



ΔΗΜΟΚΡΙΤΕΙΟ ΠΑΝΕΠΙΣΤΗΜΕΙΟ ΘΡΑΚΗΣ

ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ
ΥΠΟΛΟΓΙΣΤΩΝ

ΤΟΜΕΑΣ ΛΟΓΙΣΜΙΚΟΥ ΚΑΙ ΑΝΑΠΤΥΞΗΣ ΕΦΑΡΜΟΓΩΝ

Εφαρμογή του Διαδικτύου των Πραγμάτων σε Δίκτυα Ανεκτικά στην Καθυστερήση

Διπλωματική Εργασία

Μαρτίνης Μιχαήλ

ΑΜ: 57460

Επιβλέπων: Τσαουσίδης Βασίλειος, Καθηγητής

Ξάνθη 2022

*Αφιερώνεται στην οικογένεια μου για όλη την στήριξη που μου παρέχει στην ως
τώρα διαδρομή μου.*

Ευχαριστίες

Με την ολοκλήρωση των προπτυχιακών μου σπουδών θα ήθελα να ευχαριστήσω όλους όσους με στήριξαν άμεσα ή έμμεσα στην πραγματοποίηση της διπλωματικής μου εργασίας και γενικότερα της προπτυχιακής μου πορείας.

Θα ήθελα να ευχαριστήσω θερμά τον κ. Βασίλειο Τσαουσίδα, Καθηγητή του τμήματος Η.Μ.Μ.Υ. του Δ.Π.Θ. , για την και ανάθεση του συγκεκριμένου θέματος και για την καθοδήγηση και εμπιστοσύνη προς το πρόσωπο μου.

Επιπρόσθετα, ευχαριστώ ενθέρμως τον κ. Βασίλειο Δεμίρογλου, υποψήφιο Διδάκτορα του τμήματος Η.Μ.Μ.Υ. του Δ.Π.Θ. , για την πολύτιμη βοήθεια και τις ουσιαστικές παρατηρήσεις και προτάσεις του που συνέβαλαν στην ολοκλήρωση της παρούσας εργασίας.

Abstract

Nowadays, interconnected devices are deeply influencing human life and are evolving at a fast pace. Internet of Things (IoT) devices are characterized by a change of location and frequent change of their state from active to inactive. The common TCP / IP communication protocol is designed based on endpoints, nodes with fixed position and status. So, despite any evolution of TCP / IP, there is a need to develop a new architecture for such topologies. Disruption/Delay Tolerant Networks (DTN) are characterized by not having permanent end-to-end connections, long and variable delays, asymmetric data rates and high error rates. In this type of networks, different routing techniques are used based on the store-carry-forward mechanism, i.e., the nodes store the messages and forward them when there is a suitable opportunity for communication. This guarantees the delivery of a message, but it may take some more time. In this Diploma Thesis firstly there is an analysis of the DTN architecture at a theoretical level. The tools available for simulating a DTN topology are also presented. Finally, the evaluation of the latest Bundle Protocol 7 is done in real conditions, designing the respective scenarios and making use of real hardware that runs the needed software implementation.

Keywords:

Computer Networks, Routing, Internet of Things, Delay Tolerant Networks, Bundle Protocol 7

Περίληψη

Στην σημερινή εποχή οι διασυνδεδεμένες συσκευές έχουν εισέλθει για τα καλά στην ζωή του ανθρώπου και εξελίσσονται με γοργούς ρυθμούς. Οι συσκευές στο Διαδίκτυο των Πραγμάτων χαρακτηρίζονται από αλλαγή θέσης και συχνή εναλλαγή της κατάστασής τους από εν ενεργεία σε αδράνεια. Το παραδοσιακό πρωτόκολλο επικοινωνίας TCP/IP, έχει σχεδιαστεί με βάση τα τελικά σημεία, κόμβους με σταθερή θέση και κατάσταση. Έτσι παρά την όποια εξέλιξη του TCP/IP, υπάρχει η ανάγκη δημιουργίας μιας νέας αρχιτεκτονικής για τέτοιες τοπολογίες. Τα Δίκτυα με Ανοχή στις Διακοπές/Καθυστερήσεις (DTN) χαρακτηρίζονται από: απουσία μόνιμων συνδέσεων από άκρο σε άκρο, μεγάλες και μεταβλητές καθυστερήσεις, ασύμμετρους ρυθμούς δεδομένων και υψηλά ποσοστά σφαλμάτων. Σε αυτό το είδος δικτύων, χρησιμοποιούνται διαφορετικές τεχνικές δρομολόγησης που σαν βάση έχουν τον μηχανισμό store-carry-forward, δηλαδή οι κόμβοι αποθηκεύουν τα μηνύματα και τα προωθούν όταν υπάρξει κατάλληλη ευκαιρία επικοινωνίας. Έτσι εγγυάται την παράδοση ενός μηνύματος, αλλά υπάρχει η πιθανότητα να καθυστερήσει. Στην παρούσα διπλωματική εργασία, αρχικά γίνεται ανάλυση της αρχιτεκτονικής DTN σε θεωρητικό επίπεδο. Ακόμα γίνεται παρουσίαση των εργαλείων που υπάρχουν διαθέσιμα για την προσομοίωση μιας τοπολογίας DTN. Τέλος πραγματοποιείται η αξιολόγηση του πιο πρόσφατου Bundle Protocol 7 σε πραγματικές συνθήκες, σχεδιάζοντας τα αντίστοιχα σενάρια και κάνοντας χρήση πραγματικού hardware που εκτελεί την απαιτούμενη υλοποίηση.

Λέξεις κλειδιά

Δίκτυα Υπολογιστών, Δρομολόγηση, Διαδίκτυο των Πραγμάτων, Δίκτυα Ανεκτικά στην Καθυστερήση, Bundle Protocol 7

Πίνακας Περιεχομένων

Πίνακας περιεχομένων

Ευχαριστίες	iii
Abstract.....	iv
Περίληψη	v
Πίνακας Περιεχομένων	vi
Κατάλογος Εικόνων	viii
Κατάλογος Διαγραμμάτων	ix
Κατάλογος Πινάκων.....	x
Κατάλογος Συντομογραφιών	xi
Κεφάλαιο 1: Εισαγωγή.....	1
1.1 Περιγραφή του Θέματος	1
1.2 Στόχοι της Εργασίας.....	1
1.3 Διάρθρωση της Εργασίας.....	2
Κεφάλαιο 2: Θεωρητικό Υπόβαθρο	3
2.1 Το Διαδίκτυο	3
2.2 Το Διαδίκτυο των Αντικειμένων	4
2.3 Delay Tolerant Networks	5
2.3.1 Αρχιτεκτονική DTN.....	5
2.3.2 Store-carry and forward	7
2.3.3 Custody Transfer	8
2.3.4 Endpoint ID	9
2.3.5 Convergence Layer	9
2.3.6 Bundle Protocol 7.....	10
2.4 DTN Routing.....	10
2.4.1 Direct Delivery.....	11
2.4.2 Epidemic	12
2.4.3 Spray and Wait	13
2.4.4 PRoPHET	14
2.4.5 DTLSR	15
2.4.6 Άλλες Μορφές Routing Protocols	16
Κεφάλαιο 3: Σχετική Έρευνα	18
Κεφάλαιο 4: Εργαλεία	22

4.1 Υπάρχουσες υλοποιήσεις	22
4.2 dtn7-go	24
Κεφάλαιο 5: Μεθοδολογία	26
5.1 Σενάρια	26
5.2 Τοπολογία Δικτύου	27
5.3 Ενδεικτικά Σενάρια	28
Κεφάλαιο 6: Πειραματικό Μέρος	30
6.1 Εισαγωγή	30
6.2 Μετρικές	31
6.2.1 Delivery Ratio	31
6.2.2 Latency Time	31
6.2.3 Overhead Ratio	31
6.3 Πείραμα 1	32
6.4 Πείραμα 2	34
6.5 Πείραμα 3	36
6.6 Πείραμα 4	38
Κεφάλαιο 7: Συμπεράσματα	41
Κεφάλαιο 8: Προοπτικές	43
Βιβλιογραφικές Αναφορές	44
Παράρτημα	48
Π.1 Διαδικασία διεξαγωγής των πειραμάτων	48
Π1.1 Πείραμα 1	48
Π1.2 Πείραμα 2	64
Π1.3 Πείραμα 3	72
Π1.4 Πείραμα 4	72
Π.2 Αναλυτικά Αποτελέσματα Πειραμάτων	74
Π2.1 Πείραμα 1	74
Π2.2 Πείραμα 2	74
Π2.3 Πείραμα 3	75
Π2.4 Πείραμα 4	77

Κατάλογος Εικόνων

Εικόνα 1.	Μέρη που απαρτίζουν το IoT [8].....	4
Εικόνα 2.	Παράδειγμα Bundle [13].....	6
Εικόνα 3.	Η διαφορά μεταξύ στοίβας TCP/IP και DTN [15].....	6
Εικόνα 4.	Σύγκριση TCP/IP με DTN [4].....	7
Εικόνα 5.	Δομή ενός DTN Gateway. Πολλαπλά CLAs παρέχουν μια κοινή διεπαφή στον προωθητή μηνυμάτων [6].....	9
Εικόνα 6.	Το Epidemic Protocol όταν δύο κόμβοι A,B βρίσκονται σε επαφή μεταξύ τους [27].....	12
Εικόνα 7.	Παράδειγμα Binary Spray and Wait [24].....	13
Εικόνα 8.	Εξάρτηση μεταξύ DTN και IoT [38].....	19
Εικόνα 9.	Αρχιτεκτονική και ροή δεδομένων στον DTN7 [13].....	25
Εικόνα 10.	Bundle από dtn7-go.....	25
Εικόνα 11.	Raspberry Pi Zero W [72].....	27
Εικόνα 12.	Raspberry Pi 3 Model B [72].....	28
Εικόνα 13.	Άμεση επικοινωνία των 2 κόμβων.....	28
Εικόνα 14.	Έμμεση επικοινωνία των 2 κόμβων.....	29
Εικόνα 15.	Διάγραμμα Ροής request-response_sender_SIMPLE.sh	58
Εικόνα 16.	Διάγραμμα Ροής request-response_receiver_SIMPLE.sh	63
Εικόνα 17.	Διάγραμμα Ροής request-response_sender_size.sh	68
Εικόνα 18.	Διάγραμμα Ροής request-response_receiver_size.sh	71

Κατάλογος Διαγραμμάτων

Διάγραμμα 1. Πείραμα 1, RTT-Contact/Sleep Time.....	32
Διάγραμμα 2. Πείραμα 1, Delivery Ratio-Contact/Sleep Time.....	33
Διάγραμμα 3. Πείραμα 2, Διαφορετικό Μέγεθος Bundle, RTT- Contact/Sleep Time.....	34
Διάγραμμα 4. Πείραμα 2, Διαφορετικό Μέγεθος Bundle, Delivery Ratio- Contact/Sleep Time.....	35
Διάγραμμα 5. Πείραμα 3, Διαφορετικό Πλήθος Bundles, RTT- Contact/Sleep Time.....	36
Διάγραμμα 6. Πείραμα 3, Διαφορετικό Πλήθος Bundles, Delivery Ratio- Contact/Sleep Time.....	37
Διάγραμμα 7. Πείραμα 4, Σύγκριση Πρωτόκολλων Δρομολόγησης, RTT-Max Contact/Sleep Time.....	38
Διάγραμμα 8. Πείραμα 4, Σύγκριση Πρωτόκολλων Δρομολόγησης, Delivery Ratio-Max Contact/Sleep Time.....	39
Διάγραμμα 9. Πείραμα 4, Σύγκριση Πρωτόκολλων Δρομολόγησης, Overhead Ratio-Max Contact/Sleep Time.....	40

Κατάλογος Πινάκων

Πίνακας 1.	Σύνοψη των Πρωτόκολλων Δρομολόγησης.....	17
Πίνακας 2.	Μετρήσεις Πειράματος 1.....	74
Πίνακας 3.	Μετρήσεις Πειράματος 2, 1kB.....	74
Πίνακας 4.	Μετρήσεις Πειράματος 2, 10kB.....	74
Πίνακας 5.	Μετρήσεις Πειράματος 2, 100kB.....	74
Πίνακας 6.	Μετρήσεις Πειράματος 2, 1MB.....	75
Πίνακας 7.	Μετρήσεις Πειράματος 2, 5MB.....	75
Πίνακας 8.	Μετρήσεις Πειράματος 2, 10MB.....	75
Πίνακας 9.	Μετρήσεις Πειράματος 3, 10s.....	75
Πίνακας 10.	Μετρήσεις Πειράματος 3, 20s.....	76
Πίνακας 11.	Μετρήσεις Πειράματος 3, 40s.....	76
Πίνακας 12.	Μετρήσεις Πειράματος 3, 80s.....	76
Πίνακας 13.	Μετρήσεις Πειράματος 3, 160s.....	76
Πίνακας 14.	Μετρήσεις Πειράματος 4, EPIDEMIC.....	77
Πίνακας 15.	Μετρήσεις Πειράματος 4, PRoPHET.....	77
Πίνακας 16.	Μετρήσεις Πειράματος 4, DTLSR.....	77
Πίνακας 17.	Μετρήσεις Πειράματος 4, SPRAY AND WAIT.....	77

Κατάλογος Συντομογραφιών

BP	Bundle Protocol
CLA	Convergence Layer Adapter
DTN	Disruption/Delay Tolerant Networks
IoT	Internet of Things
IP	Internet Protocol
SBC	Single-Board Computer
SoC	System on a chip
TCP	Transmission Control Protocol
WSN	Wireless Sensor Network

Κεφάλαιο 1:

Εισαγωγή

1.1 Περιγραφή του Θέματος

Το Διαδίκτυο αποτελεί αναπόσπαστο κομμάτι της καθημερινότητας του ανθρώπου. Η μορφή του Διαδικτύου έχει αλλάξει πολύ σε σχέση με την αρχική του σύλληψη και έχει εισαχθεί ποικιλία συσκευών όπως smartphones, φορητοί υπολογιστές, δίκτυα αισθητήρων, “έξυπνες” συσκευές κ.α. Το Διαδίκτυο των Πραγμάτων (Internet of Things, IoT) αποτελεί μια επέκταση του Διαδικτύου, η οποία εισάγει δυναμικά σενάρια και τοπολογίες. Στο IoT τα μέλη του δικτύου μπορεί να βρίσκονται εν κινήσει ή να μην είναι πάντα ενεργά. Τέτοιες παράμετροι δεν είχαν ληφθεί υπόψιν κατά την σχεδίαση του TCP/IP και αποτελούν πρόκληση για την από άκρο σε άκρο αρχιτεκτονική του Διαδικτύου. Για την αντιμετώπιση των αναγκών τέτοιων δικτύων, έχουν δημιουργηθεί αρχιτεκτονικές που υπόσχονται να αντιμετωπίσουν τέτοια Οπορτουниστικά Δίκτυα [1]. Η αρχιτεκτονική Disruption/Delay-Tolerant Networking (DTN) υπόσχεται την επίλυση του θέματος εισάγοντας μηχανισμούς όπως την αποθήκευση των μηνυμάτων στους κόμβους μέχρι να υπάρξει ευκαιρία επικοινωνίας με τον στόχο-προορισμό. Το DTN ενώ αρχικά είχε σαν κύριο στόχο την εφαρμογή σε δίκτυα στο διάστημα, πλέον η εφαρμογή του προσανατολίζεται και σε άλλα δίκτυα. Η μεγάλη πρόκληση για την αρχιτεκτονική DTN πέρα από τους κινούμενους κόμβους, είναι πως ανταποκρίνεται σε ετερογενή δίκτυα [2]. Ακόμα ιδιαίτερο ακαδημαϊκό ενδιαφέρον αποτελεί η αξιολόγηση της αρχιτεκτονικής σε πραγματικές συνθήκες, όπως αυτή των IoT.

1.2 Στόχοι της Εργασίας

Κύριο στόχο της παρούσας εργασίας αποτελεί η μελέτη και η αξιολόγηση της νεότερης έκδοσης του Bundle Protocol (BP), και του DTN κατ' επέκταση, σε πραγματικά σενάρια λειτουργίας τοπολογίας IoT. Έπειτα από την θεωρητική μελέτη της αρχιτεκτονικής DTN, στόχος ήταν η εύρεση της κατάλληλης υλοποίησης που θα παρέχει τις δυνατότητες που απαιτεί η συγκεκριμένη μελέτη, καθώς και του αντίστοιχου συμβατού hardware για την πραγματοποίηση των πειραμάτων με πραγματικές συσκευές.

Σημαντικό παράγοντα αποτελεί η σχεδίαση και διεξαγωγή των κατάλληλων IoT σεναρίων και αντίστοιχων τοπολογιών για η αξιολόγηση του όλου εγχειρήματος.

Στη βασική της δομή, η παρούσα διπλωματική εργασία πραγματεύεται την απόδοση του Bundle Protocol 7 και στόχος είναι να γίνει χρήση όλων των διαθέσιμων μέσων και παραμέτρων για την

1.3 Διάρθρωση της Εργασίας

Η παρούσα εργασία αποτελείται από οχτώ κεφάλαια.

- Στο πρώτο κεφάλαιο γίνεται μια εισαγωγή στο θέμα και στους στόχους της εργασίας.
- Στο δεύτερο κεφάλαιο αναλύεται το βασικό θεωρητικό υπόβαθρο για την κατανόηση των εννοιών που θα ακολουθήσουν στην υπόλοιπη εργασία. Αρχικά εξετάζεται το Διαδίκτυο γενικότερα, έπειτα το IoT και τέλος το DTN.
- Στο τρίτο κεφάλαιο εξετάζονται παλαιότερες μελέτες σχετικές με τον συνδυασμό IoT και DTN.
- Στο τέταρτο κεφάλαιο γίνεται μια ανασκόπηση των εργαλείων DTN που υφίστανται και μια πιο αναλυτική περιγραφή της υλοποίησης που επιλέχθηκε για την παρούσα εργασία.
- Στο πέμπτο κεφάλαιο περιγράφονται οι διάφορες τοπολογίες και μεθοδολογίες που αναπτύχθηκαν για την παρούσα εργασία. Αναλύεται το hardware που επιλέχθηκε και η λειτουργία των εκάστοτε κόμβων.
- Στο έκτο κεφάλαιο παρουσιάζονται τα πειράματα και γίνεται σχολιασμός των αποτελεσμάτων τους.
- Στο έβδομο κεφάλαιο γίνεται μια σύνοψη των συμπερασμάτων που προκύπτουν από την παρούσα έρευνα.
- Στο όγδοο κεφάλαιο εκφράζονται ιδέες για μελλοντική διερεύνηση.
- Έπειτα ακολουθούν οι βιβλιογραφικές αναφορές.
- Και τέλος το παράρτημα που περιέχει τμήμα κώδικα και ανάλυσή του, καθώς και τα αναλυτικά αποτελέσματα των μετρήσεων.

Κεφάλαιο 2: Θεωρητικό Υπόβαθρο

2.1 Το Διαδίκτυο

Η σημασία του Διαδικτύου την σημερινή εποχή είναι αδιαμφισβήτητη. Από τον αρχικό σκοπό λειτουργίας του για την σύνδεση συσκευών, πλέον έχει μετατραπεί στον κύριο τρόπο επικοινωνίας μεταξύ των ανθρώπων. Πέρα από επικοινωνία το Διαδίκτυο επεκτείνεται σε μια πληθώρα δραστηριοτήτων, από αγορές μέχρι και διαστημικές αποστολές [3].

Το Διαδίκτυο αποτελείται από πολλά υποδίκτυα, για την επικοινωνία και την διασύνδεση των οποίων απαιτείται η χρήση ενός πρωτόκολλου επικοινωνίας. Από την απαρχή της δημιουργίας του Διαδικτύου μέχρι και σήμερα τις ανάγκες επικοινωνίας εξυπηρετεί το TCP/IP. Η χρησιμότητα της παραδοσιακής τεχνολογίας επικοινωνίας δεδομένων, όπως της καθιερωμένης σουίτας πρωτοκόλλου Διαδικτύου TCP/IP, γίνεται με την προϋπόθεση ότι υπάρχουν συνεχείς και αμφίδρομες διαδρομές από άκρο σε άκρο, σύντομοι χρόνοι μετ' επιστροφής, συμμετρικοί ρυθμοί δεδομένων και χαμηλά ποσοστά σφάλματος [4]. Το IP πρωτόκολλο είναι υπεύθυνο για τη διευθυνσιοδότηση των κόμβων και την δρομολόγηση ενός πακέτου από την πηγή στον προορισμό του. Το IP κάνει την δρομολόγηση χωρίς όμως να εγγυάται για την παράδοση του πακέτου. Το TCP λειτουργεί πάνω από το IP και είναι ο τρόπος επικοινωνίας μεταξύ κόμβων. Το TCP εγγυάται για την παράδοση του πακέτου ενημερώνοντας τον αποστολέα σχετικά με αυτή, και παρέχει έλεγχο συμφόρησης μεταβάλλοντας τον ρυθμό αποστολής των πακέτων [5].

Στην πλειονότητα των σημερινών δικτύων, ισχύουν οι προϋποθέσεις ότι [6]:

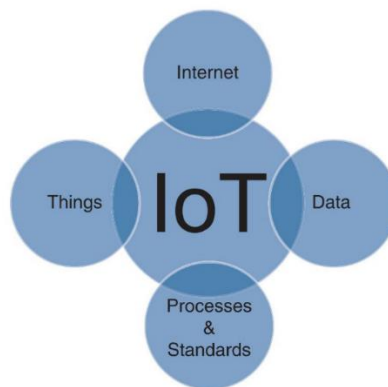
- i. Υπάρχει μια σαφώς ορισμένη διαδρομή από άκρο σε άκρο (end-to-end) για το πως θα φτάσει ένα μήνυμα στον προορισμό του.
- ii. Η επιλογή αυτής της διαδρομής είναι επαρκής για την επίτευξη αποδεκτών επιδόσεων δικτύου.
- iii. Υφίσταται μια διαρκής και αδιάλειπτη συνδεσιμότητα στο Διαδίκτυο για τις κινητές και σταθερές συσκευές .

Η συνεχής εξέλιξη και η ευρεία διαθεσιμότητα του διαδικτύου έχει ως αποτέλεσμα την ολοένα αύξηση των συσκευών του δικτύου. Οι φορητές συσκευές έχουν αλλάξει την μορφή του διαδικτύου και τις απαιτήσεις του. Πλέον οι χρήστες του διαδικτύου καταναλώνουν και μοιράζονται τεράστιο όγκο δεδομένων εν κινήσει.

Το TCP/IP με μικρές προσθήκες, όπως το MOBILE IP [7], προσπαθεί να προσαρμοστεί σε τυχόν νέες τεχνολογίες. Αυτές οι προσπάθειες δεν έχουν όμως πάντα το επιθυμητό αποτέλεσμα και απόδοση.

2.2 Το Διαδίκτυο των Αντικειμένων

Οι χρήστες του διαδικτύου συνέχεια αναζητούν καινοτόμες και αποδοτικές υπηρεσίες για την βελτίωση της καθημερινότητάς τους. Η τεχνολογία έχει πλέον περάσει από την εποχή της διασύνδεσης των δικτύων των υπολογιστών στην διασύνδεση των πραγμάτων (IoT). Η έννοια των πραγμάτων που εντάσσονται στο IoT δεν είναι αυστηρά προκαθορισμένη, περιλαμβάνει μια πληθώρα συσκευών από αισθητήρες αυτοκινήτων μέχρι και οικιακές συσκευές. Το IoT αποτελεί μια τομή του Διαδικτύου, των δεδομένων και των πραγμάτων και όπως είναι λογικό, με την πρόοδο στο Διαδίκτυο υπάρχει πρόοδος και στο IoT [8].



Εικόνα 1. Μέρη που απαρτίζουν του IoT [8]

Ο όρος Internet of Things, αποδόθηκε από τον Kevin Ashton το 1999 που οραματίστηκε πως όλες οι συσκευές της καθημερινότητας θα ήταν εξοπλισμένες με πρωτόκολλα αναγνωριστικών όπως το RFID [9] και δυνατότητα σύνδεσης μεταξύ τους, αλλά και με το Διαδίκτυο για την παρακολούθησή τους. Αυτό το όραμα δεν έχει αλλάξει κατά πολύ ακόμα και τώρα. Σήμερα σκοπός του IoT είναι οι διασύνδεση δισεκατομμυρίων χρηστών και συσκευών, με ανταλλαγή και επεξεργασία δεδομένων [10].

Κάποια από τα βασικά χαρακτηριστικά του IoT είναι: η επεκτασιμότητα και η δυναμικότητα του δικτύου, η ανομοιογένεια και η προσπάθεια ελαχιστοποίησης του κόστους. Λόγω της φύσης των συσκευών και των δυνατοτήτων τους, σκοπός της σχεδίασης μιας IoT τοπολογίας είναι η ελαχιστοποίηση του κόστους της ενέργειας και των πόρων [11]. Εξαιτίας του μικρού χώρου στην μνήμη, μπορούν να ανταλλάξουν περιορισμένο αριθμό μηνυμάτων και για να λειτουργήσουν αποδοτικά πρέπει να εντάσσονται σε δίκτυο με όσο το δυνατόν μικρότερο Overhead. Οι συσκευές IoT, καθώς πολλές τροφοδοτούνται με μπαταρία, προσπαθούν να εξοικονομήσουν όσο το δυνατόν περισσότερη ενέργεια. Για να το επιτύχουν αυτό πολλές φορές μπαίνουν σε λειτουργία αδρανείας για μεγάλο διάστημα, ή απενεργοποιούν τις κεραίες και τα μέσα επικοινωνίας τους. Έτσι, όταν υπάρξει ευκαιρία επικοινωνίας αυτή πρέπει να είναι άμεση και αποδοτική και δίχως απώλειες. Επιπρόσθετα, πέρα από την αποδοτική χρήση των ενεργειακών πόρων, αναγκαία είναι και η αποδοτική χρήση της περιορισμένης επεξεργαστικής ισχύς χρησιμοποιώντας εφαρμογές ειδικά σχεδιασμένες για αρχιτεκτονικές περιορισμένων δυνατοτήτων. Στο IoT λόγω του αυξανόμενου αριθμού συσκευών που το απαρτίζουν, δημιουργείται και μεγάλος αριθμός συνδέσεων. Αυτό σε συνδυασμό με το γεγονός ότι αυτές οι συσκευές μπορούν να κινηθούν, δημιουργεί μια δυναμική τοπολογία. Η δυναμικότητα του δικτύου σε συνδυασμό

με την ετερογένεια του, καθώς στο IoT οι συσκευές είναι συνήθως διαφορετικές μεταξύ, είναι κάτι που είναι απαραίτητο να αντιμετωπίζεται με κάποιον μηχανισμό από τα πρωτόκολλα του IoT.

Το σημερινό πρωτόκολλο Διαδικτύου TCP/IP έχει σαν προϋπόθεση την αδιάλειπτη επικοινωνία των δύο κόμβων. Όμως, όπως γίνεται εύκολα αντιληπτό, η προϋπόθεση αυτή δεν τηρείται πάντα στο IoT, δεδομένου της μεγάλης και δυναμικής τοπολογίας υπάρχουν πολλές περιπτώσεις που ο αποστολέας και παραλήπτης ενός μηνύματος δεν θα είναι ταυτόχρονα συνδεδεμένοι καθ' όλη την διάρκεια της συναλλαγής. Έτσι, έχει δημιουργηθεί η ανάγκη για ανάπτυξη νέων πρωτοκόλλων και αρχιτεκτονικών διασύνδεσης για τα μέλη του IoT, που να μην επηρεάζονται από τις καθυστερήσεις και διακοπές και να ανταποκρίνονται στις απαιτήσεις του IoT.

2.3 Delay Tolerant Networks

Οι προϋποθέσεις που αναλύθηκαν στην Ενότητα 2.1 δεν συναντιούνται πάντα. Ορισμένα δίκτυα παρουσιάζουν μεγάλες και μεταβλητές καθυστερήσεις, σε συνδυασμό με απροσδιόριστες αποσύνδεσης των μελών και ασυμμετρία στην αμφίδρομη επικοινωνία [4]. Παραδείγματα τέτοιων δικτύων είναι αυτά των δικτύων κινητής τηλεφωνίας, αισθητήρων, τα δίκτυα που χρησιμοποιούνται από τις στρατιωτικές υπηρεσίες, δίκτυα που έχουν καταστραφεί από φυσικές καταστροφές ή σε δυσπρόσιτες περιοχές και τα υποθαλάσσια καθώς και διαπλανητικά δίκτυα. Για να αντιμετωπιστούν τα προβλήματα δρομολόγησης που προκύπτουν από τέτοιες τοπολογίες, και επιπλέον να καταστεί δυνατή η επικοινωνία και αλληλεπίδραση μεταξύ ανόμοιων δικτύων, αναπτύχθηκαν τα Disruption/Delay Tolerant Networks (DTN).

2.3.1 Αρχιτεκτονική DTN

Το DTN χαρακτηρίζεται από το ότι λειτουργεί σε δίκτυα που δεν έχουν μόνιμες συνδέσεις από άκρο σε άκρο και ως εκ τούτου έχουν μεγάλες και μεταβλητές καθυστερήσεις, υψηλά ποσοστά σφαλμάτων και διακοπτόμενη συνδεσιμότητα. Στο DTN οι διαδρομές αποτελούνται από μια αλληλουχία εξαρτώμενων από το χρόνο επαφών (ευκαιρίες επικοινωνίας) που χρησιμοποιούνται για τη μεταφορά μηνυμάτων από την προέλευσή τους προς τους προορισμούς τους. Για να σταλεί ένα μήνυμα από το ένα σημείο στο άλλο, μπορεί να χρειαστούν λεπτά, εβδομάδες ή και μήνες.

Στο πρωτόκολλο μετάδοσης TCP για την επίτευξη επικοινωνίας απαιτούνται αρκετά RTTs που πιθανότατα σε ένα απαιτητικό σενάριο δεν θα ολοκληρωνόταν ποτέ η επικοινωνία με επιτυχία, γι' αυτό προτάθηκε ένα ειδικό πρωτόκολλο επικοινωνίας για το DTN, που ονομάζεται Bundle Protocol (BP) [12].

Με το BP, το request μαζί με όλες τις απαιτούμενες πληροφορίες ενθυλακώνονται σε ένα μήνυμα bundle, και το ίδιο συμβαίνει και στην απάντηση που θα λάμβανε πίσω. Στο DTN οποιαδήποτε επιβεβαίωση από τον κόμβο λήψης είναι προαιρετική, ανάλογα με την επιλεγμένη κατηγορία υπηρεσίας [4].

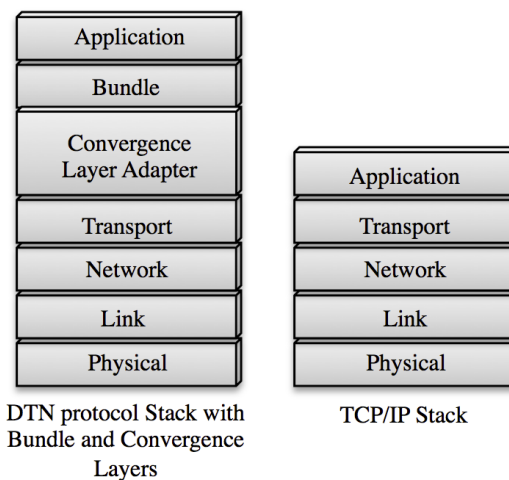
Στην Εικόνα 2 φαίνεται η δομή ενός bundle. Αυτό αποτελείται από [13]:

- i. Primary Block: περιέχει βασικές πληροφορίες όπως η έκδοση του BP, Control Flags για την παροχή πληροφοριών στο bundle, CRC Checksum για έλεγχο εγκυρότητας, τον κόμβο πηγή και προορισμό, την διάρκεια ζωής του bundle.
- ii. Canonical Blocks: στην συγκεκριμένη περίπτωση ένα Payload Block που έχει τα δεδομένα που πρέπει να σταλούν μέσω του bundle. Υπάρχουν και τα Extension Blocks, Hop Count στην προκειμένη περίπτωση, που περιέχουν περαιτέρω πληροφορίες σχετικές με την δρομολόγηση του DTN ανάλογα με τη διαμόρφωσή του.

Bundle		
Primary Block	Hop Count Block	Payload Block
Version: 7 Control Flags: <i>Status requested for reception</i> CRC Type: <i>None</i> Destination EID: dtn:sink/lux Source node EID: dtn:b2 Report-to EID: dtn:b2 Creation Timestamp: (0, 23) Lifetime: 3600000	Type Code: 9 Number: 2 Control Flags: <i>None</i> CRC Type: <i>None</i> Data: (64, 42)	Type Code: 1 Number: 1 Control Flags: <i>None</i> CRC Type: <i>None</i> Data: 0E C6

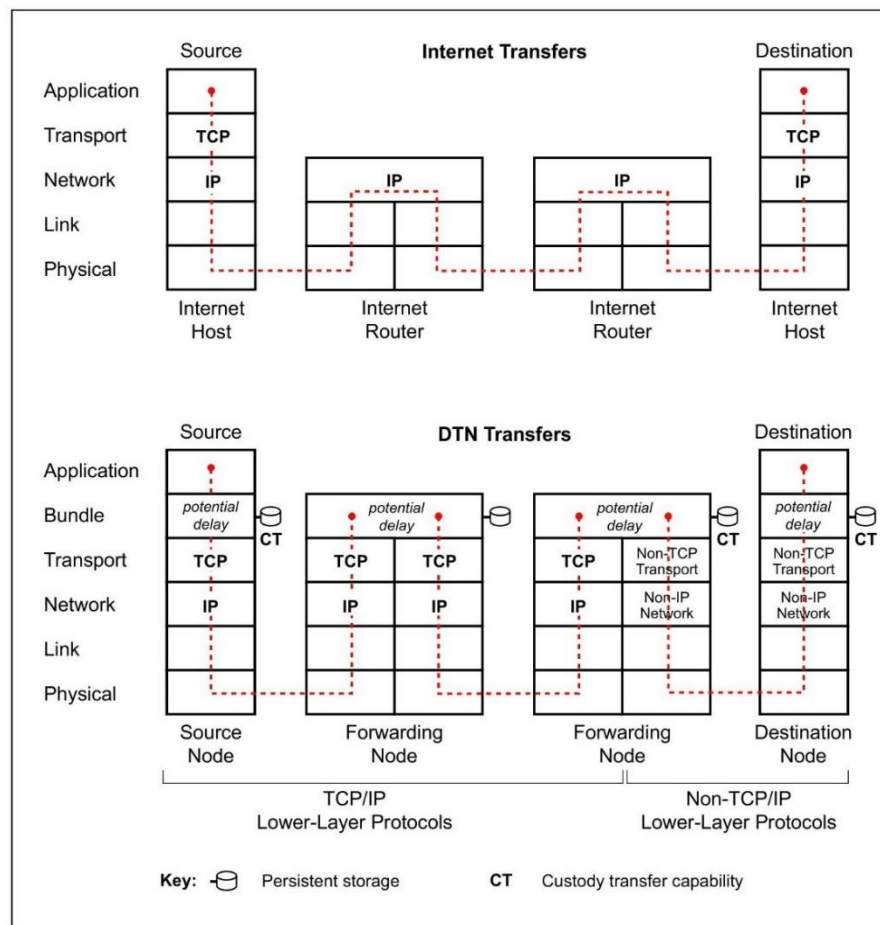
Εικόνα 2. Παράδειγμα Bundle [13]

Το DTN αποτελεί μία αρχιτεκτονική επικάλυψης η οποία λειτουργεί πάνω από τις υπάρχουσες στοίβες άλλων πρωτοκόλλων διασύνδεσης, π.χ. επικαλύπτοντας το TCP. Το BP όπως φαίνεται στην Εικόνα 3 εισάγει ένα νέο επίπεδο, το Bundle Layer μεταξύ του Επιπέδου Μεταφοράς και Εφαρμογής (Transport και Application Layer), επιτρέποντας τη διασύνδεση των ετερογενών τμημάτων ενός μεγαλύτερου δικτύου [14]. Το Bundle Protocol είναι πάνω από το Convergence Layer Protocol stack, το οποίο υποστηρίζει την ανταλλαγή bundles. Το όριο μεταξύ του BP και των πρωτόκολλων χαμηλού επιπέδου (lower-layer) ονομάζεται Convergence Layer Interface.



Εικόνα 3. Η διαφορά μεταξύ στοίβας TCP/IP και DTN [15]

Στις συνηθισμένες τοπολογίες Διαδικτύου, το TCP λειτουργεί στα άκρα της διαδρομής, ενώ το IP κατά μήκος της διαδρομής. Εν αντιθέσει, στο DTN όλοι οι κόμβοι εφαρμόζουν τόσο το BP όσο και ένα πρωτόκολλο μεταφοράς κατώτερου επιπέδου (lower-layer transport protocol).



Εικόνα 4. Σύγκριση TCP/IP με DTN [4]

Οι βασικοί μηχανισμοί και χαρακτηριστικά του BP είναι:

- i. Αποθήκευση-μεταφορά-και-προώθηση (Store-carry-and-forward) και custody transfer για την διασφάλιση παράδοσης του bundle.
- ii. Χαρακτηριστικά των κόμβων όπως το Endpoint ID και το Time to Live (TTL, δηλαδή την διάρκεια ζωής ενός bundle).
- iii. Convergence Layer για την επικοινωνία.
- iv. Ασφάλεια. Παρότι το BP δεν παρέχει άμεσα μηχανισμό ασφαλείας, χρησιμοποιεί ένα μπλοκ επέκτασης που εισάγεται στο bundle. Το Bundle Protocol Security (BPsec) [16] παρέχει ακεραιότητα δεδομένων, αυθεντικοποίηση και εμπιστευτικότητα.

2.3.2 Store-carry and forward

Σε κάθε δεδομένη χρονική στιγμή οι συνδέσεις ή οι ευκαιρίες για προώθηση ενός μηνύματος μπορεί να διαφέρουν, λόγω της κίνησης των κόμβων. Για την επίτευξη της τελικής παράδοσης σε μια δυναμική τοπολογία μπορεί να απαιτούνται κάποιοι ενδιάμεσοι κόμβοι για να μεταφέρουν τα μηνύματα στον επόμενο κόμβο ή στον τελικό προορισμό. Αυτό αποτελεί τον μηχανισμό store-carry-and-forward. Οι

κόμβοι αποθηκεύουν τα μηνύματά τους, όταν δεν υπάρχει ευκαιρία επικοινωνίας. Μόλις δημιουργηθεί μια επαφή, αποστέλλεται το μήνυμα και μπορεί να διατηρηθεί ένα αντίγραφο του στον κόμβο. Ο κόμβος που έλαβε το μήνυμα θα το μεταφέρει μέσω του δικτύου μέχρι να γίνει μια νέα επαφή, επαναλαμβάνοντας τον μηχανισμό store-carry-forward μέχρι να επιτευχθεί η παράδοση του μηνύματος στον προορισμό του .

Στο DTN υπάρχει η δυνατότητα αποθήκευσης μηνυμάτων επ'αόριστον, έως ότου βρεθεί ο αποδέκτης, το οποίο ορίζεται ως επίμονη αποθήκευση (persistent storage). Σε αντίθεση με την βραχυπρόθεσμη αποθήκευση που παρέχεται από τους buffer των κοινών δρομολογητών Διαδικτύου για να αποθηκεύουν σε μια ουρά εισερχόμενα πακέτα για μερικά χιλιοστά του δευτερολέπτου [4].

Για την διασφάλιση της παράδοσης του Bundle υπάρχει το return receipt, που αποτελεί μια επιβεβαίωση από τον προορισμό προς την πηγή-προέλευση ότι το bundle έχει ληφθεί από την εφαρμογή προορισμού και παρέχει διασφάλιση παράδοσης από άκρο σε άκρο.

2.3.3 Custody Transfer

Με δεδομένο ότι η σύνδεση από άκρο σε άκρο δεν είναι πάντα εγγυημένη, το DTN εισάγει έναν μηχανισμό βασισμένο στο custody, για την επίτευξη αξιόπιστης σύνδεσης. Αυτή η πολιτική αναμετάδοσης καθιστά έναν κόμβο DTN υπεύθυνο για τη διατήρηση ενός bundle έως ότου αναλάβει αυτή την ευθύνη κάποιος άλλος κόμβος ή λήξει το TTL του bundle. Για την επίτευξη ανθεκτικότητας το πρωτόκολλο έχει μηχανισμούς κατακερματισμού, μέσω των οποίων επιτυγχάνεται η τμηματική μεταφορά του bundle σε μικρό χρονικό διάστημα. Διαφορετικά το μεγάλο μέγεθος των bundles θα μπορούσε να προκαλέσει προβλήματα στην μετάδοση [12]. Σημαντική λεπτομέρεια είναι ότι η παράδοση ενός bundle είναι ανεξάρτητη από την παράδοση ενός άλλου.

Η ιδέα της μεταφοράς κηδεμονίας (custody transfer) είναι θεμελιώδης για την αρχιτεκτονική. Με αυτόν τον μηχανισμό μειώνονται αισθητά οι απώλειες. Επιπρόσθετα, ωφελούνται οι κόμβοι που είναι πιο αδύναμοι σε πόρους, καθώς δεν απαιτείται πλέον από αυτούς η διατήρηση της κατάστασης σύνδεσης από άκρο σε άκρο [6].

Ο τρόπος που δουλεύει ο μηχανισμός είναι με την αποστολή ενός bundle από τον τωρινό κόμβο που είναι υπεύθυνος για την παράδοση του μηνύματος (bundle custodian) στον επόμενο κηδεμόνα (custodian), ζητώντας να αναλάβει αυτός την ευθύνη της παράδοσης και εκκινεί ένα time-to-acknowledge retransmission timer. Εάν ο επόμενος κόμβος δεχθεί τη κηδεμονία (custody), επιστρέφει μια επιβεβαίωση στον αποστολέα, διαφορετικά, γίνεται προσπάθεια επαναποστολής του bundle, αν δεν έχει παρέλθει ο χρόνος του timer. Η τιμή που έχει αντιστοιχιστεί στο time-to-acknowledge retransmission timer μπορεί να υπολογιστεί με βάση τις επαφές ή ανάλογα την δρομολόγηση [4].

Σημαντική παρατήρηση είναι ότι το custody transfer ενισχύει την αξιοπιστία από άκρο σε άκρο, αλλά δεν την εγγυάται.

2.3.4 Endpoint ID

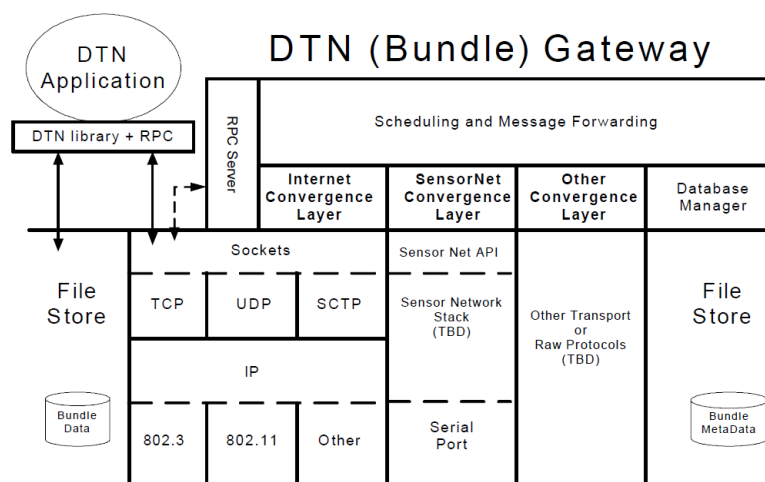
Το DTN χρησιμοποιεί έναν μηχανισμό ονοματοδοσίας και διευθυνσιοδότησης, μέσω του οποίου κάθε κόμβος αναγνωρίζεται μέσω ενός Αναγνωριστικού Τελικού Σημείου EID (Endpoint Identifier). Για τα EID ορίζεται ο περιορισμός ότι πρέπει να ανήκουν στην κατηγορία Ενιαίο Αναγνωριστικό Πόρων (URI) [15]. Η μορφή του είναι: <όνομα_σχήματος>:<τμήμα_του_σχήματος>, όπου το όνομα του σχήματος είναι συνήθως dtm ή ipn [17].

Τα bundles απευθύνονται σε Endpoint Identifiers, τα οποία ενδέχεται να μην είναι τμήμα του δικτύου την στιγμή που γίνεται η προσπάθεια επικοινωνίας. Ένα bundle endpoint είναι ένα σύνολο μηδενικών (null) ή περισσότερων κόμβων που όλοι τακτοποιούνται με το ίδιο endpoint ID. Τα endpoints ανήκουν συχνά σε ομάδες ή αποτελούνται από ομάδες. Δηλαδή ένα endpoint μπορεί να αποτελείται από πολλούς κόμβους, και αντίστοιχα οι κόμβοι μπορεί να είναι μέλη σε πολλά endpoints. Συνοψίζοντας τα endpoints είναι ένα σύνολο κόμβων που αποτελεί την πηγή ή τον προορισμό των bundles [17].

2.3.5 Convergence Layer

Το DTN είναι ένα δυναμικό και ετερογενές δίκτυο, έτσι τα bundles ανταλλάσσονται μέσω συνδέσεων μεταξύ κόμβων διαφορετικών χαρακτηριστικών και με συνδέσεις που μπορεί να διαφοροποιούνται ως προς την κατεύθυνση, την ταχύτητα μετάδοσης και το εύρος ζώνης.

Τα Convergence Layer (CL) Protocols (CLP) είναι τα πρωτόκολλα που χρησιμοποιούνται από το BP για την παράδοση των μηνυμάτων και το Convergence Layer Adapter (CLA) αποτελεί μια υλοποίηση ενός CLP. Από την ομάδα IETF DTN ορίζονται δύο CLPs για την ανταλλαγή bundles με την χρήση μιας σύνδεση TCP, το αμφίδρομο TCP Convergence Layer Protocol (TCPCL) [18] και το μιας κατεύθυνσης Minimal TCP Convergence Layer Protocol (MTCP) [19]. Τα CLP ενισχύουν υπάρχοντα πρωτόκολλα μεταφοράς και τα κάνουν συμβατά με το DTN, π.χ. το TCP χρειάζεται ενίσχυση καθώς δεν έχει όριο στα μηνύματα [6]. Το Bundle Protocol βασίζεται στις δυνατότητες του CLP για την ελαχιστοποίηση της συμφόρησης στο επικαλυπτόμενο δίκτυο [17].



Εικόνα 5. Δομή ενός DTN Gateway. Πολλαπλά CLAs παρέχουν μια κοινή διεπαφή στον προωθητή μηνυμάτων [6]

2.3.6 Bundle Protocol 7

Όπως είναι φυσικό, το BP7 έχει διατηρήσει ορισμένα βασικά χαρακτηριστικά ίδια με τον προκάτοχό του, το BP6. Ωστόσο, το BP7 φέρνει και ορισμένες αλλαγές που το καθιστούν μη συμβατό με το BP6 και υπόσχονται να βελτιώσουν την συμπεριφορά του.

Το BP7 διαφοροποιείται ως προς την μορφή των δεδομένων και τον τρόπο κωδικοποίησης των bundles, χρησιμοποιώντας το πρωτόκολλο CBOR [20] αντί του SDNV [21]. Το CBOR μοιάζει σαν ιδανική λύση για το DTN, καθώς σαν σκοπό έχει να επιτύχει υψηλότερο ποσοστό μετάδοσης και επεξεργασίας. Επιπρόσθετα, το CBOR έχει σχεδιαστεί για συστήματα με περιορισμένους πόρους, άρα είναι ιδανικό για χρήση σε IoT [22].

Μια άλλη αλλαγή είναι ότι ο μηχανισμός custody transfer πλέον ενσωματώνεται στο Bundle-in-Bundle-Encapsulation Protocol αντί για το BP [17]. Αυτό έγινε καθώς η χρήση custody transfer δεν είναι πάντα απαραίτητη και σε ορισμένες περιπτώσεις έχει αρνητικά αποτελέσματα όταν επιλέγεται custodian που δεν είναι τελικά ιδανικός για την παράδοση. Έτσι, το BP7 έχει το προνόμιο να ενεργοποιεί αυτόν τον μηχανισμό μόνο όταν είναι απαραίτητο.

Ακόμα, στο BP7 εισάγεται ο όρος node ID, με τον οποίο ένας κόμβος πρέπει να παραμείνει σε ένα singleton endpoint όταν αυτός χρησιμοποιείται για την επικοινωνία. Αυτό λύνει το πρόβλημα που υπήρχε στο BP6 όταν κάποιος κόμβος έστελνε ένα bundle και εν συνεχεία αποχωρούσε από το endpoint του με αποτέλεσμα τα μηνύματα που γυρνούσαν (όπως αναφορά παράδοσης) προς αυτόν να χάνονται.

Στο BP7 το Primary Block κάθε bundle πρέπει να παραμένει αμετάβλητο μέχρι την παράδοση του. Επιπλέον υπάρχει η προσθήκη της προαιρετικής χρήσης CRC Checksum για την ακεραιότητα των μηνυμάτων, κάτι που έλειπε σε προηγούμενες εκδόσεις. Επιπρόσθετα, έγιναν αλλαγές στα Canonical Block προσθέτοντας το πεδίο block ID number για την υποστήριξη του BPsec [16]. Ακόμα στα extension blocks προστέθηκαν τα πεδία: bundle age, previous node και hop count, τα οποία είναι ιδιαίτερα χρήσιμα σε θέματα δρομολόγησης των bundles, καθώς δείχνουν αν ένα bundle έχει λήξει, από που προήλθε το bundle και αν ο κόμβος πρέπει να ρίξει κάποιο bundle.

Πέρα από τις παραπάνω, το BP7 έχει ορισμένες ακόμα προσθήκες, οι οποίες όμως είναι πιο εξειδικευμένες. Στόχος αυτής της εργασίας είναι να εκμεταλλευτεί τα έξτρα εργαλεία που δίνει το BP7 έτσι ώστε να διασφαλιστεί το καλύτερο δυνατό αποτέλεσμα με χρήση DTN και ιδιαίτερα σε IoT.

2.4 DTN Routing

Βασικό χαρακτηριστικό κάθε πρωτοκόλλου δικτύου επικοινωνίας είναι η δυνατότητα δρομολόγησης πληροφοριών από την πηγή-αποστολέα προς τον προορισμό-παραλήπτη. Στα συνηθισμένα πρωτόκολλα δικτύωσης, η αναπαράσταση της τοπολογίας του δικτύου γίνεται με την βοήθεια ενός γραφήματος στο οποίο ο αλγόριθμος δρομολόγησης επιλέγει τη συντομότερη διαδρομή συμβατή με την πολιτική του πρωτοκόλλου για την δρομολόγηση του μηνύματος, αντιμετωπίζοντας τις

διακοπές ως αποτυχίες σύνδεσης [23]. Εν αντιθέσει με τα συνήθη πρωτόκολλα που η τοπολογίες είναι τετριμμένες, στη δρομολόγηση στο DTN υπάρχει η επιπλέον πρόκληση της εύρεσης ενός κόμβου ιδανικού για την προώθηση των μηνυμάτων, καθώς πολλές φορές η επικοινωνία από άκρο σε άκρο δεν είναι δυνατή για όλο το χρονικό διάστημα, λόγω της κινητικότητας των κόμβων. Άλλοι παράγοντες που καθιστούν δύσκολη την σχεδίαση πρωτόκολλων δρομολόγησης (routing protocols) για το DTN είναι η περιορισμένη χωρητικότητα μνήμης του κόμβου, ο παράγοντας της χωρητικότητα ενέργειας των συσκευών, το bandwidth και η ετερογένεια του δικτύου [24]. Καθώς η βασική αρχή λειτουργίας του DTN είναι το store-carry-and-forward [6], η ορθή επιλογή των κόμβων που θα προωθηθούν ή θα παραμείνουν (αναμένοντας κάποια καλύτερη ευκαιρία προώθησης) τα μηνύματα είναι άκρατα συνδεδεμένη με την απόδοση του δικτύου.

Επιπρόσθετα, η ποικιλομορφία των τοπολογιών που καλύπτονται από το DTN παρακινεί την ανάπτυξη στρατηγικών προσαρμοσμένων στις συγκεκριμένες περιπτώσεις αυτού του τύπου δικτύων. Έτσι έχει αναπτυχθεί μια πληθώρα πρωτόκολλων δρομολόγησης για Δίκτυα Ανεκτικά στην Καθυστέρηση, ορισμένα από τα οποία είναι απλά και δεν κάνουν κανένα είδος μαθηματικών υπολογισμών, ενώ άλλα μπορεί να κάνουν κάποιους υπολογισμούς για να επιλέξουν καλύτερα τους ενδιάμεσους κόμβους που θα αναμεταδώσουν ένα μήνυμα μεταξύ της πηγής και του προορισμού.

Αρχικά τα πρωτόκολλα που προτάθηκαν για τα DTN δεν ήταν πολύπλοκα και δεν χρησιμοποιούσαν τις πληροφορίες από τις επαφές με άλλους κόμβους. Αυτά τα πρωτόκολλα μπόρεσαν να επιτύχουν υψηλό ποσοστό επιτυχίας παράδοσης, αλλά ο αριθμός των αντιγράφων ενός μηνύματος στο δίκτυο ήταν επίσης υψηλός. Αυτό είναι ένα σημαντικό μειονέκτημα, επειδή οι κόμβοι μπορούν να αντιπροσωπεύουν, για παράδειγμα, συσκευές αισθητήρων που έχουν περιορισμένους πόρους. Προκειμένου να μειωθεί ο αριθμός των αντιγράφων, άρχισαν να εισάγονται πιο πολύπλοκοι μηχανισμοί [25].

Υπάρχουν διάφοροι τρόποι κατηγοριοποίησης των πρωτόκολλων δρομολόγησης στο DTN, με παράγοντες όπως: ο μηχανισμός αναπαραγωγής (replication) μηνυμάτων, το πιθανό ποσοστό Overhead κατά μια αποστολή, ακόμα και η επίγνωση της εικόνας του δικτύου που έχει ο κόμβος.

Η διαδικασία δρομολόγησης μπορεί να ταξινομηθεί ανάλογα τον προορισμό και την κατεύθυνση των μηνυμάτων. Η δρομολόγηση Unicast βασίζεται στην αρχή ότι ο προορισμός είναι μοναδικός, ενώ η Multicast δρομολόγηση υποδηλώνει ότι ο προορισμός είναι μια ομάδα κόμβων. Η δρομολόγηση Anycast, με τη σειρά της, είναι μια μέθοδος στην οποία ο προορισμός είναι οποιοσδήποτε κόμβος σε μια ομάδα πιθανών αποδεκτών [24].

2.4.1 Direct Delivery

Το Direct Delivery [26] είναι το πιο απλό πρωτόκολλο δρομολόγησης, στο οποίο κάθε μήνυμα παραμένει αποθηκευμένο στην κόμβο-πηγή δημιουργίας του μέχρι να συναντήσει τον κόμβο-προορισμό. Ο κόμβος δεν λαμβάνει υπόψιν τους υπόλοιπους κόμβους που συναντάει πέρα από τον τελικό προορισμό του μηνύματος. Ως επακόλουθο, αφού αναζητά έναν μόνο αποδέκτη, συνολικά εκτελείται μόνο μία μετάδοση ανά μήνυμα, αν βρει αυτόν τον τελικό αποδέκτη. Το Direct Delivery

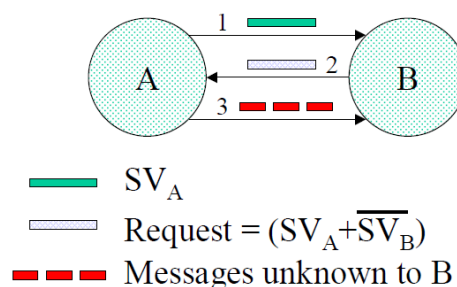
καταναλώνει λιγότερο bandwidth, λιγότερο χώρο από το buffer και λιγότερο overheard, αφού πραγματοποιείται το πολύ μια μετάδοση ανά μήνυμα. Για να είναι αποδοτικό αυτό το πρωτόκολλο θα πρέπει τα bundles να έχουν μεγάλη τιμή TTL και το δίκτυο να είναι πυκνό, καθώς αν το δίκτυο είναι πιο εκτεταμένο ή γενικότερα δεν συναντηθούν σύντομα οι δυο κόμβοι, θα περάσει το περιθώριο του TTL και θα διαγραφεί το bundle.

2.4.2 Epidemic

Το Epidemic πρωτόκολλο δρομολόγησης [27] βασίζεται στο σχήμα flooding. Η βασική ιδέα του πρωτοκόλλου είναι η απεριόριστη αναπαραγωγής (replication) και διανομή μηνυμάτων σε όλο το δίκτυο, σε κάθε πιθανή ευκαιρία επικοινωνίας. Σε ένα δεδομένο χρονικό πλαίσιο, κάθε κόμβος που έχει ένα μήνυμα προς παράδοση θα αναμεταδώσει ένα αντίγραφο του σε κάθε διαθέσιμο γειτονικό κόμβο που βρίσκεται εντός του εύρους επικοινωνίας του. Αυτός ο μηχανισμός θα σταματήσει μόνο όταν το μήνυμα φτάσει στον προορισμό του ή όταν το μήνυμα φτάσει σε ένα προκαθορισμένο μέγιστο hop count.

Το hop count καθορίζει τον μέγιστο αριθμό Epidemic ανταλλαγών στις οποίες υπόκειται ένα συγκεκριμένο μήνυμα. Μεγαλύτερες τιμές για το hop count θα διανέμουν ένα μήνυμα μέσω του δικτύου πιο γρήγορα. Αυτό συνήθως θα μειώσει τον μέσο χρόνο παράδοσης, αλλά θα αυξήσει επίσης τη συνολική κατανάλωση πόρων κατά την παράδοση μηνυμάτων.

Για να αποφευχθούν περιττές συνδέσεις, κάθε κόμβος διατηρεί μια προσωρινή μνήμη-λίστα με κόμβους με τους οποίους επικοινωνήσε πρόσφατα. Όταν συναντηθούν δύο κόμβοι ανταλλάσσουν τα διανύσματα συνοπτικής κατάστασης (summary vectors) τους για να προσδιορίσουν ποια μηνύματα που υπάρχουν στον άλλον κόμβο δεν διαθέτει ο τωρινός κόμβος. Στη συνέχεια, κάθε κόμβος ζητά αντίγραφα των μηνυμάτων που δεν έχει ακόμη στην μνήμη του.



Εικόνα 6. Το Epidemic Protocol όταν δύο κόμβοι A,B

βρίσκονται σε επαφή μεταξύ τους [27]

Σε ένα υποθετικό περιβάλλον, όπου ο χώρος αποθήκευσης μηνυμάτων είναι απεριόριστος και οι επαφές μεταξύ των κόμβων είναι αρκετά μεγάλες, αυτή η προσέγγιση μπορεί να θεωρηθεί ως η βέλτιστη λύση για την ελαχιστοποίηση της καθυστέρησης παράδοσης των μηνυμάτων και τη μεγιστοποίηση του ποσοστού παράδοσης τους [28].

Στην πραγματικότητα όμως τέτοια σενάρια με τόσο μεγάλους χώρους αποθήκευσης σπανίζουν και το σενάριο αυτό μπορεί να οδηγήσει σε συμφόρηση, η οποία έχει ως αποτέλεσμα να διαγράφονται

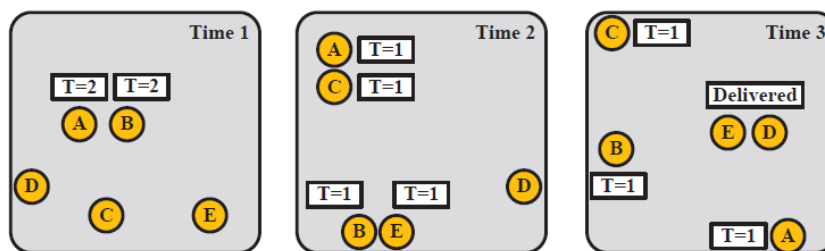
παλαιότερα μηνύματα για να προστεθούν τα νέα στον buffer. Γενικότερα όμως, το Epidemic εξακολουθεί να έχει μεγαλύτερο ρυθμό παράδοσης μηνυμάτων σε σύγκριση με πρωτόκολλα δίχως flooding, αλλά στον αντίποδα έχει μεγάλο Overhead και ιδιαίτερα εκτεταμένη χρήση του buffer κάθε κόμβου.

2.4.3 Spray and Wait

Το Spray and Wait [29] συνδυάζει την ταχύτητα του Epidemic Routing με την απλότητα και λιτότητα του Direct Delivery. Το Spray and Wait έχει δύο εκδοχές, το Normal και το Binary mode. Και για τις δύο εκδοχές το πρωτόκολλο αποτελείται από δύο ξεχωριστές φάσεις:

- i. Ψεκασμού (Spray): Ο κόμβος πηγή προωθεί έναν διαμορφώσιμο αριθμό αντιγράφων μηνυμάτων, T , στον ίδιο αριθμό διαφορετικών κόμβων αναμετάδοσης.
- ii. Αναμονής (Wait): Εάν ο προορισμός δεν βρεθεί κατά την φάση του ψεκασμού, καθένας από τους T κόμβους που φέρει ένα αντίγραφο μηνύματος εκτελεί δρομολόγηση Direct Delivery, δηλαδή θα προωθήσει το μήνυμα μόνο στον προορισμό του.

Όσον αφορά τη φάση του ψεκασμού, μπορούν να επινοηθούν αρκετοί μηχανισμοί για την αποδοτική διανομή των bundles, ο απλούστερος είναι τα T αντίγραφα να προωθηθούν στους πρώτους T κόμβους. Ένας πιο ιδιαίτερος τρόπος είναι αυτός του Binary Spray and Wait, οποιοσδήποτε κόμβος που μεταφέρει $n > 1$ αντίγραφα του μηνύματος προωθεί $\left\lfloor \frac{n}{2} \right\rfloor$ αντίγραφα στον πρώτο κόμβο που θα συναντήσει, που δεν έχει ήδη αντίγραφα, και κρατάει $\left\lfloor \frac{n}{2} \right\rfloor$ στην μνήμη του. Αυτή η διαδικασία επαναλαμβάνεται έως ότου ο σχετικός κόμβος έχει μόνο ένα αντίγραφο του μηνύματος στην προσωρινή μνήμη του, και σε αυτό το σημείο προχωρά στη φάση αναμονής.



Εικόνα 7. Παράδειγμα Binary Spray and Wait [24]

Όταν όλοι οι κόμβοι κινούνται με τρόπο Ανεξάρτητο και Πανομοιότυπα Κατανεμημένο (IID, Independent and Identically Distributed), η δρομολόγηση Binary Spray and Wait είναι η βέλτιστη, δηλαδή έχει την ελάχιστη αναμενόμενη καθυστέρηση μεταξύ όλων των αλγορίθμων δρομολόγησης Spray και Wait [29].

Το Spray and Wait θα μπορούσε να θεωρηθεί ως ανταλλαγή μεταξύ σχημάτων μεμονωμένων και πολλαπλών αντιγράφων.

2.4.4 PROPHET

To Probabilistic Routing Protocol using History of Encounters and Transitivity (PROPHET) [30] δημιουργήθηκε με σκοπό να αντιμετωπίσει την εκτεταμένη κατανάλωση πόρων του Epidemic Routing Protocol. Είναι πρωτόκολλο που εκμεταλλεύεται τη μη τυχαιότητα των συναντήσεων μεταξύ κόμβων, αξιοποιώντας τη γνώση του ιστορικού επαφών των κόμβων για την εκτίμηση της πιθανότητας συνάντησης μεταξύ τους. Το πρωτόκολλο δρομολόγησης PROPHET βρίσκει κυρίως εφαρμογή σε τοπολογίες με κινητούς κόμβους ή κόμβους που μπαίνουν συχνά σε αδράνεια, και έτσι δημιουργούν ορισμένα μοτίβα συνδεσιμότητας που δεν είναι εντελώς τυχαία, αλλά έχουν κάποιο βαθμό προβλεψιμότητας. Για να είναι αποδοτικό το PROPHET αυτά τα μοτίβα πρέπει να διατηρούνται για μεγάλο διάστημα, για όσο το δυνατόν καλύτερη πρόβλεψη. Στα μοτίβα που πατατηρούνται στο PROPHET υπάρχει ένας χαρακτηριστικός χρόνος (όπως η επικοινωνία δύο κόμβων που κινούνται σε μια πόλη και συναντιούνται σε ένα συγκεκριμένο φανάρι) που με τη σειρά του σχετίζεται με τον αναμενόμενο χρόνο που θα χρειαστεί το μήνυμα για να φτάσει στον προορισμό του στο τμήμα του δικτύου που χρησιμοποιεί PROPHET. Αυτός ο χαρακτηριστικός χρόνος βοηθάει στην κατάλληλη διαμόρφωση του πρωτοκόλλου PROPHET σε ένα δίκτυο. Όταν γίνει αυτό, το πρωτόκολλο PROPHET δημιουργεί αποτελεσματικά ένα τοπικό μοντέλο των αναμενόμενων μοτίβων στο δίκτυο που μπορεί να χρησιμοποιηθεί για τη βελτιστοποίηση της χρήσης πόρων, αποκλείοντας από το δίκτυο κόμβους που είναι απίθανο να οδηγήσουν σε τελική παράδοση του μηνύματος [30].

Για την λειτουργία του πρωτοκόλλου δημιουργείται σε κάθε κόμβο μια μετρική, η πιθανότητα παράδοσης, που δείχνει πόσο πιθανό είναι αυτός ο κόμβος να μπορεί να παραδώσει ένα μήνυμα στον τελικό προορισμό. Όταν συναντώνται δύο κόμβοι, ανταλλάσσουν διανύσματα συνοπτικής κατάστασης (summary vectors), και επιπλέον ένα διάνυσμα προβλεψιμότητας παράδοσης (delivery predictability vector) που περιέχει τις πληροφορίες πιθανότητας παράδοσης για προορισμούς που είναι γνωστοί από τους κόμβους. Αυτή η μετρική εξελίσσεται με τον εξής τρόπο: αύξηση της τιμής της εάν δύο κόμβοι επιτύχουν επικοινωνία, ενώ η απουσία συναντήσεων μεταξύ των κόμβων αντανakλάται στη μείωση της τιμής της με την πάροδο του χρόνου. Ακόμα υπάρχει μια μεταβατική ιδιότητα, η οποία βασίζεται στην παρατήρηση ότι εάν ο κόμβος A συναντά συχνά τον κόμβο B και ο κόμβος B συναντά συχνά τον κόμβο C, τότε ο κόμβος C πιθανώς είναι ένας καλός κόμβος για την προώθηση μηνυμάτων που προορίζονται για τον κόμβο A [31].

Το πρωτόκολλο δουλεύει με τέτοιο τρόπο που τα αντίγραφα μεταδίδονται σε κόμβους στους οποίους η πιθανότητας παράδοσης σε σχέση με τον τελικό προορισμό είναι μεγαλύτερη από την αντίστοιχη τιμή που υπάρχει στον τρέχον κόμβο που μεταφέρει το bundle. Ακόμα, ένας κόμβος μπορεί επίσης να επωφεληθεί από την προώθηση του μηνύματος σε έναν κόμβο του οποίου η πιθανότητα να βρει τον παραλήπτη είναι χαμηλή, αλλά, αντίθετα έχει μεγάλη πιθανότητα να βρει έναν άλλο κόμβο που συνδέεται συχνά με τον κόμβο-προορισμό.

Το κύριο ζήτημα του PROPHET σχετίζεται με τον χρόνο που αφιερώνεται στην ανταλλαγή πληροφοριών εκτός των μηνυμάτων, καθώς συνήθως οι επαφές έχουν πολύ μικρή διάρκεια και με την ανάγκη για εκχώρηση μνήμης αυτό καθίσταται ακόμα δυσκολότερο. Το γεγονός πως το PROPHET δεν κάνει τυχαίο flooding μειώνει τη συμφόρηση. Όμως το πρωτόκολλο πρέπει να αντιμετωπίσει το

πρόσθετο έξοδα που σχετίζεται με τη διατήρηση των εκτιμήσεων των πιθανοτήτων συνάντησης, καθώς κάθε κόμβος παρακολουθεί την προβλεψιμότητα παράδοσης των άλλων κόμβων του δικτύου [31].

2.4.5 DTLSR

Στο Delay Tolerant Link State Routing (DTLSR) [32] κάθε κόμβος διατηρεί ένα γράφημα που αντιπροσωπεύει την τρέχουσα προβολή της κατάστασης του δικτύου και χρησιμοποιεί έναν υπολογισμό βέλτιστης διαδρομής (π.χ. Dijkstra) για να βρει διαδρομές για μηνύματα. Με την πάροδο του χρόνου και καθώς αλλάζει η κατάσταση του δικτύου, νέες καταστάσεις διοχετεύονται στο δίκτυο για την ολοκληρωμένη απεικόνισή του. Κάθε κόμβος στο σύστημα εκχωρείται σε μια περιοχή διαχείρισης, και ένα υπόδειγμα πρωτοκόλλου κατάστασης σύνδεσης (link state) λειτουργεί εντός της περιοχής. Με την χρήση αυτού του μηχανισμού περιορίζεται το μέγεθος του γραφήματος του δικτύου και το εύρος των μηνυμάτων ανακοίνωσης (announcement messages). Οι κόμβοι που έχουν γείτονες σε άλλες περιοχές μαθαίνουν το σύνολο των EID που είναι προσβάσιμα μέσω της άλλης περιοχής.

Η συνδεσιμότητα που έχει ένας κόμβος στο δίκτυο γίνεται γνωστή σε αυτόν μέσω μηνυμάτων Link State Announcement (LSA). Στο DTN μια διαδρομή μπορεί να είναι έγκυρη, ακόμη και αν δεν παρέχει τρέχουσα συνδεσιμότητα σε έναν συγκεκριμένο προορισμό και η έλλειψη πρόσφατης επαφής με έναν γειτονικό κόμβο δεν σημαίνει ότι ο κόμβος είναι αναγκαστικά εκτός λειτουργίας. Λόγω της αρχιτεκτονικής του DTN τα LSAs στέλνονται με διάρκεια ζωής ωρών ή ημερών και όλοι οι κόμβοι διατηρούν ένα αντίγραφο του πιο πρόσφατου LSA από άλλους κόμβους στην περιοχή. Ένας κόμβος δεν χρειάζεται να αναμεταδίδει περιοδικά LSA για να εξασφαλίσει τη διάδοση σε όλους τους κόμβους, καθώς ανταλλάσσουν LSA μεταξύ τους, όλοι οι κόμβοι που τελικά είναι προσβάσιμοι θα λάβουν όλα τα LSAs.

Το κύριο χαρακτηριστικό του πρωτοκόλλου DTLSR είναι ο υπολογισμός της συντομότερης διαδρομής. Λόγω της φύσης του DTN ο υπολογισμός είναι πολύ δύσκολος, καθώς οι κόμβοι δεν είναι συνέχεια διαθέσιμοι, αλλά μπορεί να είναι μέρος του δικτύου για συγκεκριμένο χρονικό διάστημα. Ο κόμβος υπολογίζει μια εκτίμηση για την καθυστέρηση που θα χρειαζόταν για την προώθηση ενός μηνύματος χρησιμοποιώντας τον δεδομένο σύνδεσμο (link). Γίνεται διάκριση μεταξύ συνδέσμων που πιστεύεται ότι είναι “διαθέσιμοι” από εκείνους που πιστεύεται ότι είναι “κατεστραμμένοι” κατά τον υπολογισμό διαδρομών. Για μη διαθέσιμους συνδέσμους, χρησιμοποιείται η διάρκεια της διακοπής μέχρι στιγμής ως εκτίμηση της καθυστέρησης. Αυτή η απλή ευρετική αποτυπώνει την πεποίθηση ότι οι σύνδεσμοι που είναι απενεργοποιημένοι για μεγάλο χρονικό διάστημα είναι απίθανο να εμφανιστούν σύντομα.

Συνοψίζοντας, το βασικό χαρακτηριστικό του DTLSR είναι ότι ακόμη και οι σύνδεσμοι προς κόμβους που δεν είναι προσβάσιμοι μια δεδομένη στιγμή, είναι επιλέξιμα συστατικά των καλύτερων διαδρομών. Σε αυτό το πρωτόκολλο κάθε κόμβος έχει ένα γράφημα όλου του δικτύου και προσπαθεί να το αξιοποιήσει, αντιμετωπίζοντας την πρόκληση της δυναμικότητας του δικτύου.

2.4.6 Άλλες Μορφές Routing Protocols

Πέρα από τα πρωτόκολλα που προαναφέρθηκαν, έχουν αναπτυχθεί κάποια πρωτόκολλα που λαμβάνουν υπόψιν πιο σύνθετες παραμέτρους.

Μια κατηγορία τέτοιων πρωτοκόλλων είναι τα Social-Aware. Αυτού του είδους τα πρωτόκολλα δεν ασχολούνται μόνο με δυναμικές πληροφορίες δικτύου (π.χ. στιγμιαία τοποθεσία και συναντήσεις), αλλά στοχεύουν επίσης στη διερεύνηση κοινωνικών σχέσεων μεταξύ των κόμβων. Αυτές οι πληροφορίες τείνουν να είναι πιο σταθερές με την πάροδο του χρόνου και έτσι μπορούν να παρέχουν αξιόπιστους μηχανισμούς για την επιλογή του καλύτερου κόμβου προώθησης. Τα κοινωνικά χαρακτηριστικά μπορούν να συναχθούν από την κοινωνική φύση της ανθρώπινης κινητικότητας. Οι πέντε βασικές κοινωνικές ιδιότητες είναι: Community, Centrality, Similarity, Friendship and Selfishness [25]. Δεν θα αναλυθούν περαιτέρω, καθώς η μελέτη τους δεν συνδράμει στην ανάπτυξη της παρούσας εργασίας. Το πιο χαρακτηριστικό παράδειγμα πρωτόκολλου δρομολόγησης που εκμεταλλεύεται αυτές τις παραμέτρους είναι το BubbleRap το οποίο βασίζεται στο Community και το Centrality προκειμένου να μειωθεί ο αριθμός των αντιγράφων του ίδιου μηνύματος στο δίκτυο αλλά με μεγάλη πιθανότητα παράδοσης [33].

Μια άλλη κατηγορία πρωτοκόλλων είναι το Position-Based, δηλαδή πρωτόκολλα που κάνουν χρήση της τοποθεσίας των κόμβων, πιθανότατα γνωρίζοντας τις συντεταγμένες τους. Υπάρχει ήδη σχετική έρευνα στο πεδίο από τις επικοινωνίες σε δίκτυα οχημάτων (Vehicular Communications), έτσι στο DTN μπορεί να συνεχιστεί καθώς έχουν αρκετά κοινά όπως η κινητικότητα των κόμβων που μπορεί να ακολουθούν ένα συγκεκριμένο μοτίβο, στην περίπτωση των διαδρομών δημόσιων συγκοινωνιών και των μετακινήσεων του ατόμου από την εργασία στην οικία τους, αλλά μπορεί να είναι και τυχαία μοτίβα, όπως μια βόλτα με το αυτοκίνητο για αναψυχή [34]. Σε Position-Based πρωτόκολλα δρομολόγησης γίνεται εκμετάλλευση μόνο των γεωγραφικών πληροφοριών προς έναν κόμβο (π.χ. απόσταση και κατεύθυνση προς τον προορισμό). Επομένως, η γεωγραφική δρομολόγηση μπορεί να προσαρμοστεί στην διακύμανση της τοπολογίας μέσω της γεωμετρικής συμπεριφοράς αναμετάδοσης, η οποία είναι πιο αξιόπιστη για την παράδοση μηνυμάτων, ειδικά για σενάρια VANETs [35]. Τέτοιου είδους πρωτόκολλα είναι ιδανικά για δυναμικές τοπολογίες και δίκτυα κινητών κόμβων [34]. Οι πλειονότητα των πρωτοκόλλων που εντάσσονται σε αυτή την κατηγορία διατηρούν μόνο ένα αντίγραφο του μηνύματος. Χαρακτηριστικό παράδειγμα αποτελεί το Greedy-DTN που δίνει προτεραιότητα στην προώθηση πακέτων στον κόμβο που βρίσκεται πιο κοντά στον προορισμό εκείνη τη στιγμή.

Routing Protocol	Αριθμός Αντιγράφων-Μηνυμάτων	Κύρια Μετρική Προώθησης	Σύνοψη
Direct Delivery	1	ID του Κόμβου Προορισμού	Ο κόμβος πηγή μεταδίδει το μήνυμα μόνο στον προορισμό
Epidemic	Απεριόριστος	Καμία	Σε κάθε ευκαιρία επικοινωνίας οι κόμβοι ανταλλάσσουν τα μηνύματα που τους λείπουν, ο ένας από τον άλλο
PRoPHET	Απεριόριστος	Πιθανότητα Παράδοσης	Όταν δύο κόμβοι συναντώνται, ένα νέο αντίγραφο μηνύματος θα προωθηθεί μόνο εάν ο άλλος κόμβος έχει μεγαλύτερη πιθανότητα παράδοσης για να φτάσει στον προορισμό.
Spray and Wait	Περιορισμένος	Αριθμός Αντιγράφων του Bundle, ID του Κόμβου Προορισμού	Χρησιμοποιώντας μερικώς flooding, ελαττώνει την συμφόρηση σε σχέση με το Epidemic. Όταν μείνει ένα αντίγραφο ανά κόμβο αλλάζει σε Direct Delivery
DTLSR	1	Βέλτιστη Διαδρομή	Κρατώντας ένα αντίγραφο του δικτύου, βρίσκεται η βέλτιστη διαδρομή για την προώθηση του μηνύματος

Πίνακας 1. Σύνοψη των Πρωτόκολλων Δρομολόγησης

Κεφάλαιο 3:

Σχετική

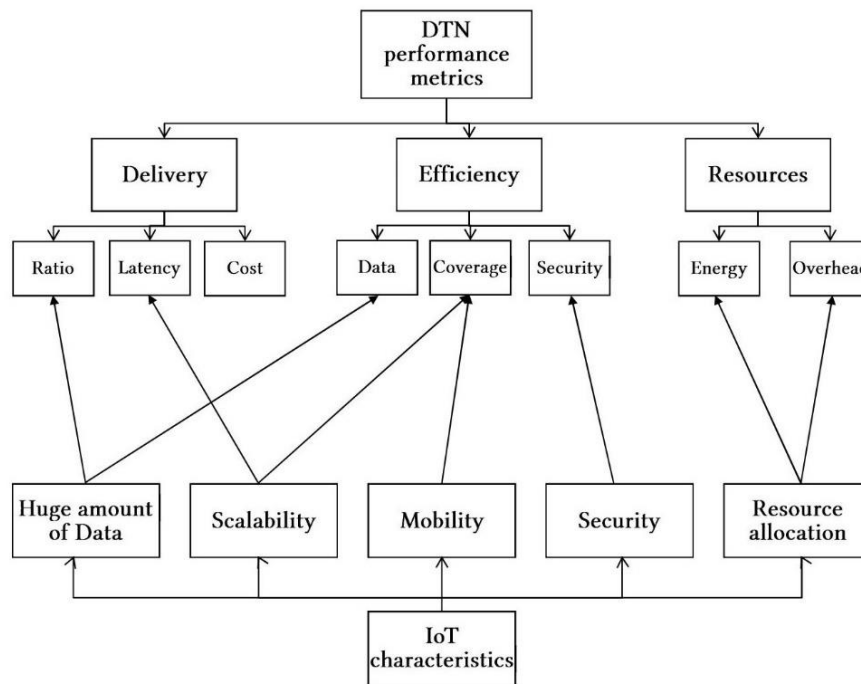
Έρευνα

Η επικοινωνία στο IoT είναι ένα απαιτητικό σενάριο, καθώς τα “έξυπνα” αντικείμενα είναι κινητά και τα δίκτυα που τα απαρτίζουν ετερογενή. Είναι γνωστό ότι τα περιβάλλοντα IoT αντιμετωπίζουν πολλές προκλήσεις όπως η διαθεσιμότητα, η αξιοπιστία και η διαχείριση ενέργειας [36]. Παρά την όποια προσπάθεια για βελτιστοποίηση της επικοινωνίας στο IoT, η διατήρηση μιας σταθερής σύνδεσης δεν επιτυγχάνεται πάντα λόγω της δυναμικότητας της τοπολογίας.

Στην πραγματικότητα, πολλές φορές τα πρωτόκολλα του IoT προσπαθούν να εφαρμόσουν έναν μηχανισμό προσωρινής αποσύνδεσης των συσκευών με τέτοιο ελεγχόμενο τρόπο ώστε το υπόλοιπο δίκτυο να πιστεύει ότι ο κόμβος είναι συνεχώς συνδεδεμένος. Σε περίπτωση διακοπής της σύνδεσης τα δεδομένα παραμένουν αποθηκευμένα στον κόμβο πηγή μέχρι να αποκατασταθεί η σύνδεση από άκρο σε άκρο. Όπως γίνεται αντιληπτό, αυτό δεν αποτελεί ιδανικό τρόπο [37], καθώς αυτή η προσέγγιση χρησιμοποιεί μόνο έναν κόμβο χωρίς να αξιοποιεί το υπόλοιπο δίκτυο.

Στην Ενότητα 2.2 αναλύθηκαν ορισμένες προϋποθέσεις και χαρακτηριστικά για να λειτουργήσει το IoT δίχως προβλήματα, όπως η συνεχής επικοινωνία από άκρο σε άκρο, η μικρή κατανάλωση, η αξιόπιστη σύνδεση, οι περιορισμένοι πόροι κ.α. Η πλειονότητα των προϋποθέσεων αυτών βρίσκει εφαρμογή στο DTN, καθώς και εκεί στόχος είναι η όσο το δυνατόν πιο αποδοτική επικοινωνία σε σύντομο χρόνο επαφής μεταξύ των κόμβων και επιπλέον με τον μηχανισμό store-carry-forward παρέχεται μια αξιόπιστη διαδρομή από άκρο σε άκρο. Ακόμα, το DTN κατεξοχήν περιλαμβάνει κινητές συσκευές και επιτυγχάνει την συνδεσιμότητά τους σε όλη την διαδρομή τους, παρά την συνεχή εναλλαγή θέσεων και κατάστασης των συσκευών, οπότε έχει ήδη τις “υποδομές” για ένα μεγάλο μέρος των συσκευών IoT. Στην Εικόνα 8 παρατηρείται η ισχυρή εξάρτηση μεταξύ των χαρακτηριστικών του IoT και των μετρικών απόδοσης του DTN. Από αυτή την αλληλεξάρτηση γίνεται εύκολα αντιληπτή η ανάγκη ενσωμάτωσης της τεχνολογίας του DTN στο IoT [38]. Οι προκλήσεις που θέτει το IoT, στην πλειοψηφία τους είναι παράγοντες που ήδη έχουν ληφθεί υπόψη στο DTN. Επομένως με την βοήθεια του DTN μπορούν να επεκταθούν οι δυνατότητες του IoT.

Ο συνδυασμός του IoT με το DTN θα επιτρέψει στα έξυπνα αντικείμενα να επικοινωνούν καλύτερα, ακόμη και με την παρουσία διακοπών στη συνδεσιμότητά τους, χρησιμοποιώντας έναν έμπιστο μηχανισμό που διασφαλίζει την τελική παράδοση των δεδομένων στον προορισμό τους. Στο DTN επιτυγχάνεται η αξιοποίηση περισσότερων κόμβων για την παράδοση ενός μηνύματος. Όταν συμβαίνουν πολλές διακοπές κατά μήκος μιας διαδρομής, τα δεδομένα αντί να περιμένουν την σύνδεση με τον έναν μοναδικό κόμβο-προορισμό, διαμοιράζονται στους ενδιάμεσους κόμβους και φτάνουν εν τέλει στον προορισμό τους [37].



Εικόνα 8. Εξάρτηση μεταξύ DTN και IoT [38]

Ακόμα ο μηχανισμός custody transfer του DTN, δίνει την δυνατότητα στους μικρούς αδύναμους σε πόρους κόμβους, να απαλλαχθούν από την ευθύνη παράδοσης ενός μηνύματος και να την αναθέσουν σε κάποιον πιο ισχυρό π.χ. ένα gateway. Έτσι, δεν επιβαρύνονται οι λιγοστοί πόροι τους και ακόμα μπορούν να χρησιμοποιηθούν πιο οικονομικές συσκευές, αφού αντιμετωπίζεται έως έναν βαθμό το πρόβλημα της περιορισμένης μνήμης. Επίσης με την χρήση του μηχανισμού store-carry-forward του DTN, οι IoT κόμβοι μπορούν να είναι σε κατάσταση αδρανείας για αόριστο χρόνο εξοικονομώντας μεγάλο ποσοστό ενέργειας και δίχως να διαταράσσουν την λειτουργικότητα του δικτύου.

Η λειτουργία συσκευών IoT σε καταστάσεις DTN, έχει αρκετές περιπτώσεις χρήσης. Όπως προαναφέρθηκε, το DTN μπορεί να χρησιμοποιηθεί για επικοινωνία κινητών κόμβων, όπως τα οχήματα σε μια πόλη. Ακόμα, βρίσκει εφαρμογή σε ένα δίκτυο αισθητήρων όπου μπορεί οι επιμέρους αισθητήρες να μην είναι ικανοί να είναι σε συνεχή επικοινωνία με το gateway και επομένως να πρέπει να στέλνουν μετρήσεις περιοδικά. Επίσης, το DTN ανταποκρίνεται σε ένα από τα κύρια σενάρια του IoT, δηλαδή την επικοινωνία μεταξύ κόμβων που βρίσκονται για ελάχιστο χρονικό διάστημα εν ενεργεία, εκτελούν μια λειτουργία και αλλάζουν την κατάσταση λειτουργίας τους σε αδρανή. Τέλος μια άλλη χρήση του DTN στο IoT είναι για την επικοινωνία των συσκευών έπειτα από φυσικές καταστροφές, που μπορεί να έχει καταστραφεί μέρος του δικτύου.

Η ενσωμάτωση του DTN στο IoT, αποτελεί θέμα αυξημένης ακαδημαϊκής δραστηριότητας. Πλέον υπάρχουν πολλές μελέτες σχετικά με τα σενάρια και τις χρήσεις του DTN σε συνδυασμό με IoT. Ακολουθούν ορισμένα παραδείγματα.

Ενδιαφέρον παρουσιάζει η χρήση του DTN για την παρακολούθηση της άγριας ζωής. Ένα παράδειγμα είναι το **ZebraNet** [39], που έχει σαν στόχο να παρέχει στους βιολόγους-ερευνητές πληροφορίες σχετικά με το μοτίβο κινητικότητας των ζέβρων στην Κένυα. Το σύστημα αποτελείται από

κινητούς σταθμούς βάσης και κολάρα τα οποία τοποθετούνται στις ζέβρες. Οι συσκευές αυτές αποτελούν ένα IoT δίκτυο, καθώς διαθέτουν ολοκληρωμένο σύστημα για την επεξεργασία, αποθήκευση και αποστολή των δεδομένων. Οι πληροφορίες ανταλλάσσονται με Epidemic δρομολόγηση, ανά τακτά χρονικά διαστήματα μεταξύ των κόμβων-ζέβρων. Προορισμός των μηνυμάτων είναι οι σταθμοί βάσεις που συγκεντρώνουν τις πληροφορίες για την ανάλυση του μοτίβου κίνησης.

Το εγχείρημα του **SHETLAND-NET** [40] είναι η κατασκευή ενός ερευνητικού δικτύου IoT, για την συλλογή μετρήσεων και δεδομένων από περιοχές της Ανταρκτικής με μόνιμα παγωμένο υπέδαφος (permafrost). Η κύρια πρόκληση που έχουν να αντιμετωπίσουν οι ερευνητές που οργανώνουν μελέτες στην Ανταρκτική είναι η έλλειψη έμπιστων δικτύων τηλεπικοινωνιών. Για να αντιμετωπιστεί αυτό το εμπόδιο χρησιμοποιούνται διασυνδεδεμένα απομακρυσμένα δίκτυα αισθητήρων (WSN) που χρησιμοποιούν DTN, και κάνουν την αποστολή των βραδινών μετρήσεων κατά την διάρκεια της ημέρας όταν βρεθεί κάποια ευκαιρία επικοινωνίας. Για την σύνδεση των WSNs χρησιμοποιούνται κύματα χώρου NVIS και ραδιοζεύξεις υψηλής συχνότητας, για να επιτευχθεί η χαμηλή κατανάλωση που απαιτείται από μια τοπολογία IoT.

Το DTN συχνά χρησιμοποιείται για **υποθαλάσσιες επικοινωνίες** μεταξύ WSN δικτύων [41]. Οι υποθαλάσσιες συσκευές συλλέγουν πολλές πληροφορίες σχετικά με την τοποθεσία, την κατάσταση του περιβάλλοντος τους, τις επαφές τους και άλλα. Οι υποθαλάσσιες συσκευές, πέρα από την μεταξύ τους επικοινωνία μπορούν να επικοινωνήσουν και με επίγειες συσκευές. Λόγω της διαφοροποίησης του υδάτινου από το επίγειο περιβάλλον, σε τέτοια δίκτυα οι συνδέσεις είναι ιδιαίτερα ασταθείς, με μεγάλες διακυμάνσεις και καθυστερήσεις. Για να αντιμετωπιστεί αυτό το πρόβλημα δημιουργήθηκαν πρωτόκολλα βασισμένα στο DTN, όπως το Focused Beam Routing (FBR) [42].

Το **IoB-DTN** [43] είναι μια μελέτη που σκοπό έχει την εκμετάλλευση του δημόσιου δικτύου ενοικίασης ποδηλάτων, για τον διαμοιρασμό δεδομένων. Τα ποδήλατα αυτά με τον τρόπο κατασκευής τους αποτελούν ένα δίκτυο IoT καθώς ενσωματώνουν ένα ολοκληρωμένο κύκλωμα διασύνδεσης. Σε αυτό το δίκτυο χρησιμοποιείται ένα πρωτόκολλο δρομολόγησης DTN flooding. Έτσι κάθε ποδήλατο έχει την δυνατότητα να παράγει και να αποθηκεύει δεδομένα, καθώς και να τα μεταφέρει από ποδήλατο σε ποδήλατο μέχρι τα δεδομένα φτάσουν στον προορισμό τους.

Μια άλλη χρήση του DTN σε συνδυασμό με το IoT είναι η δημιουργία δικτύου στις αναπτυσσόμενες χώρες. Αρχικά το **DakNet** [44] χρησιμοποίησε τα λεωφορεία για να συνδέσει τα σημεία πρόσβασης στο Διαδίκτυο, των απομονωμένων χωριών της Ινδίας, και να ανταλλάξει δεδομένα μεταξύ τους. Εν συνεχεία το **KioskNet** [45] θέλοντας να επεκτείνει το DakNet και τις δυνατότητες του DTN, δημιούργησε μια τεχνολογία για την διασύνδεση των περιπτέρων στην Ινδία με την χρήση λεωφορείων και αυτοκινήτων σαν ενδιάμεσους κόμβους για την αποθήκευση και μεταφορά των μηνυμάτων. Αυτό είναι πολύ σημαντικό, καθώς ο ρόλος των περιπτέρων σε τέτοιες περιοχές είναι μείζονος χαρακτήρα, αφού εξυπηρετούν τους πολίτες με υπηρεσίες όπως email, έκδοση πιστοποιητικών, τηλεφωνία κ.α. και η σύνδεση τους αποτελεί μεγάλη ανάγκη.

Επιπλέον υπάρχουν έρευνες για την δημιουργία DTN δικτύων με την χρήση **μη επανδρωμένων εναέριων οχημάτων (UAV)** [46]. Με αυτόν τον τρόπο συσκευές σε αραιά δίκτυα μπορούν να ανταλλάσσουν πληροφορίες μεταξύ τους χωρίς να χρειάζεται να μετακινηθούν οι ίδιες, αυτό μπορεί να

φανεί χρήσιμο στην επικοινωνία μεταξύ δύο gateway και την ανταλλαγή πληροφοριών σχετικά με τους κόμβους τους. Ακόμα τέτοιες τεχνολογίες μπορούν να χρησιμοποιηθούν για την διασύνδεση περιοχών που είναι δυσπρόσιτες και δεν έχουν καλή σύνδεση στο Διαδίκτυο ή έχουν πληγεί από φυσικές καταστροφές [47].

Το **mHealth** [48] είναι μια εφαρμογή που στόχο έχει την απομακρυσμένη παρακολούθηση ασθενών σε απομονωμένες περιοχές. Η εφαρμογή αυτή κάνει χρήση αισθητήρων για την συλλογή και αποθήκευση των μετρήσεων του ασθενή. Εν συνεχεία τα μέσα μαζικής μεταφοράς, τα οποία ενσωματώνουν ένα δίκτυο επικοινωνίας DTN, μεταφέρουν τα δεδομένα σε κέντρα υγείας ή νοσοκομεία.

Ένας άλλος τρόπος σύνδεσης των IoT με το DTN είναι με βάση τις **κοινωνικές σχέσεις** που αναπτύσσουν οι άνθρωποι. Σε ένα κλασικό δίκτυο IoT, σκοπός είναι να συνδεθούν όλες οι συσκευές μεταξύ τους κατά τον ίδιο τρόπο. Πολλές φορές αυτό όμως δεν είναι δυνατό λόγω της δυναμικότητας του δικτύου. Μελέτες [49] προσπαθούν να εκμεταλλευτούν τις κοινωνικές επαφές και τα μοτίβα που παρουσιάζει η ανθρώπινη επικοινωνία για την καλύτερη ανταλλαγή δεδομένων των IoT συσκευών, χρησιμοποιώντας την τεχνολογία του DTN.

Συνοψίζοντας υπάρχουν ποικίλες εφαρμογές της αρχιτεκτονικής IoT σε DTN. Οι εφαρμογές αυτές διαφέρουν ως προς τον σκοπό τους, αλλά έχουν σαν κοινό ότι χρησιμοποιούν παρεμφερές hardware και βασίζονται στην ίδια αρχή λειτουργίας. Μετά την διαμόρφωση του σεναρίου και το πως θα γίνει η χρήση των κόμβων, σημαντική πρόκληση αποτελεί η επιλογή των σωστών εργαλείων και συσκευών για την υλοποίηση της ιδέας.

Κεφάλαιο 4: Εργαλεία

Η τεχνολογία του DTN παρουσιάζει ιδιαίτερο ερευνητικό ενδιαφέρον. Κατά καιρούς έχει αναπτυχθεί πληθώρα υλοποιήσεων, ανάλογα με τις ανάγκες και τις τοπολογίες της μελέτης, για την αξιολόγηση και υλοποίηση του DTN και του BP. Αυτές οι υλοποιήσεις μπορεί να διαφέρουν στην γλώσσα προγραμματισμού, στις υποστηριζόμενες πλατφόρμες ή την έκδοση του BP που βασίζονται.

4.1 Υπάρχουσες υλοποιήσεις

Το **DTN2** [50] είναι η βασική υλοποίηση αναφοράς του BP που έχει αναπτυχθεί από την ομάδα IETF DTN. Είναι μια υλοποίηση που καλύπτει τα βασικά χαρακτηριστικά του προτύπου. Μπορεί να διαμορφωθεί και να διαχειρίζεται από μια κονσόλα TCL και αρχεία διαμόρφωσης. Το DTN2 διαθέτει δύο διαφορετικές ενσωματωμένες μονάδες αποθήκευσης bundle: Αποθηκευτικό χώρο που βασίζεται στη μνήμη και χώρο αποθήκευσης που βασίζεται σε σκληρό δίσκο. Υποστηρίζει σαν CLA τα πρότυπα TCP, UDP και Ethernet. Δεν παρέχει επέκταση ασφαλείας. Σαν πρωτόκολλο δρομολόγησης υποστηρίζει DTLSR και PROPHET. Πλέον δεν είναι αρκετά ενημερωμένο και βασίζεται στο RFC 4838 [51], επομένως δεν ενδείκνυται για σύγχρονη έρευνα.

Το **ION** (Interplanetary Overlay Network) [52] είναι μια υλοποίηση του BP που κυρίως στοχεύει στις διαπλανητικές αποστολές. Αναπτύχθηκε από την ομάδα του Jet Propulsion Laboratory (JPL), για να “καλύψει” τα κενά του DTN2 σε σχέση με τους περιορισμούς που έχουν οι διαστημικές αποστολές και τις κάνουν να διαφέρουν από αυτούς που χαρακτηρίζουν τις επίγειες επικοινωνίες. Κύριος σκοπός σχεδίασής του είναι για να λειτουργεί σε ρομποτικά διαστημόπλοια σε λειτουργικά συστήματα σε πραγματικό χρόνο (RTOS, Real-time operating systems), αλλά μπορεί να λειτουργήσει κανονικά και σε συστήματα x86 Linux. Το ION υποστηρίζει custody transfer, έλεγχο συμφόρησης και αναφορά παράδοσης. Χρησιμοποιεί το TCP σαν CLA. Το ION, καθώς δεν είναι ο κύριος σκοπός του τα οπιορτουριστικά δίκτυα, βασίζεται σε προγραμματισμένες ασύρματες επαφές αντί να υποστηρίζει τον μηχανισμό ανακάλυψης γειτονικών κόμβων (Peer Discovery). Έτσι το ION έχει μια ντετερμινιστική προσέγγιση ως προς την επικοινωνία και εκμεταλλεύεται τις προκαθορισμένες ευκαιρίες επικοινωνίας σύμφωνα με την τροχιά γύρω από την Γη [53]. Από την στιγμή που δεν υποστηρίζεται η ανακάλυψη γειτονικών κόμβων η συγκεκριμένη υλοποίηση κρίνεται ακατάλληλη για σενάρια IoT, όπου οι ευκαιρίες επικοινωνίας δεν ακολουθούν πάντα κάποιο μοτίβο. Υπάρχει η επιλογή του BP7 ή του BP6.

Το **IBR-DTN** [54] είναι μια ελαφριά υλοποίηση του DTN, που ενδείκνυται για επίγεια χρήση και ενσωματωμένα συστήματα. Η υλοποίηση απαιτεί λειτουργικό σύστημα βασισμένο στο Linux, με κύρια έμφαση στο OpenWRT. Είναι γραμμένο σε γλώσσα C. Υποστηρίζει διάφορα πρωτόκολλα δρομολόγησης όπως Static, Epidemic και υποστηρίζει τον μηχανισμό εξερεύνησης γειτονικών κόμβων [55]. Τα bundle αποθηκεύονται ανάλογα με την περίπτωση είτε στην μνήμη ή σε αποθηκευτικό χώρο, όπως ο σκληρός δίσκος [56]. Τα CLA που υποστηρίζονται είναι: TCPCL, UDPCL, HTTPCL και

LowPANCL. Το IBR-DTN υλοποιεί το BP6.

Το **μPCN** (Micro Planetary Communication Network) [57] έχει δημιουργηθεί έτσι ώστε να εφαρμόσει την ιδέα του Ring Road. Η βασική ιδέα του Ring Road είναι να παρέχει μια υποδομή διαδικτύου βασισμένη στο Δορυφορικό δίκτυο Χαμηλής τροχιά της γης (Low-Earth Orbit, LEO), για να συνδέσει διαφορετικές περιοχές του πλανήτη. Αυτό επιτυγχάνεται κάνοντας χρήση δορυφόρων χαμηλού κόστους, που ενσωματώνουν κάποιον μικροελεγκτή χαμηλής ενέργειας. Η υλοποίηση είναι συμβατή με μικροελεγκτές της σειράς ARM Cortex STM32F4. Υποστηρίζεται η ανακάλυψη γειτονικών κόμβων με την χρήση Beacon. Είναι γραμμένο σε γλώσσα C, λαμβάνοντας υπόψιν την περιορισμένη μνήμη και επεξεργαστική ισχύ ενός μικροελεγκτή. Υλοποιεί το BP6, καθώς και το BP7 αλλά όχι την πιο ανανεωμένη έκδοσή του.

Το **Terra** [58] είναι μια υλοποίηση χαμηλών απαιτήσεων για το BP7. Υποστηρίζεται από το λειτουργικό σύστημα Linux και είναι γραμμένο σε Java. Σαν CLA χρησιμοποιεί το STCP. Η δοκιμές σχετικά με αυτή την υλοποίηση είναι ελάχιστες. Το Terra υλοποιεί το ζητούμενο BP7, αλλά μια παλαιότερη έκδοσή του.

Το **Forban** [59] [60] είναι εφαρμογή για διαμοιρασμό δεδομένων από χρήστη σε χρήστη (peer-to-peer). Το Forban χρησιμοποιεί πρωτόκολλα σύνδεσης όπως το IP and HTTP για να μεταφέρει τα αρχεία με μια προσέγγιση ανεκτική στις καθυστερήσεις. Το Forban λειτουργεί αυτόνομα σε τοπικά δίκτυα, χρησιμοποιώντας τις δυνατότητές τους όπως ανακάλυψη, αναζήτηση και διαμοιρασμός αρχείων. Αποτελεί ένα οπportunιστικό δίκτυο. Έχει δύο λειτουργίες για τους κόμβους του opportunistic and shared. Το opportunistic είναι σαν μια epidemic δρομολόγηση για την αντιγραφή της πληροφορίας από έναν κόμβο σε ένα άλλον. Το Forban έχει το δικό του πρωτόκολλο επικοινωνίας που δεν είναι συμβατό με το BP [13].

Το **Serval** [61] επικεντρώνεται στην κινητικότητα των κόμβων παρέχοντας υλοποιήσεις που εκτελούνται σε smartphone. Κύριο σκοπό έχει την λειτουργία σε καταστάσεις ακραίων φυσικών φαινομένων, καθώς και σε δυσπρόσιτες περιοχές με αδύναμη κάλυψη διαδικτύου. Το Serval εκτελείται από smartphones με Android σαν λειτουργικό σύστημα. Το πρόγραμμα αυτό δημιουργεί τα λεγόμενα “Serval Mesh”, που είναι κινητά δίκτυα Ad Hoc (Mobile Ad Hoc Network, MANET), και επιτρέπει τα smartphones-κόμβους να επικοινωνούν μεταξύ τους. Το Serval έχει και αυτό δικό του πρωτόκολλο, άρα δεν είναι συμβατό με το BP [13].

Το **μD3TN** [62] είναι μια υλοποίηση δοκιμασμένη σε καταστάσεις διαδικτύου. Υποστηρίζεται από λειτουργικά συστήματα POSIX, Linux και STM32 μικροελεγκτές με FreeRTOS. Για την εγκατάσταση της υλοποίησης απαραίτητα είναι τα πακέτα για C και Python3 (για τα εργαλεία που περιλαμβάνει η υλοποίηση). Σαν CLA υποστηρίζει το MTCP, TCPCv3 και CCSDS Space Packet Protocol. Υλοποιεί τόσο το BP6, όσο και την τελευταία έκδοση του BP7. Στο μD3TN, περιλαμβάνεται το pyD3TN μια υλοποίηση BP γραμμένη σε python, που πολλές φορές αναφέρεται μόνο του στην βιβλιογραφία.

Το **dtm7-kotlin** [63] Το dtm7-kotlin είναι η νεότερη υλοποίηση του DTN7. Υλοποιεί το BP7 καθώς και ενσωματώνει το Bundle Protocol Security Specification [16]. Είναι επεκτάσιμο και είναι συμβατό με την σουίτα brn7 του dtm7-go. Το dtm7-kotlin είναι αρκετά δοκιμασμένο κατά την ανάπτυξη του, αλλά δεν

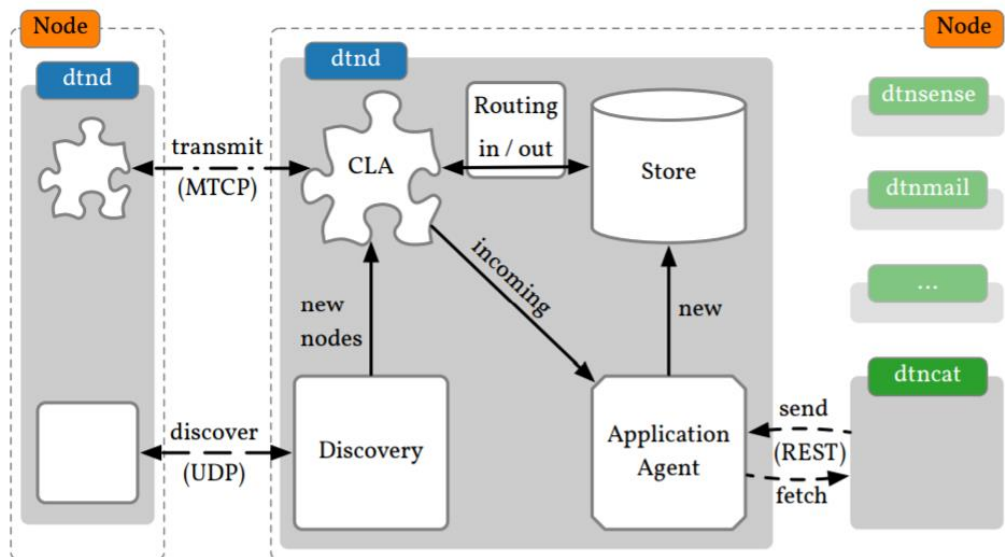
υπάρχει ακόμα επαρκής μελέτη για την αποδοτικότητα του σε σχέση με τις υλοποιήσεις του BP6 ή του BP7.

Το **dtm7-rs** [64] [65] είναι μια από τις πιο πρόσφατες και ενημερωμένες υλοποιήσεις του DTN. Η υλοποίηση στοχεύει στις μικρές συσκευές και για αυτό και για λόγους ασφαλείας είναι γραμμένη στην γλώσσα Rust, που είναι ασφαλής επιλογή στο κομμάτι της μνήμης και δεν επιφέρει μεγάλο πρόσθετο υπολογιστικό βάρος. Απαιτείται το πακέτο “cargo” για την εγκατάσταση από τον πηγαίο κώδικά της. Σαν CLA υποστηρίζει TCPV4, MTCP και HTTP. Σαν πρωτόκολλα δρομολόγησης υποστηρίζει Epidemic, Flooding και Sink. Είναι επεκτάσιμο και διανέμεται ξεχωριστά η υλοποίηση του BP7 που χρησιμοποιεί για την κωδικοποίηση και αποκωδικοποίηση των bundles. Επιπρόσθετα υπάρχει μια επέκταση για το DTN και την βιβλιοθήκη του client το **bp7-plus-rs** [66]. Επιπλέον έχουν γίνει πετυχημένες δοκιμές χρησιμοποιώντας το **dtm7-rs** και το πρωτόκολλο LoRA [67] [68]. Το **dtm7-rs** υλοποιεί το πιο πρόσφατο BP7.

4.2 dtm7-go

Ως βασικό εργαλείο για την παρούσα εργασίας επιλέχθηκε το **dtm7-go** [69], και πιο συγκεκριμένα η έκδοση 0.7.1. Η επιλογή του έγινε με κριτήριο ότι υποστηρίζει το νέο BP7, και την χρονική περίοδο που έγινε η επιλογή του φαινόταν ως η υλοποίηση του BP7 με την πιο ενεργή υποστήριξη και ερευνητική δραστηριότητα, καθώς και ότι ενσωμάτωνε τα περισσότερα χαρακτηριστικά (όπως πληθώρα CLA και πρωτόκολλων δρομολόγησης). Όπως γίνεται αντιληπτό από το όνομα η υλοποίηση είναι γραμμένη στην γλώσσα Go [70], η οποία προσφέρει μια ευρεία βιβλιοθήκη και επιτρέπει την εύκολη μεταγλώττιση (compile) σε διάφορα λειτουργικά συστήματα και αρχιτεκτονικές επεξεργαστών. Το **dtm7-go** υποστηρίζεται από την πλειονότητα των λειτουργικών συστημάτων (όπως Windows, Mac OS και Linux) και των αρχιτεκτονικών hardware (όπως x86, ARM, και MIPS) [13]. Το **dtm7-go** υποστηρίζει τα πιο δημοφιλή πρωτόκολλα δρομολόγησης, όπως DTLSR, Epidemic, Spray and Wait και PROPHET. Υποστηρίζονται τα CLA: MTCP, TCPv4 και το Bundle Broadcasting Connector, ένα CLA για την διασύνδεση με πρωτόκολλο LoRa [68] χρησιμοποιώντας το modem rf95. Για να υποστηριχθεί η ανταλλαξιμότητα των επιμέρους στοιχείων του DTN7, όπως ορίζεται από το DTN, η υλοποίηση δομήθηκε με bundles και τα αντίστοιχα τους πρωτόκολλα: για αποθήκευση, CLA, Ανακάλυψη Γειτονικών Κόμβων (Peer Discovery), Application Agent, Routing και το Βασικό (Core) πακέτο που απαιτείται για τη σύνδεση των επιμέρους πακέτων. Επιπρόσθετα, υπάρχει η δυνατότητα για μια ανεξάρτητη εφαρμογή να χρησιμοποιήσει μέρη του DTN7 σαν βιβλιοθήκη, π.χ. για την δημιουργία bundle, χρησιμοποιώντας το αντίστοιχο πακέτο. Έτσι το **dtm7-go** μπορεί να είναι συμβατό με άλλες υλοποιήσεις, όπως του kotlin.

Στην Εικόνα 9 φαίνονται τα συστατικά στοιχεία του DTN7 και πως αλληλεπιδρούν. Διακρίνεται η σύνδεση μεταξύ των κόμβων. Ο μηχανισμός του Peer Discovery γνωστοποιεί την ύπαρξη του κόμβου στο δίκτυο και συνέχεια ανιχνεύει για πιθανούς γειτονικούς κόμβους.



Εικόνα 9. Αρχιτεκτονική και ροή δεδομένων στον DTN7 [13]

Τα κύρια εργαλεία που παρέχει το dtn7-go είναι:

- i. Το dtnd, που είναι ο DTN daemon για την αποθήκευση και ανταλλαγή bundles και την διασύνδεση με τις εφαρμογές. Ο daemon είναι αυτός που αντιστοιχεί τον κόμβο στο δίκτυο. Χρησιμοποιεί ένα εξωτερικό αρχείο toml, εντός του οποίου περιγράφονται ρυθμίσεις όπως: ο φάκελος αποθήκευσης των bundles, το ID του κόμβου, τον χρόνο που μεσολαβεί μεταξύ των μηνυμάτων, τις ρυθμίσεις για την σύνδεση στον διακομιστή, το CLA, το πρωτόκολλο δρομολόγησης κ.α. Οι γειτονικοί κόμβοι είτε προκαθορίζονται στο toml αρχείο ρυθμίσεων είτε μπορούν να ανιχνευτούν μέσω του Peer Discovery.
- ii. Το dtn-tool, που συντίθεται από τα επιμέρους πρόσθετα εργαλεία: create, exchange, show και ring (στην νεότερη έκδοση). Το create δημιουργεί ένα bundle από τον αποστολέα με προορισμό τον παραλήπτη, σαν περιεχόμενο έχει το αρχείο που ορίζεται από τον χρήστη και σαν όνομα αρχείου αυτό που δίνει ο χρήστης ή τυχαίο δεκαεξαδικό σε περίπτωση που δεν οριστεί όνομα. Το exchange που είναι υπεύθυνο για την επικοινωνία με το δίκτυο, ορίζεται σαν agent στο websocket και αποθηκεύει τα λαμβανόμενα bundles σε έναν τοπικό φάκελο και μεταφέρει στο δίκτυο τα bundles που δημιουργούνται από τον κόμβο. Το ring στέλνει συνεχόμενα bundles από τον αποστολέα στον παραλήπτη. Το show εμφανίζει τις πληροφορίες του bundle, όπως τον προορισμό, τον αποστολέα, την διάρκεια ζωής του bundle, την χρονική σήμανση της δημιουργίας του bundle κ.α.

```
michmart@MichalisBook: /mnt/c/Users/micha/dtn7-go$ ./dtn-tool show 64746e3a2f2f53454e44455
22f2d3639323134323736313332302d30
{"primaryBlock":{"bundleControlFlags":["REQUESTED_DELIVERY_STATUS_REPORT"],"destination":
"dtn://RECEIVER/", "source":"dtn://SENDER/", "reportTo":"dtn://SENDER/", "creationTimestamp"
:{"date":"2021-12-06 21:52:41.320", "sequenceNo":0, "lifetime":86400000}, "canonicalBlocks"
:[{"blockNumber":2, "blockTypeCode":10, "blockType":"Hop Count Block", "blockControlFlags":[
"REPLICATE_BLOCK"], "data":{"limit":64, "count":0}}, {"blockNumber":1, "blockTypeCode":1, "blo
ckType":"Payload Block", "blockControlFlags":null, "data":"dGhpcyBpcyBhIHRLc3QK"}]}
```

Εικόνα 10. Bundle από dtn7-go

Κεφάλαιο 5: Μεθοδολογία

5.1 Σενάρια

Αυτή η εργασία αφορά την μελέτη και την απόδοση της έβδομης και νεότερης έκδοσης του Bundle Protocol. Πιο συγκεκριμένα την εφαρμογή της σε τοπολογία IoT από άκρο σε άκρο, όπου οι κόμβοι δεν έχουν άμεση επικοινωνία και η μεταφορά των δεδομένων γίνεται μέσω ενδιάμεσων DTN κόμβων. Για τις ανάγκες της παρούσας εργασίας χρησιμοποιήθηκαν τρεις DTN κόμβοι. Ένας Αποστολέας, ένας Παραλήπτης και ένας Ενδιάμεσος.

Ο Αποστολέας ζητάει μια συγκεκριμένη μέτρηση-πληροφορία στέλνοντας ένα bundle-request. Ο Παραλήπτης ελέγχει το περιεχόμενο του bundle που λαμβάνει, αν είναι κάποια πληροφορία που έχει στην μνήμη του και άρα μπορεί να εξυπηρετήσει, στέλνει ένα bundle απάντηση στο ενδιαφέρον του Αποστολέα με το περιεχόμενο που ζητήθηκε. Οι δύο κόμβοι βρίσκονται σε κοντινή απόσταση, άρα έχουν επικοινωνία μεταξύ τους όταν είναι και οι δύο ενεργοί. Ο Ενδιάμεσος κόμβος είναι πάντα ενεργός, έτσι αναλαμβάνει να αποθηκεύσει και να προωθήσει τα μηνύματα σε περίπτωση που κάποιος από τους άλλους δύο κόμβους δεν είναι ενεργός για ορισμένο χρονικό διάστημα.

Για την διεξαγωγή των πειραμάτων, στον Ενδιάμεσο κόμβο το μόνο που εκτελείται είναι ο daemon του dtn7-go. Στον κόμβο του Αποστολέα εκτελείται ένα bash script, το οποίο είναι υπεύθυνο για την διαδικασία της αυτόματης ενεργοποίησης και απενεργοποίησης του daemon (dtn7d), προσομοιώνοντας την εναλλαγή από λειτουργία σε αδράνεια μιας συσκευής IoT. Ακόμα το bash script του Αποστολέα είναι υπεύθυνο για τον ορισμό ενός exchange agent (dtn-tool exchange) για την ανταλλαγή bundles και την δημιουργία bundles (dtn-tool create). Ένα αντίστοιχο bash script τρέχει και στον Παραλήπτη, το οποίο αντίστοιχα ενεργοποιεί και απενεργοποιεί τον daemon, και δημιουργεί bundles απαντήσεις σε αυτά που λαμβάνει από τον Αποστολέα. Επιπρόσθετα, καθώς δεν υπάρχει κάποιο dissector για το WireShark που να αναγνωρίζει τα πακέτα που δημιουργούνται από το Bundle Protocol 7, στον Αποστολέα και στον Παραλήπτη τρέχουν κάποια επιπλέον bash scripts τα οποία αναγνωρίζουν το είδος του bundle και σε ένα αρχείο κειμένου μορφής txt γράφουν το όνομα του bundle και την χρονική στιγμή που παρατηρήθηκε στον κόμβο, για την εξαγωγή των μετρικών. Για την αποκωδικοποίηση των bundles χρησιμοποιήθηκε το εργαλείο jq [71].

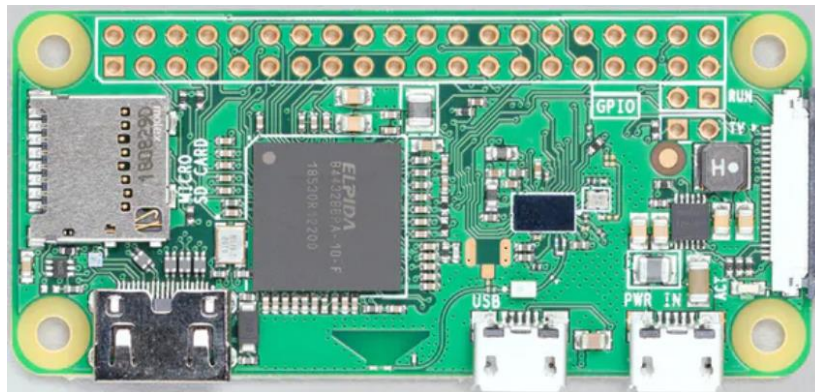
5.2 Τοπολογία Δικτύου

Για την υλοποίηση του παραπάνω σεναρίου σημαντικός παράγοντας ήταν η επιλογή των συσκευών που θα έχουν τον ρόλο των κόμβων. Για να είναι ρεαλιστικό το πείραμα επιλέχθηκε η χρήση του πρότυπου ασύρματης δικτύωσης IEEE 802.11, καθώς είναι ένα πρότυπο που χρησιμοποιείται ευρέως από συσκευές όπως κινητά τηλέφωνα, εκτυπωτές, φορητούς υπολογιστές για την σύνδεση στο Διαδίκτυο.

Ακόμα, ένα κριτήριο κατά την επιλογή των συσκευών ήταν να είναι όσο το δυνατόν προσιτά προς την ευρεία επιστημονική κοινότητα. Για τα πειράματα υλοποιήθηκαν τα εξής είδη κόμβων:

I. Αποστολέας και Παραλήπτης.

Για την υλοποίηση των κόμβων Αποστολέα και Παραλήπτη χρησιμοποιήθηκαν οι συσκευές SBC Raspberry Pi Zero W. Η επιλογή αυτή έγινε με γνώμονα το χαμηλό κόστος, το μικρό μέγεθος, την χαμηλή κατανάλωση και το ενσωματωμένο κύκλωμα διασύνδεσης (BCM43143), έτσι αντιπροσωπεύουν ικανοποιητικά τις συσκευές IoT περιορισμένων δυνατοτήτων, όπως αισθητήρες. Το Raspberry Pi Zero W διαθέτει επεξεργαστική ισχύ 1GHz αρχιτεκτονικής ARM11 και αποτελείται από το SoC Broadcom BCM2835, 512MB μνήμης RAM και αποθηκευτικό χώρο με χρήση κάρτας μνήμης Micro SD. Το Raspberry Pi Zero W τρέχει την πλήρη έκδοση λειτουργικού της εταιρείας “Raspberry Pi OS”, το οποίο είναι βασισμένο στο Debian. Το λειτουργικό σύστημα υποστηρίζει πλήρως την γλώσσα προγραμματισμού Go, επομένως και το dtn7-go.

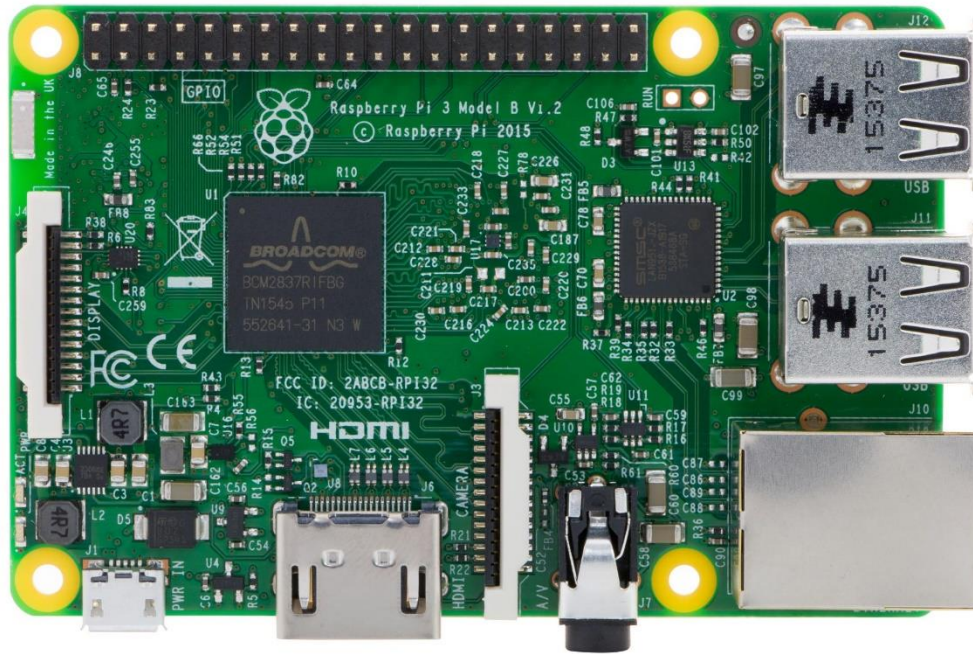


Εικόνα 11. Raspberry Pi Zero W [72]

II. Ενδιάμεσος Κόμβος.

Σαν Ενδιάμεσος κόμβος χρησιμοποιήθηκε η συσκευή SBC Raspberry Pi 3 model B. Χρησιμοποιήθηκε το Raspberry Pi 3 model B έναντι του Raspberry Pi Zero W, λόγω της μεγαλύτερης επεξεργαστικής ισχύς και μεγέθους μνήμης που διαθέτει, έτσι δεν θα έχει πρόβλημα να αποθηκεύσει και να προωθήσει μεγάλο αριθμό bundles αν αυτό χρειαστεί. Το Raspberry Pi 3 model B διαθέτει έναν επεξεργαστή τεσσάρων πυρήνων Broadcom BCM2837 64bit στα 1.2GHz, μνήμη RAM 1GB, το chip BCM43438 για σύνδεση μέσω

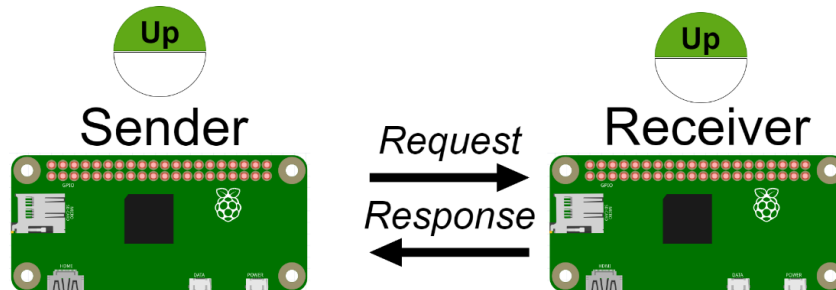
WIFI ή Bluetooth και αποθηκευτικό χώρο με χρήση κάρτας μνήμης Micro SD. Όπως και στο Raspberry Pi Zero W, στο Raspberry Pi 3 model B εγκαταστάθηκε η τελευταία έκδοση του “Raspberry Pi OS”.



Εικόνα 12. Raspberry Pi 3 Model B [72]

5.3 Ενδεικτικά Σενάρια

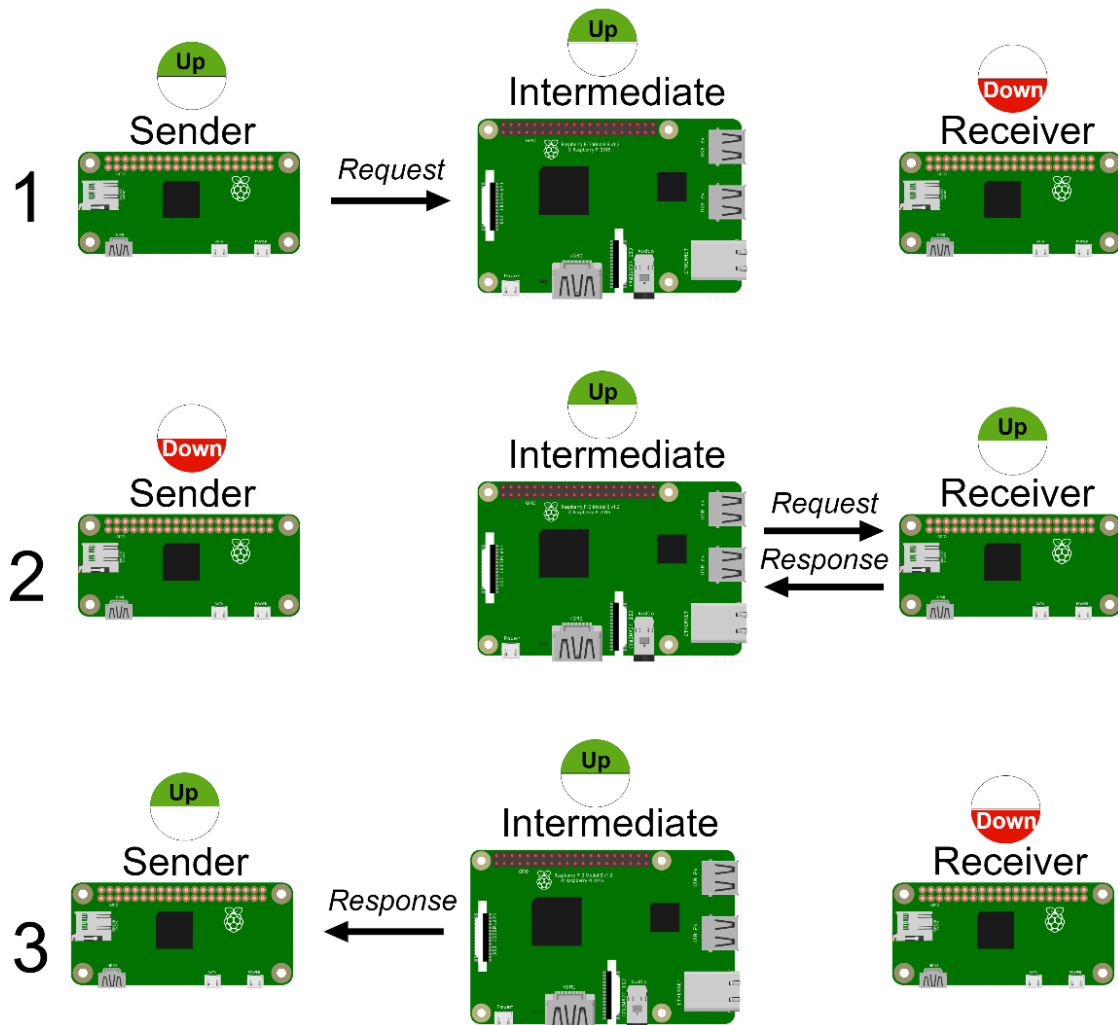
Στην Εικόνα 13 απεικονίζεται ένα ενδεικτικό σενάριο ανταλλαγής μηνυμάτων μεταξύ Αποστολέα και Παραλήπτη. Οι δύο κόμβοι είναι σε άμεση επαφή και ενεργοί. Ο Αποστολέας στέλνει ένα bundle-request, ο Παραλήπτης έχει το περιεχόμενο που ζητήθηκε και ανταποκρίνεται με ένα bundle-response.



Εικόνα 13. Άμεση επικοινωνία των 2 κόμβων

Στην Εικόνα 14 απεικονίζεται ένα άλλο σενάριο που πάλι σκοπός είναι η ανταλλαγή μηνυμάτων μεταξύ Αποστολέα και Παραλήπτη. Σε αυτό το σενάριο ο Παραλήπτης και ο Αποστολέας δεν είναι ποτέ το ίδιο χρονικό διάστημα ενεργοί. Έτσι αρχικά ο Αποστολέας δημιουργεί ένα bundle, δεν βρίσκει όμως

τον Παραλήπτη για να παραδώσει το μήνυμα, έτσι το μήνυμα προωθείται στον Ενδιάμεσο κόμβο. Εν συνεχεία απενεργοποιείται ο Αποστολέας, σε κάποια μετέπειτα χρονική στιγμή ο Παραλήπτης ενεργοποιείται και ο Ενδιάμεσος κόμβος του προωθεί το bundle. Ο Παραλήπτης δημιουργεί άμεσα ένα bundle απάντηση που ως προορισμό έχει τον Αποστολέα, αυτός όμως δεν είναι ενεργός οπότε το bundle προωθείται και αποθηκεύεται στον Ενδιάμεσο κόμβο. Εν συνεχεία απενεργοποιείται ο Παραλήπτης και μετά ξανά ενεργοποιείται ο Αποστολέας. Ο Ενδιάμεσος έχει στην μνήμη του το bundle-response από τον Παραλήπτη όπου και τελικά το προωθεί στον Αποστολέα.



Εικόνα 14. Έμμεση επικοινωνία των 2 κόμβων

Κεφάλαιο 6: Πειραματικό Μέρος

6.1 Εισαγωγή

Για την αξιολόγηση του Bundle Protocol 7 έγινε η επιλογή της πραγματοποίησης μιας σειράς πειραμάτων με κοινή βασική αρχή. Σε όλα τα πειράματα ο Ενδιάμεσος κόμβος είναι πάντα ενεργός. Ο κόμβος του Αποστολέα και του Παραλήπτη εκκινούν και απενεργοποιούν τον daemon τους ανά τακτά διαστήματα, έτσι προσομοιώνεται η συμπεριφορά μιας συσκευής IoT που λειτουργεί για μικρό χρονικό διάστημα και περνάει σε φάση αδράνειας για εξοικονόμηση ενέργειας. Καθώς προχωράνε τα πειράματα αλλάζουν ορισμένες παράμετροι για την βέλτιστη και πιο εκτεταμένη αξιολόγηση του πρωτοκόλλου. Για κάθε πείραμα η διαδικασία ανταλλαγής bundle επαναλαμβάνεται 20 φορές και ορίζεται μια μέση τιμή όλων των μετρήσεων, έτσι υπάρχει μεγαλύτερη ακρίβεια και εγγύτητα στα αποτελέσματα.

Στα Πειράματα 1, 2 και 3 όταν ο Αποστολέας είναι εν ενεργεία, ο Παραλήπτης είναι σε αδράνεια και το αντίθετο. Έτσι σε αυτά τα πειράματα ο Ενδιάμεσος κόμβος έχει τον ρόλο του διαμεσολαβητή για να προωθήσει τα μηνύματα από τον έναν κόμβο στον άλλον, όπως απεικονίζεται στην Εικόνα 14. Ουσιαστικά υλοποιήθηκε μια τοπολογία αλυσίδας, όπου οι κόμβοι είναι συνδεδεμένοι ανά ζεύγη. Ο πρώτος κόμβος στέλνει ένα bundle με προορισμό τον τελευταίο κόμβο της αλυσίδας. Η μέτρηση του συνολικού χρόνου για μια διαδρομή αντιστοιχεί: στην ανάγνωση των δεδομένων, την σειριοποίηση του bundle, την αποστολή στο δίκτυο, την αποσειριοποίηση στον παραλήπτη και την τελική παράδοση στην εφαρμογή.

Στο Πείραμα 4 ο χρόνος είναι τυχαίος παράγοντας, άρα δεν συμβαίνει πάντα η συμπεριφορά που προαναφέρθηκε και μπορεί οι δύο κόμβοι να είναι ταυτόχρονα ενεργοί και να πραγματοποιηθεί άμεση ανταλλαγή όπως στην Εικόνα 13.

Τα πειράματα που αναπτύχθηκαν βρίσκουν εφαρμογή και στον πραγματικό κόσμο, εκτός εργαστηρίου. Τοπολογίες με κόμβους που είναι σε αδράνεια για μεγάλο διάστημα και ενεργοποιούνται μόνο για λίγα δευτερόλεπτα, εφαρμόζονται κατεξοχήν στο IoT για εξοικονόμηση ενέργειας. Ένα παράδειγμα χρήσης θα ήταν η συλλογή δεδομένων από κάμερες για την παρατήρηση της άγριας ζωής σε ένα δάσος, όπου το σύστημα θα τροφοδοτείται με μπαταρίες και επομένως αναγκαία είναι η εξοικονόμηση πόρων με την αποστολή δεδομένων με αραιό ρυθμό, αλλά αποδοτικό.

6.2 Μετρικές

Για την μελέτη της απόδοσης του Bundle Protocol 7 και της υλοποίησης που χρησιμοποιήθηκε εξήφθηκαν ορισμένες μετρικές οι οποίες συμβάλλουν στην καλύτερη αξιολόγηση των αποτελεσμάτων. Οι πιο σημαντικές μετρικές είναι το Delivery Ratio, ο χρόνος καθυστέρησης μέχρι την παράδοση ενός μηνύματος (latency time) και το Overhead Ratio. Οι αναλυτικές μετρήσεις για κάθε πείραμα βρίσκονται στο Κεφάλαιο Παράρτημα.

6.2.1 Delivery Ratio

Το Delivery Ratio είναι η αναλογία μεταξύ των μηνυμάτων που παραδόθηκαν με επιτυχία προς τα συνολικά μηνύματα που δημιουργήθηκαν.

$$Delivery\ Ratio = \frac{\text{μηνύματα που παραδόθηκαν}}{\text{συνολικά μηνύματα που δημιουργήθηκαν}}$$

Εξίσωση (6.1)

Αυτή είναι μια βασική μέτρηση, καθώς ο στόχος κάθε πρωτοκόλλου επικοινωνίας και κάθε πρωτόκολλου δρομολόγησης είναι να μεγιστοποιήσει τα μηνύματα που παραδίδονται. Σε ένα ιδανικό σενάριο, όλα τα μηνύματα που αποστέλλονται παραδίδονται. Στον αριθμό των μηνυμάτων που χρησιμοποιούνται στην Εξίσωση 6.1, δεν υπολογίζονται τα μηνύματα (beacons) που δημιουργούν και ανταλλάσσουν οι κόμβοι μεταξύ τους για την ανίχνευση της τοπολογίας του δικτύου.

6.2.2 Latency Time

Το Latency Time γενικότερα είναι ο χρόνος που μεσολαβεί από τη δημιουργία του μηνύματος μέχρι την επιτυχή παράδοσή του στον τελικό προορισμό. Ειδικότερα στα πειράματα που εκτελέστηκαν για τις ανάγκες της παρούσας εργασίας επειδή αποστέλλεται ένα μήνυμα και δημιουργείται ένα άλλο μήνυμα-απάντηση, σαν μετρική χρησιμοποιείται ο χρόνος μετάβασης και επιστροφής (Round-Trip Time, RTT). Δηλαδή ο χρόνος που μεσολαβεί από τη δημιουργία του μηνύματος από τον Αποστολέα μέχρι την επιτυχή παράδοσή της απάντησης από τον Παραλήπτη στον Αποστολέα.

6.2.3 Overhead Ratio

Το Overhead Ratio υποδεικνύει την περίσσεια των μηνυμάτων που μεταδόθηκαν επιτυχώς σε σύγκριση με τον συνολικό αριθμό των μηνυμάτων που παραδόθηκαν.

$$Overhead\ Ratio = \frac{(\text{μηνύματα που μεταδόθηκαν}) - (\text{μηνύματα που παραδόθηκαν})}{\text{μηνύματα που παραδόθηκαν}}$$

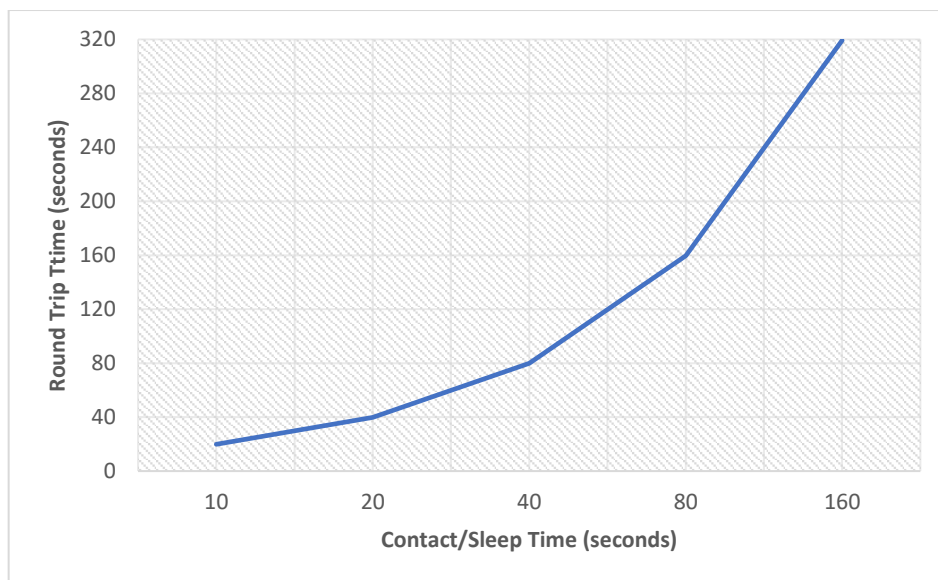
Εξίσωση (6.2)

Ένα μεγάλο Overhead Ratio είναι μια ένδειξη ότι μεταδίδεται υπερβολική ποσότητα μηνυμάτων. Αυτό είναι σημαντικό, καθώς σε ορισμένα πρωτόκολλα δρομολόγησης, αυτό μπορεί να οφείλεται σε

απόρριψη μηνυμάτων (message dropping) λόγω συμφόρησης. Ωστόσο, σε ορισμένα πρωτόκολλα δρομολόγησης, χρησιμοποιούνται επιπλέον μηνύματα για την καλύτερη αντίληψη του δικτύου και των γειτονικών κόμβων, κάτι που εκμεταλλεύονται αυτά τα πρωτόκολλα. Στον αριθμό των μηνυμάτων που χρησιμοποιούνται στην Εξίσωση 6.2, δεν υπολογίζονται τα μηνύματα (beacons) που δημιουργούν και ανταλλάσσουν οι κόμβοι μεταξύ τους για την ανίχνευση της τοπολογίας του δικτύου.

6.3 Πείραμα 1

Στο Πείραμα 1 μεταβάλλεται ο χρόνος που παραμένουν σε λειτουργία ή αδράνεια οι δύο κόμβοι (Αποστολέας και Παραλήπτης). Δηλαδή, μεταβάλλεται ο χρόνος που έχουν ευκαιρία για επίτευξη επικοινωνίας (contact time) όσο είναι ενεργοί και ο χρόνος που βρίσκονται σε αδράνεια (sleep time). Οι τιμές των δύο χρόνων ταυτίζονται στα συγκεκριμένα πειράματα. Η τιμή αυτή ξεκινάει από τα 10 δευτερόλεπτα και σταδιακά διπλασιάζεται κάθε φορά έως τα 160 δευτερόλεπτα. Σαν CLA χρησιμοποιείται το πρωτόκολλο MTCP για τους τρεις κόμβους, επιλογή που έγινε με γνώμονα την απλότητά του. Σαν πρωτόκολλο δρομολόγησης χρησιμοποιείται το Epidemic. Σαν lifetime των bundle ορίζεται η προεπιλεγμένη τιμή των ρυθμίσεων.



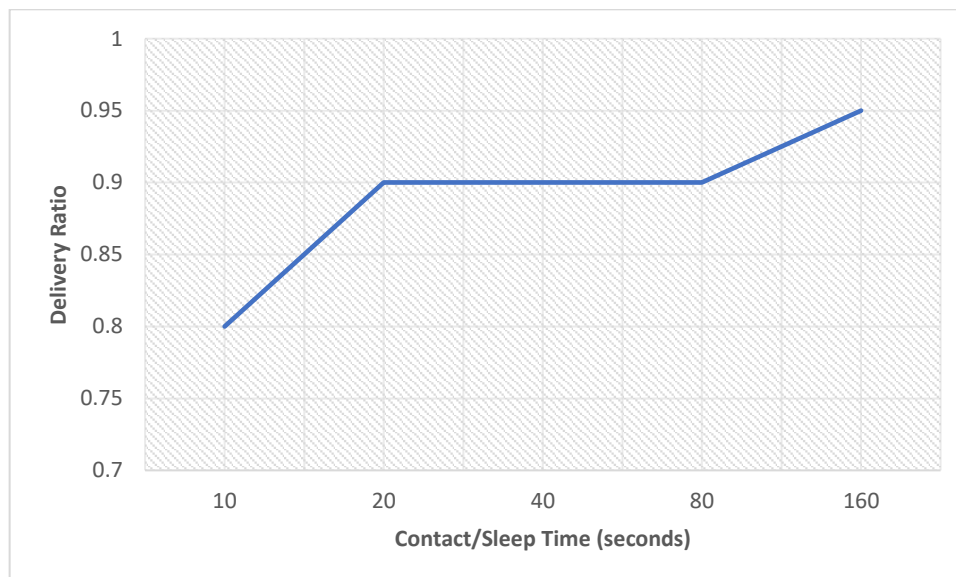
Διάγραμμα 1. Πείραμα 1, RTT-Contact/Sleep Time

Στο Διάγραμμα 1 παρατηρείται η γραμμική εξέλιξη του RTT Time καθώς αυξάνεται ο χρόνος που είναι σε αδράνεια και σε λειτουργία οι δύο κόμβοι. Ο χρόνος μετάβασης και επιστροφής του μηνύματος

είναι περίπου ο διπλάσιος από τον χρόνο αδράνειας. Το αποτέλεσμα αυτό είναι απόλυτα λογικό, καθώς η διαδικασία που ακολουθείται είναι:

- i. Ο Αποστολέας δημιουργεί ένα bundle και προσπαθεί να το προωθήσει στον Παραλήπτη, ο οποίος είναι σε αδράνεια, άρα το bundle προωθείται στον Ενδιάμεσο κόμβο.
- ii. Ο Αποστολέας μετά από x δευτερόλεπτα απενεργοποιεί τον daemon του και ο Παραλήπτης εκκινεί τον δικό του.
- iii. Ο Ενδιάμεσος κόμβος προωθεί στον Παραλήπτη το bundle που έχει αποθηκευμένο στην μνήμη του.
- iv. Ο Παραλήπτης λαμβάνει το μήνυμα και δημιουργεί ένα bundle απάντηση στον Αποστολέα, ο οποίος δεν είναι ενεργός. Άρα το bundle προωθείται και αποθηκεύεται στον Ενδιάμεσο κόμβο.
- v. Μετά από x δευτερόλεπτα ο Παραλήπτης απενεργοποιεί τον daemon του και ο Αποστολέας ενεργοποιεί τον δικό του.
- vi. Τελικά ο Αποστολέας λαμβάνει μέσω του Ενδιάμεσου κόμβου το bundle που δημιούργησε ο Παραλήπτης και ολοκληρώνεται η μεταφορά. Επειδή το μήνυμα έχει μικρό μέγεθος οι μεταφορές είναι σχεδόν άμεσες, οπότε ο συνολικός χρόνος είναι $2x$. Όπου x ο χρόνος που έχει διαθέσιμο για επίτευξη επικοινωνίας ο κόμβος του Αποστολέα.

Η παραπάνω διαδικασία είναι η έμμεση ανταλλαγή Bundle που περιγράφεται στην Ενότητα 5.3 και μπορεί να γίνει καλύτερα αντιληπτή με την Εικόνα 14.



Διάγραμμα 2. Πείραμα 1, Delivery Ratio-Contact/Sleep Time

Στο Διάγραμμα 2 παρατηρείται πως με περισσότερο διαθέσιμο χρόνο αδράνειας των 2 κόμβων επιτυγχάνεται καλύτερο Delivery Ratio. Πρέπει να γίνει αντιληπτό ότι ο χρόνος αδράνειας ισούται και με τον χρόνο που είναι σε λειτουργία ο κόμβος και άρα μπορεί να αναμεταδώσει ή να δεχθεί ένα μήνυμα. Με αυξημένο contact time δίνεται μεγαλύτερη δυνατότητα στους κόμβους να επεξεργαστούν τα bundles που έχουν στην μνήμη. Έτσι είναι πιο αποδοτική η συνολική ανταλλαγή μηνυμάτων καθώς και τα επιμέρους κομμάτια της, δηλαδή από τον Αποστολέα στον Ενδιάμεσο κόμβο και από εκεί στον Παραλήπτη και η αντίστοιχη διαδρομή της απάντησης του Παραλήπτη στον Αποστολέα. Άξια σχολιασμού είναι η διαφορά στο Delivery Ratio μεταξύ του χρόνου 80 και 160 δευτερολέπτων. Η διαφορά αυτή δεν είναι αντιπροσωπευτική για αυτό είναι και μικρή. Για χρόνο 80 και 160 δευτερόλεπτα η διαφορά στο Delivery Ratio θα έπρεπε να είναι μηδαμινή, καθώς τα 80 δευτερόλεπτα επαφής είναι αρκετά για την επεξεργασία και ανταλλαγή μηνυμάτων. Αν το πείραμα επαναλαμβανόταν για αριθμό επαναλήψεων πολύ μεγαλύτερο των 20, το αποτέλεσμα θα ήταν πιο αντιπροσωπευτικό και θα φαινόταν η σύγκλιση των αποτελεσμάτων για τους δύο χρόνους, κάτι που δεν ήταν δυνατό στην παρούσα εργασία λόγω του χρόνου των πειραμάτων και της ανάγκης για συνεχή επίβλεψή τους.

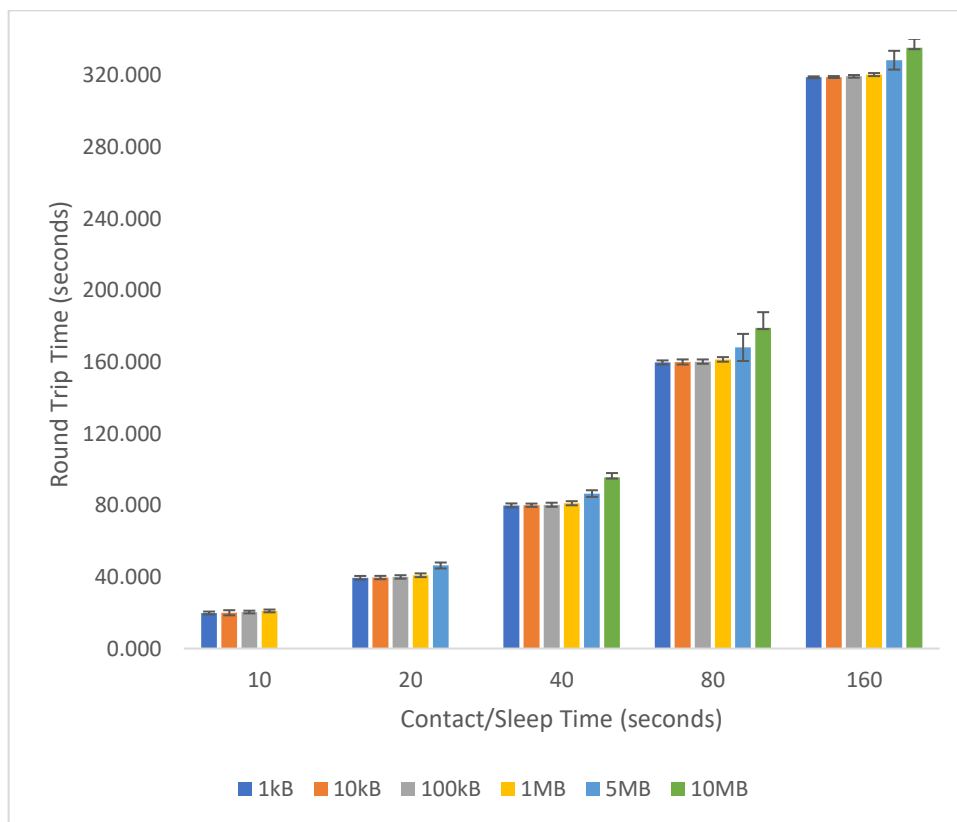
Παρατηρήσεις:

Στο συγκεκριμένο πείραμα δεν εξάχθηκε η μετρική του Overhead Ratio, καθώς τα μηνύματα σε κάθε ανταλλαγή στέλνονται με την βοήθεια του Ενδιάμεσου κόμβου χωρίς να αλλάζει σε κάποια περίπτωση αυτό. Έτσι το Overhead Ratio είναι σταθερό για όλες τις τιμές του Contact/Sleep Time. Το μέγεθος του μηνύματος σε αυτή την περίπτωση είναι 10 bytes.

6.4 Πείραμα 2

Το Πείραμα 2 έχει παρόμοια τοπολογία με το πρώτο, με την διαφορά ότι κάθε φορά διαφοροποιείται το μέγεθος του αρχικού μηνύματος καθώς και του μηνύματος απάντηση, ξεκινώντας από 1kB έως και 10MB. Πάλι ο χρόνος αδρανείας-λειτουργίας του κόμβου του Παραλήπτη και του Αποστολέα εξελίσσεται από 10 δευτερόλεπτα έως 160 δευτερόλεπτα. Οι λοιπές ρυθμίσεις παραμένουν οι ίδιες.

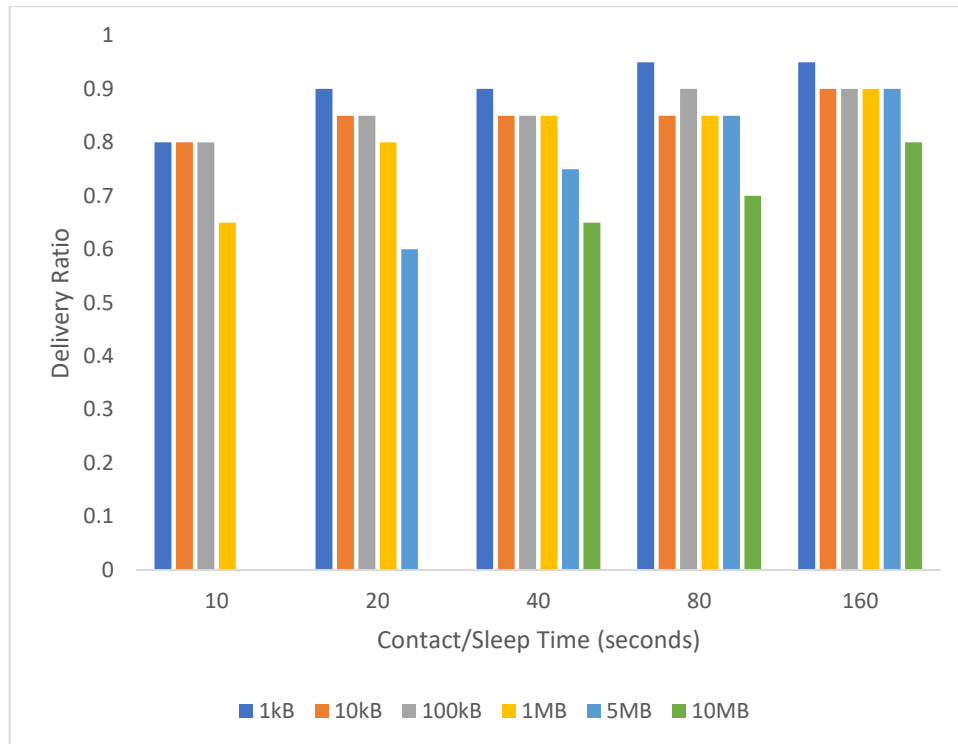
Σκοπός του πειράματος αυτού είναι να ελέγξει τα όρια της υλοποίησης και του BP7 ως προς την αποδοτική αποστολή διάφορων ειδών αρχείων. Το αρχείο μεγέθους 1kB θα μπορούσε να είναι ένα αρχείο κειμένου που αναπαριστά μια καταγραφή εντοπισμού κάποιου άγριου ζώου. Αντίστοιχα το αρχείο 100kB μια πολύ σύντομη καταγραφή κελαδήματος ενός πτηνού, το αρχείο 1MB μια συμπιεσμένη εικόνα και το αρχείο μεγέθους 10MB ένα βίντεο του ζώου.



Διάγραμμα 3. Πείραμα 2, Διαφορετικό Μέγεθος Bundle, RTT-Contact/Sleep Time

Στο Διάγραμμα 3 παρατηρείται ότι όσο μεγαλώνει το μέγεθος του bundle τόσο αυξάνεται το Round

Trip Time, κάτι αναμενόμενο καθώς η διαδικασία δημιουργίας ενός bundle και αποστολής του διαρκεί περισσότερο όταν ο κόμβος έχει να επεξεργαστεί bundle μεγαλύτερου μεγέθους.



Διάγραμμα 4. Πείραμα 2, Διαφορετικό Μέγεθος Bundle, Delivery Ratio-Contact/Sleep Time

Στο Διάγραμμα 4 παρατηρείται πως για όλα τα μεγέθη bundle η συμπεριφορά του πρωτοκόλλου είναι αυτή που παρατηρήθηκε και στο πρώτο πείραμα στο Διάγραμμα 2, δηλαδή με μεγαλύτερο χρόνο αδράνειας και λειτουργίας των δύο κόμβων επιτυγχάνεται καλύτερο Delivery Ratio. Αυτή την φορά η διαφορά στο Delivery Ratio στις μεγαλύτερες τιμές χρόνου, είναι δικαιολογημένη, καθώς πλέον τα αρχεία έχουν υπολογίσιμο μέγεθος και η επεξεργασία τους είναι πιο χρονοβόρα. Ακόμα παρατηρείται πως όσο αυξάνεται το μέγεθος του bundle τόσο μειώνεται το Delivery Ratio, καθώς το δίκτυο επιβαρύνεται περισσότερο με την αποστολή μεγαλύτερων bundle.

Παρατηρήσεις:

Για ορισμένους συνδυασμούς μεγέθους bundle και χρόνου αδράνειας-λειτουργίας των 2 κόμβων, όπως 10 δευτερόλεπτα Contact/Sleep Time και μέγεθος 5MB και 10MB ή 20 δευτερόλεπτα Contact/Sleep Time και μέγεθος 10MB, ο χρόνος μιας σύνδεσης δεν επαρκεί για την ανταλλαγή των απαραίτητων πακέτων. Αυτό συμβαίνει καθώς οι συγκεκριμένοι χρόνοι δεν επαρκούν για την ενθυλάκωση του μηνύματος και την αποστολή του μέσω του Ενδιάμεσου κόμβου.

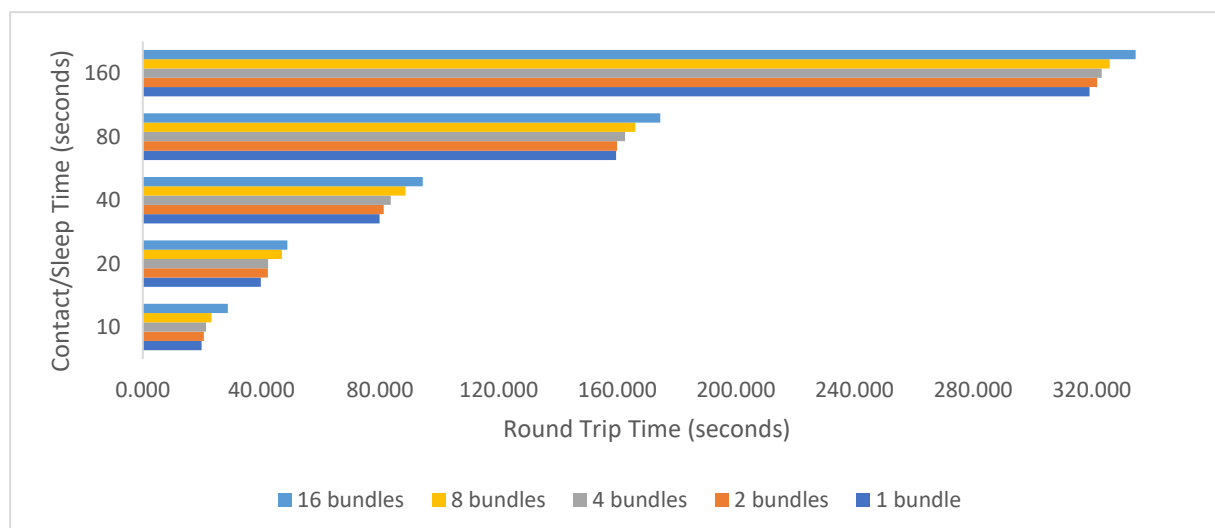
Επιπρόσθετα, όπως και στο πρώτο πείραμα δεν έχει κάποια χρησιμότητα η εξαγωγή της μετρικής του Overhead Ratio για τους ίδιους λόγους.

Τέλος στο συγκεκριμένο πείραμα με την υλοποίηση (dtn7-go) που χρησιμοποιείται για την παρούσα εργασία δεν ήταν δυνατή η εκτέλεση 20 συνεχόμενων μετρήσεων. Έτσι πραγματοποιήθηκε συνδυασμός πλήθους μεμονωμένων μετρήσεων (ομάδες των 4 ή 5 συνεχόμενων μετρήσεων) για να ληφθούν συνολικά 20 μετρήσεις. Αυτό είναι απόρροια μιας ασυμβατότητας μεταξύ της υλοποίησης και του bash script που αναπτύχθηκε στα πλαίσια του συγκεκριμένου πειράματος, που δυστυχώς δεν ήταν δυνατή η επίλυσή της.

6.5 Πείραμα 3

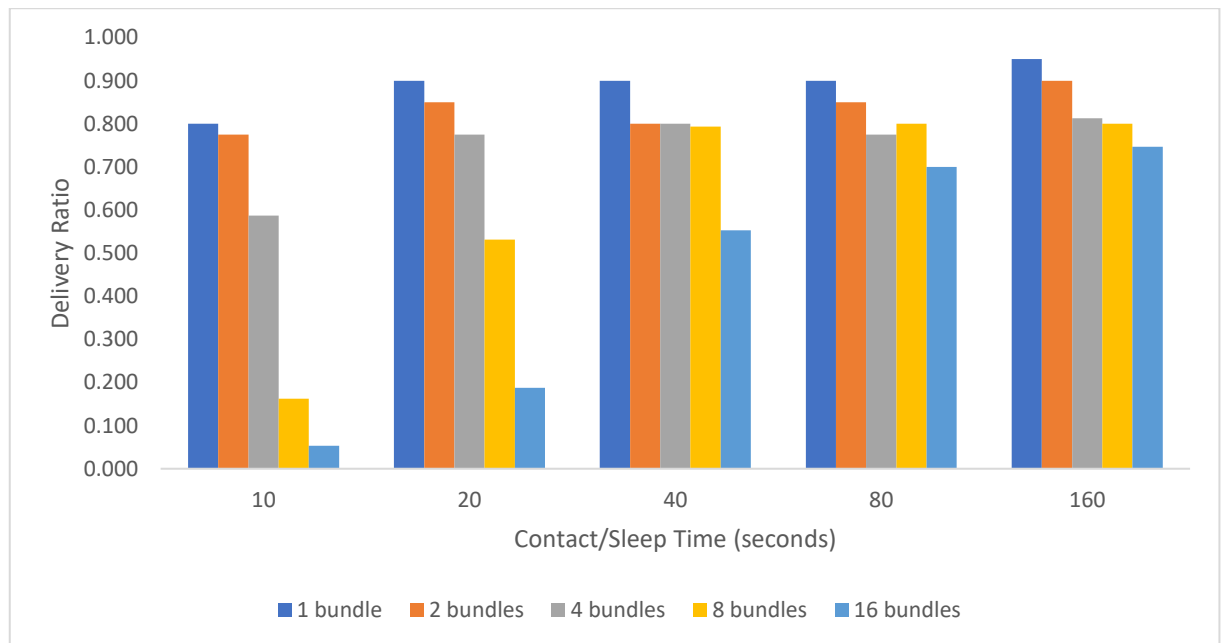
Η διαφοροποίηση του Πειράματος 3 σε σχέση με το πρώτο είναι το πλήθος των bundles που αποστέλλονται και λαμβάνονται σε κάθε ανταλλαγή. Το πείραμα ξεκινάει με την αποστολή ενός μηνύματος και επιστροφής μιας απάντησης και σταδιακά διπλασιάζεται ο αριθμός των μηνυμάτων έως τα 16 μηνύματα ανά αποστολή. Και σε αυτό το πείραμα η ανταλλαγή των x μηνυμάτων ανά φορά γίνεται με την βοήθεια του Ενδιάμεσου κόμβου, καθώς οι δύο κόμβοι δεν είναι ταυτόχρονα σε λειτουργία.

Μια εφαρμογή αυτού του σεναρίου θα μπορούσε να ήταν η ενσωμάτωσή του στο σύστημα ασφαλείας ενός αυτοκινήτου και σε περίπτωση ατυχήματος την αποστολή μηνυμάτων ενημέρωσης σε διάφορους φορείς όπως νοσοκομείο, αστυνομία και ασφαλιστική εταιρεία, αλλά και την ενημέρωση των συστημάτων των υπόλοιπων διερχομένων οχημάτων σχετικά με την κατάσταση. Ακόμα και αν το όχημα δεν έχει πρόσβαση στο δίκτυο άμεσα, με την προώθηση στα υπόλοιπα οχήματα, αυτά θα αναλάβουν την αναμετάδοση στις σχετικές αρχές.



Διάγραμμα 5. Πείραμα 3, Διαφορετικό Πλήθος Bundles, RTT-Contact/Sleep Time

Στο Διάγραμμα 5 παρατηρείται πως αυξάνεται το Round Trip Time καθώς αυξάνεται ο αριθμός των bundles. Λογικό επακόλουθο είναι όταν αποστέλλονται πολλά μηνύματα ταυτοχρόνως να διαρκεί παραπάνω η επεξεργασία τους από τον εκάστοτε κόμβο. Έτσι απαιτείται παραπάνω διάρκεια για την δημιουργία μηνυμάτων-απαντήσεων από τον Παραλήπτη στα αιτήματα του Αποστολέα και διαρκεί παραπάνω η όλη ανταλλαγή.



Διάγραμμα 6. Πείραμα 3, Διαφορετικό Πλήθος Bundles, Delivery Ratio-Contact/Sleep Time

Στο Διάγραμμα 6 παρατηρείται πως μεταβάλλεται το Delivery Ratio με την αποστολή μεγαλύτερου πλήθους bundles. Για ορισμένες περιπτώσεις η ανταλλαγή μηνυμάτων είναι προβληματική για μεγάλο πλήθος μηνυμάτων και μικρό χρόνο λειτουργίας του κόμβου, όπως για Contact/Sleep Time 10 ή 20 δευτερολέπτων και αποστολή 8 ή 16 αρχείων ταυτοχρόνως, όπου οι τιμές του Delivery Ratio που επιτεύχθηκε είναι αρκετά μικρές. Γενικότερα παρατηρείται ότι η υλοποίηση που επιλέχθηκε για την παρούσα εργασία, δεν ανταποκρίνεται ικανοποιητικά σε ανταλλαγή πολλών μηνυμάτων ταυτοχρόνως, αυτό αποτελεί μια αδυναμία της παρούσας υλοποίησης και όχι του DTN γενικότερα.

Παρατηρήσεις:

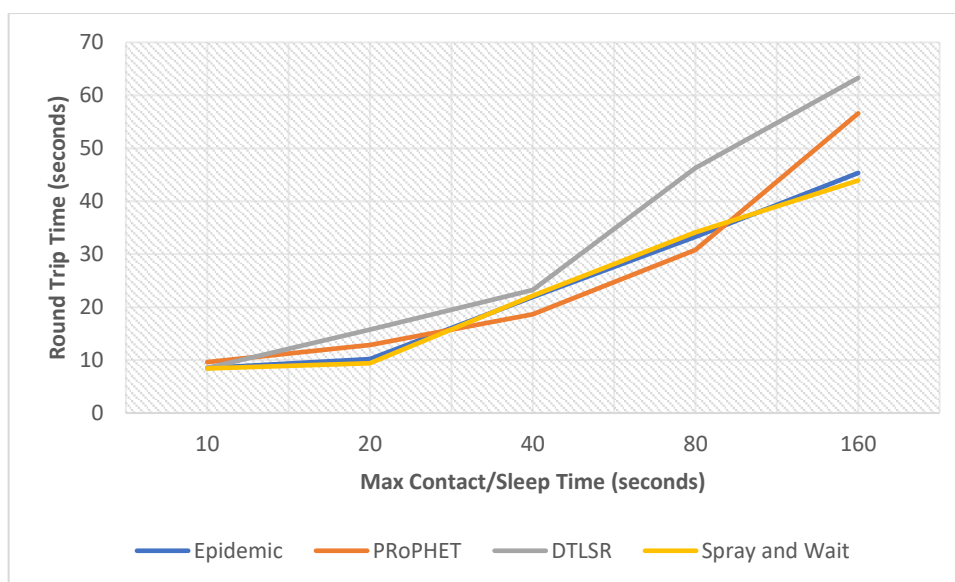
Υπήρχε το πρόβλημα της ασυμβατότητας όπως και στο δεύτερο πείραμα, έτσι για το σύνολο των 20 μετρήσεων λήφθηκε πλήθος μεμονωμένων ομάδων μετρήσεων.

Και σε αυτό το πείραμα δεν έχει κάποια χρησιμότητα η εξαγωγή της μετρικής του Overhead Ratio, καθώς αυτό παραμένει σταθερό.

6.6 Πείραμα 4

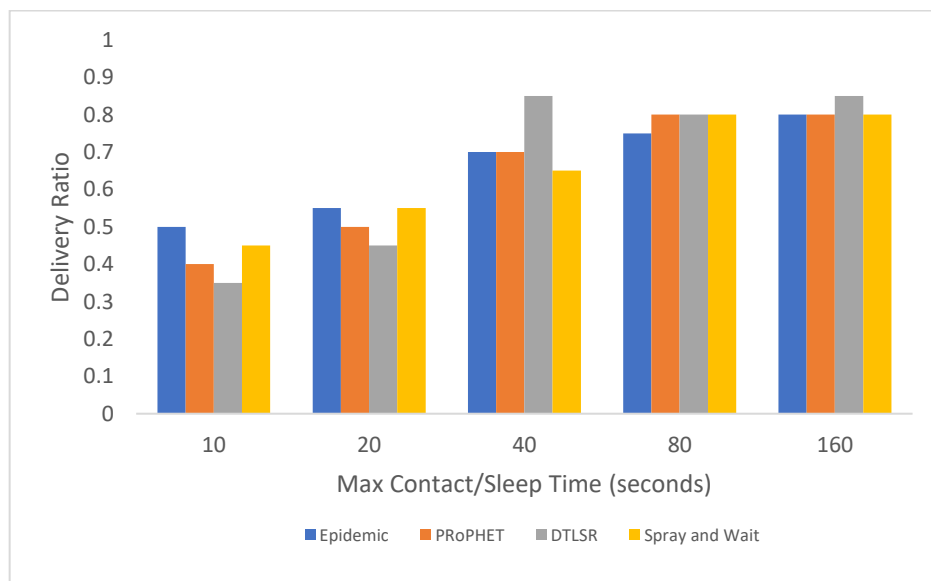
Στο Πείραμα 4 γίνεται σύγκριση μεταξύ διαφορετικών πρωτόκολλων δρομολόγησης (routing protocols). Εν αντιθέσει με τα προηγούμενα πειράματα, ο χρόνος αδρανείας-λειτουργίας των κόμβων Αποστολέα και Παραλήπτη δεν είναι σταθερός, αλλά τυχαίος. Έτσι υπάρχουν περιπτώσεις που οι κόμβοι βρίσκονται σε άμεση επαφή και η ανταλλαγή μηνυμάτων γίνεται ακαριαία, αλλά και περιπτώσεις που οι κόμβοι δεν είναι την ίδια χρονική στιγμή σε λειτουργία και τα μηνύματα προωθούνται έμμεσα μέσω του Ενδιάμεσου κόμβου. Οι ρυθμίσεις είναι ίδιες με αυτές του πρώτου πειράματος, πέρα από το πρωτόκολλο δρομολόγησης που μεταβάλλεται. Για τα διάφορα πρωτόκολλα δρομολόγησης η επιλογή των επιμέρους παραμέτρων έγινε θέτοντας τις προεπιλεγμένες ρυθμίσεις που υπάρχουν στο αρχείο "configuration.toml" που διαθέτουν οι δημιουργοί του dtn7-go.

Σκοπός αυτού του πειράματος είναι η εξέλιξη των προηγούμενων μετρήσεων, κάνοντας τις παραμέτρους του πειράματος ακόμα πιο αντιπροσωπευτικές ως προς τις πραγματικές συνθήκες, έχοντας τυχαίο χρόνο λειτουργίας και αδρανείας. Σε ένα τέτοιο δίκτυο η δρομολόγηση είναι πιο απαιτητική καθώς υπάρχει ποικιλία στην επικοινωνία, οι κόμβοι μπορεί να συναντιούνται συχνά ή και καθόλου. Η πραγματική πρόκληση είναι η επιλογή ενός πρωτόκολλου δρομολόγησης που θα εγγυάται την παράδοση των μηνυμάτων, χωρίς μεγάλη επιβάρυνση του δικτύου.



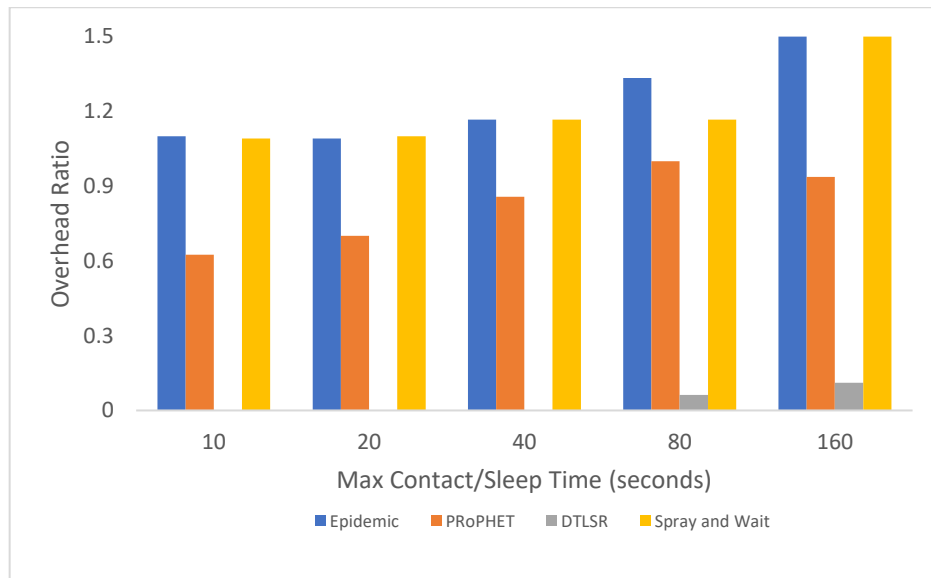
Διάγραμμα 7. Πείραμα 4, Σύγκριση Πρωτόκολλων Δρομολόγησης, RTT-Max Contact/Sleep Time

Στο Διάγραμμα 7 συγκρίνεται το RTT που επιτυγχάνεται με κάθε πρωτόκολλο δρομολόγησης. Παρατηρείται ότι το DTLSR εμφανίζει παραπάνω καθυστέρηση σε σχέση με τα άλλα πρωτόκολλα. Στο DTLSR ο κόμβος του Αποστολέα διατηρεί ένα αντίγραφο του διαγράμματος του δικτύου για να στείλει το bundle χρησιμοποιώντας έναν αλγόριθμο βέλτιστης διαδρομής. Επειδή από τις ρυθμίσεις το πόσο συχνά θα ανανεωθεί αυτό το διάγραμμα, έχει οριστεί στα 30 δευτερόλεπτα, για μικρούς χρόνους contact-sleep το πρωτόκολλο λειτουργεί πρακτικά σαν το Direct Delivery, καθώς ο κόμβος θα βλέπει ότι η διαδρομή προς τον Παραλήπτη είναι ενεργή ακόμα και αν δεν είναι και επομένως δεν θα προωθεί το Bundle στον Ενδιάμεσο κόμβο. Δηλαδή ο κόμβος του Αποστολέα θα κρατήσει το bundle μέχρι να συναντηθεί με τον Παραλήπτη και να του το προωθήσει ο ίδιος, και δεν θα το προωθήσει στον Ενδιάμεσο κόμβο. Επίσης παρατηρείται ότι οι χρόνοι για το Epidemic και το Spray and Wait είναι παρόμοιοι. Αυτό συμβαίνει καθώς λόγω της τοπολογίας που υλοποιείται υπάρχει μόνο ένα μήνυμα και ένας κόμβος που προωθείται, το Spray and Wait πρακτικά περνάει άμεσα από την φάση Spray στην φάση Wait και επομένως η συμπεριφορά του είναι σαν το Epidemic κάνοντας Flooding τα bundles σε όλους τους κόμβους. Επίσης παρατηρείται ότι το PRoPHET έχει μεγαλύτερο Latency από το Epidemic και το Spray and Wait. Αυτό γίνεται καθώς την πρώτη φορά που θα επιχειρήσει να στείλει ένα μήνυμα ο Αποστολέας κατά πάσα πιθανότητα θα είναι και οι δύο κόμβοι (Ενδιάμεσος και Παραλήπτης) ενεργοί, επομένως η τιμή της πιθανότητας παράδοσης θα παραμείνει σταθερή. Όταν επιχειρήσει ο Αποστολέας να στείλει το επόμενο bundle, σε περίπτωση που δει τον Ενδιάμεσο Κόμβο ενεργό, αλλά τον Παραλήπτη μη ενεργό, δεν θα προωθήσει το Bundle στον Ενδιάμεσο κόμβο καθώς έχουν την ίδια τιμή πιθανότητας παράδοσης. Αυτό όμως δεν είναι αποδοτικό στην τοπολογία που χρησιμοποιείται σε αυτό το πείραμα, καθώς αν στην επόμενη χρονική στιγμή τεθεί σε αδράνεια ο Αποστολέας και ενεργοποιηθεί ο Παραλήπτης, δεν θα γίνει η παράδοση του bundle, αφού δεν το έχει ο Ενδιάμεσος κόμβος λόγω της πολιτικής της δρομολόγησης. Στην ίδια περίπτωση με Epidemic δρομολόγηση το Bundle έχει προωθηθεί στον Ενδιάμεσο κόμβο και από εκεί στον Παραλήπτη, για αυτό και η διαφορά στο Round Trip Time.



Διάγραμμα 8. Πείραμα 4, Σύγκριση Πρωτόκολλων Δρομολόγησης, Delivery Ratio-Max Contact/Sleep Time

Στο Διάγραμμα 8 παρατηρείται πως μεταβάλλεται το Delivery Ratio ανάλογα με το πρωτόκολλο δρομολόγησης που επιλέγεται. Παρατηρείται ότι το DTLSR έχει σταθερά μεγαλύτερο Delivery Ratio για μεγαλύτερους χρόνους. Αυτό συμβαίνει καθώς η διάρκεια των πειραμάτων είναι μικρή επομένως ακόμα και με πρωτόκολλο που μοιάζει σε συμπεριφορά με το Direct Delivery, το όριο του TTL δεν συναντάται και διατηρούνται τα bundles. Οι διαφορές των άλλων 3 πρωτοκόλλων δεν είναι σημαντικές.



Διάγραμμα 9. Πείραμα 4, Σύγκριση Πρωτοκόλλων Δρομολόγησης, Overhead Ratio-Max Contact/Sleep Time

Στο Διάγραμμα 9 γίνεται σύγκριση του Overhead Ratio του κάθε πρωτοκόλλου δρομολόγησης. Το DTLSR έχει πρακτικά μηδενικό Overhead καθώς λόγω της τιμής του recomputetime που έχει τεθεί στα 30 δευτερόλεπτα, πρακτικά λειτουργεί σαν Direct Delivery, πέρα από ορισμένες περιπτώσεις για χρόνους μεταξύ 1 και 80 ή 160 δευτερόλεπτα. Όπως ήταν αναμενόμενο το Epidemic και το Spray and Wait έχουν παρόμοιο Overhead Ratio, ενώ το PROPHET ελαφρώς καλύτερο αφού δεν προωθεί πάντα τα μηνύματα στον Ενδιάμεσο κόμβο.

Παρατηρήσεις:

Ο χρόνος Max Contact/Sleep Time είναι το ανώτατο όριο της τυχαίας τιμής του χρόνου που μπορεί να είναι σε αδράνεια ή λειτουργία ο κόμβος. Γενικά η τιμή του χρόνου αδρανείας-λειτουργίας κυμαίνεται στο εύρος 1 μέχρι Max Contact/Sleep Time.

Για το πρωτόκολλο Spray and Wait χρησιμοποιήθηκε το Binary Mode, που είναι το πιο δημοφιλές, παρότι δεν θα είχε διαφορά η χρήση του Normal Mode στην συγκεκριμένη τοπολογία.

Κεφάλαιο 7:

Συμπεράσματα

Κύριος σκοπός της παρούσας εργασίας είναι η μελέτη της δικτύωσης σε τοπολογίες IoT περιορισμένης, δικτυακής υποδομής και διακοπτόμενης συνδεσιμότητας. Αρχικά έγινε μια μικρή ανάλυση της τωρινής αρχιτεκτονικής TCP/IP, η οποία όμως δεν είναι σχεδιασμένη για αυτόν τον σκοπό και άρα δεν είναι αποδοτική. Εν συνεχεία έγινε μια πιο εκτεταμένη θεωρητική ανάλυση στην αρχιτεκτονική DTN, που υπόσχεται την επίτευξη αξιόπιστης επικοινωνίας μεταξύ φορητών συσκευών και συσκευών μη αδιάληπτης επικοινωνίας. Μετά έγινε μια ανάλυση των μελετών και εργαλείων που προϋπάρχουν σχετικά με το DTN και το IoT.

Στο κύριο μέρος γίνεται εξομοίωση σεναρίων με διακοπτόμενη συνδεσιμότητα, πιο συγκεκριμένα τα σενάρια αξιοποιούν την νεότερη έκδοση BP7, με την χρήση του εργαλείου dtn7-go. Γίνεται η ανάλυση της μεθοδολογίας και αναπτύσσονται τέσσερα σενάρια.

Οι τρεις βασικές μετρικές που εξήχθησαν και αποτελούν κριτήριο για την αξιολόγηση των αποτελεσμάτων είναι: το Round-Trip-Time, το Delivery Ratio και το Overhead Ratio. Αυτές είναι μετρικές που υπολογίστηκαν σε όλα τα πειράματα, όμως το Overhead έχει νόημα σαν κριτήριο σύγκρισης μόνο στο τελευταίο πείραμα.

Πιο συγκεκριμένα σχετικά με τα πειράματα:

- i. Στο πρώτο πείραμα μεταβλήθηκε ο χρόνος λειτουργίας των κόμβων. Από αυτό το αρχικό πείραμα αποδείχθηκε ότι το dtn7-go μπορεί να λειτουργήσει σε συνθήκες διακοπτόμενης συνδεσιμότητας για αρκετές τιμές καθυστέρησης. Επιπρόσθετα από τα αποτελέσματα προκύπτει το συμπέρασμα ότι όσο μεγαλύτερος είναι αυτός ο χρόνος τόσο πιο αποδοτική είναι η επικοινωνία.
- ii. Έπειτα έχοντας το Πείραμα 1 ως βάση, η μελέτη συνεχίστηκε με την πιο απαιτητική διεργασία της αποστολής πακέτων με μεγαλύτερο μέγεθος. Το dtn7-go ανταποκρίθηκε σε αυτή την πρόκληση, αλλά δεν ήταν για όλους τους συνδυασμούς μεγέθους πακέτου-χρόνου λειτουργίας δυνατή η επίτευξη αποδοτικής επικοινωνίας. Για χρόνο λειτουργίας 10 δευτερολέπτων και μέγεθος bundle 5MB και 10MB ή χρόνο 20 δευτερολέπτων και μέγεθος αρχείου 10MB, ο χρόνος μιας σύνδεσης δεν ήταν επαρκής για την ανταλλαγή των απαραίτητων πακέτων. Επομένως το dtn7-go θα ήταν χρήσιμο για την ανταλλαγή αρχείων μικρού μεγέθους όπως αρχεία κειμένου ή φωτογραφίες μεσαίας ανάλυσης, αλλά δεν θα μπορούσε να ανταποκριθεί σε σενάρια που σκοπός είναι η αποστολή μεγαλύτερων αρχείων όπως αρχεία βίντεο.
- iii. Εν συνεχεία στο Πείραμα 3, πάλι με βάση το Πείραμα 1, αυξήθηκε σταδιακά το πλήθος των bundle που αποστέλλονται σε κάθε ευκαιρία επικοινωνίας. Και εδώ υπήρχαν τα επιθυμητά αποτελέσματα επικοινωνίας, αλλά κυρίως για μεγάλους χρόνους λειτουργίας των κόμβων. Για

χρόνο λειτουργίας 10 ή 20 δευτερολέπτων και αποστολή 8 ή 16 αρχείων ταυτοχρόνως παρατηρείται ότι το Delivery Ratio που επιτυγχάνεται με το dtn7-go είναι αρκετά μικρό. Σε γενικές γραμμές η παρούσα υλοποίηση χρειάζεται ένα σχετικά μεγάλο χρόνο λειτουργίας των κόμβων, 40 δευτερόλεπτα και άνω, για να εγγυηθεί την παράδοση της πλειονότητας των μηνυμάτων. Επομένως η χρήση του dtn7-go δεν θα ήταν ιδανική σε κάποια κατάσταση κινδύνου, όπου πολλά μηνύματα πρέπει να αποσταλούν με πολύ σύντομο χρονικό περιθώριο.

- iv. Στο τέταρτο και τελευταίο πείραμα, αλλάζει λίγο η μεθοδολογία και εισάγεται ο παράγοντας της τυχαιότητας του χρόνου λειτουργίας και αδρανείας των κόμβων. Επιπρόσθετα, γίνεται μια προσπάθεια σύγκρισης των πρωτόκολλων δρομολόγησης που περιλαμβάνει το dtn7-go. Τα αποτελέσματα είναι τα αναμενόμενα, σύμφωνα με τις λεπτομέρειες για την δρομολόγηση που αναλύθηκαν στην Ενότητα 2.4. Πιο συγκεκριμένα φαίνεται το DTLSR να πετυχαίνει ελαφρά καλύτερο Delivery Ratio από τα υπόλοιπα πρωτόκολλα και έχοντας πολύ μικρότερο Overhead Ratio, με το μόνο μειονέκτημα να είναι ότι έχει το μεγαλύτερο latency. Το καλύτερο latency πετυχαίνουν το Epidemic και το Spray and Wait, με μεγάλο Delivery Ratio, αλλά και μεγάλο Overhead Ratio αφού κάνουν flooding πακέτων στο δίκτυο. Το PRoPHET φαίνεται ως το πιο χρήσιμο για ένα μέσο δίκτυο που οι πόροι δεν είναι τεράστιοι, καθώς πετυχαίνει καλό Delivery Ratio, και έχει μικρότερο Overhead Ratio από το Epidemic, καταναλώνοντας λίγο παραπάνω χρόνο για να παραδώσει τα bundles.

Από την παρούσα μελέτη προκύπτει το συμπέρασμα πως όντως το DTN μπορεί να ανταπεξέλθει σε τοπολογίες με διακοπτόμενη συνδεσιμότητα. Έτσι η αρχιτεκτονική μπορεί πράγματι να εφαρμοστεί σε πραγματικά σενάρια IoT με τις αντίστοιχες απαιτήσεις, αλλά και παραχωρήσεις.

Όσον αφορά το BP7 πιο συγκεκριμένα, δυστυχώς η παρούσα μελέτη δείχνει πως είναι σε πρώιμο επίπεδο ακόμα, τουλάχιστον βάσει των αποτελεσμάτων που εξήχθησαν με την υλοποίηση που επιλέχθηκε. Τα αποτελέσματα της παρούσας εργασίας δείχνουν πως όντως το BP7 ανταποκρίνεται σε καταστάσεις που δεν θα ήταν δυνατό με το TCP/IP. Όμως αν γίνει σύγκριση με παλαιότερες μελέτες που χρησιμοποίησαν υλοποιήσεις βασισμένες στο BP6, προκύπτει το συμπέρασμα ότι οι επιδόσεις της παρούσας υλοποίησης δεν είναι τόσο αποδοτικές. Το dtn7-go χωλαίνει όσον αφορά το κομμάτι του Delivery Ratio, καθώς για μικρά πακέτα και μεγάλο χρόνο επαφής δεν είναι αναμενόμενη η απώλεια πακέτων όπως παρατηρείται στα αποτελέσματα αυτής της εργασίας. Οι αδυναμίες στις υλοποιήσεις είναι εν μέρη δικαιολογημένες, καθώς το BP7 είναι πρόσφατο πρωτόκολλο σε πειραματική φάση, που συνεχώς αναπτύσσεται. Έτσι και οι υλοποιήσεις συνεχώς βελτιώνονται και αντιμετωπίζουν τις πρώιμες ελλείψεις τους. Επομένως τα αποτελέσματα που προέκυψαν δεν είναι τα απολύτως αναμενόμενα, αλλά είναι ενθαρρυντικά για μετέπειτα διερεύνηση του BP7, ανάλογα και με την ανάπτυξή του και των εργαλείων που το υλοποιούν.

Κεφάλαιο 8: Προοπτικές

Με την περάτωση της παρούσας διπλωματικής, έχει ολοκληρωθεί ένα δείγμα μελέτης του BP7 με την βοήθεια του dtm7-go. Υπάρχουν αρκετοί τομείς για μελλοντική διερεύνηση.

Στην παρούσα διπλωματική εργασία έγινε μια προσπάθεια σύγκρισης των πρωτόκολλων δρομολόγησης. Αυτό επιτεύχθηκε σε έναν βαθμό, αλλά καθώς ο αριθμός των κόμβων ήταν μικρός και οι επαφές του κάθε κόμβου δεν διαφοροποιούνταν δραματικά τα αποτελέσματα είναι χρήσιμα, αλλά όχι αντιπροσωπευτικά για κάθε πραγματική τοπολογία. Ενδιαφέρον θα παρουσίαζε η αναλυτικότερη σύγκριση των πρωτόκολλων δρομολόγησης σε ένα πιο πυκνό δίκτυο με περισσότερους κόμβους.

Ακόμα, όσον αφορά την δρομολόγηση θα ήταν χρήσιμη η μελέτη και άλλων πρωτόκολλων, κυρίως αυτών που εκμεταλλεύονται τις κοινωνικές σχέσεις ή την γεωγραφική τοποθεσία των κόμβων, αν αυτά προστεθούν μελλοντικά σε μια υλοποίηση του BP7.

Ακόμα χρήσιμη θα ήταν η μελέτη μια τοπολογίας με κινητούς κόμβους (VANETs), καθώς είναι κάτι που ολοένα και εξελίσσεται και περισσότερες κινητές συσκευές (όπως αυτοκίνητα, λεωφορεία κ.λπ.) ενσωματώνουν τεχνολογίες διασύνδεσης. Σε μια τοπολογία IoT θα μπορούσε να συνδυαστεί η μελέτη VANETs με τα πρωτόκολλα δρομολόγησης που προαναφέρθηκαν.

Επιπρόσθετα, χρήσιμη θα ήταν η αξιολόγηση της κατανάλωσης ενέργειας του DTN7, ιδίως για την αποδοτικότητά του σε IoT.

Τέλος ενδιαφέρον θα παρουσίαζε η ανάπτυξη της παρούσας μελέτης, αλλά με διαφορετική υλοποίηση του BP7. Πλέον το dtm7-rs λαμβάνει συνεχείς ενημερώσεις και φαίνεται πιο ελπιδοφόρο και υγιές.

Βιβλιογραφικές Αναφορές

- [1] L. Lilien, Z. Kamal, V. Bhuse, and A. Gupta, "Opportunistic networks: the concept and research challenges in privacy and security," *Mob. Wirel. Netw. Secur. Priv.*, Jan. 2007, doi: 10.1007/978-0-387-71058-7_5.
- [2] I. Psaras, L. Wood, and R. Tafazolli, "Delay-/Disruption-Tolerant Networking: State of the Art and Future Challenges," Dec. 2009.
- [3] S. Burleigh, V. G. Cerf, J. Crowcroft, and V. Tsaoussidis, "Space for Internet and Internet for space," *Ad Hoc Netw.*, vol. 23, pp. 80–86, 2014, doi: <https://doi.org/10.1016/j.adhoc.2014.06.005>.
- [4] F. Warthman, "Delay- and Disruption-Tolerant Networks (DTNs) - A Tutorial, Version 3.2." DTN Research Group (DTN-RG), 2015.
- [5] C. Hunt, *TCP/IP Network Administration*. O'Reilly Media, Inc., 2002.
- [6] K. Fall, "A delay-tolerant network architecture for challenged internets," in *Proceedings of the 2003 conference on Applications, technologies, architectures, and protocols for computer communications - SIGCOMM '03*, Karlsruhe, Germany, 2003, p. 27. doi: 10.1145/863955.863960.
- [7] C. E. Perkins, "Mobile IP," *Int. J. Commun. Syst.*, vol. 11, no. 1, pp. 3–20, 1998, doi: 10.1002/(SICI)1099-1131(199801/02)11:1<3::AID-DAC351>3.0.CO;2-6.
- [8] A. Rayes and S. Salam, *Internet of Things From Hype to Reality: The Road to Digitization*, 2nd ed. 2019. Cham: Springer International Publishing : Imprint: Springer, 2019. doi: 10.1007/978-3-319-99516-8.
- [9] K. Finkenzeller, *RFID Handbook: Fundamentals and Applications in Contactless Smart Cards, Radio Frequency Identification and Near-Field Communication*. John Wiley & Sons, 2010.
- [10] S. Madakam, R. Ramaswamy, and S. Tripathi, "Internet of Things (IoT): A Literature Review," *J. Comput. Commun.*, vol. 3, pp. 164–173, Apr. 2015, doi: 10.4236/jcc.2015.35021.
- [11] K. Patel, S. Patel, P. Scholar, and C. Salazar, *Internet of Things-IOT: Definition, Characteristics, Architecture, Enabling Technologies, Application & Future Challenges*. 2016.
- [12] K. Scott and S. Burleigh, "Bundle Protocol Specification," RFC Editor, RFC5050, Nov. 2007. doi: 10.17487/rfc5050.
- [13] A. Penning, L. Baumgärtner, J. Höchst, A. Sterz, M. Mezini, and B. Freisleben, "DTN7: An Open-Source Disruption-Tolerant Networking Implementation of Bundle Protocol 7," in *Ad-Hoc, Mobile, and Wireless Networks*, vol. 11803, M. R. Palattella, S. Scanzio, and S. Coleri Ergen, Eds. Cham: Springer International Publishing, 2019, pp. 196–209. doi: 10.1007/978-3-030-31831-4_14.
- [14] K. Fall and S. Farrell, "DTN: an architectural retrospective," *IEEE J. Sel. Areas Commun.*, vol. 26, no. 5, pp. 828–836, Jun. 2008, doi: 10.1109/JSAC.2008.080609.
- [15] P. Muri, J. McNair, J. Antoon, A. Gordon-Ross, K. Cason, and N. Fitz-Coy, "User Datagram and Bundle Protocol for Distributed Small Satellite Topologies," *J. Wirel. Netw. Commun.*, vol. 3, no. 3, pp. 19–28, 2013.
- [16] E. J. Birrane and K. McKeever, "Bundle Protocol Security Specification," Internet Engineering Task Force, Internet Draft draft-ietf-dtn-bpsec-27, Feb. 2021. Accessed: Dec. 07, 2021. [Online]. Available: <https://datatracker.ietf.org/doc/draft-ietf-dtn-bpsec>
- [17] S. Burleigh, K. Fall, and E. J. Birrane, "Bundle Protocol Version 7," Internet Engineering Task Force, Internet Draft draft-ietf-dtn-bpbis-31, Jan. 2021. Accessed: Dec. 02, 2021. [Online]. Available: <https://datatracker.ietf.org/doc/draft-ietf-dtn-bpbis-31>
- [18] B. Sipos, M. Demmer, J. Ott, and S. Perreault, "Delay-Tolerant Networking TCP Convergence Layer Protocol Version 4," Internet Engineering Task Force, Internet Draft draft-ietf-dtn-tcpclv4-28, Oct. 2021. Accessed: Dec. 06, 2021. [Online]. Available: <https://datatracker.ietf.org/doc/draft-ietf-dtn-tcpclv4>

- [19] S. C. Burleigh, "Minimal TCP Convergence-Layer Protocol," Internet Engineering Task Force, Internet Draft draft-ietf-dtn-mtcpcl-01, Apr. 2019. Accessed: Dec. 06, 2021. [Online]. Available: <https://datatracker.ietf.org/doc/draft-ietf-dtn-mtcpcl>
- [20] C. Bormann and P. E. Hoffman, "Concise Binary Object Representation (CBOR)," Internet Engineering Task Force, Request for Comments RFC 8949, Dec. 2020. doi: 10.17487/RFC8949.
- [21] W. Eddy and E. B. Davies, "Using Self-Delimiting Numeric Values in Protocols," Internet Engineering Task Force, Request for Comments RFC 6256, May 2011. doi: 10.17487/RFC6256.
- [22] M. Beyer, "Comparison of Bundle Protocol version 6 and 7," p. 7.
- [23] S. Jain, K. Fall, and R. Patra, "Routing in a delay tolerant network," in *Proceedings of the 2004 conference on Applications, technologies, architectures, and protocols for computer communications - SIGCOMM '04*, Portland, Oregon, USA, 2004, p. 145. doi: 10.1145/1015467.1015484.
- [24] Y. Cao and Z. Sun, "Routing in Delay/Disruption Tolerant Networks: A Taxonomy, Survey and Challenges," *IEEE Commun. Surv. Tutor.*, vol. 15, no. 2, pp. 654–677, 2013, doi: 10.1109/SURV.2012.042512.00053.
- [25] Y. Zhu, B. Xu, X. Shi, Y. Wang, and S. Member, "A survey of social-based routing in delay tolerant networks: positive and negative social effects," in *Communications Surveys & Tutorials, IEEE 15.1*, 2013, pp. 387–401.
- [26] T. Spyropoulos, K. Psounis, and C. S. Raghavendra, "Single-copy routing in intermittently connected mobile networks," in *2004 First Annual IEEE Communications Society Conference on Sensor and Ad Hoc Communications and Networks, 2004. IEEE SECON 2004.*, Santa Clara, CA, USA, 2004, pp. 235–244. doi: 10.1109/SAHCN.2004.1381922.
- [27] A. Vahdat and D. Becker, "Epidemic Routing for Partially-Connected Ad Hoc Networks," p. 14.
- [28] T. Spyropoulos, K. Psounis, and C. S. Raghavendra, "Efficient Routing in Intermittently Connected Mobile Networks: The Multiple-Copy Case," *IEEEACM Trans. Netw.*, vol. 16, no. 1, pp. 77–90, Feb. 2008, doi: 10.1109/TNET.2007.897964.
- [29] T. Spyropoulos, K. Psounis, and C. S. Raghavendra, "Spray and wait: an efficient routing scheme for intermittently connected mobile networks," in *Proceeding of the 2005 ACM SIGCOMM workshop on Delay-tolerant networking - WDTN '05*, Philadelphia, Pennsylvania, USA, 2005, pp. 252–259. doi: 10.1145/1080139.1080143.
- [30] A. Lindgren, A. Doria, E. Davies, and S. Grasic, "Probabilistic Routing Protocol for Intermittently Connected Networks," RFC Editor, RFC6693, Aug. 2012. doi: 10.17487/rfc6693.
- [31] A. Lindgren, A. Doria, and O. Schelén, "Probabilistic routing in intermittently connected networks," *ACM SIGMOBILE Mob. Comput. Commun. Rev.*, vol. 7, no. 3, pp. 19–20, Jul. 2003, doi: 10.1145/961268.961272.
- [32] M. Demmer and K. Fall, "DTLSR: delay tolerant routing for developing regions," in *Proceedings of the 2007 workshop on Networked systems for developing regions - NSDR '07*, Kyoto, Japan, 2007, p. 1. doi: 10.1145/1326571.1326579.
- [33] P. Hui, J. Crowcroft, and E. Yoneki, "BUBBLE Rap: Social-Based Forwarding in Delay-Tolerant Networks," *IEEE Trans. Mob. Comput.*, vol. 10, no. 11, pp. 1576–1589, Nov. 2011, doi: 10.1109/TMC.2010.246.
- [34] S. M. Bilal, C. J. Bernardos, and C. Guerrero, "Position-based routing in vehicular networks: A survey," *J. Netw. Comput. Appl.*, vol. 36, no. 2, pp. 685–697, Mar. 2013, doi: 10.1016/j.jnca.2012.12.023.
- [35] T. Wang, Y. Cao, Y. Zhou, and P. Li, "A Survey on Geographic Routing Protocols in Delay/Disruption Tolerant Networks," *Int. J. Distrib. Sens. Netw.*, vol. 12, no. 2, p. 3174670, Feb. 2016, doi: 10.1155/2016/3174670.
- [36] S. Bounsiar, F. Z. Benhamida, A. Henni, D. L. de Ipiña, and D. C. Mansilla, "How to Enable Delay Tolerant Network Solutions for Internet of Things: From Taxonomy to Open Challenges," *Proceedings*, vol. 31, no. 1, Art. no. 1, 2019, doi: 10.3390/proceedings2019031024.
- [37] A. P. da Silva, S. C. Burleigh, and K. Obraczka, *Delay and disruption tolerant networks:*

interplanetary and earth-bound -- architecture, protocols, and applications. 2019. Accessed: Dec. 12, 2021. [Online]. Available: <https://www.books24x7.com/marc.asp?bookid=138741>

- [38] F. Z. Benhamida, A. Bouabdellah, and Y. Challal, "Using delay tolerant network for the Internet of Things: Opportunities and challenges," in *2017 8th International Conference on Information and Communication Systems (ICICS)*, Apr. 2017, pp. 252–257. doi: 10.1109/IACS.2017.7921980.
- [39] Philo Juang, H. Oki, Y. Wang, M. M. Tonosi, Li-Shiuan Peh, and D. Rubenstein, "Energy-Efficient Computing for Wildlife Tracking: Design Tradeoffs and Early Experiences with ZebraNet," 2002, doi: 10.13140/RG.2.1.2314.6326.
- [40] A. Mallorquí, A. Zaballo, and A. Briones, "DTN Trustworthiness for Permafrost Telemetry IoT Network," *Remote Sens.*, vol. 13, no. 22, p. 4493, Nov. 2021, doi: 10.3390/rs13224493.
- [41] M. C. Domingo, "An overview of the internet of underwater things," *J. Netw. Comput. Appl.*, vol. 35, no. 6, pp. 1879–1890, Nov. 2012, doi: 10.1016/j.jnca.2012.07.012.
- [42] E. Kulla, "Performance evaluation of Focused Beam Routing for IoT applications in underwater environment," *Internet Things*, vol. 17, p. 100471, Mar. 2022, doi: 10.1016/j.iot.2021.100471.
- [43] Y. Zguira and H. Rivano, "Evaluation of IoB-DTN protocol for mobile IoT Application," p. 22.
- [44] A. Pentland, R. Fletcher, and A. Hasson, "DakNet: rethinking connectivity in developing nations," *Computer*, vol. 37, no. 1, pp. 78–83, Jan. 2004, doi: 10.1109/MC.2004.1260729.
- [45] A. Seth, D. Kroeker, M. Zaharia, S. Guo, and S. Keshav, "Low-cost communication for rural internet kiosks using mechanical backhaul," in *Proceedings of the 12th annual international conference on Mobile computing and networking*, New York, NY, USA, Sep. 2006, pp. 334–345. doi: 10.1145/1161089.1161127.
- [46] C. Barroca, A. Grilo, and P. R. Pereira, "Improving Message Delivery in UAV-based Delay Tolerant Networks," in *2018 16th International Conference on Intelligent Transportation Systems Telecommunications (ITST)*, Lisboa, Oct. 2018, pp. 1–7. doi: 10.1109/ITST.2018.8566956.
- [47] R. Udriou, A. M. Deaconu, and C.-Ş. Nanau, "Data Delivery in a Disaster or Quarantined Area Divided into Triangles Using DTN-Based Algorithms for Unmanned Aerial Vehicles," *Sensors*, vol. 21, no. 11, p. 3572, May 2021, doi: 10.3390/s21113572.
- [48] E. Yaacoub, K. Abualsaud, T. Khattab, and A. Chehab, "Secure Transmission of IoT mHealth Patient Monitoring Data from Remote Areas Using DTN," *IEEE Netw.*, vol. 34, no. 5, pp. 226–231, Sep. 2020, doi: 10.1109/MNET.011.1900627.
- [49] B. Guo, D. Zhang, Z. Wang, Z. Yu, and X. Zhou, "Opportunistic IoT: Exploring the harmonious interaction between human and the internet of things," *J. Netw. Comput. Appl.*, vol. 36, no. 6, pp. 1531–1539, Nov. 2013, doi: 10.1016/j.jnca.2012.12.028.
- [50] *DTN Reference Implementation*. delay-tolerant-networking, 2021. Accessed: Dec. 07, 2021. [Online]. Available: <https://github.com/delay-tolerant-networking/DTN2>
- [51] L. Torgerson *et al.*, "Delay-Tolerant Networking Architecture," Internet Engineering Task Force, Request for Comments RFC 4838, Apr. 2007. doi: 10.17487/RFC4838.
- [52] S. Burleigh, "Interplanetary Overlay Network: An Implementation of the DTN Bundle Protocol," in *2007 4th IEEE Consumer Communications and Networking Conference*, Jan. 2007, pp. 222–226. doi: 10.1109/CCNC.2007.51.
- [53] N. Bezirgiannidis, F. Tsapeli, S. Diamantopoulos, and V. Tsaoussidis, "Towards flexibility and accuracy in space DTN communications," in *Proceedings of the 8th ACM MobiCom workshop on Challenged networks*, New York, NY, USA, Sep. 2013, pp. 43–48. doi: 10.1145/2505494.2505499.
- [54] M. Doering, S. Lahde, J. Morgenroth, and L. Wolf, "IBR-DTN: an efficient implementation for embedded systems," in *Proceedings of the third ACM workshop on Challenged networks*, New York, NY, USA, Sep. 2008, pp. 117–120. doi: 10.1145/1409985.1410008.
- [55] S. Schildt, J. Morgenroth, W.-B. Pöttner, and L. Wolf, "IBR-DTN: A lightweight, modular and highly portable Bundle Protocol implementation," *Electron. Commun. EASST*, p. Volume 37: Kommunikation in Verteilten Systemen 2011, Feb. 2011, doi: 10.14279/TUJ.ECEASST.37.512.
- [56] W.-B. Pöttner, J. Morgenroth, S. Schildt, and L. Wolf, "An Empirical Performance Comparison of

DTN Bundle Protocol Implementations,” p. 21.

- [57] M. Feldmann and F. Walter, “ μ PCN — A bundle protocol implementation for microcontrollers,” in *2015 International Conference on Wireless Communications Signal Processing (WCSP)*, Oct. 2015, pp. 1–5. doi: 10.1109/WCSP.2015.7341252.
- [58] Terra. RightMesh, 2021. Accessed: Dec. 08, 2021. [Online]. Available: <https://github.com/RightMesh/Terra>
- [59] A. Dulaunoy, Forban. 2021. Accessed: Dec. 08, 2021. [Online]. Available: <https://github.com/adulau/Forban>
- [60] A. Dulaunoy, “Forban a simple link-local opportunistic p2p free software - or sharing your bookshelf everywhere,” p. 25.
- [61] D. P. Gardner-Stephen, “The Serval Project: Practical Wireless Ad-Hoc Mobile Telecommunications,” p. 29.
- [62] “ μ D3TN | D3TN.” <https://d3tn.com/ud3tn.html#top> (accessed Dec. 08, 2021).
- [63] *dtm7-kotlin*. Nodle, 2021. Accessed: Dec. 08, 2021. [Online]. Available: <https://github.com/NodleCode/dtm7-kotlin>
- [64] *dtm7-rs*. dtm7, 2021. Accessed: Dec. 08, 2021. [Online]. Available: <https://github.com/dtm7/dtm7-rs>
- [65] L. Baumgartner, J. Hochst, and T. Meuser, “B-DTN7: Browser-based Disruption-tolerant Networking via Bundle Protocol 7,” in *2019 International Conference on Information and Communication Technologies for Disaster Management (ICT-DM)*, Paris, France, Dec. 2019, pp. 1–8. doi: 10.1109/ICT-DM47966.2019.9032944.
- [66] *dtm7-plus*. dtm7, 2021. Accessed: Dec. 09, 2021. [Online]. Available: <https://github.com/dtm7/dtm7-plus-rs>
- [67] L. Baumgärtner, P. Lieser, J. Zobel, B. Bloessl, R. Steinmetz, and M. Mezini, “LoRAgent: A DTN-based Location-aware Communication System using LoRa,” in *2020 IEEE Global Humanitarian Technology Conference (GHTC)*, Oct. 2020, pp. 1–8. doi: 10.1109/GHTC46280.2020.9342886.
- [68] “LoRa PHY | Semtech.” <https://www.semtech.com/lora/what-is-lora> (accessed Dec. 07, 2021).
- [69] *dtm7-go*. dtm7, 2021. Accessed: Dec. 02, 2021. [Online]. Available: <https://github.com/dtm7/dtm7-go>
- [70] “GoLang.” <https://go.dev/> (accessed Dec. 03, 2021).
- [71] “jq.” <https://stedolan.github.io/jq/> (accessed Dec. 06, 2021).
- [72] Raspberry Pi Foundation, “Raspberry Pi,” *Raspberry Pi*. <https://www.raspberrypi.com/> (accessed Dec. 01, 2021).

Παράρτημα

Π.1 Διαδικασία διεξαγωγής των πειραμάτων

Π1.1 Πείραμα 1

Π1.1.1 Ενδιάμεσος Κόμβος

Στον Ενδιάμεσο Κόμβο εκκινείται ο daemon, με την εντολή “./dtnd configuration_int.toml”

Τα αρχεία “configuration.toml” περιέχουν τις ρυθμίσεις για τις παραμέτρους του DTN και είναι πανομοιότυπα για τους τρεις κόμβους.

Ένα παράδειγμα τέτοιου αρχείου:

```
# SPDX-FileCopyrightText: 2019 Markus Sommer
# SPDX-FileCopyrightText: 2019, 2020, 2021 Alvar Penning
#
# SPDX-License-Identifier: GPL-3.0-or-later

# The core is the main module of the delay-tolerant networking daemon.
[core]
# Path to the bundle storage. Bundles will be saved in this directory to be
# present after restarting dtnd.
store = "store"

# Allow inspection of forwarding bundles, containing an administrative
record.

# This allows deletion of stored bundles after being received.
inspect-all-bundles = false

# The node's ID, which should be a dtn-URI. Each node's endpoint ID should
```

```

be

# an URI based on the given node-id.

node-id = "dtn://INTERMEDIATE/"

# If a signature-private entry exists, all outgoing bundles created at this
# node will be signed with the following key. Such a key can be created by:
#   $ xxd -l 64 -p -c 64 /dev/urandom
# Please DO NOT use the following key or a variation of it. I am serious.
# signature-private =
"2d5b59df9e860636ee392fc7833d957543cd7e47e95b8a2800224408840242a8edfflaafc1
0af23ae32a6868e2c31cbbcf3157a706accae2eb7faa7a1d7ee84e"

#signature-private =
"18b5105e95a69fadbc9e3dfcec2e20f06f1b79f8cf1444dbb6996de0b101e95202c497e0c
b4b7305ac9086dddf4c4d9ffa1b8595b074fd73ec6c1cdedd11aac"

# Configure the format and verbosity of dtnd's logging.

[logging]

# Should be one of, sorted from silence to verbose:
# panic,fatal,error,warn,info,debug,trace
level = "info"

# Show the calling method and its file in the logs
# report-caller = true

# Could be "text" for human readable output or "json".
# format = "json"

# The peer/neighbor discovery searches the (local) network for other dtnd
nodes

# and tries to establish a connection to the promoted CLAs.

[discovery]

```

```

ipv4 = true

ipv6 = false

# Interval between two messages in seconds, defaults to 10.
interval = 1

# Agents are applications or interfaces for sending or receiving bundles.
[agents]

# Web server based agent with an own HTTP server for third party tools.
[agents.webserver]

# Address to bind the server to.
address = "localhost:8080"

# Create a WebSocket endpoint at "ws://localhost:8080/ws"
websocket = true

# Create a RESTful endpoints at "http://localhost:8080/rest/"
rest = true

# Each listen is another convergence layer adapter (CLA). Multiple
[[listen]]

# blocks are usable.
[[listen]]

# Protocol to use, one of tcpclv4, tcpclv4-ws, mtcp, bbc.
protocol = "mtcp"

# Address to bind this CLA to.
endpoint = ":4552"

```

```

# Another example based on the WebSocket variant of the TCPCLv4.

# [[listen]]

# protocol = "tcpclv4-ws"

# # Webserver on port 8081 with a WebSocket endpoint at
# "ws://HOST:8081/tcpclv4".

# endpoint = ":8081"


# Another example for a Bundle Broadcasting Connector with a rf95modem.

# [[listen]]

# protocol = "bbc"

# endpoint = "bbc://rf95modem/dev/ttyUSB0"


# Multiple [[peers]] might be configured.

# [[peer]]

# # Protocol to use, one of tcpclv4, tcpclv4-ws, mtcp.

# protocol = "tcpclv4"

# # Address to connect to this CLA.

# endpoint = "10.0.0.2:4556"


# [[peer]]

# protocol = "tcpclv4-ws"

# endpoint = "ws://HOST:PORT/tcpclv4"


# Another peer example..

# [[peer]]

```

```

# # The name/endpoint ID of this peer, as MTCP does not support any
# introduction.

# node = "dtn://gamma/"

# protocol = "mtcp"

# endpoint = "[fc23::2]:35037"


# Specify routing algorithm

[routing]

# One of "epidemic", "spray", "binary_spray", "dtlsr", "prophet",
# "sensor-mule"

algorithm = "epidemic"


# Config for spray routing

#algorithm = "binary_spray"

# [routing.sprayconf]

# multiplicity = 10


# Config for dtlsr

#algorithm = "dtlsr"

# [routing.dtlsrconf]

# recomputetime = "30s"

#broadcasttime = "30s"

#purgetime = "10m"


# Config for prophet

#algorithm = "prophet"

# [routing.prophetconf]

```

```

# # pinit ist the prophet initialisation constant (default value provided
by the PROPHEET-paper)

# pinit = 0.75

#

# # beta is the prophet scaling factor for transitive predictability
(default value provided by the PROPHEET-paper)

# beta = 0.25

#

# # gamma is the prophet ageing factor (default value provided by the
PROPHEET-paper)

# gamma = 0.98

#

# ageinterval = "1m"

```

Σε αυτό το αρχείο παρατηρούνται οι διάφορες παραμέτρους για τον DTN κόμβο. Κάποιες από αυτές τις παραμέτρους είναι: το Node ID, ο φάκελος αποθήκευσης των bundles, ο χρόνος που διαμεσολαβεί μεταξύ των μηνυμάτων και της ανταλλαγής beacon μεταξύ των κόμβων, το Convergence Layer Adapter, το πρωτόκολλο δρομολόγησης κ.α.

Π1.1.2 Κόμβος Αποστολέα

Για τις ανάγκες του πρώτου πειράματος στον Αποστολέα τρέχει το bash script “exp1_SENDER_simple Scenario.sh” :

```

#!/bin/sh

#exec 1> sender_10.log 2>&1

#set -x

# make a log file for Sender

echo just started time: $(date +%T.%N)

#launch the request-response watchdog of Sender

lxtterminal -e sh request-response_sender.sh

#Make the exchange 20 times, to obtain accurate results

for i in {1..20}

```



```

do

    #launch the daemon-UPTIME

    lxterminal -e dtnd configuration_sender.toml && echo $i daemon start
time: $(date +%T.%N)

    #register an agent for exchanging bundles

    lxterminal -e dtn-tool exchange ws://localhost:8080/ws dtn://SENDER/
/home/pizw2/dtn7-go-0.7.1/

    #set a small sleep time before the the creation of the bundle to set
up the agent

    sleep 2s

    #create a bundle

    dtn-tool create dtn://SENDER/ dtn://RECEIVER/ request.txt bundle_no$i

    #set a sleep time for the purpose of our determenistic topology( x
seconds UPTIME and x seconds DOWNTIME)

    sleep 5.77s

    #kill the daemon-DOWNTIME

    pkill -KILL dtnd && echo $i daemon close time: $(date +%T.%N)

    sleep 10s

done

#launch the daemon and register the agent to receive a final response from
the RECEIVER

lxterminal -e dtnd configuration_sender.toml && echo 21 daemon open time:
$(date +%T.%N)

lxterminal -e dtn-tool exchange ws://localhost:8080/ws dtn://SENDER/
/home/pizw2/dtn7-go-0.7.1/

sleep 7.6s

#after a total of x seconds uptime terminate the daemon

pkill -KILL dtnd && echo 21 daemon close time: $(date +%T.%N)

```

Αυτό το bash script είναι υπεύθυνο για την εκκίνηση του dtn7-go daemon με τις ρυθμίσεις που περιέχονται στο αρχείο "configuration_sender.toml". Μετά ορίζει έναν agent για την ανταλλαγή bundles, ο οποίος χρησιμοποιεί το websocket "ws://localhost:8080/ws" σαν κανάλι επικοινωνίας. Εν συνεχεία, δημιουργεί ένα bundle που το περιεχόμενο του είναι αυτό που περιέχεται στο αρχείο "request.txt",

δηλαδή η λέξη "temperature" σαν να ζητάει μια μέτρηση θερμοκρασίας. Μετά από ένα συγκεκριμένο χρονικό διάστημα τερματίζει τον daemon. Αυτή η διαδικασία επαναλαμβάνεται 20 φορές, άρα και ανταλλάσσει 20 bundles.

Επιπρόσθετα καλεί το script "request-response_sender.sh" :

```
#!/bin/bash

#clear the store( is the folder that bundles are saved, by configuration)
from previous run

rm -r store

#and create the directory again

mkdir -p store/bndl

#temp, make an alternative directory to move the bundles that would be
needed for the tests

#remove previous run

rm -r temp

mkdir temp

#set as target the dir that the bundles would be saved

TARGET=/home/pizw2/dtn7-go-0.7.1/

#set as Processed the dir that the temp would move the bundles that we need

PROCESSED=/home/pizw2/dtn7-go-0.7.1/temp

#create a file to save the timestamp of requests and responses

#remove previous measures

rm timestamp.txt

touch timestamp.txt

#count variables

data_packet_count=1

request_count=1


#monitor(-m) for events(until script is terminated) for event(-e)

#when a file is created or mover in the target directory

inotifywait -m -e create -e moved_to --format "%f" $TARGET \
```

```

| while read FILENAME

#read the file

do

    echo Detected bundle $FILENAME

    #pass the payload of the bundle to a variable

    #I am using two variables, because depending on the
the bundle the payload could be on first or second canonical block

    var1=$(./dtn-tool show $TARGET/$FILENAME | jq -r
.canonicalBlocks[-1].data | base64 --decode)

    var2=$(./dtn-tool show $TARGET/$FILENAME | jq -r
.canonicalBlocks[-2].data | base64 --decode)

    #check if the payload of the bundle is a data_packet-
response to your request

    #I am comparing the var with Ldata_packet instead of
data_packet, because the implementation face some problems with decoding
the payload

    if [ "Ldata_packet" = "$var1" ] || [ "Ldata_packet"
= "$var2" ]; then

        #copy the data packet bundle to the temp dir
        cp "$TARGET/$FILENAME" "$PROCESSED/$FILENAME"

        #write the time that the bundle was received to a
file

        echo "time: $(date +%T.%N) type:data packet
number:$data_packet_count name:$FILENAME" >> timestamp.txt

        #just announce that you received a bundle
data packet to your request

        echo "the content is: data packet,this is the
no:$data_packet_count data packet to my request"

        #increase the count variable since you received a
bundle

        data_packet_count=$((data_packet_count+1))

        #check if the payload of the bundle meets an request that
created by you

        #I am comparing the var with Ltemperature instead of

```

temperature, because the implementation face some problems with decoding the payload

```
elif [ "Ltemperature" = "$var1" ] || [ "Ltemperature" =
"$var2" ]; then

    #copy the bundle to the temp dir

    cp "$TARGET/$FILENAME" "$PROCESSED/$FILENAME"

    #write the time that the bundle was created
to a file

    echo "time: $(date +%T.%N) type:request
number:$request_count name:$FILENAME" >> timestamp.txt

    echo "the content is: request,this is the
no:$request_count request that I send"

    #increase the count variable since you send a
bundle

    request_count=$((request_count+1))

    #if the payload of the bundle is different from the
request that you created or not a response to your request

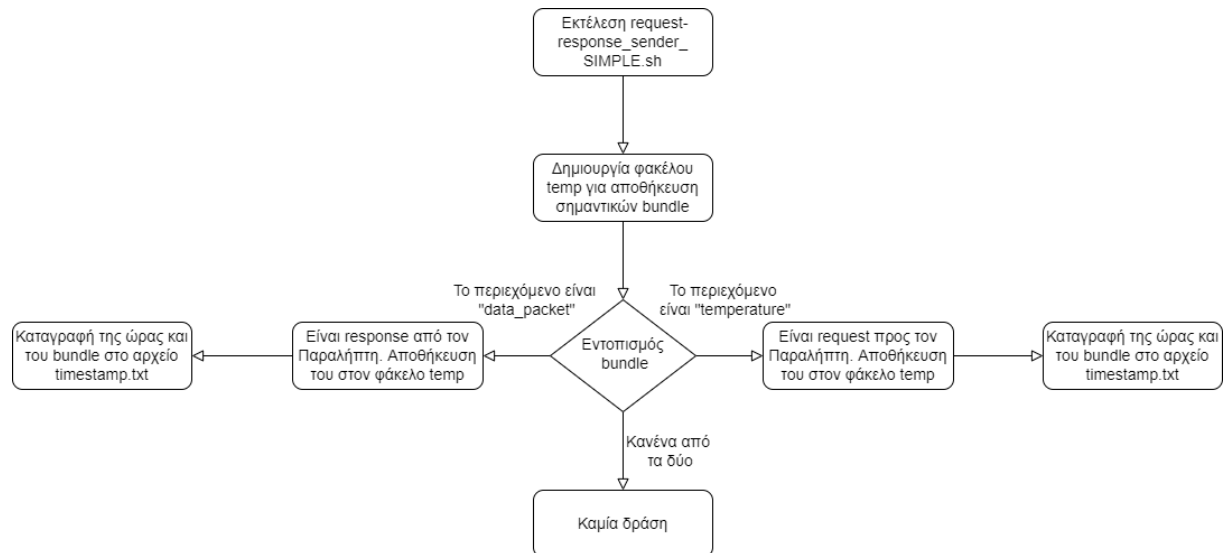
    else

        echo I cant assist with this request

    fi

done
```

Αυτό το bash script είναι υπεύθυνο για την αναγνώριση των bundles. Πιο συγκεκριμένα όταν δημιουργηθεί κάποιο bundle στον φάκελο που έχουμε ορίσει με το dtn-tool exchange, το script αυτό αποκωδικοποιεί το περιεχόμενο του αρχείου. Αν το περιεχόμενο του αρχείου είναι κάτι που το ενδιαφέρει όπως request (temperature) ή response (data_packet) τότε αποθηκεύει το bundle σε έναν άλλο φάκελο, επιπρόσθετα σε ένα αρχείο "timestamp.txt" γράφει την ώρα που εντοπίστηκε το bundle και το όνομα του. Έτσι μετά με την βοήθεια των μετρήσεων που περιέχονται στο "timestamp.txt" μπορεί να υπολογιστεί η χρονική διάρκεια της ανταλλαγής. Σημαντικό κομμάτι είναι η αποκωδικοποίηση του bundle, επειδή το bundle αποτελείται από δύο blocks, κάνοντας σύγκριση με τις μεταβλητές var1, var2 ελέγχονται και τα δύο Canonical Block για το να εντοπιστεί αυτό που περιέχει το Payload.



Εικόνα 15. Διάγραμμα Ροής request-response_sender_SIMPLE.sh

Π1.1.3 Κόμβος Παραλήπτη

Στον Παραλήπτη εκτελείται ένα script παρόμοιο με αυτό του αποστολέα. Το bash script αυτό ονομάζεται “exp1_RECEIVER_simple Scenario.sh”:

```

#!/bin/sh

#exec 1> receiver_160.log 2>&1

#set -x

# make a log file for Receiver
echo just started time: $(date +%T.%N)

#launch the request-response watchdog of Receiver
lxtterminal -e sh request-response_receiver.sh

#Start from sleep for the purpose of the scenario (when Sender is UP
Receiver is DOWN)

sleep 161.19s

#Make the exchange 20 times, to obtain accurate results
for i in {1..20}
do

    #launch the daemon-UPTIME

    lxtterminal -e ./dtnd configuration_receiver.toml && echo $i daemon
  
```

```

open time: $(date +%T.%N)

#register an agent for exchanging bundles

lxtterminal -e ./dtn-tool exchange ws://localhost:8080/ws
dtn://RECEIVER/ /home/pizw/dtn7-go-0.7.1/

#Give some time to receive the bundle and send a response

sleep 158.23s

#kill the daemon-DOWNTIME

pkill -KILL dtnd && echo $i daemon close time: $(date +%T.%N)

sleep 159.41s

done

```

Όπως και στον Αποστολέα αυτό το script είναι υπεύθυνο για εκκίνηση του daemon και τον ορισμό ενός agent για την ανταλλαγή bundle. Και σε αυτό το script η διαδικασία επαναλαμβάνεται 20 φορές για ανταλλαγή 20 bundles.

Επιπρόσθετα και ο παραλήπτης καλεί ένα δεύτερο script το “request-response_receiver.sh”:

```

#!/bin/bash

#clear the store( is the folder that bundles are saved, by configuration)
from previous run

rm -r store

#and create the directory again

mkdir -p store/bndl

#temp, make an alternative directory to move the bundles that would be
needed for the tests

#remove previous run

rm -r temp

mkdir temp

#set as target the dir that the bundles would be saved

TARGET=/home/pizw/dtn7-go-0.7.1/

#set as Processed the dir that the temp would move the bundles that we need

PROCESSED=/home/pizw/dtn7-go-0.7.1/temp

#create a file to save the timestamp of requests and responses

```

```

#remove previous measures

rm timestamp.txt

touch timestamp.txt

#count variables

data_packet_count=1

request_count=1


#monitor(-m) for events(until script is terminated) for event(-e)

#when a file is created or mover in the target directory

inotifywait -m -e create -e moved_to --format "%f" $TARGET \

    | while read FILENAME

    #read the file

        do

            echo Detected bundle $FILENAME

            #pass the payload of the bundle to a variable

            #I am using two variables, because depending
on the bundle the payload could be on first or second canonical block

            var1=$(./dtm-tool show $TARGET/$FILENAME | jq -r
.canonicalBlocks[-1].data | base64 --decode)

            var2=$(./dtm-tool show $TARGET/$FILENAME | jq
-r .canonicalBlocks[-2].data | base64 --decode)

            #check if the payload of the bundle meets an
request you can serve(temperature), check the store_content to see which
request you can serve

            #I am comparing the var with Ltemperature
instead of temperature, because the implementation face some problems with
decoding the payload

            if ([ "Ltemperature" = "$var1" ] || [ "Ltemperature" =
"$var2" ]) && [ "temperature" = "$(cat content_store.txt)" ]; then

                #copy the bundle to the temp dir

                cp "$TARGET/$FILENAME" "$PROCESSED/$FILENAME"

                #write the time that the bundle was
received to a file

```

```

        echo "time: $(date +%T.%N) type:request
number:$request_count name:$FILENAME" >> timestamp.txt

        #just announce that you received a bundle with a
request that you can serve

        echo "the content is: temperature,this
is the no:$request_count temperature request,I would make a data packet"

        #leave 1 second sleeping time to set-up

        sleep 1s

        #create a data packet to the request

        ./dtn-tool create dtn://RECEIVER/
dtn://SENDER/ data_packet.txt /home/pizw/dtn7-go-
0.7.1/data_packet$data_packet_count

        #write the time that the data packet
bundle was created to a file

        #echo "time:$(date +%T.%N) type:data
packet number:$data_packet_count name:data_packet$data_packet_count" >>
timestamp.txt

        echo "the content is: request,this is
the no:$request_count request that I receive"

        #increase the count variable since you
received a bundles

        request_count=$((request_count+1))

        #check if the payload of the bundle is a
data_packet-response to the Sender

        #I am comparing the var with Ldata_packet
instead of data_packet, because the implementation face some problems with
decoding the payload

        elif [ "Ldata_packet" = "$var1" ] || [
"Ldata_packet" = "$var2" ]; then

        #copy the data packet bundle to the
temp dir

        cp "$TARGET/$FILENAME" "$PROCESSED/$FILENAME"

        echo this is a data packet bundle

        #write the time that the data packet

```


bundle was created to a file

```
                                echo "time: $(date +%T.%N) type:data
packet number:$data_packet_count name:data packet$data_packet_count" >>
timestamp.txt

                                echo "the content is: data packet,this
is the no:$data_packet_count data packet I send as response to Sender"

                                #increase the count variables since you
send a bundle

                                data_packet_count=$((data_packet_count+1))

                                #if the payload of the bundle is different
from the request that you can serve(temperature) or not a response to the
request

                                else

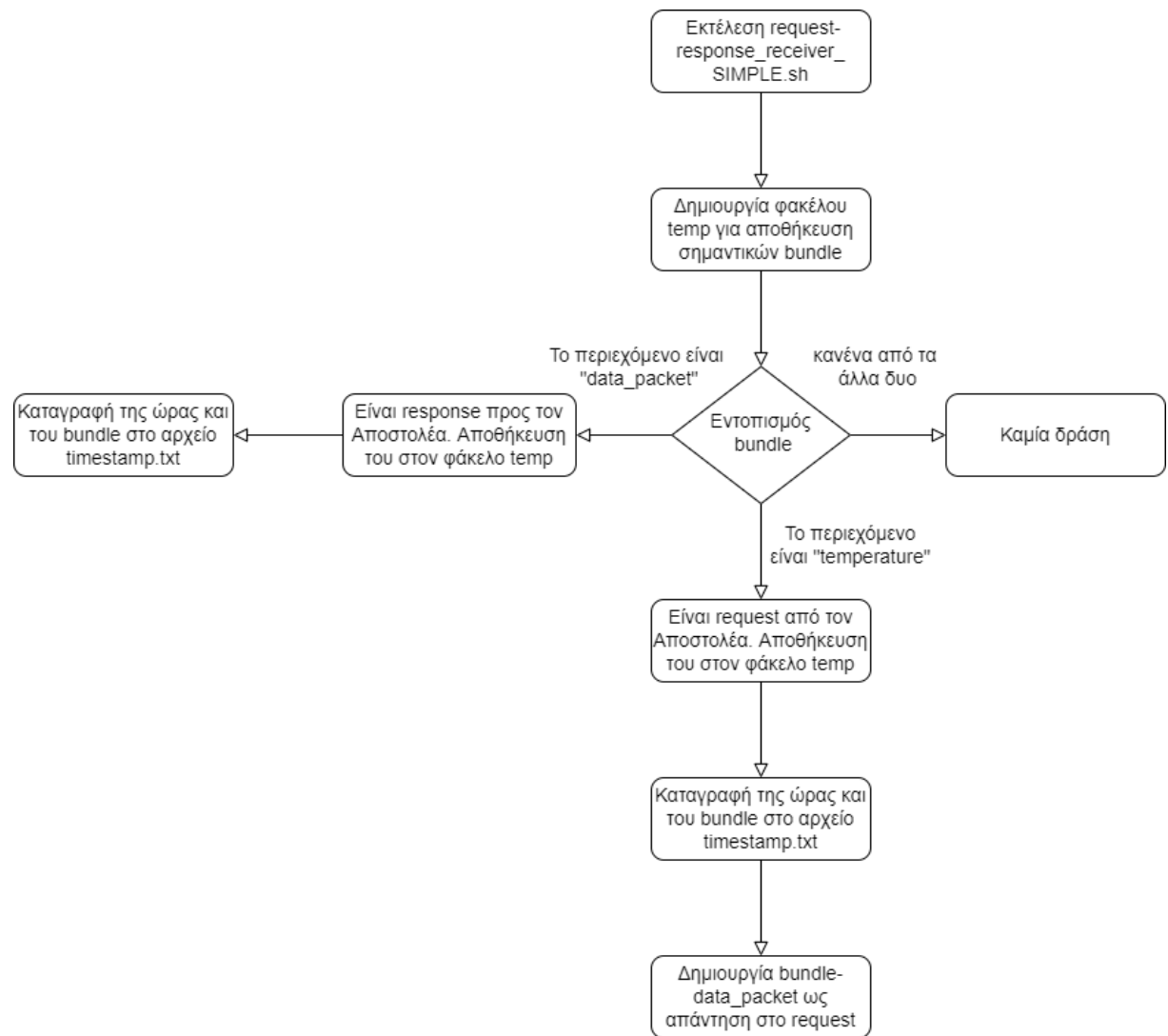
                                #if the payload of the bundle is different from
my request that you can serve(temperature)

                                echo I cant assist with this request

                                fi

                                done
```

Αυτό το script όπως και το αντίστοιχο του Αποστολέα, αποκωδικοποιεί το περιεχόμενο των bundle που δημιουργούνται ή λαμβάνονται στον Παραλήπτη. Επιπρόσθετα, αν το περιεχόμενο του bundle είναι "temperature", δηλαδή πρόκειται για request, ελέγχει αν το περιεχόμενο του content_store είναι ίδιο με το request και αν ναι δημιουργεί ένα bundle απάντηση στον Αποστολέα με την μέτρηση που ζητήθηκε. Τέλος και σε αυτό bash script αποθηκεύει τα σημαντικά bundles σε έναν φάκελο και δημιουργεί ένα αρχείο timestamp.txt με τα ονόματα και την χρονική στιγμή ανάγνωσης αυτών των bundles.



Εικόνα 16. Διάγραμμα Ροής request-response_receiver_SIMPLE.sh

Π1.2 Πείραμα 2

Π1.2.1 Ενδιάμεσος Κόμβος

Στον Ενδιάμεσο κόμβο χρησιμοποιείται το ίδιο αρχείο ρυθμίσεων “configuration.toml”, όπως και στο Πείραμα 1. Το ίδιο ισχύει και για τα αρχεία configuration των υπόλοιπων κόμβων.

Π1.2.2 Κόμβος Αποστολέα

Για τις ανάγκες του δεύτερου πειράματος στον Αποστολέα τρέχει το bash script “exp2_SENDER_dif_bund_size.sh”. Αυτό είναι παρόμοιο με το αντίστοιχο του πρώτου πειράματος, είναι υπεύθυνο για την εκκίνηση του dtn7-go daemon με τις ρυθμίσεις που περιέχονται στο αρχείο configuration, μετά ορίζει έναν agent για την ανταλλαγή bundles, ο οποίος χρησιμοποιεί το websocket “ws://localhost:8080/ws” σαν κανάλι επικοινωνίας, έπειτα δημιουργεί ένα bundle το οποίο όμως αυτή την φορά έχει όνομα το μέγεθος του αρχείου (π.χ. 1kb.txt, αντί για request.txt που ήταν το αντίστοιχο αρχείο στο πρώτο πείραμα) και μετά από ένα συγκεκριμένο χρονικό διάστημα τερματίζει τον daemon.

Συνοψίζοντας η διαφορά σε σχέση με το αντίστοιχο bash script του πρώτου πειράματος είναι ότι αυτό δημιουργεί ένα αρχείο της μορφής “number_size.txt” αντί για “request.txt”. Το αρχείο αυτό δημιουργείται στο terminal του κόμβου με την εντολή: “truncate -s 1k 1kb.txt” για να δημιουργηθεί ένα αρχείο μεγέθους ακριβώς 1kb.

Επιπρόσθετα καλείται το script “request-response_sender_size.sh” :

```
#!/bin/bash

#clear the store( is the folder that bundles are saved, by configuration)
from previous run

rm -r store

#and create the directory again

mkdir -p store/bndl

#temp, make an alternative directory to move the bundles that would be
needed for the tests

#remove previous run

rm -r temp

mkdir temp

#set as target the dir that the bundles would be saved

TARGET=/home/pizw2/dtn7-go-0.7.1/

#set as Processed the dir that the temp would move the bundles that we need
```

```

PROCESSED=/home/pizw2/dtn7-go-0.7.1/temp

#create a file to save the timestamp of request and responses

#remove previous measures

rm timestamp.txt

touch timestamp.txt

#count variables

data_packet_count=1

request_count=1


#monitor(-m) for events(until script is terminated) for event(-e)

#when a file is created or mover in the target directory

inotifywait -m -e create -e moved_to --format "%f" $TARGET \

    | while read FILENAME

        #read the file

            do

                echo Detected bundle $FILENAME

                case $FILENAME in

                    2521762850414e49433d537472696e67206d6574686f643a2072756e74696d6520657

                    2726f723a20696e76616c6964206d656d6f72792061646472657373206f72206e696c20706f

                    696e7465722064657265666572656e6365292d302d30)

                        #first case, this bundle is

                        auto generated by the daemon, is not useful for us.

                        echo I cant assist with

                        this request

                        ;;

                        bundle_no*)

                            #bundle_no* are the name of

                            the bundles that are generated by the Sender( that's the name we determined

                            for them )

                            #copy the bundle to the

                            temp dir

                            cp "$TARGET/$FILENAME"

```

```

"$PROCESSED/$FILENAME"

                                #write the time that the bundle was
created to a file

                                echo "time: $(date +%T.%N)
type:request number:$request_count name:$FILENAME" >> timestamp.txt

                                echo "the content is:
request,this is the no:$request_count request that I send"

                                #increase the count
variable since you send a bundle

                                request_count=$((request_count+1))

                                ;;

                                *)

                                #if the payload is any different then
it's a data_packet-response

                                #copy the data packet bundle to the
temp dir

                                cp "$TARGET/$FILENAME"

"$PROCESSED/$FILENAME"

                                #write the time that the
bundle was received to a file

                                echo "time: $(date +%T.%N)
type:data packet number:$data_packet_count name:$FILENAME" >> timestamp.txt

                                #just announce that you received a
bundle data packet to your request

                                echo "the content is: data
packet,this is the no:$data_packet_count data packet to my request"

                                #increase the count
variables since you received bundle

                                data_packet_count=$((data_packet_count+1))

                                ;;

                                esac

done

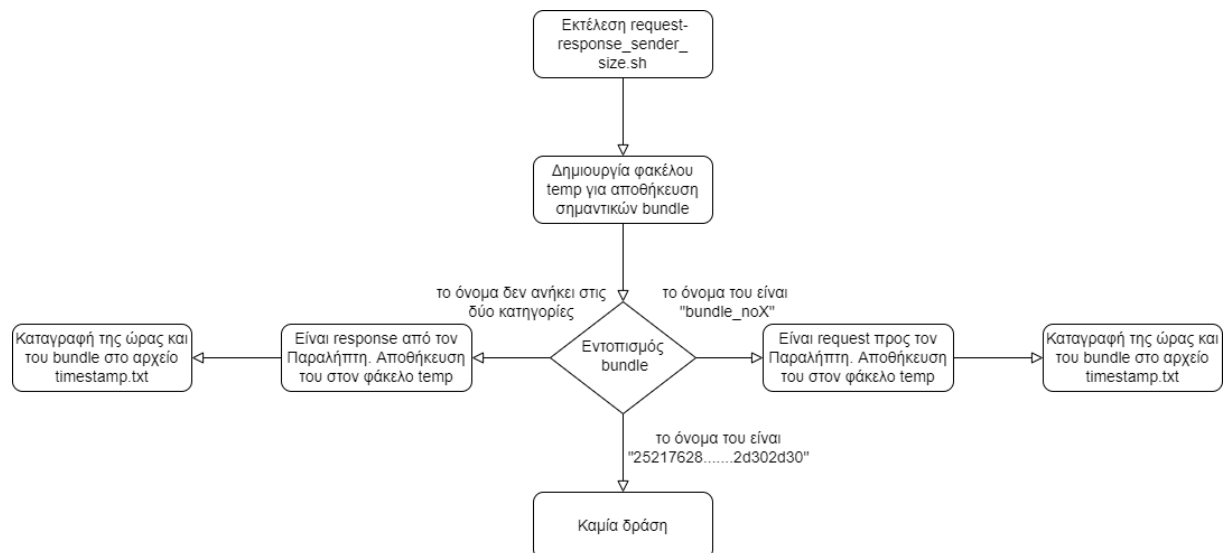
```

Εδώ υπάρχει μια διαφορετική αντιμετώπιση σε σχέση με το αντίστοιχο bash script του πρώτου πειράματος. Αυτό συμβαίνει καθώς το αρχείο που δημιουργείται με την εντολή “truncate” είναι ένας πίνακας και δεν μπορεί να αποκωδικοποιηθεί ορθά, λόγω του προβλήματος με την κωδικοποίηση κειμένου που αντιμετωπίζει η συγκεκριμένη έκδοση (0.7.1) του dtn7-go. Έτσι αντί να ελέγχεται το περιεχόμενο του μηνύματος, ελέγχεται το όνομά του αρχείου του bundle.

Σε αυτή την ανταλλαγή δημιουργούνται 3 είδη bundles:

- i. Ένα bundle που δημιουργείται μόνο του από τον daemon με χαρακτηριστικό όνομα: “2521762850414e49433d537472696e67206d6574686f643a2072756e74696d65206572726f723a20696e76616c6964206d656d6f72792061646472657373206f72206e696c20706f696e7465722064657265666572656e6365292d302d30” και δεν είναι ελεγχόμενο από τον χρήστη.
- ii. Bundles που δημιουργούνται από τον Αποστολέα με προορισμό τον Παραλήπτη. Το όνομα τους είναι της μορφής bundle_noX, το οποίο ορίζεται στο προηγούμενο script και X είναι ο αριθμός της επανάληψης.
- iii. Τα bundles που λαμβάνονται από τον Ενδιάμεσο κόμβο και έχουν δημιουργηθεί από τον Παραλήπτη. Σε αυτή την περίπτωση το όνομα δεν είναι κάποιο συγκεκριμένο μοτίβο, οπότε το script κατατάσσει σε αυτή την κατηγορία όποια bundle δεν έχουν όνομα που ανήκει στις 2 πρώτες κατηγορίες.

Και σε αυτή την περίπτωση τα bundles που είναι σημαντικά για το πείραμα (δηλαδή όσα δεν ανήκουν στην 1^η κατηγορία) αποθηκεύονται σε έναν ξεχωριστό φάκελο και καταγράφεται σε ένα αρχείο “timestamp.txt” η χρονική στιγμή που εντοπίστηκε το bundle και το όνομα του. Μετέπειτα χρησιμοποιώντας αυτές τι καταγραφές υπολογίζεται η συνολική διάρκεια της αποστολής και λήψης μηνυμάτων.



Εικόνα 17. Διάγραμμα Ροής request-response_sender_size.sh

Π1.2.3 Κόμβος Παραλήπτη

Το αρχικό script που εκτελείται στον Παραλήπτη είναι πανομοιότυπο με αυτό του Πειράματος 1. Πέρα από το να θέτει σε λειτουργία και αδράνεια τον daemon και να δημιουργεί κανάλι επικοινωνίας μέσω του dtn-tool exchange, καλεί και το δεύτερο bash script “request-response_receiver_size.sh”:

```
#!/bin/bash

#clear the store( is the folder that bundles are saved, by configuration)
from previous run

rm -r store

#and create the directory again

mkdir -p store/bndl

#temp, make an alternative directory to move the bundles that would be
needed for the tests

#remove previous run

rm -r temp

mkdir temp

#set as target the dir that the bundles would be saved

TARGET=/home/pizw/dtn7-go-0.7.1/

#set as Processed the dir that the temp would move the bundles that we need

PROCESSED=/home/pizw/dtn7-go-0.7.1/temp

#create a file to save the timestamp of requests and responses

#remove previous measures

rm timestamp.txt

touch timestamp.txt

#count variables

data_packet_count=1

request_count=1

#monitor(-m) for events(until script is terminated) for event(-e)

#when a file is created or mover in the target directory

inotifywait -m -e create -e moved_to --format "%f" $TARGET \
```

```

| while read FILENAME

#read the file

do

    echo Detected bundle $FILENAME

    case $FILENAME in

        2521762850414e49433d537472696e67206d6574686f643a2072756e74696d6520657
        2726f723a20696e76616c6964206d656d6f72792061646472657373206f72206e696c20706f
        696e7465722064657265666572656e6365292d302d30)

            #first case, this bundle is auto
            generated by the daemon, is not useful for us.

            echo I cant assist with this
            request

            ;;

            data_packet*)

                #data packet is a bundle that was
                created in this node(so I know the name of the bundle) as a response to the
                bundle I received from the SENDER

                #copy the data packet bundle to
                the temp dir

                cp "$TARGET/$FILENAME" "$PROCESSED/$FILENAME"

                echo "the content is: data
                packet,this is the no:$data_packet_count data packet that I create"

                #write the time that the data
                packet bundle was created to a file

                echo "time: $(date +%T.%N)
                type:data packet number:$data_packet_count name:data
                packet$data_packet_count" >> timestamp.txt

                #increase the count variables
                since you send a bundle

                data_packet_count=$((data_packet_count+1))

                ;;

            *)

```



```

#if the name is any different it
means is a bundle from SENDER

cp "$TARGET/$FILENAME"

"$PROCESSED/$FILENAME"

#write the time that the bundle
was received to a file

echo "time: $(date +%T.%N)
type:request number:$request_count name:$FILENAME" >> timestamp.txt

#just announce that you received a
bundle with a request that you can serve

echo "the content is:
temperature,this is the no:$request_count temperature request,I would make
a data packet"

#create a data packet as response to
the request

#give 1 second sleep to set-up
sleep 1s

./dtn-tool create dtn://RECEIVER/
dtn://SENDER/ 10kb.txt /home/pizw/dtn7-go-
0.7.1/data_packet$data_packet_count

#increase the count variables
since you received a bundles

request_count=$((request_count+1))

;;

esac

done

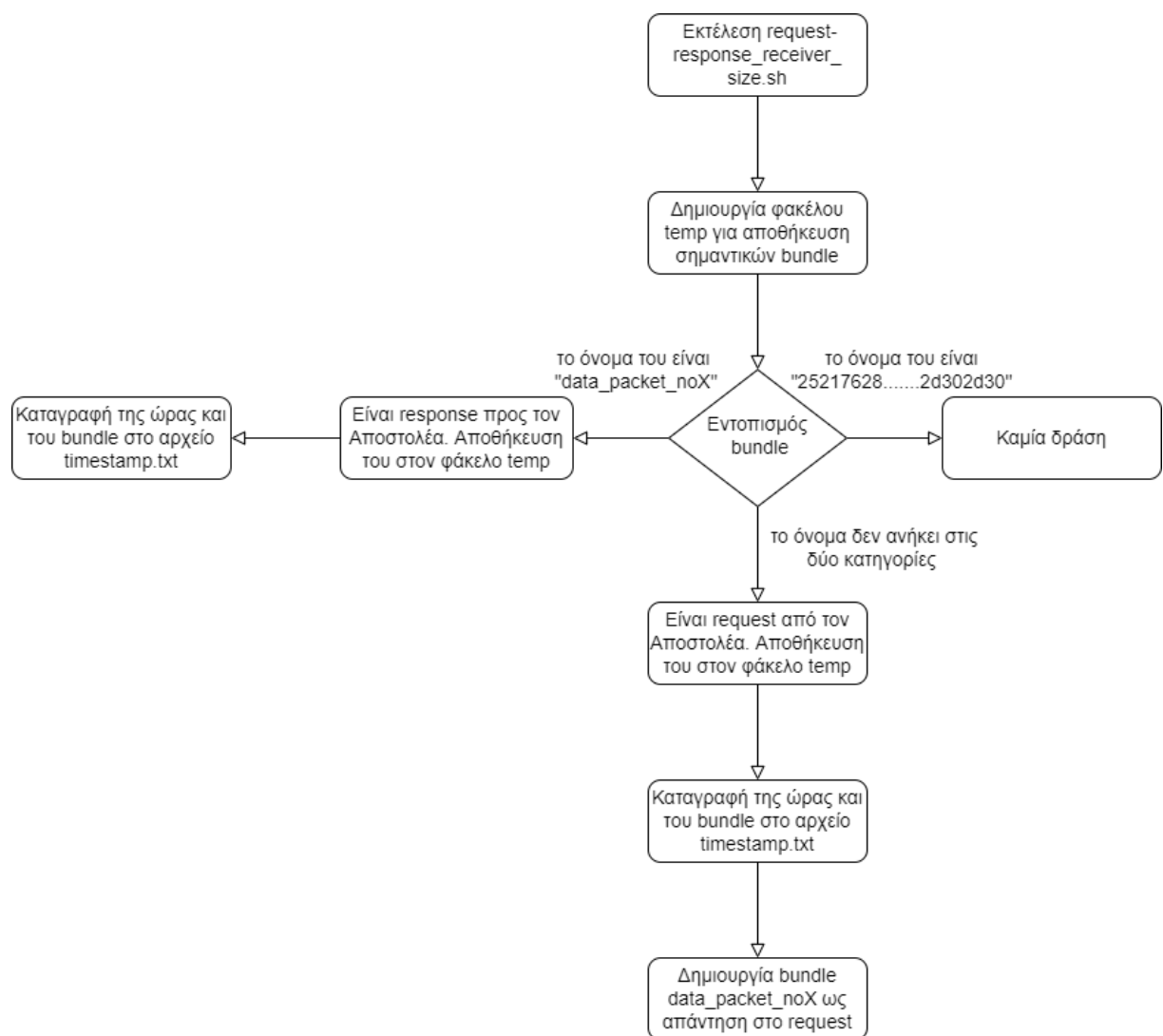
```

Αυτό το script είναι υπεύθυνο για την αναγνώριση του bundle. Όπως και στην περίπτωση του Αποστολέα, αναγκαστικά άλλαξε ο τρόπος λειτουργίας του script και αντί να ξεχωρίζει τα bundle με βάση το περιεχόμενο του μηνύματος, η κατηγοριοποίηση γίνεται με βάση το όνομα του αρχείου. Όπως και στην περίπτωση του Αποστολέα υπάρχουν 3 κατηγορίες μηνυμάτων:

- i. Ένα bundle που δημιουργείται μόνο του από τον daemon με χαρακτηριστικό όνομα: "2521762850414e49433d537472696e67206d6574686f643a2072756e74696d65206572726f723a20696e76616c6964206d656d6f72792061646472657373206f72206e696c20706f696e7465722064657265666572656e6365292d302d30".

- ii. Bundles που δημιουργούνται από τον ίδιο τον κόμβο σαν απάντηση στα αιτήματα του Αποστολέα. Το όνομα τους είναι της μορφής `data_packetX`, όπου `X` είναι ο αριθμός της επανάληψης.
- iii. Τα bundles που λαμβάνονται από τον Ενδιάμεσο κόμβο και έχουν δημιουργηθεί από τον Αποστολέα. Το όνομα τους δεν ανήκει στις προηγούμενες κατηγορίες και έτσι διαχωρίζονται.

Αν το όνομα του bundle δείχνει ότι είναι από τον κόμβο Αποστολέα, τότε ο κόμβος Παραλήπτης δημιουργεί ένα bundle απάντηση. Το bundle-απάντηση έχει διαφορετικό μέγεθος ανάλογα την φορά που εκτελείται, άλλα ίδιο σε μέγεθος με το bundle που λαμβάνει, κυμαίνεται από 1kB έως 10MB και δημιουργείται με την εντολή `"truncate"`. Τέλος αποθηκεύονται τα bundles που δεν ανήκουν στην 1^η κατηγορία σε έναν φάκελος και οι μετρήσεις σε ένα αρχείο `"timestamp.txt"`.



Εικόνα 18. Διάγραμμα Ροής `request-response_receiver_size.sh`

Π1.3 Πείραμα 3

Στο Πείραμα 3 τα bash script για όλους τους κόμβους είναι παρόμοια με αυτά στο Πείραμα 1, οπότε δεν συνεισφέρει η παράθεσή τους. Η μόνη διαφοροποίηση είναι πως ανάλογα με το πόσα bundles πρέπει να αποσταλούν ανά συναλλαγή αυξάνεται η αντίστοιχη μεταβλητή στην επανάληψη που ορίζει το πόσα bundles θα δημιουργηθούν. Αυτό συμβαίνει στο script του Κόμβου Αποστολέα που δημιουργεί τα αρχικά bundles, καθώς και στο script του Κόμβου Παραλήπτη που δημιουργεί bundles-απαντήσεις στα αρχικά bundles.

Π1.4 Πείραμα 4

Στο Πείραμα 4 υπάρχουν κάποιες διαφοροποιήσεις για όλους τους κόμβους στα αρχεία configuration. Ανάλογα με το πρωτόκολλο που δοκιμάζεται ανά εκτέλεση, επιλέγεται και η αντίστοιχη μεταβλητή “algorithm” στο configuration.toml. Για τα Routing Protocols που έχουν επιπρόσθετες παραμέτρους, χρησιμοποιούνται αυτές που έχουν δοθεί από τους δημιουργούς της υλοποίησης.

Επιπρόσθετα στους Κόμβους Αποστολέα και Παραλήπτη διαφοροποιείται ο χρόνος λειτουργίας και αδρανείας και αντί να είναι σταθερός, είναι τυχαίος με τις τιμές του να είναι εντός συγκεκριμένου διαστήματος.

Ενδεικτικά παρακάτω ακολουθεί το script του Αποστολέα:

```
#!/bin/sh

#exec 1> sender_160_random.log 2>&1

#set -x

# make a log file for Sender
echo just started time: $(date +%T.%N)

#launch the request-response watchdog of Sender
lxterminal -e sh request-response_sender.sh

#Make the exchange 20 times, to obtain accurate results
for i in {1..20}
do

    #launch the daemon-UPTIME

    lxterminal -e dtnd configuration_sender.toml && echo $i daemon start
time: $(date +%T.%N)

    #register an agent for exchanging bundles

    lxterminal -e dtn-tool exchange ws://localhost:8080/ws dtn://SENDER/
/home/pizw2/dtn7-go-0.7.1/

    #set a small sleep time before the the creation of the bundle to set
```

up the agent

```
sleep 2s

#create a bundle

dtn-tool create dtn://SENDER/ dtn://RECEIVER/ request.txt bundle_no$i

#set a sleep time for the purpose of our random scenario( the time is
in the range of (1 to x) seconds UPTIME and (1 to x) seconds DOWNTIME)

sleep $(( $RANDOM % 160 + 1 ))

#kill the daemon-DOWNTIME

pskill -KILL dtnd && echo $i daemon close time: $(date +%T.%N)

sleep $(( $RANDOM % 160 + 1 ))

done

#launch the deamon and register the agent to receive a final response from
the RECEIVER

lxterminal -e dtnd configuration_sender.toml && echo 21 daemon open time:
$(date +%T.%N)

lxterminal -e dtn-tool exchange ws://localhost:8080/ws dtn://SENDER/
/home/pizw2/dtn7-go-0.7.1/

sleep $(( $RANDOM % 160 + 1 ))

#after a total of time period in rang of (1 to x) seconds uptime terminate
the daemon

pskill -KILL dtnd && echo 21 daemon close time: $(date +%T.%N)
```

Το παραπάνω script είναι όμοιο με αυτό του Πειράματος 1, μόνο που αντί για χρόνο sleep 160 δευτερόλεπτα, έχει “\$((\$RANDOM % 160 + 1))” που μεταφράζεται σε μια τιμή από 1 έως 160 δευτερόλεπτα.

Π.2 Αναλυτικά Αποτελέσματα Πειραμάτων

Π2.1 Πείραμα 1

Sleep time	Round Trip Time	Standard Deviation (for RTT)	Delivery Ratio
10	19.86588444	1.130294755	0.8
20	39.80252394	0.796609632	0.9
40	79.88817493	1.039431709	0.9
80	159.6613886	0.906784841	0.9
160	319.2844178	0.649031689	0.95

Πίνακας 2. Μετρήσεις Πειράματος 1

Π2.2 Πείραμα 2

1kB	Sleep Time	Round Trip Time	Standard Deviation (for RTT)	Delivery Ratio
	10	19.871	0.795	0.8
	20	39.492	1.048	0.9
	40	79.933	1.107	0.9
	80	159.781	1.130	0.95
	160	319.021	0.510	0.95

Πίνακας 3. Μετρήσεις Πειράματος 2, 1kB

10kB	Sleep Time	Round Trip Time	Standard Deviation (for RTT)	Delivery Ratio
	10	19.980	1.428	0.8
	20	39.651	0.927	0.85
	40	80.031	0.928	0.85
	80	160.045	1.422	0.85
	160	319.153	0.538	0.9

Πίνακας 4. Μετρήσεις Πειράματος 2, 10kB

100kB	Sleep Time	Round Trip Time	Standard Deviation (for RTT)	Delivery Ratio
	10	20.392	0.750	0.8
	20	39.982	0.969	0.85
	40	80.302	1.139	0.85
	80	160.236	1.214	0.9
	160	319.536	0.730	0.9

Πίνακας 5. Μετρήσεις Πειράματος 2, 100kB

1MB	Sleep Time	Round Trip Time	Standard Deviation (for RTT)	Delivery Ratio
	10	21.003	0.771	0.65
	20	40.922	1.039	0.8
	40	81.137	1.142	0.85
	80	161.472	1.269	0.85
	160	320.526	0.851	0.9

Πίνακας 6. Μετρήσεις Πειράματος 2, 1MB

5MB	Sleep Time	Round Trip Time	Standard Deviation (for RTT)	Delivery Ratio
	10	-	-	-
	20	46.379	1.670	0.6
	40	86.536	1.847	0.75
	80	168.161	7.564	0.85
	160	328.561	5.245	0.9

Πίνακας 7. Μετρήσεις Πειράματος 2, 5MB

10MB	Sleep Time	Round Trip Time	Standard Deviation (for RTT)	Delivery Ratio
	10	-	-	-
	20	-	-	-
	40	95.570	2.425	0.65
	80	179.089	8.756	0.7
	160	335.510	4.958	0.8

Πίνακας 8. Μετρήσεις Πειράματος 2, 10MB

Π2.3 Πείραμα 3

10s	Number of Bundles	Round Trip Time	Standard Deviation (for RTT)	Delivery Ratio
	1	19.866	1.130	0.800
	2	20.628	1.854	0.775
	4	21.349	2.633	0.588
	8	23.232	2.746	0.163
	16	28.714	3.444	0.053

Πίνακας 9. Μετρήσεις Πειράματος 3, 10s

20s	Number of Bundles	Round Trip Time	Standard Deviation (for RTT)	Delivery Ratio
	1	39.803	0.797	0.900
	2	42.191	2.011	0.850
	4	42.308	2.397	0.775
	8	46.880	3.010	0.531
	16	48.738	5.044	0.188

Πίνακας 10. Μετρήσεις Πειράματος 3, 20s

40s	Number of Bundles	Round Trip Time	Standard Deviation (for RTT)	Delivery Ratio
	1	79.888	1.039	0.900
	2	81.207	2.096	0.800
	4	83.618	5.834	0.800
	8	88.607	4.405	0.794
	16	94.424	6.425	0.553

Πίνακας 11. Μετρήσεις Πειράματος 3, 40s

80s	Number of Bundles	Round Trip Time	Standard Deviation (for RTT)	Delivery Ratio
	1	159.661	0.907	0.900
	2	159.983	1.321	0.850
	4	162.697	2.388	0.775
	8	166.058	3.635	0.800
	16	174.471	5.764	0.700

Πίνακας 12. Μετρήσεις Πειράματος 3, 80s

160s	Number of Bundles	Round Trip Time	Standard Deviation (for RTT)	Delivery Ratio
	1	319.284	0.649	0.950
	2	321.888	5.775	0.900
	4	323.409	4.713	0.813
	8	326.116	3.812	0.800
	16	334.843	3.050	0.747

Πίνακας 13. Μετρήσεις Πειράματος 3, 160s

Π2.4 Πείραμα 4

EPIDEMIC					
Time range of Sleep-Awake	Average Sleep time	Round Trip Time	Standard Deviation(for RTT)	Delivery Ratio	Overhead
1-10	5.742350736	8.506285737	4.571900587	0.5	1.1
1-20	11.12147248	10.2171842	8.086401786	0.55	1.090909091
1-40	21.63716512	21.92483405	21.7040667	0.7	1.166666667
1-80	39.52973372	33.28813696	33.84153241	0.75	1.333333333
1-160	70.338	45.34496083	43.35220508	0.8	1.5

Πίνακας 14. Μετρήσεις Πειράματος 4, EPIDEMIC

PRoPHET					
Time range of Sleep-Awake	Average Sleep time	Round Trip Time	Standard Deviation(for RTT)	Delivery Ratio	Overhead
1-10	6.362948701	9.622082191	8.702240573	0.4	0.625
1-20	11.43734433	12.83315865	12.79843094	0.5	0.7
1-40	19.08780346	18.6400271	21.32194623	0.7	0.857142857
1-80	44.83390812	30.81028226	40.5068984	0.8	1
1-160	69.941	56.59499712	66.25841005	0.8	0.9375

Πίνακας 15. Μετρήσεις Πειράματος 4, PRoPHET

DTLSR					
Time range of Sleep-Awake	Average Sleep time	Round Trip Time	Standard Deviation(for RTT)	Delivery Ratio	Overhead
1-10	6.239897898	8.513692833	6.579813512	0.35	0
1-20	10.39419538	15.7855826	16.42733219	0.45	0
1-40	20.84911184	23.23502257	21.61553807	0.85	0
1-80	38.44248666	46.26869741	45.50063353	0.8	0.0625
1-160	83.648	63.29410004	70.42334724	0.85	0.111111111

Πίνακας 16. Μετρήσεις Πειράματος 4, DTLSR

SPRAY AND WAIT					
Time range of Sleep-Awake	Average Sleep time	Round Trip Time	Standard Deviation(for RTT)	Delivery Ratio	Overhead
1-10	5.6321312	8.42352352	4.970919309	0.45	1.090909091
1-20	11.39324654	9.43221321	6.052997279	0.55	1.1
1-40	22.09231133	22.18543542	10.47405931	0.65	1.166666667
1-80	40.43916436	34.13232189	38.1589534	0.8	1.166666667
1-160	71.48348382	43.92854354	74.9822618	0.8	1.5

Πίνακας 17. Μετρήσεις Πειράματος 4, SPRAY AND WAIT