup> Interest  | 87.8439     | 86.4571     | 135.036     |
| 6 <sup>th</sup> Interest  | 158.01      | 158.202     | 325.045     |
| 7 <sup>th</sup> Interest  | 102.049     | 99.9142     | 95.0646     |
| 8 <sup>th</sup> Interest  | 115.061     | 85.796      | 0.000168407 |
| 9 <sup>th</sup> Interest  | 0.000159584 | 87.9752     | 217.98      |
| 10 <sup>th</sup> Interest | 157.982     | 172.337     | 62.0489     |
| 11 <sup>th</sup> Interest | 71.0207     | 144.115     | 215.094     |
| 12 <sup>th</sup> Interest | 144.028     | 200.039     | 124.824     |
| 13 <sup>th</sup> Interest | 196.954     | 67.4946     | 138.452     |
| 14 <sup>th</sup> Interest | 123.119     | 112.541     | 98.3906     |
| 15 <sup>th</sup> Interest | 140.076     | 225.043     | 168.468     |
| 16 <sup>th</sup> Interest | 84.9868     | 0.000262102 | 150.036     |
| 17 <sup>th</sup> Interest | 66.5526     | 118.774     | 87          |
| 18 <sup>th</sup> Interest | 188.47      | 136.277     | 148.023     |
| 19 <sup>th</sup> Interest | 95.061      | 93.4493     | 83.0225     |
| 20 <sup>th</sup> Interest | 60.5288     | 67.8802     | 117.114     |

## 2.Manhattan

| Average Delay    |          |          |         |
|------------------|----------|----------|---------|
|                  | Flooding | Epidemic | PRoPHET |
| Lifetime 60 sec  | 183.847  | 199.905  | 234.247 |
| Lifetime 120 sec | 188.419  | 182.725  | 229.187 |
| Lifetime 180 sec | 174.195  | 174.257  | 234.183 |
| Lifetime 240 sec | 174.139  | 174.178  | 216.046 |
| Lifetime 300 sec | 183.255  | 174.205  | 190.459 |
| Lifetime 360 sec | 167.459  | 174.287  | 232.06  |

| Cache Hit Ratio  |          |          |         |
|------------------|----------|----------|---------|
|                  | Flooding | Epidemic | PRoPHET |
| Lifetime 60 sec  | 0        | 0        | 0.05    |
| Lifetime 120 sec | 0        | 0        | 0.05    |
| Lifetime 180 sec | 0.1      | 0.05     | 0.05    |
| Lifetime 240 sec | 0.05     | 0.05     | 0.15    |
| Lifetime 300 sec | 0        | 0.05     | 0.15    |
| Lifetime 360 sec | 0.05     | 0.05     | 0.05    |

| Success Ratio    |          |          |          |
|------------------|----------|----------|----------|
|                  | Flooding | Epidemic | PRoPHET  |
| Lifetime 60 sec  | 0.273973 | 0.25641  | 0.222222 |
| Lifetime 120 sec | 0.487805 | 0.487805 | 0.408163 |
| Lifetime 180 sec | 0.625    | 0.689655 | 0.512821 |
| Lifetime 240 sec | 0.869565 | 0.869565 | 0.714286 |
| Lifetime 300 sec | 0.952381 | 0.952381 | 0.952381 |
| Lifetime 360 sec | 1        | 1        | 0.909091 |

| Lifetime 60 sec           |             |          |            |
|---------------------------|-------------|----------|------------|
|                           | Flooding    | Epidemic | PRoPHET    |
| 1 <sup>st</sup> Interest  | 190.908     | 183.171  | 509.624    |
| 2 <sup>nd</sup> Interest  | 325.335     | 323.424  | 289.084    |
| 3 <sup>rd</sup> Interest  | 170.183     | 171.587  | 188.633    |
| 4 <sup>th</sup> Interest  | 174.585     | 117.637  | 269.685    |
| 5 <sup>th</sup> Interest  | 135.519     | 191.804  | 231.886    |
| 6 <sup>th</sup> Interest  | 174.938     | 172.129  | 0.00034966 |
| 7 <sup>th</sup> Interest  | 184.095     | 122.112  | 240.077    |
| 8 <sup>th</sup> Interest  | 141.723     | 205.808  | 203.665    |
| 9 <sup>th</sup> Interest  | 240.904     | 240.19   | 166.447    |
| 10 <sup>th</sup> Interest | 194.267     | 195.044  | 137.311    |
| 11 <sup>th</sup> Interest | 151         | 175.012  | 304.433    |
| 12 <sup>th</sup> Interest | 0.000212953 | 139.41   | 175.731    |
| 13 <sup>th</sup> Interest | 163.168     | 289.4    | 196.924    |
| 14 <sup>th</sup> Interest | 290.303     | 189.787  | 194.379    |
| 15 <sup>th</sup> Interest | 189.729     | 196.592  | 241.635    |
| 16 <sup>th</sup> Interest | 192.087     | 194.213  | 461.446    |
| 17 <sup>th</sup> Interest | 200.872     | 238.381  | 187.157    |

|                      |         |         |         |
|----------------------|---------|---------|---------|
| <b>18th Interest</b> | 236.913 | 242.432 | 224.706 |
| <b>19th Interest</b> | 135.446 | 223.669 | 256.08  |
| <b>20th Interest</b> | 184.978 | 186.305 | 206.061 |

| <b>Lifetime 120 sec</b>        |                 |                 |                |
|--------------------------------|-----------------|-----------------|----------------|
|                                | <b>Flooding</b> | <b>Epidemic</b> | <b>PRoPHET</b> |
| <b>1<sup>st</sup> Interest</b> | 182.279         | 184.98          | 513.233        |
| <b>2<sup>nd</sup> Interest</b> | 325.041         | 325.098         | 290.119        |
| <b>3<sup>rd</sup> Interest</b> | 170.086         | 170.058         | 203.843        |
| <b>4<sup>th</sup> Interest</b> | 120.072         | 120.076         | 271.31         |
| <b>5th Interest</b>            | 190.093         | 187.969         | 215.049        |
| <b>6th Interest</b>            | 172.997         | 174.968         | 241.918        |
| <b>7th Interest</b>            | 122.273         | 137.92          | 202.191        |
| <b>8th Interest</b>            | 208.77          | 189.962         | 165.04         |
| <b>9th Interest</b>            | 175.048         | 176.404         | 200.88         |
| <b>10th Interest</b>           | 105.027         | 188.729         | 0.000262975    |
| <b>11th Interest</b>           | 303.665         | 220.052         | 229.277        |
| <b>12th Interest</b>           | 161.486         | 165.078         | 190.052        |
| <b>13th Interest</b>           | 292.65          | 290.234         | 191.118        |
| <b>14th Interest</b>           | 187.393         | 189.84          | 198.179        |
| <b>15th Interest</b>           | 160.64          | 151.599         | 245.865        |
| <b>16th Interest</b>           | 104.068         | 138.479         | 203.031        |
| <b>17th Interest</b>           | 126.889         | 178.721         | 361.665        |
| <b>18th Interest</b>           | 238.582         | 158.25          | 230.065        |
| <b>19th Interest</b>           | 213.734         | 213.165         | 170.493        |
| <b>20th Interest</b>           | 207.612         | 92.9445         | 260.427        |

| <b>Lifetime 180 sec</b>        |                 |                 |                |
|--------------------------------|-----------------|-----------------|----------------|
|                                | <b>Flooding</b> | <b>Epidemic</b> | <b>PRoPHET</b> |
| <b>1<sup>st</sup> Interest</b> | 183.346         | 185.369         | 507.245        |
| <b>2<sup>nd</sup> Interest</b> | 325.591         | 324.188         | 222.088        |
| <b>3<sup>rd</sup> Interest</b> | 165.089         | 166.908         | 275.788        |
| <b>4<sup>th</sup> Interest</b> | 123.963         | 123.626         | 250.756        |
| <b>5th Interest</b>            | 186.871         | 186.51          | 232.832        |
| <b>6th Interest</b>            | 174.251         | 175.012         | 176.194        |
| <b>7th Interest</b>            | 0.000233092     | 197.577         | 187.07         |
| <b>8th Interest</b>            | 122.068         | 132.341         | 222.195        |
| <b>9th Interest</b>            | 206.105         | 174.76          | 200.122        |
| <b>10th Interest</b>           | 181.018         | 190.059         | 178.09         |
| <b>11th Interest</b>           | 185.073         | 220.073         | 264.742        |
| <b>12th Interest</b>           | 220.078         | 160.527         | 264.725        |
| <b>13th Interest</b>           | 165.108         | 293.876         | 206.452        |
| <b>14th Interest</b>           | 289.176         | 186.257         | 367.724        |
| <b>15th Interest</b>           | 0.000177754     | 95.9869         | 253.024        |
| <b>16th Interest</b>           | 190.906         | 171.017         | 188.138        |
| <b>17th Interest</b>           | 190.115         | 203.099         | 221.273        |
| <b>18th Interest</b>           | 254.13          | 0.000162405     | 0.000232888    |
| <b>19th Interest</b>           | 184.987         | 159.075         | 259.488        |
| <b>20th Interest</b>           | 136.041         | 138.905         | 205.727        |

| Lifetime 240 sec         |            |             |             |
|--------------------------|------------|-------------|-------------|
|                          | Flooding   | Epidemic    | PRoPHET     |
| 1 <sup>st</sup> Interest | 182.736    | 183.55      | 510.458     |
| 2 <sup>nd</sup> Interest | 325.113    | 323.393     | 289.423     |
| 3 <sup>rd</sup> Interest | 167.245    | 221.069     | 211.04      |
| 4 <sup>th</sup> Interest | 122.846    | 120.81      | 152.616     |
| 5th Interest             | 188.805    | 138.981     | 185.048     |
| 6th Interest             | 176.302    | 172.916     | 318.989     |
| 7th Interest             | 132.21     | 135.995     | 0.000176599 |
| 8th Interest             | 192.891    | 191.239     | 188.669     |
| 9th Interest             | 179.059    | 177.665     | 220.06      |
| 10th Interest            | 112.754    | 125.231     | 165.045     |
| 11th Interest            | 296.736    | 283.946     | 0.000826589 |
| 12th Interest            | 159.404    | 162.779     | 288.546     |
| 13th Interest            | 291.457    | 0.000255947 | 349.262     |
| 14th Interest            | 189.975    | 290.803     | 232.404     |
| 15th Interest            | 0.00021124 | 189.348     | 244.131     |
| 16th Interest            | 94.5438    | 192.434     | 416.353     |
| 17th Interest            | 195.597    | 155.263     | 148.753     |
| 18th Interest            | 178.177    | 121.442     | 230.062     |
| 19th Interest            | 161.897    | 158.719     | 0.000178858 |
| 20th Interest            | 135.041    | 138.002     | 170.081     |

| Lifetime 300 sec         |          |             |             |
|--------------------------|----------|-------------|-------------|
|                          | Flooding | Epidemic    | PRoPHET     |
| 1 <sup>st</sup> Interest | 185.492  | 184.104     | 509.103     |
| 2 <sup>nd</sup> Interest | 323.425  | 325.078     | 286.566     |
| 3 <sup>rd</sup> Interest | 166.718  | 168.036     | 206.897     |
| 4 <sup>th</sup> Interest | 121.802  | 122.046     | 157.208     |
| 5th Interest             | 188.333  | 188.001     | 186.101     |
| 6th Interest             | 175.012  | 0.000227389 | 0.000184212 |
| 7th Interest             | 135.084  | 175.065     | 286.7       |
| 8th Interest             | 195.024  | 194.331     | 157.381     |
| 9th Interest             | 175.057  | 132.821     | 283.077     |
| 10th Interest            | 185.058  | 177.695     | 248.928     |
| 11th Interest            | 225.064  | 187.419     | 133.189     |
| 12th Interest            | 160.045  | 222.726     | 263.658     |
| 13th Interest            | 219.385  | 161.396     | 189.6       |
| 14th Interest            | 127.048  | 290.194     | 0.000212494 |
| 15th Interest            | 136.122  | 189.975     | 199.216     |
| 16th Interest            | 191.189  | 150.929     | 247.82      |
| 17th Interest            | 200.516  | 197.188     | 0.00125271  |
| 18th Interest            | 235.082  | 122.173     | 0.00080891  |
| 19th Interest            | 139.141  | 159.901     | 203.12      |
| 20th Interest            | 180.517  | 135.04      | 250.636     |

| Lifetime 360 sec          |             |             |             |
|---------------------------|-------------|-------------|-------------|
|                           | Flooding    | Epidemic    | PRoPHET     |
| 1 <sup>st</sup> Interest  | 183.843     | 188.063     | 506.977     |
| 2 <sup>nd</sup> Interest  | 325.072     | 324.075     | 319.082     |
| 3 <sup>rd</sup> Interest  | 168.912     | 167.206     | 175.004     |
| 4 <sup>th</sup> Interest  | 121.155     | 120.801     | 256.091     |
| 5 <sup>th</sup> Interest  | 187.661     | 190.843     | 230.684     |
| 6 <sup>th</sup> Interest  | 172.419     | 171.298     | 178.434     |
| 7 <sup>th</sup> Interest  | 138.004     | 122.428     | 188.51      |
| 8 <sup>th</sup> Interest  | 0.000210007 | 207.644     | 220.087     |
| 9 <sup>th</sup> Interest  | 192.066     | 180.044     | 161.573     |
| 10 <sup>th</sup> Interest | 180.041     | 185.048     | 295.135     |
| 11 <sup>th</sup> Interest | 100.022     | 220.064     | 281.567     |
| 12 <sup>th</sup> Interest | 305.067     | 164.088     | 223.44      |
| 13 <sup>th</sup> Interest | 165.037     | 289.119     | 180.215     |
| 14 <sup>th</sup> Interest | 215.052     | 0.000190025 | 295.096     |
| 15 <sup>th</sup> Interest | 129.711     | 191.013     | 297.597     |
| 16 <sup>th</sup> Interest | 135.347     | 190.529     | 190.505     |
| 17 <sup>th</sup> Interest | 149.991     | 72.8528     | 221.089     |
| 18 <sup>th</sup> Interest | 140.112     | 204.355     | 215.569     |
| 19 <sup>th</sup> Interest | 176.454     | 158.413     | 204.562     |
| 20 <sup>th</sup> Interest | 163.237     | 137.876     | 0.000270199 |

### 3.Random Direction

| Average Delay    |          |          |         |
|------------------|----------|----------|---------|
|                  | Flooding | Epidemic | PRoPHET |
| Lifetime 60 sec  | 246.066  | 262.688  | 329.193 |
| Lifetime 120 sec | 250.693  | 248.171  | 320.348 |
| Lifetime 180 sec | 243.711  | 243.693  | 294.313 |
| Lifetime 240 sec | 250.248  | 246.421  | 329.409 |
| Lifetime 300 sec | 262.804  | 243.594  | 326.123 |
| Lifetime 360 sec | 230.208  | 250.439  | 342.28  |
| Lifetime 420 sec | 243.494  | 243.715  | 342.73  |

| Cache Hit Ratio  |          |          |         |
|------------------|----------|----------|---------|
|                  | Flooding | Epidemic | PRoPHET |
| Lifetime 60 sec  | 0        | 0.05     | 0       |
| Lifetime 120 sec | 0.05     | 0.05     | 0       |
| Lifetime 180 sec | 0.1      | 0.05     | 0.1     |
| Lifetime 240 sec | 0.05     | 0        | 0       |
| Lifetime 300 sec | 0        | 0.1      | 0.05    |
| Lifetime 360 sec | 0.1      | 0        | 0       |
| Lifetime 420 sec | 0.1      | 0.05     | 0.05    |

| Lifetime 60 sec           |          |             |         |
|---------------------------|----------|-------------|---------|
|                           | Flooding | Epidemic    | PRoPHET |
| 1 <sup>st</sup> Interest  | 114.692  | 116.747     | 450.965 |
| 2 <sup>nd</sup> Interest  | 277.942  | 275.091     | 223.553 |
| 3 <sup>rd</sup> Interest  | 167.229  | 166.162     | 291.038 |
| 4 <sup>th</sup> Interest  | 400.063  | 402.239     | 388.471 |
| 5 <sup>th</sup> Interest  | 390.157  | 386.479     | 281.174 |
| 6 <sup>th</sup> Interest  | 280.043  | 284.152     | 135.033 |
| 7 <sup>th</sup> Interest  | 170.082  | 134.993     | 411.643 |
| 8 <sup>th</sup> Interest  | 140.199  | 232.515     | 437.414 |
| 9 <sup>th</sup> Interest  | 148.536  | 390.493     | 272.437 |
| 10 <sup>th</sup> Interest | 293.382  | 424.619     | 529.266 |
| 11 <sup>th</sup> Interest | 433.196  | 240.051     | 171.123 |
| 12 <sup>th</sup> Interest | 151.922  | 255.568     | 283.456 |
| 13 <sup>th</sup> Interest | 340.034  | 279.979     | 234.973 |
| 14 <sup>th</sup> Interest | 278.462  | 279.59      | 495.359 |
| 15 <sup>th</sup> Interest | 282.34   | 239.697     | 289.79  |
| 16 <sup>th</sup> Interest | 236.54   | 490.294     | 365.875 |
| 17 <sup>th</sup> Interest | 270.408  | 0.000166317 | 304.732 |
| 18 <sup>th</sup> Interest | 222.971  | 280.405     | 324.686 |
| 19 <sup>th</sup> Interest | 273.715  | 43.0927     | 520.178 |
| 20 <sup>th</sup> Interest | 49.4246  | 331.608     | 172.704 |

| Lifetime 120 sec          |             |             |         |
|---------------------------|-------------|-------------|---------|
|                           | Flooding    | Epidemic    | PRoPHET |
| 1 <sup>st</sup> Interest  | 90.1539     | 82.7246     | 393.072 |
| 2 <sup>nd</sup> Interest  | 310.083     | 311.312     | 279.114 |
| 3 <sup>rd</sup> Interest  | 163.79      | 164.717     | 291.06  |
| 4 <sup>th</sup> Interest  | 401.409     | 400.991     | 386.59  |
| 5 <sup>th</sup> Interest  | 392.352     | 0.000142606 | 282.628 |
| 6 <sup>th</sup> Interest  | 280.074     | 389.044     | 170.127 |
| 7 <sup>th</sup> Interest  | 0.000285196 | 281.467     | 349.358 |
| 8 <sup>th</sup> Interest  | 307.895     | 135.09      | 463.91  |
| 9 <sup>th</sup> Interest  | 150.118     | 233.115     | 271.045 |
| 10 <sup>th</sup> Interest | 294.859     | 385.811     | 424.946 |
| 11 <sup>th</sup> Interest | 428.878     | 429.005     | 272.666 |
| 12 <sup>th</sup> Interest | 154.953     | 202.391     | 287.745 |
| 13 <sup>th</sup> Interest | 340.348     | 293.597     | 236.283 |
| 14 <sup>th</sup> Interest | 275.463     | 275.901     | 547.806 |
| 15 <sup>th</sup> Interest | 284.694     | 283.335     | 214.935 |
| 16 <sup>th</sup> Interest | 237.343     | 240.128     | 135.01  |
| 17 <sup>th</sup> Interest | 267.84      | 261.764     | 296.38  |
| 18 <sup>th</sup> Interest | 225.044     | 228.73      | 263.419 |
| 19 <sup>th</sup> Interest | 270.24      | 271.376     | 322.108 |
| 20 <sup>th</sup> Interest | 138.344     | 92.932      | 518.771 |

| Lifetime 180 sec          |             |             |             |
|---------------------------|-------------|-------------|-------------|
|                           | Flooding    | Epidemic    | PRoPHET     |
| 1 <sup>st</sup> Interest  | 115.527     | 117.885     | 392.501     |
| 2 <sup>nd</sup> Interest  | 278.566     | 276.221     | 167.109     |
| 3 <sup>rd</sup> Interest  | 165.039     | 169.496     | 401.203     |
| 4 <sup>th</sup> Interest  | 403.431     | 399.244     | 387.232     |
| 5 <sup>th</sup> Interest  | 387.222     | 391.012     | 284.977     |
| 6 <sup>th</sup> Interest  | 280.907     | 279.834     | 167.312     |
| 7 <sup>th</sup> Interest  | 313.423     | 169.481     | 200.037     |
| 8 <sup>th</sup> Interest  | 145.488     | 141.442     | 385.075     |
| 9 <sup>th</sup> Interest  | 294.933     | 145.143     | 430.082     |
| 10 <sup>th</sup> Interest | 430.08      | 0.000164986 | 0.000176709 |
| 11 <sup>th</sup> Interest | 204.292     | 297.628     | 285.905     |
| 12 <sup>th</sup> Interest | 289.961     | 431.011     | 482.16      |
| 13 <sup>th</sup> Interest | 279.965     | 151.41      | 287.157     |
| 14 <sup>th</sup> Interest | 280.138     | 450.312     | 336.326     |
| 15 <sup>th</sup> Interest | 240.059     | 166.825     | 448.691     |
| 16 <sup>th</sup> Interest | 265.144     | 286.205     | 0.000121472 |
| 17 <sup>th</sup> Interest | 225.056     | 234.993     | 281.369     |
| 18 <sup>th</sup> Interest | 0.000217393 | 266.112     | 318.688     |
| 19 <sup>th</sup> Interest | 275.008     | 226.333     | 307.367     |
| 20 <sup>th</sup> Interest | 0.000208303 | 273.291     | 323.092     |

| Lifetime 240 sec          |           |          |         |
|---------------------------|-----------|----------|---------|
|                           | Flooding  | Epidemic | PRoPHET |
| 1 <sup>st</sup> Interest  | 85.9697   | 87.5496  | 512.789 |
| 2 <sup>nd</sup> Interest  | 305.496   | 310.07   | 169.9   |
| 3 <sup>rd</sup> Interest  | 170.011   | 165.03   | 285.018 |
| 4 <sup>th</sup> Interest  | 399.342   | 404.285  | 386.033 |
| 5 <sup>th</sup> Interest  | 390.176   | 384.838  | 285.027 |
| 6 <sup>th</sup> Interest  | 280.019   | 284.977  | 170.045 |
| 7 <sup>th</sup> Interest  | 310.413   | 132.589  | 352.203 |
| 8 <sup>th</sup> Interest  | 0.0006416 | 175.006  | 230.061 |
| 9 <sup>th</sup> Interest  | 145.123   | 147.735  | 718.505 |
| 10 <sup>th</sup> Interest | 300.085   | 299.95   | 479.606 |
| 11 <sup>th</sup> Interest | 430.222   | 430.136  | 285.703 |
| 12 <sup>th</sup> Interest | 200       | 150.009  | 238.499 |
| 13 <sup>th</sup> Interest | 294.225   | 341.163  | 490.119 |
| 14 <sup>th</sup> Interest | 271.818   | 276.314  | 274.976 |
| 15 <sup>th</sup> Interest | 288.127   | 283.009  | 135.018 |
| 16 <sup>th</sup> Interest | 235.312   | 239.886  | 303.079 |
| 17 <sup>th</sup> Interest | 265.152   | 266.213  | 338.361 |
| 18 <sup>th</sup> Interest | 228.437   | 224.054  | 382.14  |
| 19 <sup>th</sup> Interest | 271.507   | 274.981  | 376.044 |
| 20 <sup>th</sup> Interest | 133.548   | 50.6526  | 175.071 |



| Lifetime 300 sec          |          |             |             |
|---------------------------|----------|-------------|-------------|
|                           | Flooding | Epidemic    | PRoPHET     |
| 1 <sup>st</sup> Interest  | 116.412  | 117.169     | 392.193     |
| 2 <sup>nd</sup> Interest  | 278.016  | 275.458     | 165.675     |
| 3 <sup>rd</sup> Interest  | 165.343  | 165.388     | 0.000187718 |
| 4 <sup>th</sup> Interest  | 401.88   | 402.266     | 404.183     |
| 5 <sup>th</sup> Interest  | 390.116  | 386.021     | 387.627     |
| 6 <sup>th</sup> Interest  | 284.322  | 0.000141013 | 282.482     |
| 7 <sup>th</sup> Interest  | 165.823  | 286.547     | 167.135     |
| 8 <sup>th</sup> Interest  | 140.084  | 170.036     | 378.07      |
| 9 <sup>th</sup> Interest  | 147.403  | 195.433     | 210.037     |
| 10 <sup>th</sup> Interest | 296.688  | 0.000134778 | 427.051     |
| 11 <sup>th</sup> Interest | 430.132  | 387.253     | 287.643     |
| 12 <sup>th</sup> Interest | 201.749  | 427.905     | 480.533     |
| 13 <sup>th</sup> Interest | 293.411  | 154.026     | 287.1       |
| 14 <sup>th</sup> Interest | 275.869  | 340.016     | 237.256     |
| 15 <sup>th</sup> Interest | 282.672  | 276.724     | 491.003     |
| 16 <sup>th</sup> Interest | 281.722  | 284.473     | 345.093     |
| 17 <sup>th</sup> Interest | 448.112  | 238.25      | 367.784     |
| 18 <sup>th</sup> Interest | 272.775  | 264.649     | 336.449     |
| 19 <sup>th</sup> Interest | 132.365  | 225.935     | 604.237     |
| 20 <sup>th</sup> Interest | 251.216  | 274.354     | 270.919     |

| Lifetime 360 sec          |             |          |         |
|---------------------------|-------------|----------|---------|
|                           | Flooding    | Epidemic | PRoPHET |
| 1 <sup>st</sup> Interest  | 86.1876     | 119.227  | 509.393 |
| 2 <sup>nd</sup> Interest  | 312.951     | 279.073  | 165.516 |
| 3 <sup>rd</sup> Interest  | 165.039     | 164.865  | 285.023 |
| 4 <sup>th</sup> Interest  | 400.097     | 400.271  | 389.755 |
| 5 <sup>th</sup> Interest  | 390.095     | 387.399  | 281.567 |
| 6 <sup>th</sup> Interest  | 282.294     | 283.8    | 309.13  |
| 7 <sup>th</sup> Interest  | 167.818     | 133.558  | 240.07  |
| 8 <sup>th</sup> Interest  | 140.075     | 175.441  | 205.05  |
| 9 <sup>th</sup> Interest  | 150.075     | 148.238  | 506.928 |
| 10 <sup>th</sup> Interest | 0.00016996  | 297.059  | 524.208 |
| 11 <sup>th</sup> Interest | 294.988     | 429.03   | 169.742 |
| 12 <sup>th</sup> Interest | 429.101     | 202.183  | 283.528 |
| 13 <sup>th</sup> Interest | 202.267     | 294.123  | 235.057 |
| 14 <sup>th</sup> Interest | 0.000225839 | 270.89   | 492.352 |
| 15 <sup>th</sup> Interest | 292.847     | 286.331  | 297.748 |
| 16 <sup>th</sup> Interest | 276.28      | 238.026  | 405.961 |
| 17 <sup>th</sup> Interest | 283.854     | 269.005  | 261.617 |
| 18 <sup>th</sup> Interest | 240.056     | 222.414  | 322.756 |
| 19 <sup>th</sup> Interest | 263.583     | 272.338  | 636.015 |
| 20 <sup>th</sup> Interest | 226.574     | 135.54   | 324.203 |

| Lifetime 420 sec          |             |             |             |
|---------------------------|-------------|-------------|-------------|
|                           | Flooding    | Epidemic    | PRoPHET     |
| 1 <sup>st</sup> Interest  | 114.862     | 120.241     | 391.706     |
| 2 <sup>nd</sup> Interest  | 277.073     | 277.298     | 281.741     |
| 3 <sup>rd</sup> Interest  | 167.291     | 165.031     | 288.408     |
| 4 <sup>th</sup> Interest  | 400.701     | 401.785     | 387.386     |
| 5 <sup>th</sup> Interest  | 390.124     | 390.64      | 650.396     |
| 6 <sup>th</sup> Interest  | 282.213     | 280.065     | 385.028     |
| 7 <sup>th</sup> Interest  | 130.503     | 170.155     | 429.849     |
| 8 <sup>th</sup> Interest  | 236.2       | 140.789     | 285.51      |
| 9 <sup>th</sup> Interest  | 388.46      | 146.002     | 490.751     |
| 10 <sup>th</sup> Interest | 427.346     | 296.513     | 279.032     |
| 11 <sup>th</sup> Interest | 202.826     | 431.287     | 236.907     |
| 12 <sup>th</sup> Interest | 295.022     | 152.596     | 490.127     |
| 13 <sup>th</sup> Interest | 270.905     | 0.000121849 | 368.632     |
| 14 <sup>th</sup> Interest | 0.000144886 | 339.834     | 334.819     |
| 15 <sup>th</sup> Interest | 0.00010949  | 276.047     | 347.026     |
| 16 <sup>th</sup> Interest | 286.235     | 286.765     | 379.883     |
| 17 <sup>th</sup> Interest | 238.058     | 235.052     | 0.000293947 |
| 18 <sup>th</sup> Interest | 265.774     | 266.581     | 380.089     |
| 19 <sup>th</sup> Interest | 224.394     | 225.897     | 115.028     |
| 20 <sup>th</sup> Interest | 271.915     | 271.744     | 332.299     |

| Success Ratio    |          |          |          |
|------------------|----------|----------|----------|
|                  | Flooding | Epidemic | PRoPHET  |
| Lifetime 60 sec  | 0.224719 | 0.206186 | 0.169492 |
| Lifetime 120 sec | 0.377358 | 0.377358 | 0.3125   |
| Lifetime 180 sec | 0.540541 | 0.555556 | 0.465116 |
| Lifetime 240 sec | 0.625    | 0.645161 | 0.540541 |
| Lifetime 300 sec | 0.833333 | 0.8      | 0.625    |
| Lifetime 360 sec | 0.869565 | 0.869565 | 0.740741 |
| Lifetime 420 sec | 0.952381 | 0.952381 | 0.833333 |

## Πείραμα 5

| Average Delay |          |
|---------------|----------|
| 0% Loss       | 111.691  |
| 10% Loss      | 148.191  |
| 20% Loss      | 147.701  |
| 30% Loss      | 256.455  |
| 40% Loss      | 633.927  |
| 50% Loss      | 1507.601 |

| Cach Hit Ratio |      |
|----------------|------|
| 0% Loss        | 0.05 |
| 10% Loss       | 0    |
| 20% Loss       | 0.05 |
| 30% Loss       | 0.1  |
| 40% Loss       | 0.05 |
| 50% Loss       | 0    |

|                           | 0.00%       | 10.00%  | 20.00%      | 30.00%      | 40.00%      | 50.00%  |
|---------------------------|-------------|---------|-------------|-------------|-------------|---------|
| 1 <sup>st</sup> Interest  | 88.6794     | 98.355  | 89.5158     | 120.279     | 271.468     | 471.696 |
| 2 <sup>nd</sup> Interest  | 81.9606     | 106.146 | 112.932     | 74.124      | 186.513     | 433.407 |
| 3 <sup>rd</sup> Interest  | 188.705     | 163.728 | 144.821     | 164.165     | 378.955     | 567.101 |
| 4 <sup>th</sup> Interest  | 105.025     | 109.967 | 164.387     | 151.483     | 260.576     | 976.344 |
| 5 <sup>th</sup> Interest  | 55.8739     | 173.837 | 112.72      | 133.92      | 276.889     | 863.441 |
| 6 <sup>th</sup> Interest  | 64.1988     | 187.281 | 209.181     | 177.925     | 784.905     | 1365.5  |
| 7 <sup>th</sup> Interest  | 89.9813     | 86.9953 | 230.024     | 241.827     | 253.335     | 1147.66 |
| 8 <sup>th</sup> Interest  | 135.036     | 171.475 | 102.459     | 311.757     | 918.855     | 1102.54 |
| 9 <sup>th</sup> Interest  | 89.9907     | 132.414 | 132.6       | 366.671     | 468.465     | 350.233 |
| 10 <sup>th</sup> Interest | 0.000189778 | 79.2309 | 261.632     | 498.701     | 0.000233684 | 1000.67 |
| 11 <sup>th</sup> Interest | 77.3531     | 179.484 | 183.756     | 463.356     | 801.507     | 977.108 |
| 12 <sup>th</sup> Interest | 121.733     | 97.3561 | 170.75      | 410.335     | 208.612     | 384.048 |
| 13 <sup>th</sup> Interest | 120.03      | 162.583 | 151.917     | 375.208     | 521.448     | 553.388 |
| 14 <sup>th</sup> Interest | 79.9812     | 165.143 | 0.000202169 | 312.976     | 725.081     | 1295.37 |
| 15 <sup>th</sup> Interest | 200.05      | 96.8413 | 142.658     | 0.000202821 | 232.064     | 876.457 |
| 16 <sup>th</sup> Interest | 109.188     | 211.204 | 110.855     | 288.885     | 627.451     | 3589.23 |
| 17 <sup>th</sup> Interest | 135.06      | 225.297 | 90.0244     | 462.125     | 624.986     | 4099.12 |
| 18 <sup>th</sup> Interest | 165.862     | 132.42  | 162.682     | 300.032     | 266.541     | 869.06  |
| 19 <sup>th</sup> Interest | 92.2494     | 214.034 | 140.345     | 275.343     | 3545.64     | 2329.6  |
| 20 <sup>th</sup> Interest | 232.882     | 170.041 | 240.782     | 0.000158321 | 1325.27     | 6900.07 |