



ΔΙΕΘΝΕΣ
ΠΑΝΕΠΙΣΤΗΜΙΟ
ΤΗΣ ΕΛΛΑΔΟΣ

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΗΛΕΚΤΡΟΝΙΚΩΝ
ΣΥΣΤΗΜΑΤΩΝ

ΠΡΟΓΡΑΜΜΑ ΜΕΤΑΠΤΥΧΙΑΚΩΝ ΣΠΟΥΔΩΝ
ΕΥΦΥΕΙΣ ΤΕΧΝΟΛΟΓΙΕΣ ΔΙΑΔΙΚΤΥΟΥ - WEBINTELLIGENCE

**Βελτίωση της Απόδοσης του 6LoWPAN Δικτύου με τη Χρήση
της Τεχνικής Διόρθωσης Σφαλμάτων FEC**

ΜΕΤΑΠΤΥΧΙΑΚΗ ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

της

Ji Chenyang

Επιβλέπων : Χατζημίσιος Περικλής
Καθηγητής
Τμήμα Μηχανικών Πληροφορικής & Ηλεκτρονικών Συστημάτων, ΔΙΠΑΕ

Συνεπιβλέποντες: Georgios Z. Papadopoulos and Amaury Bruniaux, IMT Atlantique,
France

Θεσσαλονίκη, Φεβρουάριος 2022

Η σελίδα αυτή είναι σκόπιμα λευκή.



ΔΙΕΘΝΕΣ
ΠΑΝΕΠΙΣΤΗΜΙΟ
ΤΗΣ ΕΛΛΑΔΟΣ

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ
ΗΛΕΚΤΡΟΝΙΚΩΝ ΣΥΣΤΗΜΑΤΩΝ

ΠΡΟΓΡΑΜΜΑ ΜΕΤΑΠΤΥΧΙΑΚΩΝ ΣΠΟΥΔΩΝ
ΕΥΦΥΕΙΣ ΤΕΧΝΟΛΟΓΙΕΣ ΔΙΑΔΙΚΤΥΟΥ – WEB
INTELLIGENCE

Βελτίωση της Απόδοσης του 6LoWPAN Δικτύου με τη Χρήση της Τεχνικής Διόρθωσης Σφαλμάτων FEC

ΜΕΤΑΠΤΥΧΙΑΚΗ ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

της

Ji Chenyang

Επιβλέπων : Χατζημίσσιος Περικλής
Καθηγητής
Τμήμα Μηχανικών Πληροφορικής & Ηλεκτρονικών Συστημάτων, ΔΙΠΑΕ

Συνεπιβλέποντες: Georgios Z. Papadopoulos and Amaury Bruniaux, IMT Atlantique,
France

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή

(Υπογραφή)

(Υπογραφή)

(Υπογραφή)

.....
Όνομα Επώνυμο

.....
Όνομα Επώνυμο

.....
Όνομα Επώνυμο

Θεσσαλονίκη, Φεβρουάριος 2022

(Υπογραφή)

.....

© 2022 All rights reserved

Περίληψη

Τα τελευταία χρόνια όλο και περισσότερες συσκευές συνδέονται στο Διαδίκτυο και μεταξύ τους. Αυτό το φαινόμενο δημιούργησε πολλές καινούργιες περιπτώσεις χρήσης (use cases), μία εκ των οποίων είναι το Βιομηχανικό Διαδίκτυο των Πραγμάτων (Industrial Internet of Things - IIoT). Για τέτοιες περιπτώσεις χρήσης υπάρχουν ειδικές απαιτήσεις όπως υψηλή αξιοπιστία και χαμηλή καθυστέρηση. Το πρότυπο IEEE 802.15.4-2015 ανταποκρίνεται σε αυτές τις απαιτήσεις αλλά απαιτείται η χρήση των τεχνικών που περιγράφονται στο έγγραφο RFC 4944 για να αντιμετωπιστεί ο περιορισμός στο μέγεθος των πακέτων. Το RFC 4944 προβλέπει τον τεμαχισμό των πακέτων με το μειονέκτημα η εν λόγω μέθοδος να είναι πολύ ευάλωτη σε χαμένα δεδομένα επειδή απαιτεί επιτυχή αποστολή όλων των κομματιών (fragments) για να συναρμολογήσει το πακέτο. Για τη βελτίωση της απόδοσης, προτείνεται η τεχνική LoRaFEC η οποία χρησιμοποιεί την τεχνική διόρθωσης σφαλμάτων Forward Error Correction (FEC) που δίνει τη δυνατότητα ανάκτησης χαμένων δεδομένων από τον παραλήπτη. Η παρούσα διπλωματική έχει στόχο να προτείνει μία νέα μέθοδο που χρησιμοποιεί την τεχνική FEC για τη βελτίωση της απόδοσης και να τη συγκρίνει με άλλες υπάρχουσες μεθόδους. Για αυτή τη σύγκριση χρησιμοποιήθηκε ο 6Tisch προσομοιωτής και πραγματοποιήθηκε σύγκριση της απόδοσης βάσει της αξιοπιστίας του δικτύου, της καθυστέρησης και της επιπλέον επιβάρυνσης στο δίκτυο. Η χρήση της τεχνικής FEC οδηγεί σε αύξηση της αξιοπιστίας αλλά και της επιπλέον επιβάρυνσης στο δίκτυο, πράγμα που προκαλεί επιπλέον καθυστέρηση. Για αυτό το λόγο η παρούσα διπλωματική πραγματοποιεί μια λεπτομερή αξιολόγηση της απόδοσης και προτείνει τη βέλτιστη μέθοδο αναλόγως τις απαιτήσεις που έχουν οι διάφορες περιπτώσεις χρήσης.

Λέξεις Κλειδιά: Βιομηχανικό IIoT, 6LoWPAN, RFC 4944, IPv6, FEC

Η σελίδα αυτή είναι σκόπιμα λευκή.

Abstract

The last few years many devices are connected to the Internet and between them. This phenomenon is creating many new use cases and one of them. Such use cases have high demands in terms of reliability and latency. In order to fulfill these demands, the IEEE 802.15.4-2015 standard is used but it also requires the techniques included in the RFC 4944 to solve its payload size limitation by using fragmentation and compression. RFC 4944 is very sensitive to data loss, because it requires the reception of all fragments to successfully reassemble a packet. In order to improve reliability, FEC methods are suggested since they improve reliability by recovering lost fragments. The current thesis presents and analyzes the state-of-the-art proposals using FEC methods. We further introduce a new method, named LoRaFEC. We implement these proposals and conducted simulations by using the 6TiSCH simulator. We compare these methods based on reliability, latency and additional overhead. The use of FEC results in a tradeoff between reliability and additional overhead, which causes higher latency. For this reason, the current thesis conducts a thorough performance assessment and proposes the most efficient method according to the demands of the various use cases.

Keywords: Industrial IoT, 6LoWPAN, RFC 4944, IPv6, FEC

Η σελίδα αυτή είναι σκόπιμα λευκή.

Πίνακας περιεχομένων

1	Εισαγωγή.....	1
1.1	Περιγραφή της κατάστασης.....	1
1.2	Αντικείμενο διπλωματικής.....	2
1.2.1	Συνεισφορά.....	2
1.3	Οργάνωση κειμένου.....	3
2	Θεωρητικό υπόβαθρο.....	4
2.1	RFC 4944 Fragment Forwarding.....	4
2.2	Προβλήματα του RFC 4944 FF.....	8
2.3	Minimal Fragment Forwarding (MFF).....	12
3	Σχετικές εργασίες.....	17
3.1	Forward Error Correction (FEC).....	17
3.2	XORFEC.....	19
3.3	RFEC.....	20
3.4	NCFEC.....	21
3.5	LoRaFFEC.....	23
4	Προτεινόμενη λύση.....	25
4.1	LoRaFEC.....	25
4.2	Τελική έκδοση.....	28
4.2.1	Καθορισμός αριθμός των <i>redundant fragments</i>	28
4.2.2	Περιεχόμενο των <i>redundant fragments</i>	31
4.2.3	Μέθοδοι βελτίωσης.....	41
5	Προσομοίωση.....	45
5.1	Εργαλείο προσομοίωσης - 6Tisch προσομοιωτή.....	45
5.1.1	Σύγκριση προσομοιωτών.....	47
5.2	Παράμετροι προσομοίωσης.....	48

5.3	Πλατφόρμα Grid'5000.....	49
6	Συμπεράσματα.....	52
6.1	Παράμετροι αξιολόγησης.....	52
6.2	Οργάνωση πειραμάτων.....	52
6.3	Αποτελέσματα.....	53
6.4	Σύνοψη συμπερασμάτων αξιολόγησης.....	56
7	Επίλογος.....	57
7.1	Σύνοψη και συμπεράσματα.....	57
7.2	Μελλοντικές επεκτάσεις.....	58
	Βιβλιογραφία.....	61

1

Εισαγωγή

Στην σημερινή εποχή, όλες και περισσότερες συσκευές συνδέονται στο δίκτυο και αυτό οδήγησε στην αύξηση του ενδιαφέροντος για το Internet of Things (IoT) [1], στα ελληνικά Διαδίκτυο των Πραγμάτων. Η έννοια αυτή πρώτο εμφανίστηκε στα τέλη της δεκαετίας του 1990 και το βασικό χαρακτηριστικό του IoT είναι όλες οι συσκευές συνδέονται μεταξύ τους ώστε ο χρήστης μπορεί να τα ελέγχει μέσω μιας μηχανής, υπολογιστή ή κινητού.

Λόγω του μεγάλου αριθμού των συσκευών που συνδέονται στο δίκτυο, χρησιμοποιείται το Internet Protocol Version 6 (IPv6) [2] για την ταυτοποίηση τους. Οι πιο γνωστές περιπτώσεις χρήσης του IoT είναι το έξυπνο σπίτι, η έξυπνη πόλη και το Βιομηχανικό IoT (Industrial Internet of Things - IIoT).

Το IIoT [3] έχει στόχο την αύξηση της παραγωγικότητας και της αξιοπιστίας σε βιομηχανικούς χώρους με δυνατότητες που παρέχουν οι συσκευές IoT, δηλαδή η παρακολούθηση και ο έλεγχος σε πραγματικό χρόνο. Σε τέτοιες περιπτώσεις χρήσης είναι πολύ σημαντικό να υπάρχει αξιοπιστία των αποστολών πληροφορίας και μικρή καθυστέρηση, πράγμα που είναι δύσκολο να επιτευχθεί σε ασύρματη επικοινωνία με περιορισμένες ικανότητες.

1.1 Περιγραφή της κατάστασης

Για τις απαιτήσεις του IIoT προτείνεται η χρήση του προτύπου IEEE 802.15.4-2015 [4] που δημοσιεύτηκε το 2016. Για κάθε αποστολή ή λήψη δεδομένων, υπάρχει ξεχωριστό κανάλι για να μη υπάρχει σύγκρουση όταν πραγματοποιούνται ταυτόχρονες μεταδόσεις.

Όμως αυτό το πρωτόκολλο δεν είναι συμβατό με το IPv6. Το IPv6 απαιτεί σύνδεση με δυνατότητα να μεταφέρει πακέτα τουλάχιστον 1280 bytes, ενώ το IEEE 802.15.4-2015 έχει μέγιστο όριο 127 bytes για κάθε πακέτο. Η ομάδα Internet Engineering Task Force (IETF) έχει καθορίσει το IPv6 over Low power Wireless Personal Area Networks (6LoWPAN) στο έγγραφο RFC 4944 Fragment Forwarding (RFC 4944 FF) [5] και δίνει λύση σε αυτό το πρόβλημα.

Όμως το πρόβλημα με το RFC 4944 είναι ότι πρέπει να παραδοθούν χωρίς σφάλματα μετάδοσης όλα τα κομμάτια (fragments) ενός πακέτου για να θεωρηθεί επιτυχής η αποστολή του. Πράγμα που είναι δύσκολο να επιτευχθεί λόγω της κατάστασης του δικτύου. Το πρόβλημα είναι πιο έντονο όταν τα πακέτα που στέλνονται έχουν μεγάλο μέγεθος όπως τα πακέτα που περιέχουν πολυμεσική πληροφορία. Αυτό προκαλεί πρόβλημα στο IIoT και σε άλλες περιπτώσεις χρήσης που απαιτούν μεγάλη αξιοπιστία της μετάδοσης και μικρή καθυστέρηση παράδοσης.

Για την λύση του προβλήματος, οι ερευνητές έχουν προτείνει την τεχνική Forward Error Correction (FEC) που δημιουργεί και στέλνει πλεονασμικά ουσιαστικά δίνει την δυνατότητα ανάκτησης χαμένων fragments από τον τελικό προορισμό. Επιπλέον έχουν προταθεί τεχνικές σχετικά με εναλλακτικούς τρόπους τεμαχισμού των πακέτων [6].

1.2 Αντικείμενο διπλωματικής

Η παρούσα διπλωματική έχει στόχο (μελετώντας τις υπάρχουσες λύσεις που σχετίζονται με την χρήση της τεχνικής FEC για τη βελτίωση της απόδοσης του δικτύου), να προτείνει μια νέα λύση, την LoRaFEC.

1.2.1 Συνεισφορά

Η συνεισφορά της διπλωματικής συνοψίζεται ως εξής:

1. Παρουσίαση του RFC 4944 και αναλυτική περιγραφή του προβλήματος
2. Παρουσίαση των παρακάτω λύσεων που προσπαθούν με την τεχνική FEC να βελτιώσουν την απόδοση:
 - a. XORFEC
 - b. RFEC
 - c. NCFEC
 - d. LoRaFFEC
3. Αναλυτική περιγραφή της δικής μας πρότασης LoRaFEC
4. Σύγκριση της απόδοσης όλων των προτεινόμενων τεχνικών
5. Περιγραφή των εργαλείων που χρησιμοποιήθηκαν

6. Συμπεράσματα και προτάσεις για μελλοντική επέκταση

1.3 Οργάνωση κειμένου

Το κείμενο είναι οργανωμένο σε 7 κεφάλαια. Το 1^ο κεφάλαιο πραγματοποιεί μια σύντομη εισαγωγή του θέματος. Στο 2^ο κεφάλαιο περιέχεται η περιγραφή της τεχνολογίας που χρησιμοποιείται αυτή την στιγμή και τα προβλήματα της που χρειάζονται βελτίωση. Στο 3^ο κεφάλαιο, περιγράφουμε τις τεχνικές που έχουν προταθεί για τη λύση του προβλήματος. Στο 4^ο κεφάλαιο παρουσιάζουμε την δική μας πρόταση LoRaFEC και περιγράφουμε τη διαδικασία μέχρι την τελική έκδοση. Στο 5^ο κεφάλαιο παρουσιάζουμε τις παραμέτρους προσομοίωσης που χρησιμοποιήσαμε. Στο 6^ο κεφάλαιο περιέχει τη σύγκριση της απόδοσης την οποία πραγματοποιήσαμε χρησιμοποιώντας τα αποτελέσματα των προσομοιώσεων που εκτελέσαμε. Στο τελευταίο κεφάλαιο, παρουσιάζουμε μια σύνοψη των συμπερασμάτων της διπλωματικής εργασίας και κάποιες προτάσεις για μελλοντική επέκταση.

2

Θεωρητικό υπόβαθρο

Λόγω περιορισμού της μέγιστης μονάδας μετάδοσης (Maximum Transmission Unit – MTU), δηλαδή ένα πακέτο που στέλνεται δεν μπορεί να ξεπεράσει το 127 bytes, υπάρχει η ανάγκη για τεμαχισμό των πακέτων. Λαμβάνοντας υπόψη αυτή την ανάγκη, ο οργανισμός IETF έχει προτείνει το έγγραφο RFC 4944 Fragment Forwarding (RFC 4944 FF).

Παρακάτω γίνεται η αναλυτική περιγραφή του προτύπου και των προβλημάτων που έχει και πρέπει να επιλυθούν. Επιπλέον παρουσιάζεται μια άλλη ιδέα που πρόκειται να προτυποποιηθεί και δίνει λύσεις σε κάποια από τα προβλήματα.

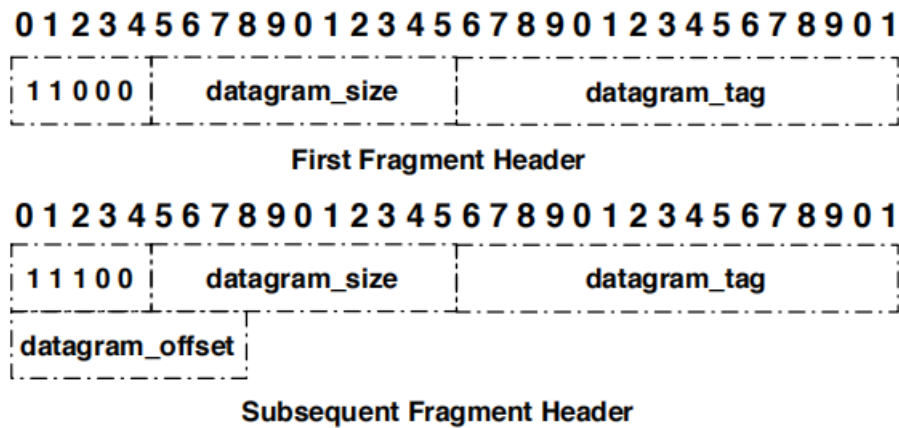
2.1 RFC 4944 Fragment Forwarding

Όπως έχουμε περιγράψει προηγουμένως, τα IPv6 πρωτόκολλα έχουν την απαίτηση οι συνδέσεις του 2ου επιπέδου να μπορούν να διαχειρίζονται πακέτα με μέγεθος τουλάχιστον 1280 bytes, αλλά οι συνδέσεις του IEEE 802.15.4-2015 μπορεί να δεχθούν μόνο μέχρι 127 bytes. Για την λύση αυτού του προβλήματος, το 6LoWPAN, χρησιμοποιώντας την τεχνική συμπίεσης σύμφωνα με την RFC 6282 [7] και άλλες τεχνικές που ορίζονται το RFC 4944, επιτρέπει σε τέτοιες συνδέσεις να επεξεργάζονται πακέτα μέχρι και 2048 bytes. Το RFC 4944 είναι το πρότυπο που καθορίζει λειτουργίες του επιπέδου προσαρμογής για να μπορεί να γίνει επιτυχή αποστολή των IPv6 πακέτων μέσω συνδέσεων του δικτύου που ακολουθεί το IEEE 802.15.4-2015. Οι λειτουργίες που καθορίζει είναι ο τεμαχισμός, η συναρμολόγηση και η προώθηση των fragments.

Η κεφαλίδα του IPv6 και του User Datagram Protocol (UDP) έχουν αρχικά μέγεθος 40 bytes και 8 bytes, αντιστοίχως. Οι κεφαλίδες συμπίεζονται σύμφωνα με το RFC 6282. Αν μετά την

συμπύεση, το μέγεθος του πακέτου είναι μεγαλύτερο από 127 bytes, τότε εφαρμόζεται τεμαχισμός του πακέτου.

Όταν γίνεται τεμαχισμός του πακέτου, όλα τα fragments που προκύπτουν περιέχουν την κεφαλίδα Medium Access Control (MAC) και το πρώτο fragment περιέχει την συμπιεσμένη κεφαλίδα του IPv6 και του UDP.



Εικόνα 1. Δομή του fragment

Η δομή της κεφαλίδας του fragment φαίνεται στην εικόνα 1. Κάθε fragment έχει μέγεθος πολλαπλάσιο των 8 bytes. Η λειτουργία του κάθε πεδίου είναι η εξής:

1. Τύπος του fragment : Είναι τα πρώτα 5 bits στην κεφαλίδα. Όταν τα fragments προέρχονται από τεμαχισμό πακέτου, το 11000 συμβολίζει ότι είναι το πρώτο fragment ενώ το 11100 συμβολίζει πως είναι από τα επόμενα fragments. Υπάρχουν και άλλοι τύποι και φαίνονται αναλυτικά στην εικόνα 2.
2. Μέγεθος πακέτου: Έχει μέγεθος 11 bits και συμβολίζει το μέγεθος του αρχικού πακέτου πριν γίνει ο τεμαχισμός. Η τιμή σε αυτό το πεδίο παραμένει σταθερή για τα fragments που προέρχονται από το ίδιο πακέτο.
3. Tag του πακέτου: Έχει μέγεθος 16 bits και παραμένει σταθερή για τα fragments που προέρχονται από το ίδιο πακέτο. Χρησιμοποιείται για να ξεχωρίσει από ποια πακέτα προέρχονται τα fragments. Αυξάνεται από τον αποστολέα σταδιακά με κάθε τεμαχισμένο πακέτο και η μέγιστη τιμή που μπορεί να πάρει είναι 65.535.
4. Offset του πακέτου: Αυτό το πεδίο έχει μέγεθος 8 bits και συμβολίζει τη θέση του fragment στο αρχικό πακέτο. Η μέτρηση της θέσης αρχίζει από την αρχή του πακέτου και είναι πάντα πολλαπλάσιο του 8 bytes. Για το πρώτο fragment το offset ισούται με μηδέν.

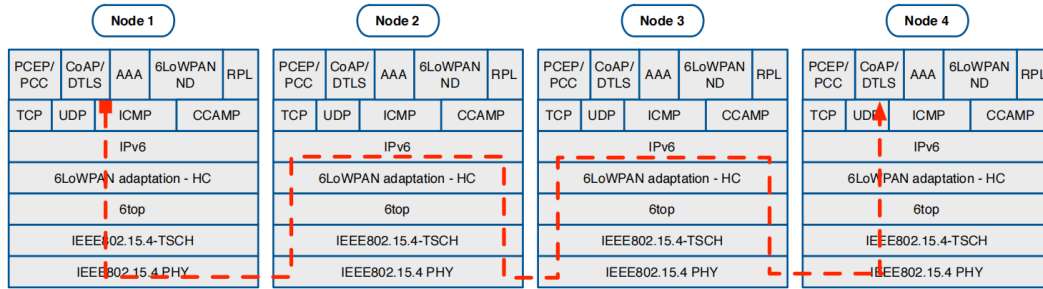
Pattern	Header Type
00 xxxxxx	NALP - Not a LoWPAN frame
01 000001	IPv6 - Uncompressed IPv6 Addresses
01 000010	LOWPAN_HC1 - LOWPAN_HC1 compressed IPv6
01 000011	reserved - Reserved for future use
...	reserved - Reserved for future use
01 001111	reserved - Reserved for future use
01 010000	LOWPAN_BC0 - LOWPAN_BC0 broadcast
01 010001	reserved - Reserved for future use
...	reserved - Reserved for future use
01 111110	reserved - Reserved for future use
01 111111	ESC - Additional Dispatch byte follows
10 xxxxxx	MESH - Mesh Header
11 000xxx	FRAG1 - Fragmentation Header (first)
11 001000	reserved - Reserved for future use
...	reserved - Reserved for future use
11 011111	reserved - Reserved for future use
11 100xxx	FRAGN - Fragmentation Header (subsequent)
11 101000	reserved - Reserved for future use
...	reserved - Reserved for future use
11 111111	reserved - Reserved for future use

Εικόνα 2. Τύποι του fragment

Επιπλέον για το μέγεθος του πακέτου, αυτή η τιμή μπορεί να χρησιμοποιηθεί όταν το fragment που φτάνει δεν είναι το πρώτο fragment. Ενημερώνει τον κόμβο για τον αριθμό των fragments που θα πρέπει να φτάσουν και έτσι ο κόμβος μπορεί να αποφασίσει πότε πρέπει να γίνει η συναρμολόγηση και πόσο αποθηκευτικό χώρο θα χρειαστεί.

Καθώς τα πακέτα είναι κομμένα σε fragments, πρέπει να γίνει προώθηση στον τελικό προορισμό. Το RFC 4944 καθορίζει 2 τρόπους προώθησης, το Mesh under και το Route over.

- Στο Mesh under η προώθηση πραγματοποιείται στο επίπεδο προσαρμογής και σε αυτή την λειτουργία δεν συμμετέχει το επίπεδο IP, δηλαδή δε δρομολογούνται πακέτα με χρήση IP διευθύνσεων. Τα πακέτα αποστέλλονται σε δίκτυα που διαθέτουν πολλούς ενδιάμεσους κόμβους και σε κάθε κόμβο δεν απαιτείται συναρμολόγηση των πακέτων επειδή οι διευθύνσεις του επιπέδου σύνδεσης περιέχονται σε όλα τα fragments. Με αυτή την ενσωμάτωση, κάθε fragment δρομολογείται ξεχωριστά.
- Στο Route over η δρομολόγηση πραγματοποιείται στο επίπεδο δικτύου και απαιτείται συναρμολόγηση και τεμαχισμό των πακέτων σε κάθε ενδιάμεσο κόμβο. Ο λόγος είναι επειδή στο RFC 4944 μόνο το πρώτο fragment περιέχει την IPv6 διεύθυνση στην κεφαλίδα του, και αυτή η πληροφορία χρησιμοποιείται για τη δρομολόγηση. Ο κόμβος πρέπει να συναρμολογήσει πρώτα το πακέτο για να πάρει την πληροφορία αυτή και μετά να δρομολογήσει σωστά τα πακέτα. Η διαδικασία αυτή αυξάνει την καθυστέρηση γιατί εκτελείται επεξεργασία των πακέτων σε κάθε ενδιάμεσο κόμβο.

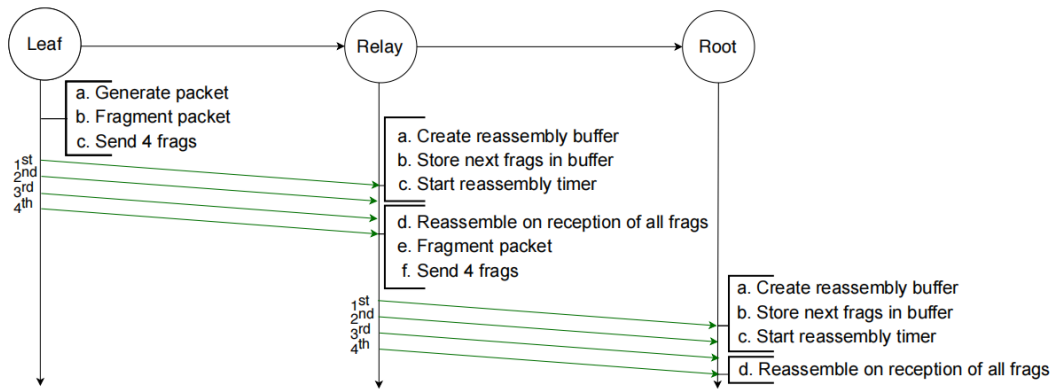


Εικόνα 3. Route over forwarding operation

Αν και συγκριτικά το Mesh under προκαλεί μικρότερο φόρτο στο δίκτυο και προσφέρει μικρότερο χρόνο καθυστέρησης, αυτό που επιλέγεται περισσότερο είναι το Route over. Το Route over μας επιτρέπει να χρησιμοποιήσουμε πρωτόκολλα δρομολόγησης, όπως για παράδειγμα το RPL που είναι το καθορισμένο πρωτόκολλο δρομολόγησης για βιομηχανικές περιπτώσεις χρήσης. Οι μέθοδοι που παρουσιάζουμε στην παρούσα διπλωματική βασίζονται στο Route over.

Για την συναρμολόγηση στο Route over, όταν ένας κόμβος λαμβάνει το πρώτο fragment, ελέγχει αν το tag του πακέτου που περιέχεται στο fragment είναι συνδεδεμένο με κάποια άλλα fragments που έχει λάβει. Αν το fragment που έλαβε είναι το πρώτο fragment με αυτό το tag, δημιουργεί στην μνήμη μία καινούργια αποθήκη με μέγεθος που ισούται με το μέγεθος του πακέτου. Αν ήδη υπάρχει το tag, τότε απλώς αποθηκεύει το fragment στην αντίστοιχη αποθήκη. Η θέση που θα αποθηκευτεί το fragment καθορίζεται από το offset του πακέτου. Όταν ο κόμβος λαμβάνει το πρώτο fragment με συγκεκριμένο tag, ξεκινάει και ένας μετρητής που αν γίνει 0 και δεν έχει γεμίσει η αποθήκη του συγκεκριμένου πακέτου, θεωρεί πως η αποστολή απέτυχε και διαγράφει τα fragments από την αποθήκη. Όταν η αποθήκη έχει γεμίσει, τότε ο κόμβος ξεκινάει την διαδικασία συναρμολόγησης. Μετά την συναρμολόγηση, ελέγχει τον τελικό προορισμό και ενεργεί αναλόγως. Σε περίπτωση που δεν είναι ο τελικός προορισμός, τεμαχίζει ξανά το πακέτο και το προωθεί στον επόμενο κόμβο.

Στην εικόνα 4 παρουσιάζουμε ένα παράδειγμα με μία αποστολή του IPv6 πακέτο από το αποστολέα στον τελικό προορισμό με έναν ενδιάμεσο κόμβο. Θεωρούμε πως το μέγεθος του πακέτου που πρέπει να αποσταλεί είναι μεγαλύτερο από το όριο και πρέπει να κοπεί σε 4 fragments. Στο παράδειγμα, και οι δύο κόμβοι που λαμβάνουν το πακέτο, δηλαδή ο ενδιάμεσος κόμβος και ο τελικός προορισμός, δημιουργεί αποθήκη όταν λαμβάνει τα fragments. Θεωρούμε σε αυτό το παράδειγμα ότι όλες οι αποστολές και οι συναρμολογήσεις έγινε επιτυχής και έφτασε με επιτυχία το πακέτο στον τελικό προορισμό.



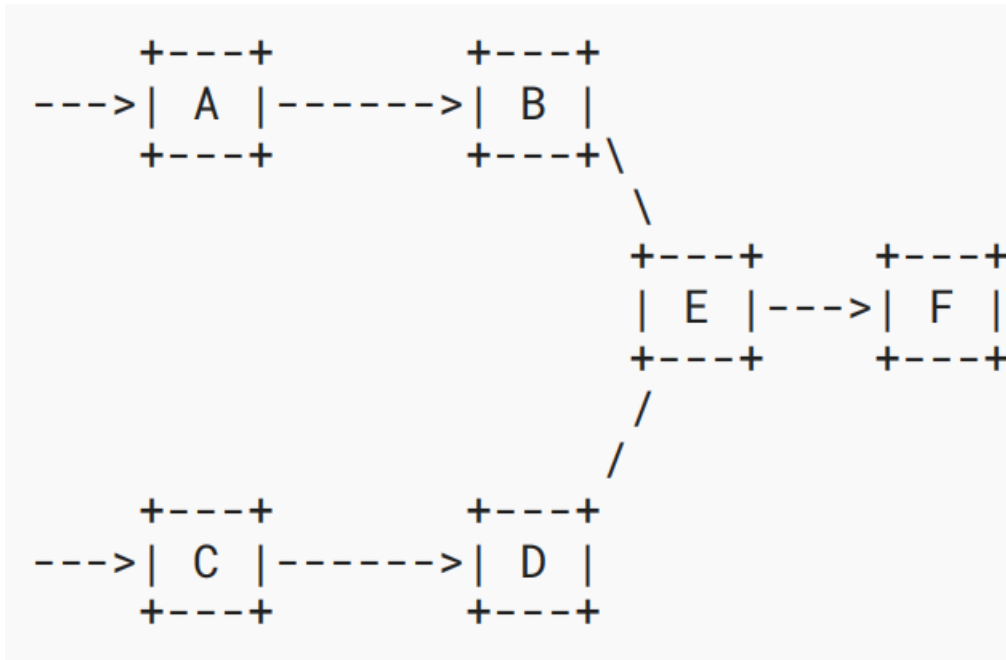
Εικόνα 4. Παράδειγμα επιτυχής αποστολής

2.2 Προβλήματα του RFC 4944 FF

Το δίκτυο που προβλέπεται στο RFC 4944 αποτελείται από πολλούς ενδιάμεσους κόμβους και στο Route over σε κάθε στάση (ενδιάμεσο κόμβο) που φτάνει το πακέτο πρέπει να γίνει συναρμολόγηση και τεμαχισμός ολόκληρου του πακέτου. Τα προβλήματα που δημιουργεί αυτός ο τρόπος λειτουργίας αφορά πολλά πεδία [8] και αναλυτικά θα τα περιγράψουμε παρακάτω.

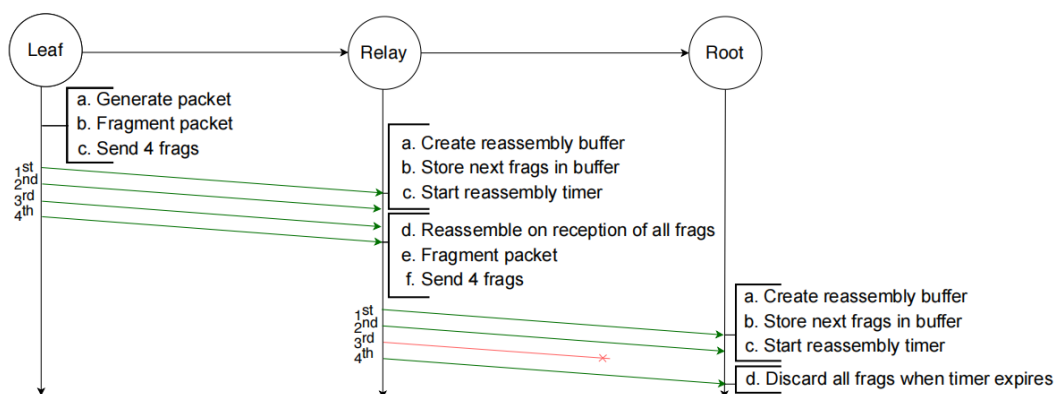
Καταρχήν, όταν ένας κόμβος λαμβάνει για πρώτη φορά ένα fragment με συγκεκριμένο tag, αρχίζει έναν αντίστροφο μετρητή που όταν φτάσει σε συγκεκριμένο χρόνο διαγράφει τα fragments με συγκεκριμένο tag. Αυτό γίνεται επειδή οι κόμβοι έχουν περιορισμένο αποθηκευτικό χώρο και απαιτείται αρχικοποίηση του για να μπορεί να δεχτεί καινούργια fragments.

Όμως η λειτουργία αυτή επηρεάζει αρνητικά την απόδοση του δικτύου. Όπως έχουμε περιγράψει προηγουμένως για να μπορεί ένας κόμβος να αρχίσει την διαδικασία της συναρμολόγησης, πρέπει να έχει φτάσει όλα τα fragment ενός πακέτου. Ακόμα και ένα fragment να λείπει, δεν μπορεί ο κόμβος να εκτελέσει την διαδικασία της συναρμολόγησης. Λαμβάνοντας υπόψη τα χαρακτηριστικά της ασύρματης επικοινωνίας, όπου τα πακέτα που αποστέλλονται είναι πολύ πιθανό να έχουμε λάθη μετάδοσης λαμβάνονται, μπορούμε να πούμε ότι αυτό το χαρακτηριστικό ενεργεί αρνητικά στην απόδοση του δικτύου.



Εικόνα 5. Το Bottleneck πρόβλημα

Το πρόβλημα είναι πιο έντονο για δίκτυα με κόμβους με διαθέτουν περιορισμένο αποθηκευτικό χώρο. Για αυτές τις περιπτώσεις ο μετρητής θα έχει μικρές τιμές για ένα fragment επειδή πρέπει να αρχικοποιείται συχνά ο αποθηκευτικός χώρος του κόμβου. Επίσης παρατηρείται έντονο αυτό το πρόβλημα σε αποστολές πακέτων που μεσολαβεί μεγάλος αριθμός ενδιάμεσων στάσεων. Όσο αυξάνεται ο αριθμός των κόμβων τόσο αυξάνεται και η πιθανότητα αποτυχίας μιας αποστολής.



Εικόνα 6. Παράδειγμα μη επιτυχής αποστολής

Άλλο πρόβλημα του RFC 4944 είναι ο περιορισμένος αποθηκευτικός χώρος των κόμβων. Όπως περιγράψαμε παραπάνω, ο κόμβος εκτελεί συναρμολόγηση μόνο όταν έχουν φτάσει όλα τα fragments. Αυτό το χαρακτηριστικό απαιτεί από τους κόμβους να αφήσουν μεγάλο χώρο στην μνήμη καθώς το μέγεθος του πακέτου μπορεί να φτάσει μέχρι και 2048 bytes.

Λαμβάνοντας υπόψη τους πιθανούς περιορισμούς των συσκευών του IoT (μνήμη, επεξεργαστική ικανότητα κτλ.), θεωρούμε πως ένας τέτοιος ενδιάμεσος κόμβος μπορεί να

αποθηκεύσει μόνο ένα ή δύο IPv6 πακέτα. Περιγράφοντας το πρόβλημα με ένα παράδειγμα, έστω ότι έχουμε τα πακέτα A και B που θα φτάσουν στον ίδιο κόμβο. Το πρώτο fragment που φτάνει είναι του πακέτου A και ο κόμβος ενεργεί ανάλογα. Δημιουργεί ένα χώρο στη μνήμη με μέγεθος ίσο με το μέγεθος του πακέτου A. Μετά λαμβάνει και κάποια fragments του πακέτου A και τα αποθηκεύει στην αντίστοιχη θέση. Τώρα, φτάνει ένα fragment του πακέτου B στον ίδιο κόμβο. Σε αυτή την περίπτωση τα fragments του πακέτου A δεν έχουν φτάσει όλα και δεν μπορεί να γίνει συναρμολόγηση, ο κόμβος χρειάζεται να αφήσει χώρο στη μνήμη με μέγεθος ίσο με το μέγεθος του πακέτου B και άρα αναγκαστικά πρέπει να διαγράψει τα fragments του πακέτου A. Αυτό το πρόβλημα μειώνει την απόδοση του δικτύου επειδή αυξάνει την πιθανότητα να χαθεί ένα πακέτο.

Το δίκτυο μπορεί να αποτελείται από πολλούς ενδιάμεσους κόμβους, όπως έχουμε αναφέρει προηγουμένως, και καταλήγουν σε κάποιο προορισμό. Η δομή του δικτύου μπορεί να δημιουργήσει συμφόρηση σε κόμβους κοντά στο τελικό προορισμό, επειδή είναι οι μόνοι που συνδέονται με τον τελικό και δεν υπάρχει άλλες διαθέσιμες διαδρομές. Το πρόβλημα στο RFC 4944 είναι ότι δεν έχει ένα μηχανισμό που διαχειρίζεται το φόρτο του δικτύου και οι κόμβοι δεν έχουν την δυνατότητα να μεταβιβάσουν το φόρτο σε άλλο κόμβο σε περίπτωση συμφόρησης.

Άλλο πρόβλημα που δημιουργεί η δομή του δικτύου είναι η διαδρομή που ακολουθούν τα πακέτα μπορεί να είναι κύκλος, δηλαδή να μη μπορούν να φτάσουν στον τελικό προορισμό. Αυτό το φαινόμενο δημιουργεί επιπλέον φόρτο στο δίκτυο. Για την λύση αυτών των προβλημάτων προτείνεται η χρήση των πρωτοκόλλων ελέγχου ροής.

Η καθυστέρηση που προκαλείται από τον τεμαχισμό και συναρμολόγηση των πακέτων σε κάθε ενδιάμεσο κόμβο αποτελεί ακόμη ένα πρόβλημα του RFC 4944. Η σχετική έρευνα έχει δείξει πως αυτή η διαδικασία δημιουργεί σημαντική αύξηση στην καθυστέρηση. Οι καθυστερήσεις προκαλούνται επίσης από τη συναρμολόγηση αλλά και από τον τεμαχισμό των πακέτων. Σε αυτή την διαδικασία όμως, δεν είναι μόνο η καθυστέρηση το πρόβλημα, αλλά και οι επιπλέον απαιτήσεις (επεξεργαστική ισχύ) που χρησιμοποιήθηκαν από τον κάθε ενδιάμεσο κόμβο. Ως αποτέλεσμα, όσο αυξάνεται ο αριθμός των ενδιάμεσων κόμβων, τόσο αυξάνεται και ο χρόνος καθυστέρησης.

Συγκρίνοντας τη μέθοδο που απαιτεί συναρμολόγηση με αυτή που δεν απαιτεί, το αποτέλεσμα φαίνεται στην εικόνα 7. Γίνεται ξεκάθαρο, η μέθοδος που δεν απαιτεί συναρμολόγηση ξοδεύει λιγότερα μονάδες χρόνου. Σε περίπτωση που δεν απαιτείται συναρμολόγηση, μπορεί να υποστηριχθούν και παράλληλες αποστολές των fragments του ίδιου πακέτου.

	Source	Forwarder 1	Forwarder 2	Destination
T=0	FFF			
T=1	FF(F)	F		
T=2	F(F)	FF		
T=3	(F)	FFF		
T=4		FF(F)	F	
T=5		F(F)	FF	
T=6		(F)	FFF	
T=7			FF(F)	F
T=8			F(F)	FF
T=9			(F)	FFF

(a) RFC 4944.

	Source	Forwarder 1	Forwarder 2	Destination
T=0	FFF			
T=1	FF(F)	F		
T=2	FF	(F)	F	
T=3	FF		(F)	F
T=4	F(F)	F		F
T=5	F	(F)	F	F
T=6	F		(F)	FF
T=7	(F)	F		FF
T=8		(F)	F	FF
T=9			(F)	FFF

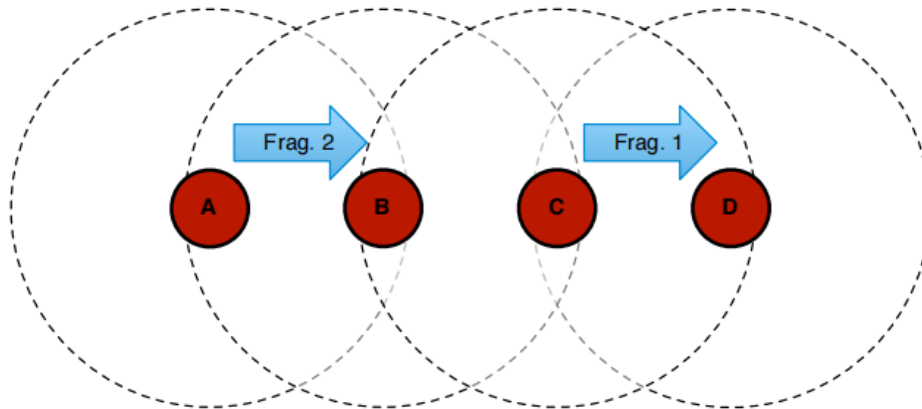
(b) Streamlining with a window of size 1.

	Source	Forwarder 1	Forwarder 2	Destination
T=0	FFF			
T=1	FF(F)	F		
T=2	FF	(F)	F	
T=3	F(F)	F	(F)	F
T=4	F	(F)	F	F
T=5	(F)	F	(F)	FF
T=6		(F)	F	FF
T=7			(F)	FFF
T=8				
T=9				

(c) Streamlining with a larger window.

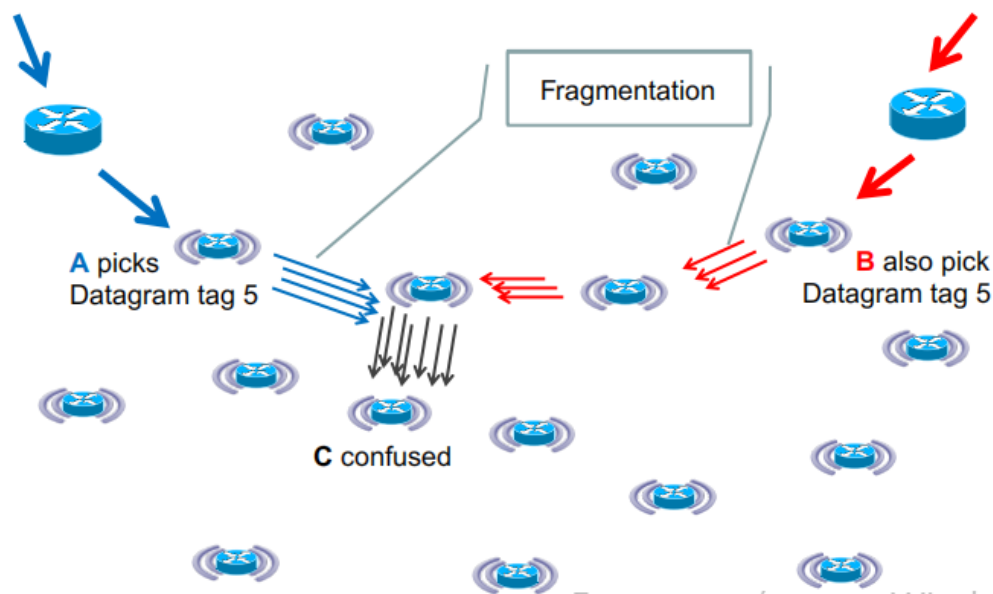
Εικόνα 7. Παράδειγμα σύγκριση αποστολής

Σε δίκτυα που αποτελούνται από κόμβους, υπάρχει ένα γνωστό πρόβλημα που είναι το πρόβλημα των κρυφών και εκτεθειμένων κόμβων. Το πρόβλημα είναι όπως φαίνεται στην εικόνα 9. Στο παράδειγμα που δίνουμε, η αποστολή που θέλουμε να πραγματοποιήσουμε από κόμβο A προς κόμβο B μπορεί να επηρεάζεται από την αποστολή από κόμβο C προς κόμβο D, επειδή οι κόμβοι B και C βρίσκονται στην ίδια εμβέλεια. Για τη λύση του προβλήματος προτείνεται η χρήση μηχανισμού ελέγχου ροής. Για παράδειγμα, αν θέσουμε χρόνο αναμονής μεταξύ των αποστολών fragments στον ίδιο κόμβο, μειώνεται η πιθανότητα σύγκρουσης.



Εικόνα 9. Παράδειγμα εκτεθειμένων κόμβων

Το τελευταίο πρόβλημα που θα περιγράψουμε είναι το πρόβλημα του tag. Όπως αναφέραμε παραπάνω, η IPv6 διεύθυνση του αποστολέα και του τελικού προορισμού υπάρχει μόνο στο πρώτο fragment. Διαχωρίζουμε τα fragments που ακολουθούν με το tag του πακέτου που κουβαλάνε. Το πρόβλημα που έχουμε εδώ είναι το tag του πακέτου για έναν fragment είναι μοναδικό μόνο όταν συνδυάζεται με την IPv6 διεύθυνση του αποστολέα και του τελικού προορισμού. Για αυτό το λόγο, για fragments που δεν είναι το πρώτα και προέρχονται από διαφορετικό αποστολέα, υπάρχει πιθανότητα να συναντήσουμε το ίδιο tag στον ενδιάμεσο κόμβο. Σε αυτή την περίπτωση, ο ενδιάμεσος κόμβος δεν έχει την ικανότητα να ξεχωρίσει σε ποιο πακέτο ανήκουν τα fragments και υπάρχει σύγχυση σχετικά με την αποθήκευση, συναρμολόγηση και προώθηση.



Εικόνα 10. Παράδειγμα πρόβλημα του tag

Στην εικόνα 10 παρουσιάζεται ένα παράδειγμα για τέτοιες περιπτώσεις όπου υπάρχουν δύο ροές, η μία συμβολίζεται με κόκκινο βέλος και η άλλη με μπλε βέλος. Οι ροές περνάνε από τους κόμβους A και B. Το πρόβλημα συμβαίνει όταν και οι δύο ροές επιλέγουν τον ίδιο tag, το 5 στο συγκεκριμένο παράδειγμα. Τότε όταν γίνεται προώθηση προς τον κόμβο C, αυτός ο κόμβος μπερδεύεται και δεν μπορεί να ξεχωρίσει τα fragments και άρα δεν μπορεί να καταλάβει ποιο πρέπει να είναι ο επόμενος ενδιαμέσος κόμβος.

Συνοψίζοντας, το χαρακτηριστικό του RFC 4944 που απαιτεί λήψη όλων των fragments για τη συναρμολόγηση του πακέτου οδηγεί στο φαινόμενο, όσο αυξάνονται οι ενδιάμεσοι κόμβοι τόσο να αυξάνεται η πιθανότητα αποτυχίας παράδοσης. Για τον λόγο αυτό υπάρχει η ανάγκη να βρούμε ένα τρόπο να μειώσουμε την πιθανότητα αποτυχίας και προτείνεται η χρήση FEC. Οι τεχνικές FEC έχουν την ικανότητα να ανακτήσουν χαμένα fragments αυξάνοντας σημαντικά την πιθανότητα για μια επιτυχή αποστολή.

2.3 Minimal Fragment Forwarding (MFF)

Η τεχνική MFF [9] δίνει λύσεις για κάποια από τα προβλήματα που παρουσιάσαμε προηγουμένως και αποτελεί μια πρόταση που σύντομα θα προτυποποιηθεί. Τα προβλήματα που λύνει η πρόταση είναι τα εξής:

1. Πρόβλημα διαγραφή του πακέτου λόγω μετρητή και μνήμης
2. Καθυστέρηση από συναρμολόγηση και τεμαχισμό σε κάθε ενδιάμεσο κόμβο
3. Πρόβλημα του tag

Στην τεχνική MFF δεν απαιτείται συναρμολόγηση και τεμαχισμό σε κάθε ενδιάμεσο κόμβο γιατί χρησιμοποιεί την τεχνική Virtual Reassembly Buffer (VRB). Η VRB έχει σκοπό την

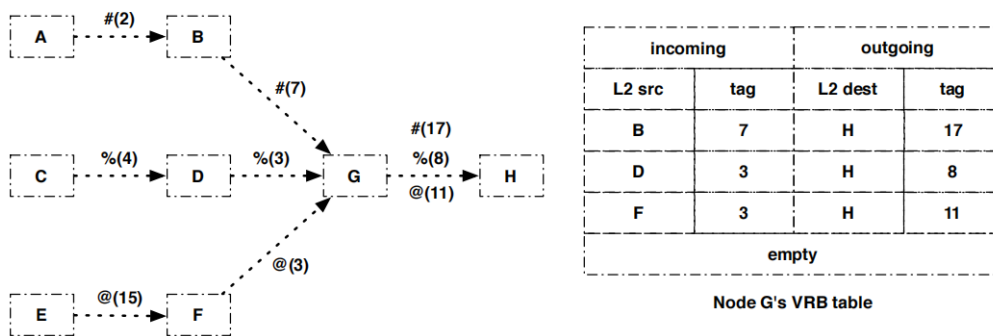
αύξηση της απόδοσης και την μείωση της καθυστέρησης. Επιτρέπει τα fragments να κινηθούν ελεύθερα στους ενδιάμεσους κόμβους χωρίς να απαιτείται συναρμολόγηση και τεμαχισμός. Για την υλοποίηση αυτής της λειτουργίας, η VRB ενεργεί με παρόμοιο τρόπο με τον πίνακα δρομολόγησης που συναντάμε και ο κάθε κόμβος αποθηκεύει κάποιες πληροφορίες για την δρομολόγηση που συγκρατείται στο πρώτο fragment.

Κάθε κόμβος με χρήση VRB συγκρατεί σε έναν πίνακα την διεύθυνση του αποστολέα, του παραλήπτη και το tag. Ο συνδυασμός αυτός είναι μοναδικός για κάθε πακέτο. Το VRB αξιοποιεί την ιδιότητα όλα τα fragments του ίδιου πακέτου ακολουθούν την ίδια διαδρομή και έτσι μπορεί να αναγνωρίσει με ακρίβεια την προέλευση των fragments.

Η αναλυτική λειτουργία περιγράφεται παρακάτω:

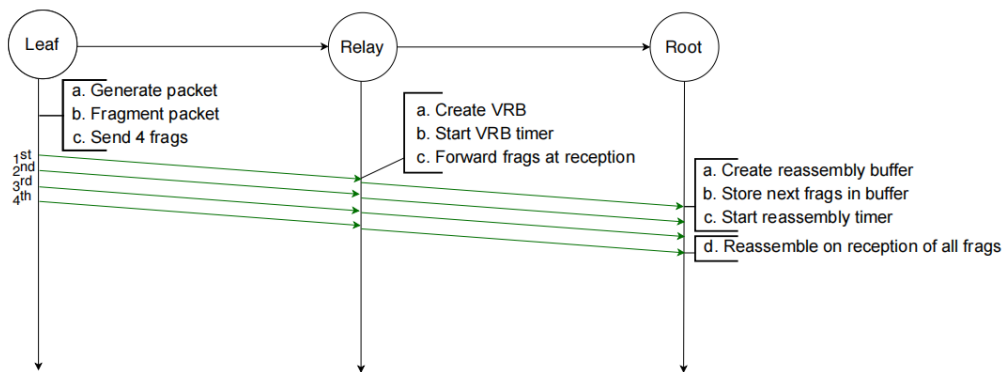
1. Πρώτα ελέγχεται αν το fragment που λαμβάνει είναι το πρώτο που περιέχει τις διευθύνσεις. Αν δεν είναι και δεν υπάρχουν αντίστοιχες καταγραφές τότε διαγράφει κατευθείαν το fragment και όλα τα ακόλουθα fragments που προέρχονται από τον ίδιο αποστολέα. Ο λόγος είναι σε τέτοιες περιπτώσεις το πρώτο fragment έχει χαθεί και δεν υπάρχει δυνατότητα συναρμολόγησης στον τελικό προορισμό.
2. Εάν το είναι το πρώτο fragment του πακέτου, τότε δημιουργεί μια καινούργια εγγραφή στον πίνακα με τις παρακάτω πληροφορίες
 - Το tag του πακέτου
 - Καινούργιο tag που θα χρησιμοποιείται όταν προωθείται προ την επόμενη στάση
 - MAC διεύθυνση του αποστολέα
 - MAC διεύθυνση της επόμενης στάσης
 - Μετρητή για την διαγραφή των fragments
3. Προωθεί το fragment στην επόμενη στάση.

Ένα πράγμα που πρέπει να τονίσουμε εδώ είναι η MAC διεύθυνση της επόμενης στάσης εξαρτάται από το πρώτο fragment και παραμένει σταθερή για όλα τα fragments που ακολουθούν. Αξιοποιώντας την ιδιότητα αυτή, με την αποθήκευση των διευθύνσεων της προηγούμενης στάσης, για κάθε fragment έχουμε ένα μοναδικό συνδυασμό και μπορούμε να τα ξεχωρίσουμε.



Εικόνα 11. VRB πίνακα

Δίνοντας ένα παράδειγμα στην εικόνα 11, έχουμε τον κόμβο G που λαμβάνει ένα fragment tag 7 από τον κόμβο B και δύο fragments από τον κόμβο D και κόμβο F με ίδιο tag 3. Όπως φαίνεται στον πίνακα VRB, δεν υπάρχει το πρόβλημα της σύγχυσης επειδή ο συνδυασμός της διεύθυνσης προηγούμενης στάσης και του tag είναι μοναδικός για κάθε fragment. Επιπλέον, για δυο fragments που έχουν τον ίδιο tag, όταν προωθούνται στην επόμενη στάση, ο κόμβος αλλάζει το tag. Αυτό δεν ισχύει μόνο για τα fragments που έχουν ίδιο tag αλλά για όλα τα fragments.



Εικόνα 12. Επιτυχή λειτουργία MFF

Όπως φαίνεται στην εικόνα 12, όταν ένας ενδιαμέσος κόμβος λαμβάνει ένα fragment, δημιουργεί καινούργια εγγραφή στο πίνακα ή αν βλέπει ότι υπάρχει εγγραφή στον πίνακα για το συγκεκριμένο πακέτο, και μετά το προωθεί κατευθείαν στην επόμενη στάση, χωρίς να περιμένει για όλα τα fragment για να εκτελέσει συναρμολόγηση. Στον τελικό προορισμό, ο κόμβος δημιουργεί χώρο στη μνήμη όπως στο RFC 4944 και συναρμολογεί το πακέτο αφού έχουν φτάσει όλα τα fragments. Με αυτόν τον τρόπο, κάθε ενδιαμέσος κόμβος φαίνεται να έχει κάνει συναρμολόγηση και τεμαχισμό, ενώ δεν απαιτείται να χρησιμοποιήσει επιπλέον πόρους.

Αν και το VRB λύνει κάποια από τα προβλήματα που έχουμε αναφέρει προηγουμένως, έχει κάποια μειονεκτήματα που πρέπει να παρουσιάσουμε. Η τεχνική MFF λύνει προβλήματα

όπως περιορισμένη μνήμη, καθυστέρηση λόγω συναρμολόγησης σε κάθε στάση και το πρόβλημα του tag, αλλά υπάρχουν 2 προβλήματα στα οποία δε δίνει λύσεις. Αυτά είναι τα εξής:

1. Δεν υπάρχει ανάκτηση των χαμένων fragments
2. Όλα τα fragments ακολουθούν την ίδια διαδρομή

Επίσης υπάρχει ακόμα το πρόβλημα της μνήμης, επειδή και για τη δημιουργία και συγκράτηση των VRB απαιτείται χρήση επιπλέον μνήμης. Σε σύγκριση με το RFC 4944 σίγουρα καταναλώνει λιγότερη μνήμη επειδή δεν απαιτείται χώρος στην μνήμη ίσος με το αρχικό μέγεθος του πακέτου. Ακόμα και με τη χρήση της VRB, ο αριθμός των εγγραφών που μπορεί ένας κόμβος να κρατήσει είναι περιορισμένος. Σε περιπτώσεις που υπάρχουν πολλές ταυτόχρονες αποστολές προς τον ίδιο κόμβο, πάλι υπάρχει μεγάλη πιθανότητα να χαθεί το πακέτο. Τέτοια προβλήματα συναντάμε σε κόμβους που είναι κοντά στο τελικό προορισμό και στον τελικό προορισμό, επειδή οι διαδρομές μέσω αυτών των κόμβων είναι καθορισμένες και δεν υπάρχουν πολλές επιλογές αλλαγής.

Ένα σημαντικό πρόβλημα που δε λύνει το MFF είναι ότι όσο αυξάνεται ο αριθμός των κόμβων τόσο αυξάνεται και η πιθανότητα αποτυχίας αποστολής για ένα πακέτο. Ο λόγος όπως είχαμε αναφέρει προηγουμένως, είναι επειδή για την συναρμολόγηση του πακέτου, απαιτείται η λήψη όλων των fragments.

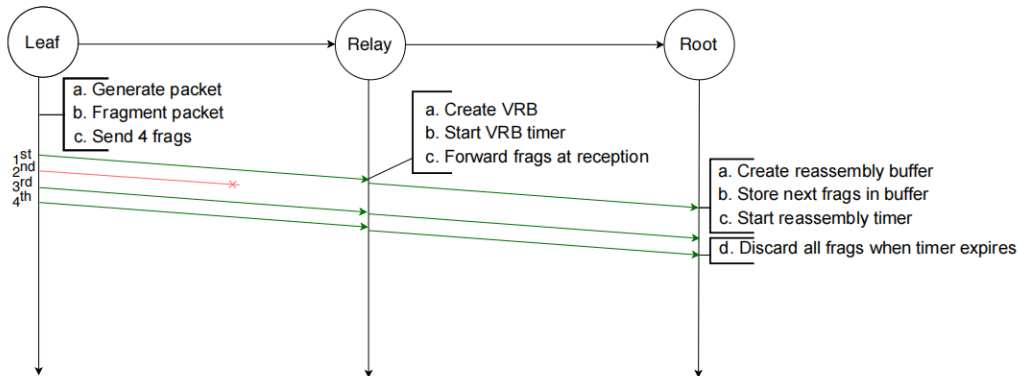
Η τεχνική MFF, με τον τρόπο προώθησης που χρησιμοποιεί, σίγουρα μειώνει την καθυστέρηση στο δίκτυο αλλά δεν εγγυάται η επιτυχή αποστολή του πακέτου. Στην λειτουργία της MFF, ένα χαμένο fragment παρατηρείται μόνο όταν είναι το πρώτο fragment. Όταν παρατηρείται, απορρίπτει όλα τα ακόλουθα fragments και έτσι δεν αποφεύγονται οι άσκοπες αποστολές για πακέτα που δεν μπορούν να συναρμολογηθούν στον τελικό προορισμό.

Στο παράδειγμα που περιγράφεται στην εικόνα 13 παρατηρούμε ότι ο ενδιάμεσος κόμβος εκτελεί την προώθηση των fragments και ας χαθούν κάποια από τα fragments. Επειδή το fragment που χάθηκε δεν είναι το πρώτο, μ δε μπορεί να παρατηρήσει αυτό το φαινόμενο.

Σε αυτές τις περιπτώσεις, εκτελούνται άσκοπες αποστολές στο δίκτυο επειδή αυτά τα πακέτα σίγουρα δεν μπορεί να συναρμολογηθούν στον τελικό προορισμό. Επιπλέον, όταν φτάσουν κάποια από τα fragments στον τελικό προορισμό, ξοδεύονται μνήμη και υπολογιστικοί πόροι για το συγκεκριμένο πακέτο επειδή ο τελικός κόμβος πρέπει να αποθηκεύσει τα fragments και να δημιουργήσει τον αντίστοιχο μετρητή.

Στο παράδειγμα της εικόνας 13 ο ενδιάμεσος κόμβος δεν παρατήρησε πως το δεύτερο fragment έχει χαθεί και ακόμα στέλνει στον τελικό προορισμό τα fragments που έλαβε. Οι

αποστολές μετά από το χαμένο fragment θεωρούνται άσκοπες αποστολές σε αυτή την περίπτωση.



Εικόνα 13. Μη επιτυχή λειτουργία MFF

Το MFF δεν λύνει το πρόβλημα, εάν χαθεί έστω και ένα fragment, τότε η αποστολή του πακέτου θεωρείται αποτυχία. Δεν υπάρχει μεθόδους ανάκτησης των χαμένων fragments, για αυτό το λόγο δόθηκε η πρόταση να χρησιμοποιηθεί τεχνικές Forward Error Correction (FEC).

Ένα άλλο πρόβλημα λόγω δομής των fragments είναι ότι τα fragments που προέρχονται από το ίδιο πακέτο μπορούν μόνο να ακολουθήσουν μία διαδρομή [10]. Αυτό είναι επειδή μόνο το πρώτο fragment κουβαλάει IPv6 κεφαλίδα που χρησιμοποιείται στη δρομολόγηση. Λαμβάνοντας υπόψη αυτόν τον περιορισμό, δεν μας δίνει την δυνατότητα για δρομολόγηση πολλαπλών διαδρομών εάν δεν στέλνουμε πολλές φορές το πρώτο fragment.

Άλλο πρόβλημα που σχετίζεται με την ασφάλεια του δικτύου είναι οι επιθέσεις Denial-of-Service (DoS) που μπορεί να δεχθεί το δίκτυο. Από την περιγραφή της λειτουργίας μπορούμε να δούμε πως ένα τέτοιο δίκτυο είναι πολύ ευάλωτο για τέτοιες επιθέσεις. Η χειρότερη περίπτωση είναι ο αποστολέας στέλνει πολλά άσκοπα πρώτα fragments χωρίς να στέλνει τα ακόλουθα fragments. Σε αυτήν την περίπτωση, σε όλους τους κόμβους μπορεί να υπάρξει το πρόβλημα της κατανάλωσης της διαθέσιμης μνήμης επειδή δεν μπορεί να συγκρατήσει στο VRB τόσες πολλές εγγραφές. Ειδικά στον τελικό προορισμό που πρέπει να αφήσει στην μνήμη αρκετό χώρο για κάθε πρώτο fragments που λαμβάνει.

3

Σχετικές εργασίες

Όπως περιγράψαμε προηγουμένως, η τεχνική MFF δε δίνει λύση για το πρόβλημα που έχουμε όταν έστω και ένα fragment χάνεται στη διάρκεια της αποστολής(τότε η αποστολή δεν είναι επιτυχής, επειδή δεν μπορεί να γίνει συναρμολόγηση του πακέτου).

Μία πρόταση που έχει γίνει για τη λύση του προβλήματος είναι η χρήση της τεχνικής FEC με την οποία έχουμε τη δυνατότητα να ανακτήσουμε τα χαμένα fragments έτσι ώστε να αυξήσουμε τη πιθανότητα επιτυχής αποστολής και άρα να βελτιώσουμε την απόδοση του δικτύου.

3.1 Forward Error Correction (FEC)

Η FEC είναι μια τεχνική που χρησιμοποιείται για τον έλεγχο σφαλμάτων στη αποστολή δεδομένων σε κανάλια επικοινωνίας που έχουν πολλές ενοχλήσεις ή όταν η ποιότητα της σύνδεσης είναι χαμηλή. Και στις δύο περιπτώσεις, η χρήση της FEC έχει στόχο τη διόρθωση σφαλμάτων και την ανάκτηση των χαμένων δεδομένων για την αύξηση της απόδοσης του δικτύου. Το κυρίως χαρακτηριστικό της FEC είναι ότι ο αποστολέας κωδικοποιεί το μήνυμα με συγκεκριμένο τρόπο, κυρίως χρησιμοποιείται μια τεχνική Error Correction Coding (ECC) .

Με την χρήση της FEC, δημιουργείται ένας πλεονασμός που βασίζεται στα υπάρχοντα δεδομένα. Αυτός ο πλεονασμός, όταν φτάσει στον τελικό προορισμό, μπορεί να χρησιμοποιηθεί από τον παραλήπτη να εντοπίσει, με περιορισμένη δυνατότητα, κάποια σφάλματα στο δεδομένα ή κάποια δεδομένα που έχουν χαθεί κατά την αποστολή. Αυτά τα λάθη μπορεί να εμφανίζονται οπουδήποτε. Σε δίκτυο που δεν εφαρμόζεται η FEC, τα δεδομένα μπορούν να διορθωθούν μόνο με νέα αποστολή. Συνεπώς χρησιμοποιώντας

τεχνικές FEC δεν απαιτείται νέα αποστολή καθώς απαιτείται περισσότερος χρόνος επεξεργασίας της πληροφορίας και επιβαρύνεται το δίκτυο αλλά αυξάνεται η αξιοπιστία του δικτύου.

Περιπτώσεις που εφαρμόζεται η τεχνική FEC είναι όταν η επανάληψη της αποστολής ξοδεύει περισσότερους πόρους από τον πλεονασμό ή όταν η επανάληψη της αποστολής είναι αδύνατη, όπως για παράδειγμα όταν έχουμε πολλαπλές αποστολές προς διαφορετικούς προορισμούς ή απαιτείται μεγάλο χρονικό διάστημα για την επαναποστολή [11].

Η τεχνική FEC μπορεί να χρησιμοποιηθεί στο σύνολο των δεδομένων ή σε τμήματα τους, αλλά το τελικό αποτέλεσμα είναι ένας πλεονασμός διαφορετικός από τα υπάρχοντα δεδομένα. Ο πλεονασμός αυτός είτε στέλνεται ξεχωριστά ή εντάσσεται σε συγκεκριμένη θέση των δεδομένων. Ο αριθμός των σφαλμάτων που μπορεί να διορθώσει ή ο αριθμός των χαμένων δεδομένων που μπορεί να ανακτήσει ο πλεονασμός εξαρτάται από την κωδικοποίηση και το μέγεθος του. Σε γενικές γραμμές, όσο μεγαλύτερος πλεονασμός δημιουργείται τόσο περισσότερα λάθη μπορεί να διορθώσει (δυστυχώς αυξάνοντας αντίστοιχα τον επιπλέον φόρτο και την καθυστέρηση).

Από την περιγραφή της FEC είναι φανερό πως η δυνατότητα που παρέχει για ανάκτηση των χαμένων δεδομένων είναι αυτό που χρειάζεται το 6LoWPAN. Τα 6LoWPAN δίκτυα συνήθως έχουν χαμηλή ποιότητα σύνδεσης και είναι πολύ πιθανό να χαθούν τα δεδομένα που στέλνονται. Δυστυχώς αποτελεί σημαντικό πρόβλημα για τα RFC 4944 και MFF, επειδή και στα δύο ισχύει εάν χαθεί έστω και ένα fragment, η αποστολή δεν είναι επιτυχής. Προσθέτοντας την τεχνική FEC, έχουμε την δυνατότητα να ανακτήσουμε τα πακέτα αν χαθούν κατά την διάρκεια της αποστολής και συνεπώς μειώνουμε την επίπτωση του προβλήματος.

Υπάρχουν πολλές διαφορετικές τεχνικές του FEC και οι μέθοδοι που θα παρουσιάσουμε εδώ δημιουργούν πλεονασμό με επανάληψη ή χρησιμοποιούν μεθόδους κωδικοποίησης. Πιο συγκεκριμένα, έχουμε το RFEC που στέλνει με επανάληψη τα δεδομένα, το XORFEC που εφαρμόζει κωδικοποίηση στο σύνολο των δεδομένων και προσθέτει μοναδικό πλεονασμό, το NCFEC που κωδικοποιεί τα δεδομένα και LoRaFFEC που εφαρμόζει κωδικοποίηση μεταξύ των δεδομένων.

Αυτό που πρέπει να σημειωθεί είναι, όπως έχουμε αναφέρει προηγουμένως, η FEC κάνει ισορροπεί μεταξύ αξιοπιστίας και επιπλέον φόρτου και καθυστέρησης. Για την αποστολή του πακέτου, ένας κόμβος πρέπει να καταναλώσει ένα μέρος των πόρων του όπως ενέργεια, χρόνος, επεξεργαστική ικανότητα. Αυτό που χρειάζεται προσοχή είναι επειδή η τεχνική FEC δημιουργεί επιπλέον φόρτο στο δίκτυο, το μέρος που αυξάνεται να μη υπερβεί το όριο του δικτύου. Αν το υπερβεί, μετά έχουμε το πρόβλημα της συμφόρησης στους κόμβους. Για αυτό το λόγο, απαιτείται ιδιαίτερη προσοχή εάν θέλουμε να χρησιμοποιήσουμε την τεχνική FEC

σε δίκτυα που ήδη έχουν μεγάλο φόρτο ή διαθέτουν πολύ περιορισμένη χωρητικότητα. Η εργασία [12] προτείνει την απενεργοποίηση της FEC όταν η αύξηση του πλεονασμού δεν επηρεάζει θετικά την απόδοση.

Για την καλύτερη περιγραφή των πακέτων, πρέπει πρώτα να παρουσιάσουμε κάποιες έννοιες για τον τύπο των fragments όπως ακολούθως:

1. Data fragments, προέρχεται από τον τεμαχισμό των πακέτων και περιέχει δεδομένα του πακέτου
2. Redundant fragments, πλεονασμός που δημιουργείται από την εφαρμογή του μεθόδου κωδικοποίησης σε μέρος ή σύνολο των δεδομένων του πακέτου.
3. Encoded fragments, μια έννοια που χρησιμοποιείται μόνο στο NCFEC. Είναι τα fragments που δημιουργούνται μετά από κωδικοποίηση και αντικαθιστούν τα data fragments.

3.2 XORFEC

Αυτή η μέθοδος είναι μια επέκταση της μεθόδου Network Coding - Mesh Under Routing (NC - MUR) [13]. Σε αυτή την μέθοδο, χρησιμοποιείται ως μέθοδο κωδικοποίησης το Exclusive OR (XOR) για την δημιουργία του πλεονασμού. Ο πλεονασμός που δημιουργείται είναι ένα επιπλέον fragment, προέρχεται από την εφαρμογή του XOR όλων των data fragments. Ο αριθμός των redundant fragments παραμένει σταθερό και είναι πάντα ίσον με ένα. Ο πλεονασμός αυτός δίνει την δυνατότητα ανάκτησης ενός χαμένο fragment, πράγμα που αυξάνει σημαντικά την απόδοση του δικτύου.

Η ανάκτηση του χαμένου data fragment βασίζεται στο XOR, που είναι ένας δυαδικός τρόπος επεξεργασίας των δεδομένων. Αναλυτικά, τα αποτελέσματα που δίνει φαίνονται στην εικόνα 14.

		XOR
0	0	0
0	1	1
1	0	1
1	1	0

Εικόνα 14. XOR υπολογισμοί

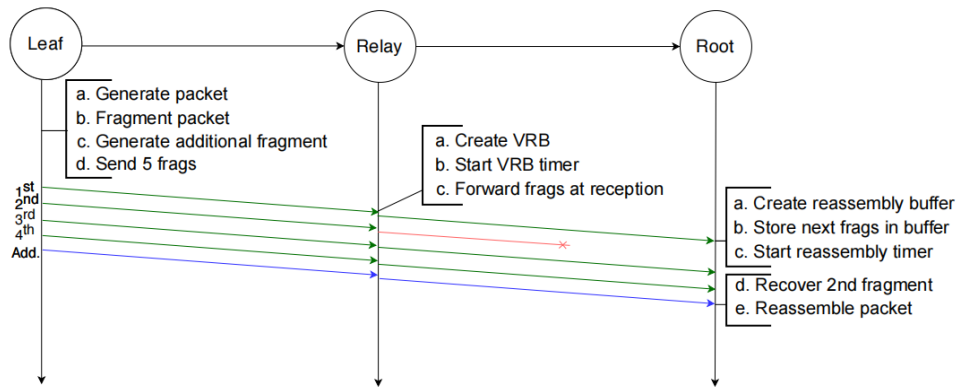
Ας δούμε ένα παράδειγμα για την ανάκτηση του χαμένου fragment. Έστω ένα πακέτο χωρίζεται σε 4 data fragments και δημιουργεί ένα redundant fragment ως πλεονασμό. Συμβολίζοντας με δυαδικές σειρές το περιεχόμενο των fragments και θεωρούμε ότι έφτασε στον τελικό προορισμό 3 data fragments και το redundant fragment. Εφαρμόζοντας την XOR στο σύνολο των fragments που έλαβε ο αποστολέας, μπορούμε να πάρουμε το χαμένο data fragment.

Όπως είχαμε αναφέρει στο παράγραφο του RFC 4944, για πακέτα που είναι μεγαλύτερο από 127 bytes απαιτείται τεμαχισμός του πακέτου σε fragments. Ο πλεονασμός, δηλαδή το redundant fragment στέλνεται μετά από την αποστολή των data fragments και το offset του fragment ορίζεται ως την συνέχεια του τελευταίου data fragment. Για παράδειγμα, έστω ένα πακέτο που έχει μέγεθος 450 bytes και το μέγεθος των fragments ορίζεται στο 90 bytes, τότε τα offset των fragments ορίζεται ως εξής, 0 - 90 - 180 - 270 - 360, και το offset του redundant fragment θα είναι 450.

Η λειτουργία του XORFEC είναι παρόμοια με αυτή της τεχνικής MFF(εικόνα 15). Όλοι οι ενδιάμεσοι κόμβοι στο δίκτυο διαθέτουν VRB που συγκρατεί για κάθε πακέτο πληροφορίες για δρομολόγηση και άρα τα fragments στο XORFEC δεν συναρμολογούνται σε κάθε στάση αλλά προωθούνται κατευθείαν στην επόμενη στάση.

Και στην περίπτωση του XORFEC, εάν το χαμένο fragment είναι το πρώτο fragment, απορρίπτονται όλα τα ακόλουθα fragments επειδή δεν υπάρχουν πληροφορίες δρομολόγησης. Αλλά εάν το fragment που χάνεται δεν είναι το πρώτο fragment, τότε συνεχίζεται η προώθηση. Στο τελικό προορισμό, εάν ο κόμβος συνειδητοποιεί ότι ο αποθηκευτικός χώρος έχει γεμίσει αλλά ένα από τα data fragments έχει αντικατασταθεί από redundant fragment, τότε εφαρμόζει XOR στο σύνολο των ληφθέντων fragments για να ανακτήσει το χαμένο fragment.

Στο παράδειγμα που παρουσιάζουμε στην εικόνα, το πακέτο χωρίζεται σε 4 data fragments , δημιουργείται 1 redundant fragment και στέλνονται με αυτή την σειρά. Όλα τα fragments έχουν φτάσει με επιτυχία στον ενδιάμεσο κόμβο, όπου δημιουργεί μία εγγραφή στον VRB αφού το πρώτο fragment έχει φτάσει. Μετά, προωθεί όλα τα ακόλουθα fragments χωρίς συναρμολόγηση και σύμφωνα με την εγγραφή του VRB. Βλέπουμε ότι κατά την αποστολή μεταξύ ενδιάμεσου κόμβου και τελικό προορισμό, χάθηκε το δεύτερο fragment. Ο τελικός προορισμός λαμβάνει το πρώτο fragment, δημιουργεί χώρο στην μνήμη ίσον με το μέγεθος του πακέτου και αποθηκεύει τα fragments που λαμβάνει. Όταν ο χώρος αυτός γεμίσει, αρχίζει την διαδικασία συναρμολόγησης. Πρώτα, συνειδητοποιεί πως υπάρχει χαμένο fragment και μετά εκτελώντας XOR υπολογισμοί, ανακτά το χαμένο fragment και γίνεται συναρμολόγηση. Αυτό που πρέπει να σημειωθεί είναι ότι με την μέθοδο XORFEC, υπάρχει δυνατότητα ανάκτηση μόνο ενός χαμένου fragment.



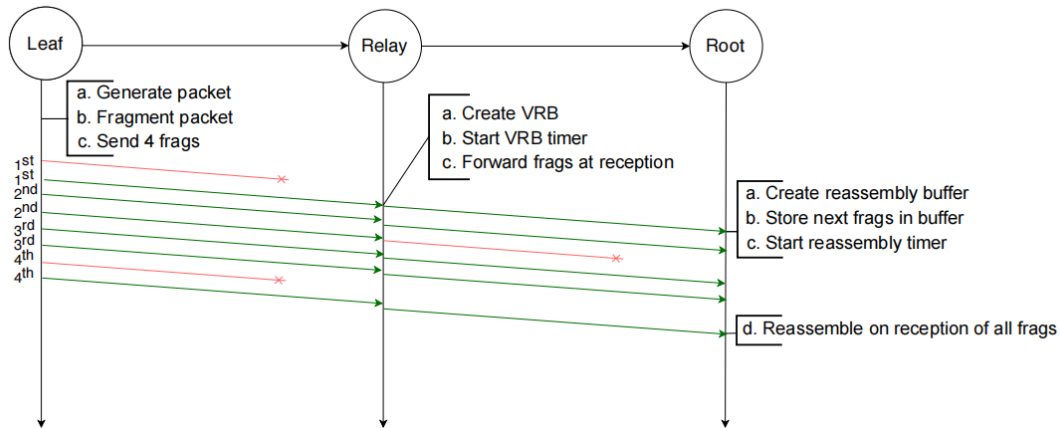
Εικόνα 15. Επιτυχή XOR λειτουργία

3.3 RFEC

Το RFEC είναι μία επέκταση της μεθόδου που περιγράφει στο [12] και η δημιουργία του πλεονασμού γίνεται με την επανάληψη της αποστολής των data fragments. Δεν χρησιμοποιεί μεθόδους κωδικοποίησης στα δεδομένα. αυτός ο τρόπος δημιουργίας του πλεονασμού είναι ο πιο απλός τρόπος και δεν έχει μεγάλη απαίτηση στην επεξεργαστική ικανότητα του κόμβου. Οι επαναλήψεις γίνονται διαδοχικά, δηλαδή επαναλαμβάνεται για κάθε data fragment ξεχωριστά.

Στο RFEC χρησιμοποιείται το VRB, άρα και σε αυτή την μέθοδο δεν απαιτείται συναρμολόγηση των πακέτων σε κάθε ενδιάμεσο κόμβο. Στην εργασία [12] υπάρχει η περιγραφή της μεθόδου, οι επαναλήψεις των αποστολών δεν ενεργοποιούνται στην αρχή αλλά μόνο όταν η απόδοση του δικτύου πέσει κάτω από συγκεκριμένο όριο. Όταν ενεργοποιείται η επανάληψη, οι αποστολές του πρωτοτύπου και της επανάληψης έχουν μεταξύ τους μια καθυστέρηση 3 δευτερολέπτων, για να μη χαθούν ταυτόχρονα και τα 2 fragments.

Η λειτουργία του RFEC φαίνεται στην εικόνα 16. Το παράδειγμα που δίνουμε στην εικόνα είναι μία αποστολή σε ένα δίκτυο με 2 στάσεις, την ενδιάμεση και την τελική. Με αυτή μέθοδο έχουμε την δυνατότητα ανάκτησης και για το πρώτο fragments. Όπως φαίνεται στην εικόνα, το πρωτότυπο πρώτο fragments χάθηκε κατά την διάρκεια της αποστολής. Με την επανάληψη, τελικά φτάνει το πρώτο fragment στην ενδιάμεση στάση.



Εικόνα 16. Επιτυχή RFEC λειτουργία

3.4 NCFEC

Η NCFEC χρησιμοποιεί μια τεχνική που ονομάζεται κωδικοποίηση δικτύου στο σύνολο των data fragments ενός πακέτου και δημιουργεί τα encoded fragments. Ο αριθμός των encoded fragments είναι μεγαλύτερο από τον αριθμό των data fragments. Η αποστολή ενός πακέτου θεωρείται επιτυχής όταν ο αριθμός των encoded fragments που έχει φτάσει στον τελικό προορισμό είναι ίσον με τον αριθμό των data fragments. Η τεχνική της κωδικοποίησης του δικτύου πρώτα παρουσιάστηκε το 2000 από το [14] και αρχικά έχει στόχο την βελτίωση της αξιοπιστίας του δικτύου δίνοντας την δυνατότητα να εκτελεί λειτουργίες κωδικοποίησης και αποκωδικοποίησης [15].

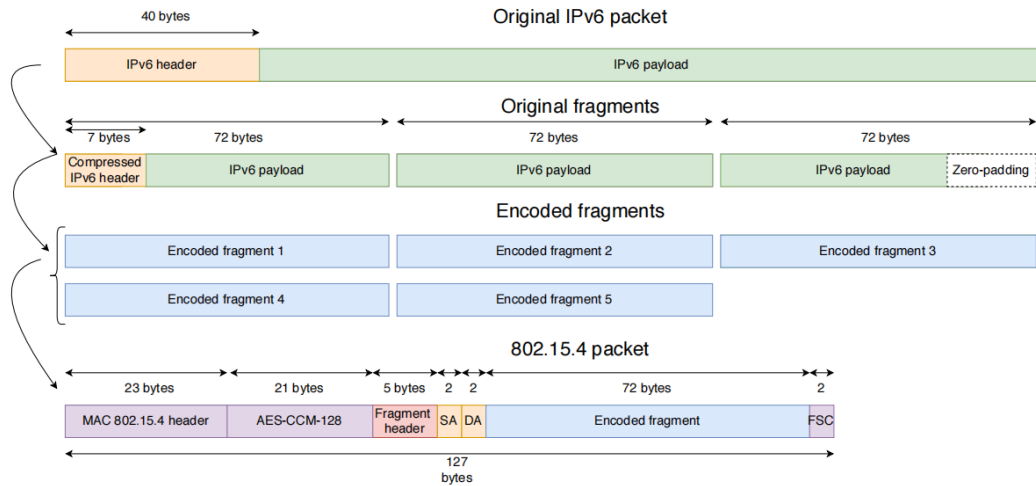
Το NCFEC δεν χρησιμοποιεί την VRB, αλλά ενσωματώνει συμπιεσμένη IPv6 διευθύνσεις σε κάθε fragment. Με αυτόν τον τρόπο, σε όλα τα fragments μπορεί να γίνει δρομολόγηση και δεν απαιτείται συναρμολόγηση και τεμαχισμό σε κάθε στάση. Στο [16], η κωδικοποίηση των data fragments γίνεται σε κάθε στάση. Αλλά με την υλοποίηση που προτείνει δεν πετυχαίνει την επιθυμητή απόδοση. Στο NCFEC [6], επισημοποιείται μηχανισμός εκτίμησης της απόδοσης για να καθορίσει τον μικρότερο αριθμό των encoded fragments που απαιτείται για να πραγματοποιήσει τη επιθυμητή απόδοση.

Συνοψίζοντας, τα χαρακτηριστικά του NCFEC είναι:

1. Ο τεμαχισμός και κωδικοποίηση γίνεται μόνο στον αποστολέα, ενώ συναρμολόγηση και αποκωδικοποίηση μόνο στον παραλήπτη. Έτσι αποφεύγεται επιπλέον χρήση πόρων και χρόνο καθυστέρησης για επεξεργασία πακέτου.
2. Κάθε fragment κουβαλάει συμπιεσμένη την IPv6 διεύθυνση. Επιτρέπει την δρομολόγηση των κάθε fragments ξεχωριστά και δεν χρειάζεται να συναρμολόγηση τα πακέτα σε κάθε στάση.

- Χρησιμοποιείται μηχανισμός εκτίμησης της απόδοσης για να καθορίσει τον αριθμό των encoded fragments. Χρησιμοποιείται πληροφορίες από τα μηνύματα επικοινωνίας του DODAG του RPL.

Για τα encoded fragments χρησιμοποιείται διαφορετικό τύπο, και είναι το 11011. Η τελική δομή των fragments φαίνεται στην εικόνα 17.

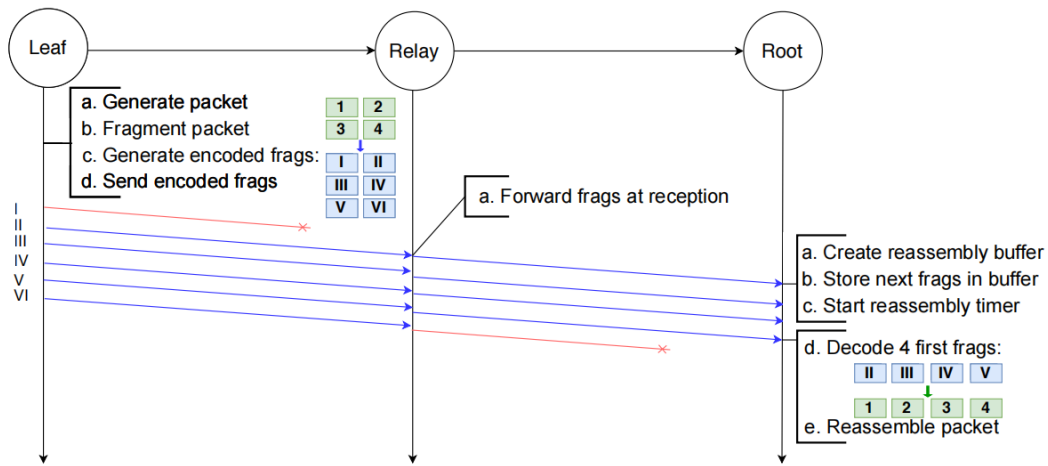


Εικόνα 17. δομή NCFEC fragments

Η λειτουργία του NCFEC είναι όπως οι άλλοι μέθοδοι, απλώς χρησιμοποιείται ενσωμάτωση πληροφορία δρομολόγησης για να αποφύγει την συναρμολόγηση σε κάθε στάση. Όταν ένα fragment φτάνει στον τελικό προορισμό, υπάρχουν 3 περιπτώσεις:

- Δεν έχει δημιουργηθεί αποθηκευτικό χώρο για το συγκεκριμένο συνδυασμό tag με διεύθυνση, άρα δημιουργείται χώρος στην μνήμη για αποθήκευση και ένας μετρητής.
- Υπάρχει χώρος στην μνήμη και αποθηκεύεται στην αντίστοιχη θέση.
- Ο αριθμός των fragments που αποθηκεύονται στη μνήμη έχει φτάσει στο όριο και γίνεται συναρμολόγηση του πακέτου.

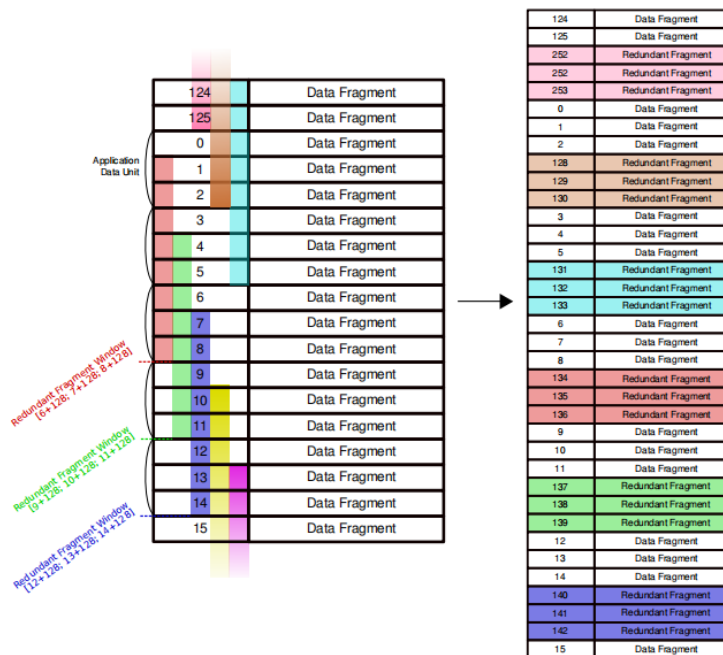
Ένα παράδειγμα για το NCFEC υπάρχει στην εικόνα 18. Έστω έχουμε ένα πακέτο που χωρίζεται σε 4 data fragments και μετά από κωδικοποίηση έχουμε 6 encoded fragments. Βλέπουμε πως κατά την αποστολή από τον αποστολέα προς τον ενδιάμεσο κόμβο, χάνεται το πρώτο encoded fragment. Αλλά αυτό δεν αποτελεί πρόβλημα για το NCFEC, για τι κάθε fragment κουβαλάει πληροφορίες δρομολόγησης και μπορεί να δρομολογηθεί ξεχωριστά. Στο τελικό προορισμό φτάνει 4 encoded fragments, ίσον με το αριθμό των data fragments αρχικά, και μπορεί να γίνει αποκωδικοποίηση του αρχικού δεδομένου. Άρα, η συναρμολόγηση του πακέτου είναι επιτυχής.



Εικόνα 18. Επιτυχή NCFEC λειτουργία

3.5 LoRaFFEC

Από αυτή την μέθοδο προέρχεται η ιδέα για τη λύση που προτείνεται στην παρούσα διπλωματική για αυτό το λόγο η προτεινόμενη ονομασία (LoRaFEC) είναι παρόμοια, . Η LoRaFFEC [17] χρησιμοποιεί μέθοδος κωδικοποίησης LDPC [18] για να δημιουργεί πλεονασμό. Για την δημιουργία του πλεονασμού εφαρμόζεται μέθοδος κωδικοποίησης στα data fragments που έχουν αποσταλεί προηγουμένως και δεν λαμβάνει υπόψιν αν τα data fragments ανήκουν στο ίδιο πακέτο. Η επιλογή για δημιουργία redundant fragments είναι όπως φαίνεται στην εικόνα 19.



Εικόνα 19. δομή LoRaFFEC fragments

Στον εξυπηρετητή, δηλαδή τον τελικό προορισμό σε ένα δίκτυο LoRaWAN, υπάρχει χώρος που αποθηκεύει όλα τα προηγούμενα fragments που έχει αποσταλεί σε ένα συγκεκριμένο πίνακα. Χρησιμοποιώντας τις πληροφορίες που υπάρχει στον εξυπηρετητή, θεωρητικά μπορεί να ανακτήσει κάθε data fragment που έχει χαθεί κατά την αποστολή.

Το μειονέκτημα της μεθόδου είναι ότι απαιτεί μεγάλη επεξεργαστική ικανότητα και μνήμη. Η μνήμη απαιτείται γιατί ο κόμβος πρέπει να κρατήσει data fragments που έχει σταλεί νωρίτερα για την δημιουργία των redundant fragments και ο τελικός προορισμός πρέπει να αποθηκεύσει όλα τα fragments που έχει σταλεί προηγουμένως. Επίσης ο κόμβος πρέπει να εκτελεί πολύπλοκους υπολογισμούς για δημιουργία των redundant fragments και για την ανάκτηση των χαμένων data fragments.

4

Προτεινόμενη λύση

Το πρόβλημα που θέλουμε να λύσουμε είναι η αύξηση της απόδοσης σε δίκτυο που υπάρχουν περιορισμοί. Η λύση που προτείνουμε μπορεί να θεωρηθεί σαν επέκταση της XORFEC και η ιδέα προέρχεται από την μέθοδο LoRaFFEC.

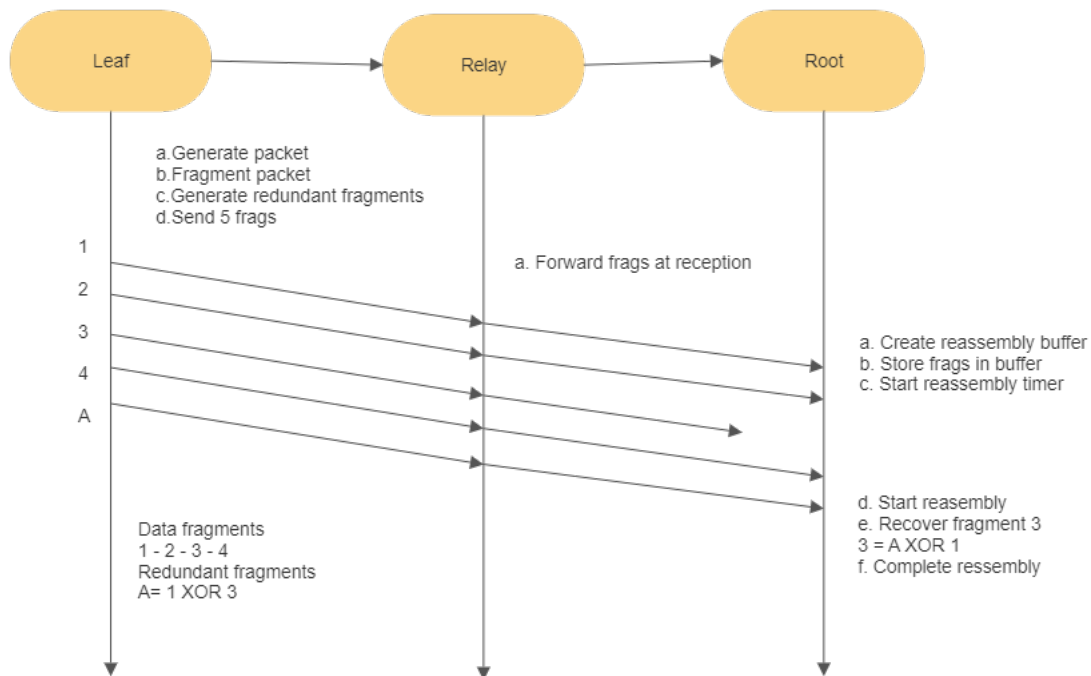
4.1 LoRaFEC

Η προτεινόμενη τεχνική LoRaFEC χρησιμοποιεί συγκεκριμένη τεχνική κωδικοποίησης η Exclusive OR (XOR), όπως και με τη XORFEC, για αυτόν το λόγο χαρακτηρίζουμε την μέθοδο μας ως επέκταση της XORFEC. Η LoRaFEC χρησιμοποιώντας την τεχνική αυτή, δημιουργεί επιπλέον fragments, αυτά τα fragments προκύπτουν από την χρήση XOR σε διάφορους συνδυασμούς των data fragments και χρησιμοποιείται για την βελτίωση της απόδοσης του δικτύου. Ο αριθμός των redundant fragments εξαρτάται από πολλούς παράγοντες όπως η ποιότητα της σύνδεσης, ο αριθμός των data fragments και το όριο που θέτουμε ως επιθυμητή απόδοση του δικτύου.

Καταρχήν, όπως έχουμε περιγράψει προηγουμένως, όταν ένας 6LoWPAN κόμβος επιθυμεί να αποσταλεί ένα IPv6 πακέτο όπου το μέγεθος του ξεπερνάει το MTU, δηλαδή 127 bytes, το πακέτο αυτό πρέπει να κομματιαστεί σε fragments. Τα fragments που προκύπτουν από το κόψιμο χαρακτηρίζεται ως data fragments. Μετά, η LoRaFEC χρησιμοποιεί την XOR σε διάφορους συνδυασμούς των data fragments και δημιουργεί επιπλέον fragments. Αυτά τα fragments χαρακτηρίζεται ως redundant fragments. Στην έρευνα μας, η έκδοση του LoRaFEC εξαρτάται από το περιεχόμενο των συνδυασμών που χρησιμοποιήθηκε για XOR και έχουμε δοκιμάσει διάφορες εκδόσεις του LoRaFEC. Η βασική έκδοσή του LoRaFEC θεωρείται αυτή

που δημιουργεί redundant fragments από τυχαίους συνδυασμούς. Τα redundant fragments στέλνονται στο τέλος αφού έχουν αποσταλεί τα data fragments. Έπειτα, αν ο τελικός προορισμός συνειδητοποιήσει ότι υπάρχουν χαμένα data fragments, ελέγχει αν από τα redundant fragments που έχει λάβει μπορεί να γίνει ανάκτηση των χαμένων fragments.

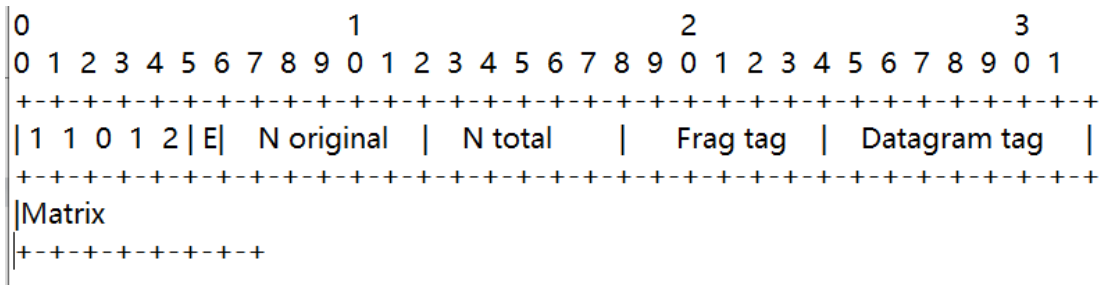
Δίνοντας ένα παράδειγμα για την λειτουργία του LoRaFEC. Έχοντας ένα πακέτο που πρέπει να κομματιαστεί σε 4 data fragments και ο αριθμός των redundant fragments είναι 1. Τα data fragments ανάλογα με την σειρά τεμαχισμού, τα ονομάζουμε 1-2-3-4, και τα redundant fragments τα ονομάζουμε A. Το redundant fragment A προκύπτει από τεχνική XOR σε συνδυασμό των data fragments 1-3. Έστω τα χαμένα fragments είναι 3.



Εικόνα 20. Παράδειγμα λειτουργίας του LoRaFEC

Όπως φαίνεται στην εικόνα 20, μπορούμε να ανακτήσουμε τα χαμένα fragments εφαρμόζοντας την τεχνική XOR σε συνδυασμό redundant fragment και data fragment. Στο παράδειγμα που δίνουμε εδώ, γίνεται ανάκτηση των χαμένων fragments 2 από XOR σε συνδυασμό redundant fragment B και data fragment 3, για το χαμένο fragments 4 από συνδυασμό redundant fragment A και data fragment 4.

Το μήκος του εξαρτάται από το αριθμό των data fragments. Για παράδειγμα ο πίνακας για ένα πακέτο που χωρίζεται σε 4 data fragment είναι όπως φαίνεται στη εικόνα 22.



Εικόνα 21. Δομή του LoRaFEC fragment

Data fragment

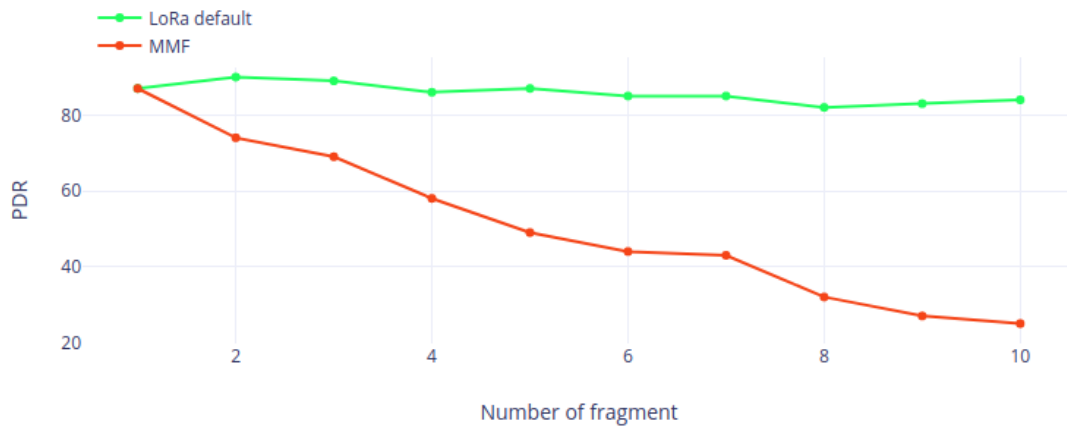
- 0 - 0 - 0 - 1
- 0 - 0 - 1 - 0
- 0 - 1 - 0 - 0
- 1 - 0 - 0 - 0

Redundant fragment

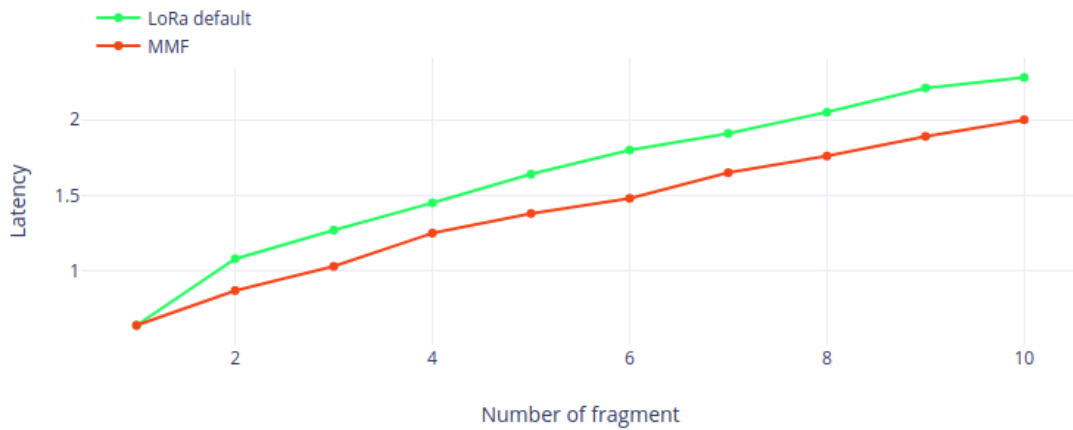
- 1 XOR 2 -» 1 - 1 - 0 - 0
- 2 XOR 3 XOR 4 -» 0 - 1 - 1 - 1

Εικόνα 22. Matrix του LoRaFEC fragment

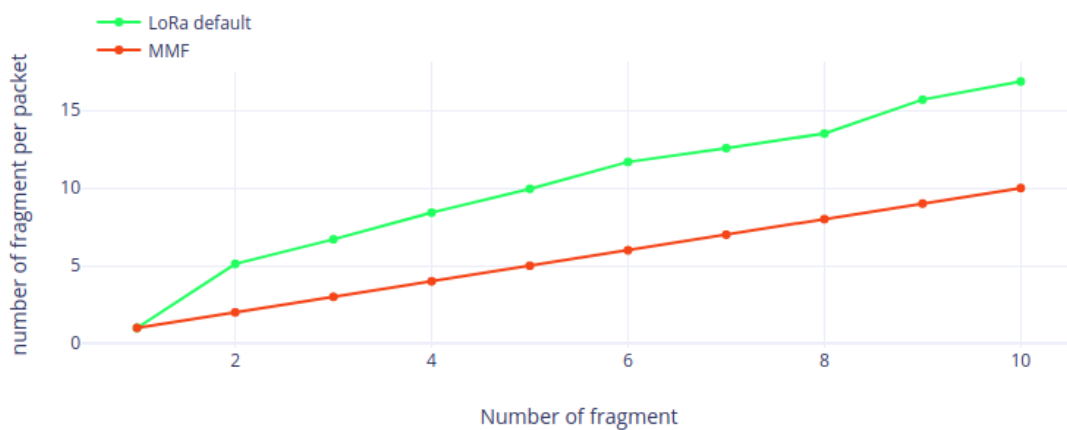
Για να ελέγξουμε την απόδοση της προτεινόμενης λύσης, συγκρίνουμε αρχικά την προτεινόμενη τεχνική με την MFF. Το αποτέλεσμα φαίνεται στην εικόνα 23.



A) απόδοση του δικτύου



B) χρόνος καθυστέρησης



Γ) συνολικός αριθμός των fragments

Εικόνα 23. Σύγκριση LoRaFEC με MFF με ποιότητα σύνδεσης 0.65

Μπορούμε να δούμε ότι η LoRaFEC αποδίδει πολύ καλύτερα από την MFF. Ο χρόνος καθυστέρησης είναι μεγαλύτερη, πράγμα που είναι λογικό, επειδή χρειάζεται περισσότερο χρόνο για την επεξεργασία των fragments. Με τον τρόπο που χρησιμοποιούμε η τεχνική FEC εδώ, είναι φυσιολογικό να δούμε την μέθοδο μας να χρησιμοποιεί περισσότερα fragments.

Αξίζει να σημειωθεί ότι η έκδοση της LoRaFEC που χρησιμοποιούμε εδώ για σύγκριση είναι η αρχική έκδοση που δεν έχει υποστεί βελτίωση. Αλλά η αύξηση στην απόδοση που επιτυγχάνει μας δείχνει ότι είναι ένας μέθοδος που αξίζει διεξοδική έρευνα.

4.2 Τελική έκδοση

Όπως αναφέρθηκε προηγουμένως, έχουμε δοκιμάσει διάφορες εκδόσεις του LoRaFEC και οι εκδόσεις διαφοροποιούνται ανάλογα με τους συνδυασμούς των data fragments που χρησιμοποιούνται για τη δημιουργία των redundant fragments. Η βασική έκδοση επιλέγει τυχαίους συνδυασμούς και ονομάζουμε την έκδοση αυτή LoRaFEC default.

Ακολουθήσαμε 3 βήματα για την βελτίωση του LoRaFEC default και για να καθαρίσουμε την τελική έκδοση του LoRaFEC. Τα βήματα είναι τα εξής:

1. Καθορίζουμε τον αριθμό των redundant fragments
2. Καθορίζουμε το περιεχόμενο των redundant fragments, δηλαδή ποιους συνδυασμούς θα χρησιμοποιείται για δημιουργία των redundant fragments.
3. Τι επιπλέον θα χρησιμοποιήσει για την βελτίωση της απόδοσης του μεθόδου

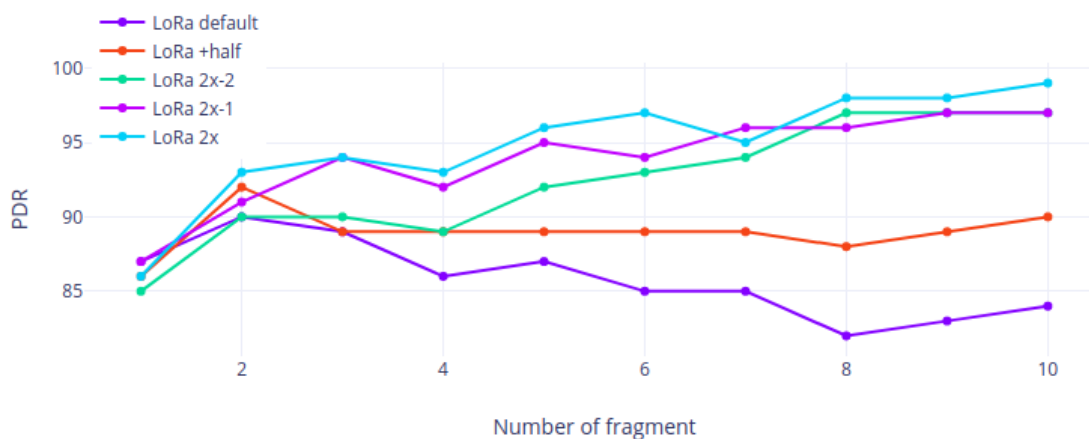
4.2.1 Καθορισμός αριθμός των redundant fragments

Υπάρχουν διαφορετικοί τρόποι για να αυξήσουμε την απόδοση του δικτύου και ο πιο απλός από αυτούς είναι να αυξήσουμε το μέγεθος των επιπλέον δεδομένων που αποστέλλονται. Για την αύξηση αυτή υπάρχουν πολλές επιλογές, όπως για παράδειγμα η επανάληψη της αποστολής που χρησιμοποιείται στο RFEC. Στην μέθοδο που προτείνουμε, καθορίζουμε την αύξηση αυτή ως αύξηση του αριθμού των redundant fragments.

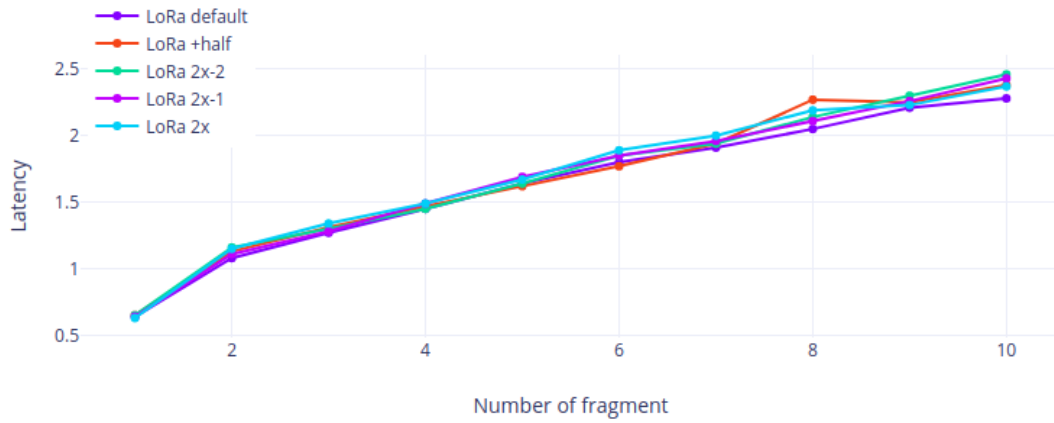
Στο 1^ο βήμα, δοκιμάσαμε διαφορετικούς σημεία έναρξης για το αριθμό των redundant fragments. Οι αριθμοί που δοκιμάστηκαν είναι τα εξής:

1. +half : αριθμός των redundant fragments ισούται με τον μισό του αριθμού των data fragments
2. $2x-2$: αριθμός των redundant fragments ισούται με τον αριθμό των data fragments μείον 2
3. $2x-1$: αριθμός των redundant fragments ισούται με τον αριθμό των data fragments μείον 1
4. $2x$: αριθμός των redundant fragments ισούται με τον αριθμό των data fragments

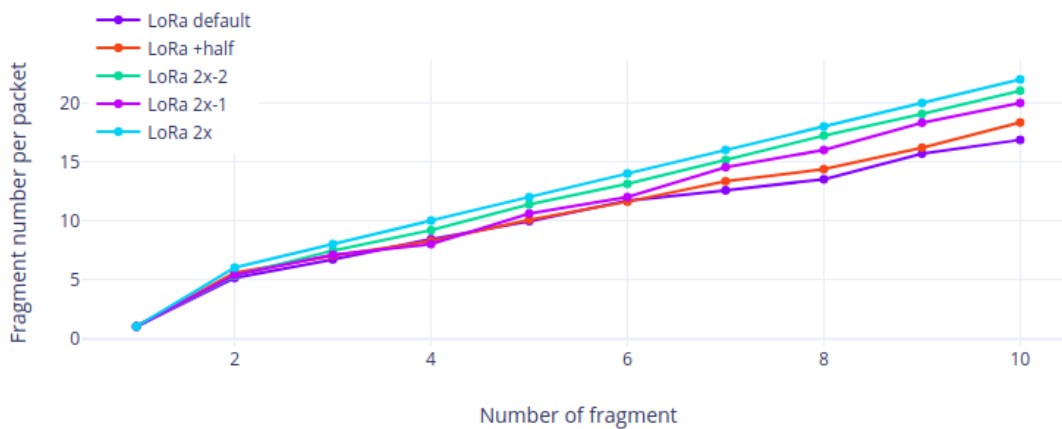
Τα αποτελέσματα φαίνονται στην εικόνα 24.



A) απόδοση του δικτύου



B) χρόνος καθυστέρησης



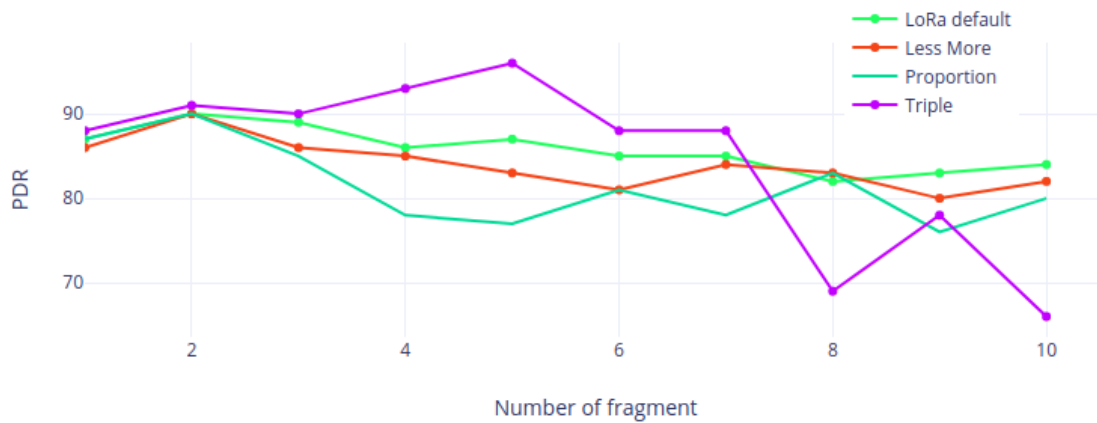
Γ) συνολικός αριθμός των fragments

Εικόνα 24. Σύγκριση LoRaFEC διαφορετικοί redundant fragments

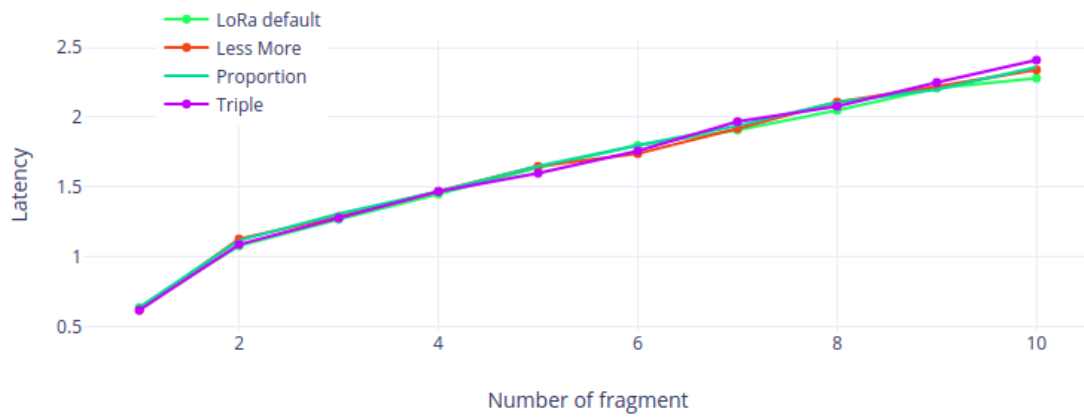
Από τα αποτελέσματα των προσομοιώσεων που τρέξαμε, συμπεράνουμε ότι οι καλύτερες επιλογές για το αριθμό των redundant fragments είναι 2x-1, για τι σε σύγκριση με την αμέσως παρακάτω επιλογή, που είναι το 2x-1, υπάρχει μεγάλη βελτίωση των αποδόσεων. Ενώ η σύγκριση της με την αμέσως παραπάνω επιλογή, που είναι η 2x, δεν υπάρχει μεγάλη διαφορά στην απόδοση. Επιπλέον, παρατηρώντας τις άλλες μετρικές που χρησιμοποιούμε για την αξιολόγηση του μεθόδου, δεν υπάρχει σημαντική διαφορά που μπορεί να επηρεάζει την επιλογή μας.

Ο λόγος που καθορίζουμε ως πρώτο βήμα την αύξηση του αριθμό των redundant fragments, είναι επειδή κάναμε σύγκριση των LoRaFEC παραλλαγών με την LoRaFEC default χωρίς να αλλάζουμε το αριθμό των redundant fragments. Η σύγκριση φαίνεται αναλυτικά στην εικόνα

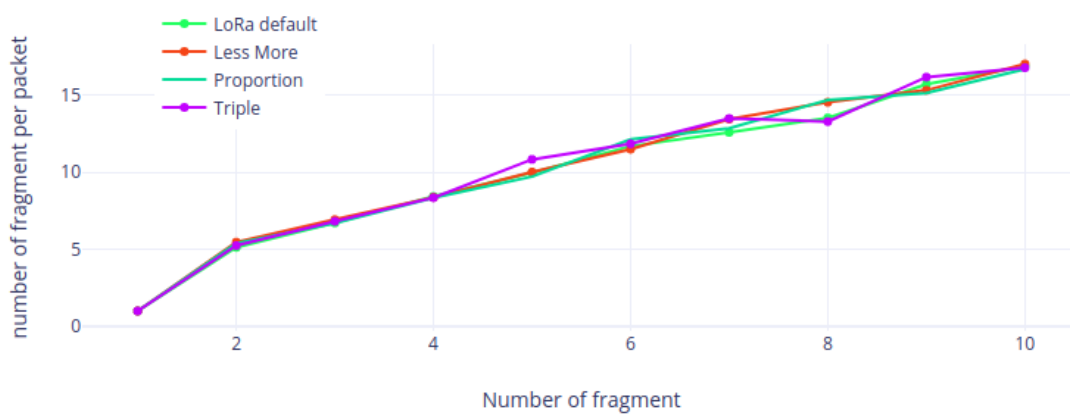
παρακάτω.



A) απόδοση του δικτύου



B) χρόνος καθυστέρησης



Γ) συνολικός αριθμός των fragments

Εικόνα 25. Σύγκριση LoRaFEC διαφορετικές παραλλαγές

Βλέπουμε ότι εάν ο αριθμός των redundant fragments παραμένει χαμηλό, τότε όποιο μέθοδο και να χρησιμοποιήσουμε, δεν υπάρχει βελτίωση στα αποτελέσματα.

4.2.2 Περιεχόμενο των redundant fragments

Ως 2ο βήμα, ορίσαμε τον καθορισμό των περιεχομένων των redundant fragments. Σε αυτό το βήμα, έχουμε 2 ιδέες και είναι οι εξής:

1. Χωρισμός των redundant fragments σε διαφορετικές ομάδες και αναλογίες
2. Καθορισμός του περιεχομένου των συνδυασμών των data fragments

Για την 1^η ιδέα, το αρχικό κριτήριο για τον χωρισμό χρησιμοποιούμε τον αριθμό των data fragments στον συνδυασμό, αναλυτικότερα οι ομάδες είναι τα εξής:

1. Κάτω από το μισό: ο αριθμός των data fragments που υπάρχει στον συνδυασμός είναι λιγότερα από το μισό του σύνολο των data fragments
2. Πάνω από το μισό: ο αριθμός των data fragments που υπάρχει στον συνδυασμός είναι περισσότερα από το μισό του σύνολο των data fragments

Δίνοντας ένα παράδειγμα για τον χωρισμό. Ένα πακέτο χωρίζεται σε 4 data fragments και ο αριθμός του redundant fragments ορίζεται στο 2. Η αναλογία που καθορίζουμε είναι 50%. Το τελικό σύνολο του fragments που δημιουργείται είναι όπως φαίνεται παρακάτω.

Less - More χωρισμός

Data fragments

1 - 2 - 3 - 4

Redundant fragments

Less

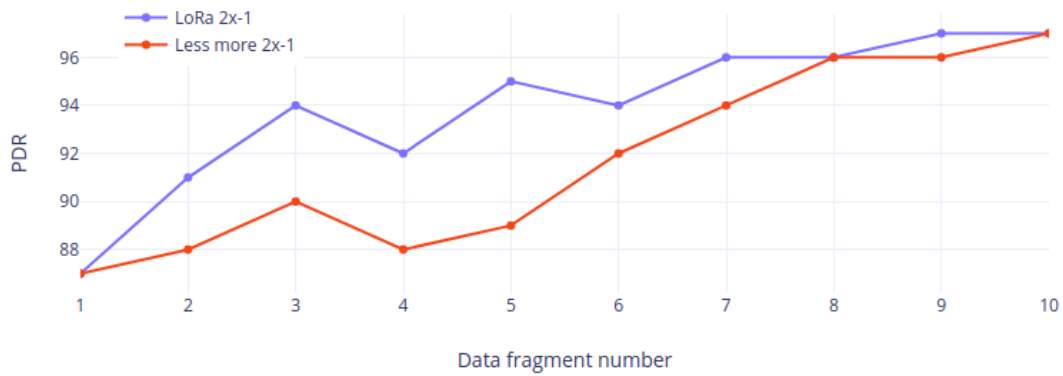
A : 1 XOR 4

More

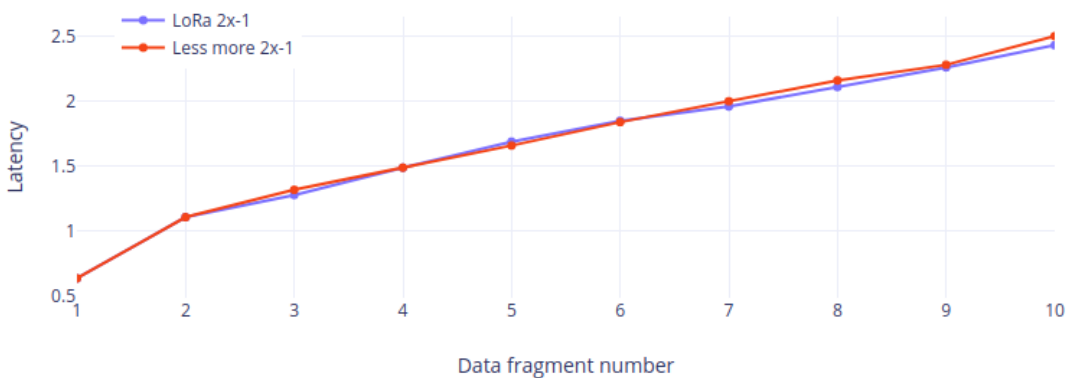
B : 1 XOR 2 XOR 4

Εικόνα 26. Χωρισμός σε ομάδες - παράδειγμα.

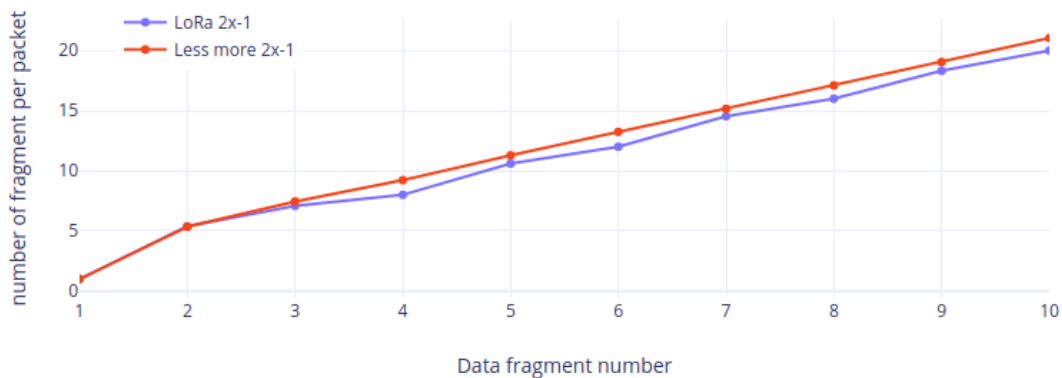
Συγκρίνοντας αυτό το μέθοδο με την LoRaFEC default με 2x-1 redundant fragments, έχουμε το παρακάτω αποτέλεσμα.



A) απόδοση του δικτύου



B) χρόνος καθυστέρησης



Γ) συνολικός αριθμός των fragments

Εικόνα 27. Σύγκριση .LoRaFEC default - less/more

Από την παραπάνω σύγκριση βλέπουμε ότι αυτή η μέθοδος δεν δίνει καλύτερο αποτέλεσμα από το LoRaFEC default. Για τη βελτίωση αυτής της ιδέας, κάναμε αναλυτική μελέτη των αποτελεσμάτων, συγκρίνοντας την αναλογία των προσομοιώσεων που δίνει το καλύτερο αποτέλεσμα και αυτά που δίνουν το χειρότερο αποτέλεσμα. Βγάζουμε συμπέρασμα ότι πρέπει να χωριστούν σε παραπάνω από 2 κατηγορίες και οι κατηγορίες είναι τα εξής:

1. Κάτω από το μισό: ο αριθμός των data fragments που υπάρχει στον συνδυασμός είναι λιγότερα από το μισό -1 του σύνολο των data fragments
2. Γύρω από το μισό: ο αριθμός των data fragments που υπάρχει στον συνδυασμός είναι ίσον με το μισό και μισό +- 1 του σύνολο των data fragments
3. Πάνω από το μισό: ο αριθμός των data fragments που υπάρχει στον συνδυασμός είναι περισσότερα από το μισό + 1 του σύνολο των data fragments

Υποθέτουμε, από την μελέτη των καλύτερων προσομοιώσεων, η αναλογία των 3 κατηγοριών είναι στο 20% - 70% - 10%.

Δίνοντας ένα παράδειγμα για το χωρισμό. Ένα πακέτο χωρίζεται σε 10 data fragments και ο αριθμός του redundant fragments ορίζεται στο 10. Η αναλογία που καθορίζουμε είναι 20% - 70% - 10%. Το τελικό σύνολο του fragments που δημιουργείται είναι όπως φαίνεται παρακάτω.

20 - 70 - 10 χωρισμός

Data fragments

1 - 2 - 3 - 4 - 5 - 6 - 7 - 8 - 9 - 10

Redundant fragments

20%

2 fragment - XOR less than 4 data fragments

70%

7 fragment - XOR with 4 or 5 or 6 data fragments

10%

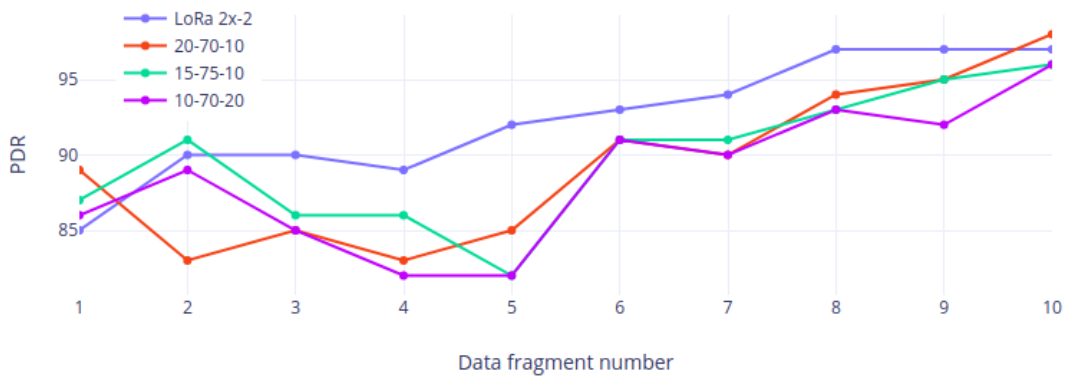
1 fragment - XOR more than 6 data fragments

Εικόνα 28. Χωρισμός σε 3 ομάδες - παράδειγμα.

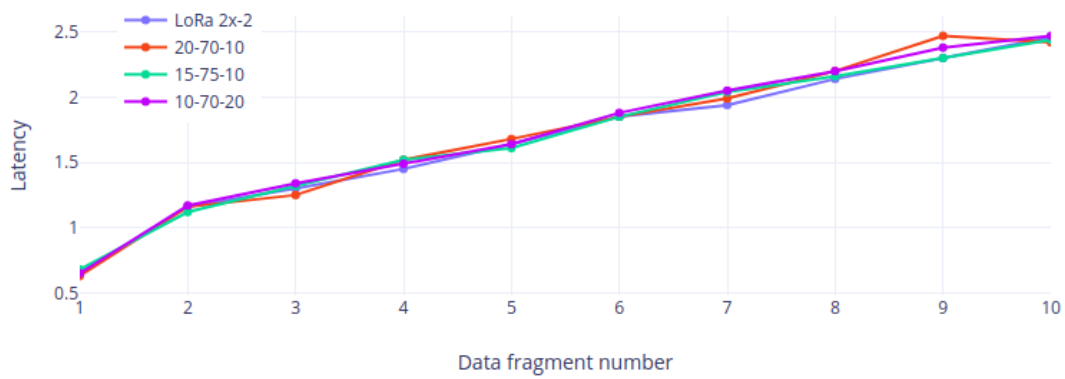
Για να βεβαιωσουμε αυτή την αναλογία, τρέξαμε προσομοιώσεις και με άλλες αναλογίες όπως:

1. 20 - 70 - 10
2. 15 - 75 - 10
3. 10 - 70 - 20

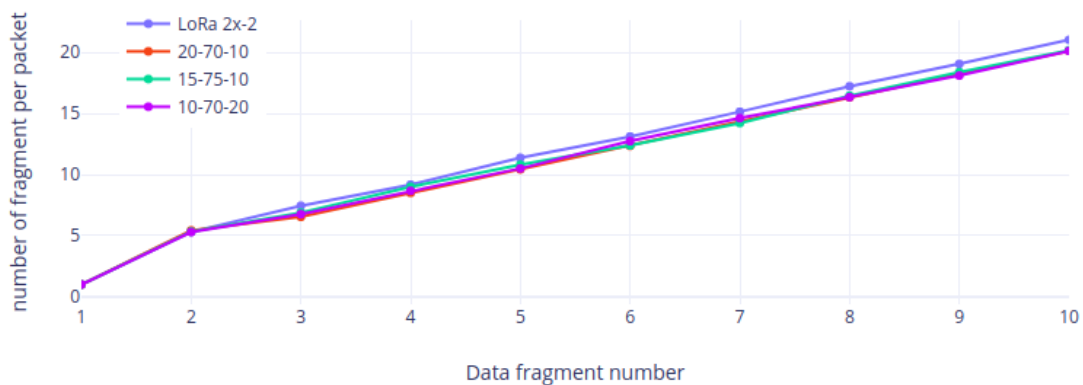
Όλες οι επιλογές χρησιμοποιούν τους ίδιους παραμέτρους προσομοίωσης και όλοι έχουν τον αριθμό του redundant fragments ίσον με $2x-1$. Το αποτέλεσμα της σύγκρισης με το LoRaFEC default φαίνεται στην εικόνα 29.



A) απόδοση του δικτύου



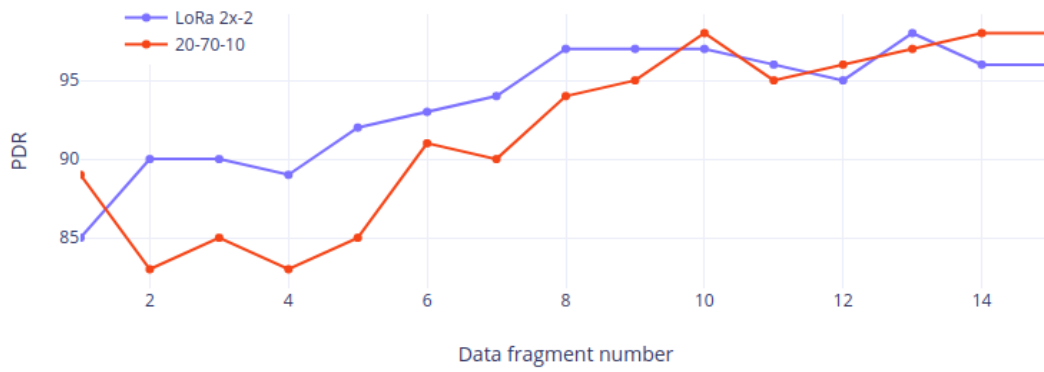
B) χρόνος καθυστέρησης



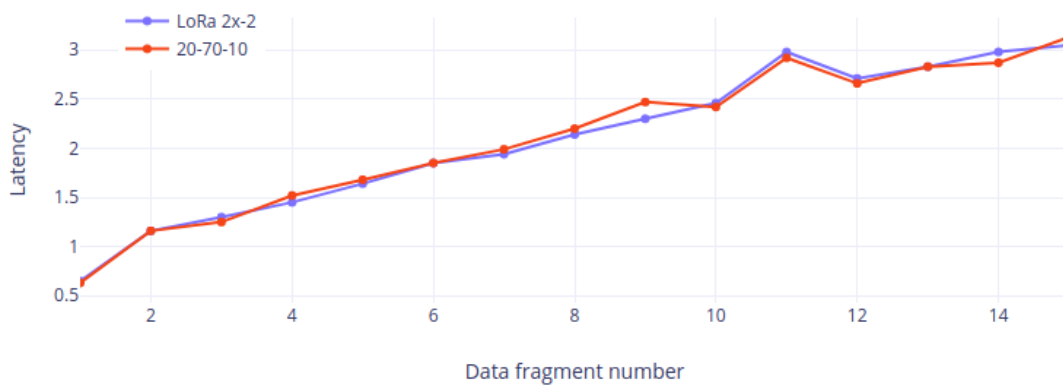
Γ) συνολικός αριθμός των fragments

Εικόνα 29. Σύγκριση LoRaFEC - αναλογίες

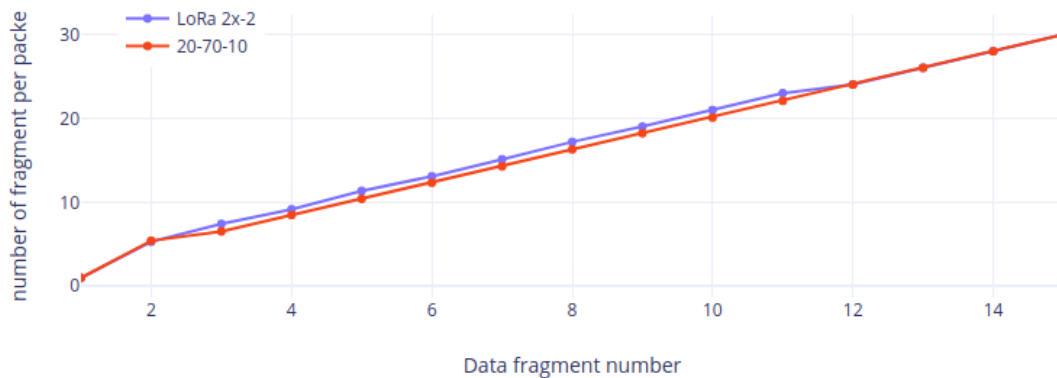
Βλέπουμε πως στα περισσότερα σημεία, η μέθοδος με αναλογία δεν δίνει καλύτερα αποτελέσματα από το LoRaFEC default, παρά μόνο όταν ο αριθμός των data fragments είναι ίσο με 10. Για να εξερευνήσουμε αν αυτό το φαινόμενο είναι τυχαίο, τρέξαμε προσομοιώσεις με αριθμό των data fragments μεγαλύτερο από το 10 και η σύγκριση φαίνεται στην εικόνα 30.



A) απόδοση του δικτύου



B) χρόνος καθυστέρησης



Γ) συνολικός αριθμός των fragments

Εικόνα 30. Σύγκριση .LoRaFEC - αναλογίες >10

Από την σύγκριση που βλέπουμε στην παραπάνω εικόνα, δε μπορούμε να πούμε ότι η μέθοδος με αναλογίες δίνει σίγουρα καλύτερα αποτελέσματα. Στα περισσότερα σημεία που δοκιμάσαμε επιτυγχάνει μια καλύτερη απόδοση, ειδικά για μεγάλο αριθμός των fragments. Μπορούμε να κρατήσουμε τον μέθοδο σαν μια επιλογή για συνδυασμό με άλλες μεθόδους.

Η άλλη ιδέα είναι να καθαρίσουμε το περιεχόμενων των συνδυασμών των data fragments που θα χρησιμοποιήσει η τεχνική XOR για την δημιουργία redundant fragments. Η αρχική υλοποίηση αυτής της ιδέας είναι ο διαδοχικός δυαδικός συνδυασμός των data fragments. Δίνοντας ένα παράδειγμα, ένα πακέτο χωρίζεται σε 4 data fragments και ο αριθμός του redundant fragments ορίζεται στο 2. Τα data fragments ονομάζονται 1-4 και τα redundant fragments ονομάζονται A και B. Το τελικό σύνολο των fragments που δημιουργείται είναι όπως φαίνεται στην εικόνα 31.

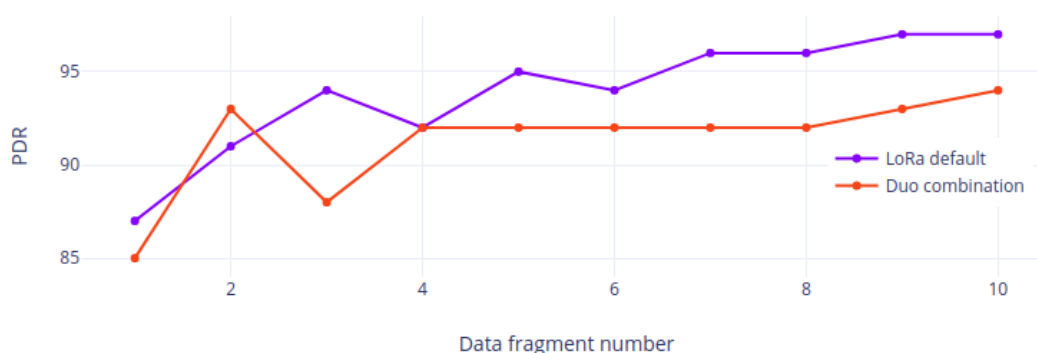
Sequence DUO χωρισμός

Data fragments
1 - 2 - 3 - 4

Redundant fragments
A : 1 XOR 2
B : 2 XOR 3

Εικόνα 31. Χωρισμός σε δυαδικό - παράδειγμα

Το αποτέλεσμα που δίνει, σε σύγκριση με την LoRaFEC default με αριθμό των redundant fragments ίσο με $2x-1$ φαίνεται στην εικόνα 32.



Εικόνα 32. Σύγκριση .LoRaFEC - δυαδικό

Η αρχική υλοποίηση, δηλαδή ο διαδοχικός δυαδικός συνδυασμός δε δίνει καλύτερα αποτελέσματα από το LoRaFEC default. Για βελτιώσουμε την ιδέα, σκεφτήκαμε διαφορετικούς συνδυασμούς, αναλυτικότερα, οι συνδυασμοί που δοκιμάστηκαν είναι τα εξής:

1. Διαδοχικός δυαδικός συνδυασμός
2. Μην διαδοχικός δυαδικός συνδυασμός
3. Διαδοχικός τριαδικός συνδυασμός

4. Μη διαδοχικός τριαδικός συνδυασμός

Δίνοντας ένα παράδειγμα για το καθένα, ένα πακέτο χωρίζεται σε 5 data fragments και ο αριθμός του redundant fragments ορίζεται στο 2. Τα data fragments ονομάζονται 1-4 και τα redundant fragments ονομάζονται A και B. Το τελικό σύνολο των fragments που δημιουργείται είναι όπως φαίνεται στην παρακάτω εικόνα.

No sequence DUO χωρισμός

Data fragments

1 - 2 - 3 - 4

Redundant fragments

A : 1 XOR 3

B : 2 XOR 4

A) Μη σειριακός δυαδικός χωρισμός

Sequence triple χωρισμός

Data fragments

1 - 2 - 3 - 4

Redundant fragments

A : 1 XOR 2 XOR 3

B : 2 XOR 3 XOR 4

B) Σειριακός τριαδικός χωρισμός

No sequence triple χωρισμός

Data fragments

1 - 2 - 3 - 4

Redundant fragments

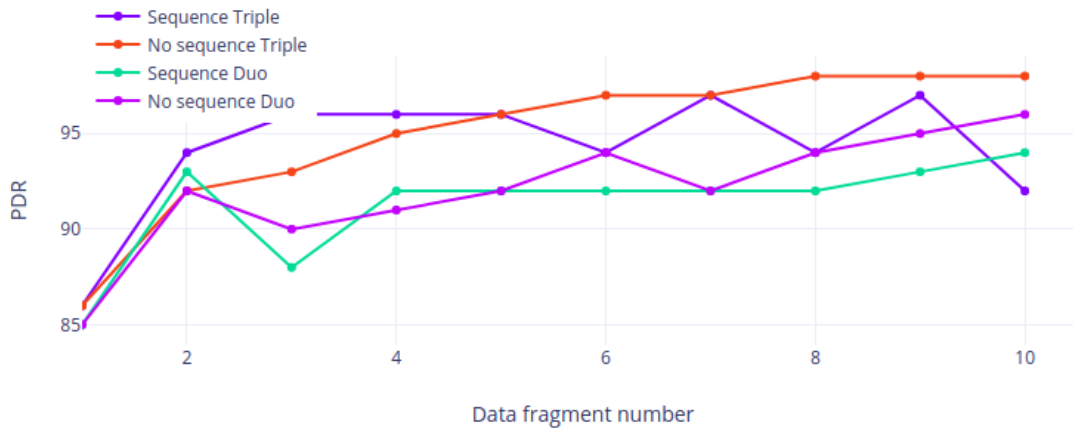
A : 1 XOR 3 XOR 4

B : 1 XOR 2 (random)

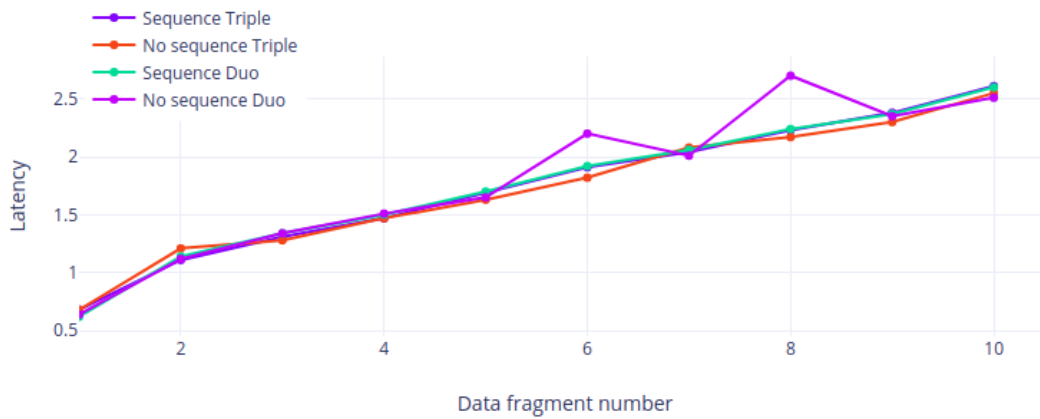
Γ) Μη σειριακός τριαδικός χωρισμός

Εικόνα 33. Χωρισμός - παράδειγμα

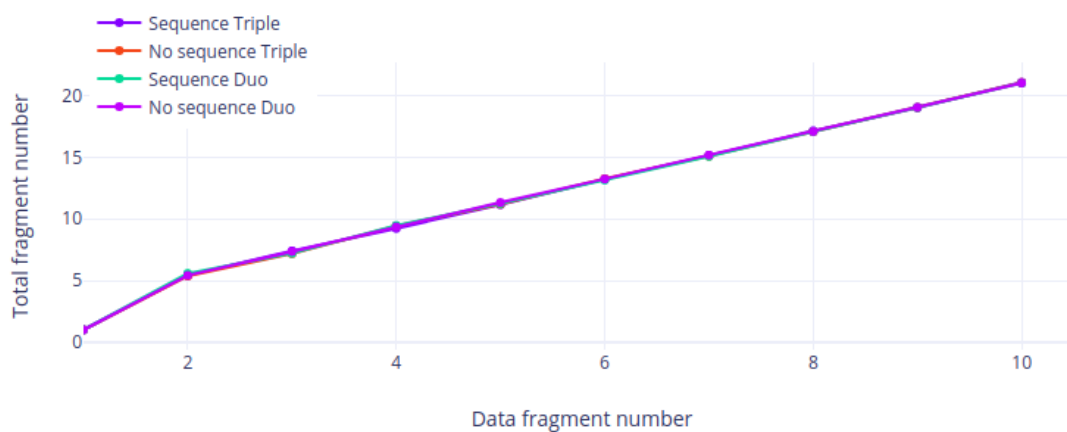
Για όλες τις κατηγορίες έχουμε τρέξει προσομοιώσεις με αριθμό του redundant fragments ίσον με $2x-1$. Η σύγκριση των μεθόδων φαίνεται από την εικόνα 34.



A) απόδοση του δικτύου



B) χρόνος καθυστέρησης



Γ) συνολικός αριθμός των fragments

Εικόνα 34. Σύγκριση . συνδυασμούς

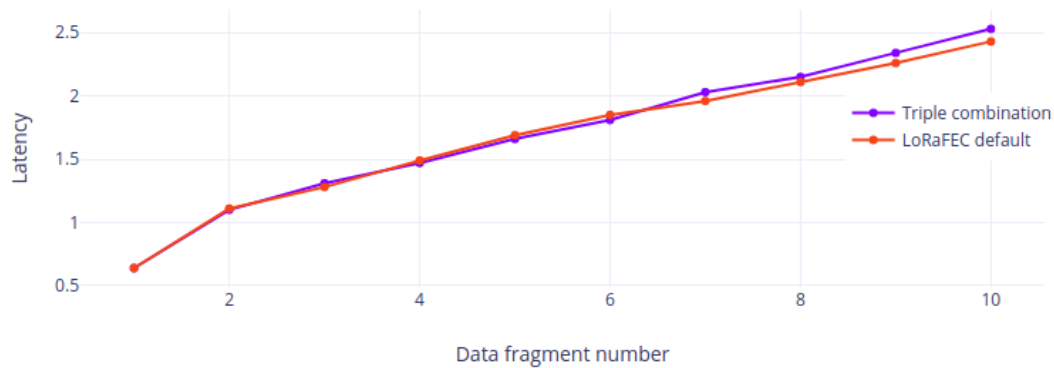
Όπως φαίνεται από τα αποτελέσματα, για διαφορετικό αριθμό των data fragments, διαφορετικοί μέθοδοι δίνουν το καλύτερο αποτέλεσμα. Για αριθμό των data fragments

μικρότερο ή ίσο με 4, η μέθοδος που αποδίδει καλύτερα είναι ο διαδοχικός τριαδικός συνδυασμός, και για αριθμό των data fragments μεγαλύτερο με 4, η μέθοδος που αποδίδει καλύτερα είναι ο μην διαδοχικός τριαδικός συνδυασμός.

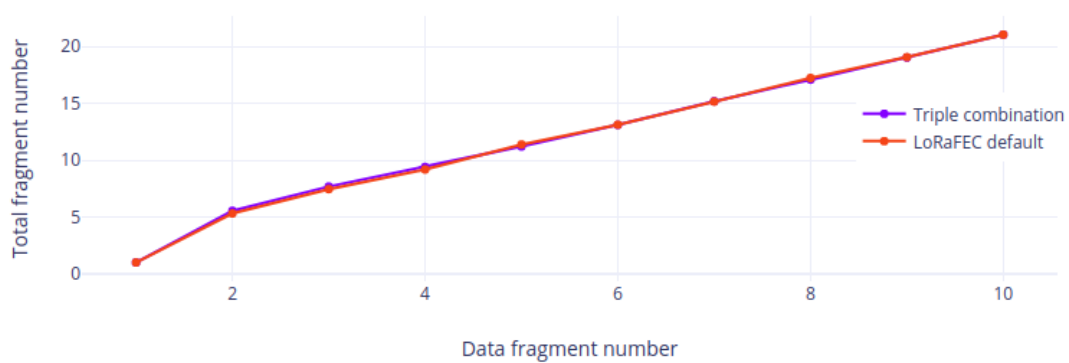
Η σύγκριση του LoRaFEC default με τον συνδυασμό που δίνει το καλύτερο αποτέλεσμα με αριθμό του redundant fragments ίσο με $2x-1$ δίνει το παρακάτω αποτέλεσμα.



A) απόδοση του δικτύου



B) χρόνος καθυστέρησης



Γ) συνολικός αριθμός των fragments

Εικόνα 35. Σύγκριση .LoRaFEC - τριαδικό συνδυασμό

Από την σύγκριση, βλέπουμε ότι δίνει πολύ καλύτερα απόδοση από το LoRaFEC default, έχοντας περίπου το ίδιο επιπλέον φόρτος και χρόνο καθυστέρησης.

4.2.3 Μέθοδοι βελτίωσης

Στο 3ο βήμα, αποφασίζουμε αν θα χρησιμοποιήσουμε επιπλέον τεχνικές για την βελτίωση του μεθόδου. Οι 2 μέθοδοι που έχουμε υπόψιν μας είναι τα εξής:

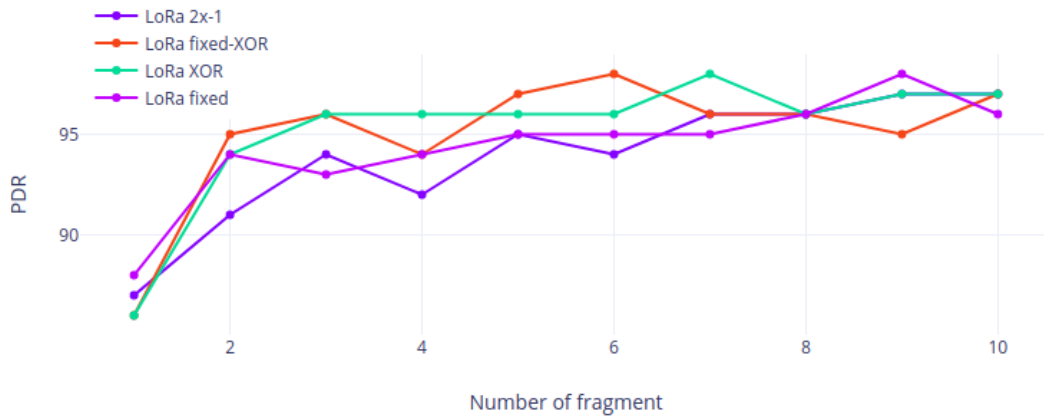
1. Να υπάρχει 1 συγκεκριμένο data fragment που θα περιλαμβάνεται σε όλους τους συνδυασμούς των redundant fragments για ένα πακέτο
2. Να υπάρχει 1 redundant fragment από συγκεκριμένο συνδυασμό των data fragments

Δίνοντας ένα παράδειγμα για το καθένα, ένα πακέτο χωρίζεται σε 5 data fragments και ο αριθμός του redundant fragments ορίζεται στο 2. Τα data fragments ονομάζονται 1-5 και τα redundant fragments ονομάζονται A και B. Στην 1η κατηγορία, ορίζουμε το fragment 1 ως το data fragment που πρέπει να υπάρχει σε κάθε redundant fragments. Στην 2^η κατηγορία, ορίζουμε τον συνδυασμό XOR όλων των data fragments ως το redundant fragment που πρέπει να υπάρχει σε κάθε σύνολο. Το τελικό σύνολο των fragments που δημιουργείται είναι όπως φαίνεται στην παρακάτω εικόνα.

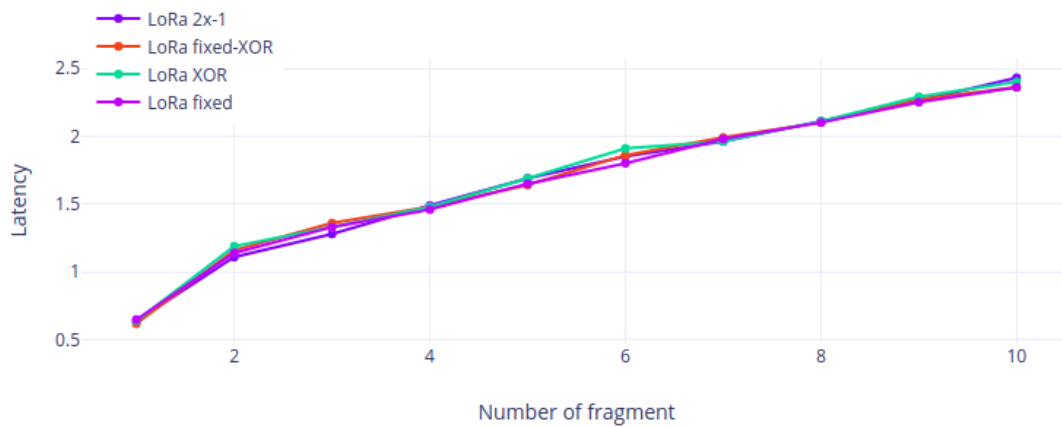
Fixed	XORall
Data fragments 1 - 2 - 3 - 4 - 5	Data fragments 1 - 2 - 3 - 4 - 5
Redundant fragments A : 1 XOR 4 B : 1 XOR 2	Redundant fragments A : 1 XOR 2 XOR 3 XOR 4 XOR 5 B : 3 XOR 4

Εικόνα 36. Χωρισμός - παράδειγμα

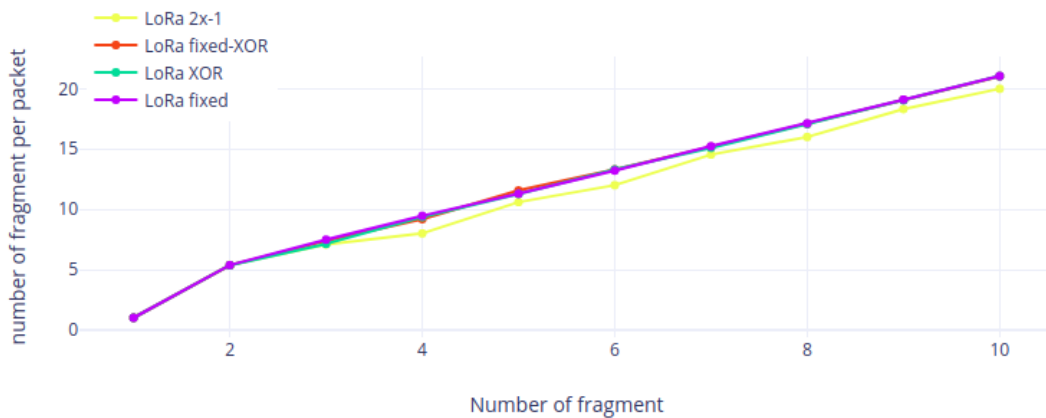
Στο 1^ο στάδιο, συγκρίναμε το LoRaFEC default με αριθμό του redundant fragments ίσο με $2x-1$ και το ίδιο LoRaFEC default χρησιμοποιώντας είτε το ένα είτε το άλλο μέθοδο βελτίωσης και βλέπουμε την σύγκριση στην εικόνα 37.



A) απόδοση του δικτύου



B) χρόνος καθυστέρησης



Γ) συνολικός αριθμός των fragments

Εικόνα 37. Σύγκριση των μεθόδων για ποιότητα του δικτύου ίσον με 0.65

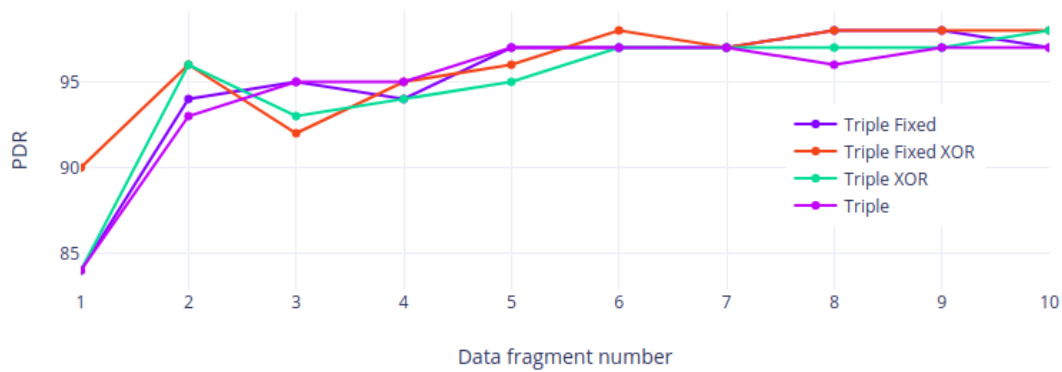
Βλέπουμε από τα αποτελέσματα ότι και οι δυο μέθοδοι βελτίωσης αυξάνουν σε ορισμένο βαθμό την απόδοση. Ειδικότερα η προσθήκη XOR fragment αυξάνει σημαντικά την απόδοση του δικτύου, χωρίς να υπάρχει μεγάλη αλλαγή στο φόρτο και την καθυστέρηση.

Στο 2ο στάδιο, εφαρμόζουμε τους μεθόδους βελτίωσης στην έκδοση της LoRaFEC που δοκιμάσαμε προηγουμένως και δίνει το καλύτερο αποτέλεσμα, δηλαδή η ένωση των διαδοχικών και τριαδικό συνδυασμών.

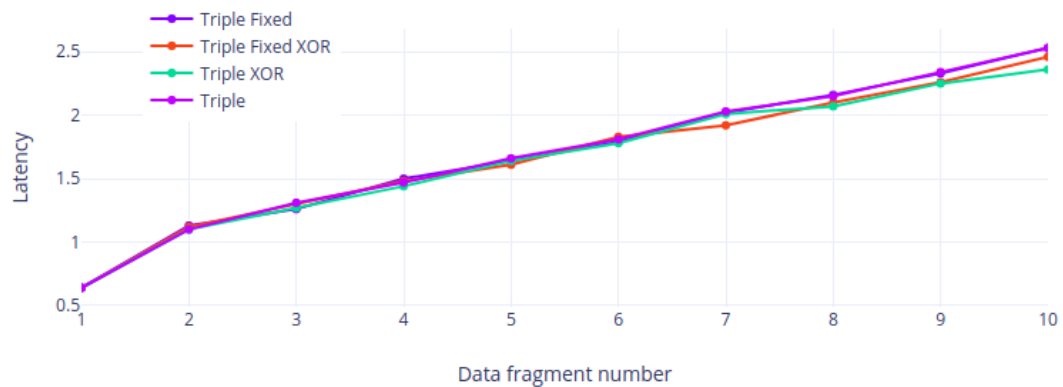
Κάνουμε σύγκριση των τεσσάρων μεθόδων που είναι τα εξής:

1. Χωρίς μεθόδους βελτίωσης
2. Με 1^η μέθοδο βελτίωση, δηλαδή fixed data fragment
3. Με την 2^η μέθοδο βελτίωσης, δηλαδή XOR όλων των redundant fragments
4. Συνδυασμός 1ης και 2ης μεθόδου

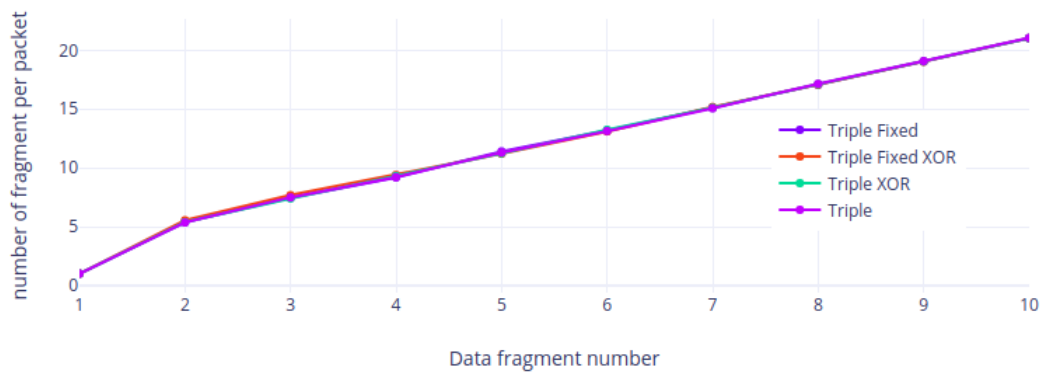
Το αποτέλεσμα της σύγκρισης φαίνεται στην παρακάτω εικόνα. Όλες οι προσομοιώσεις εκτελούνται με ίδιους παραμέτρους και με αριθμό των redundant fragments ίσο με $2x-1$.



A) απόδοση του δικτύου



B) χρόνος καθυστέρησης



Γ) συνολικός αριθμός των fragments

Εικόνα 38. Σύγκριση των μεθόδων για ποιότητα του δικτύου ίσον με 0.65

Ανάμεσα στις 4 μεθόδους, αυτή που δίνει καλύτερα αποτελέσματα στα περισσότερα σημεία είναι ο τριαδικός συνδυασμός που χρησιμοποιεί και τις 2 μεθόδους βελτίωσης. Η καθυστέρηση είναι περίπου ίδια για όλες τις μεθόδους. Για αριθμό fragments ίσο με 10 παρατηρούμε ότι η μέθοδος του τριαδικού συνδυασμού που χρησιμοποιεί μέθοδο βελτίωσης XOR έχει τη χαμηλότερη καθυστέρηση, ενώ δίνει την ίδια απόδοση. Ο λόγος που απορρίπτουμε αυτή που χρησιμοποιεί μέθοδο βελτίωσης XOR είναι επειδή στα περισσότερα σημεία δεν μπορεί να δώσει την καλύτερη απόδοση.

Η τελική έκδοση του LoRaFEC που επιλέγουμε είναι ο τριαδικός συνδυασμός που χρησιμοποιεί και τις 2 μεθόδους βελτίωσης. Η έκδοση αυτή, ανάμεσα σε όλες τις εκδόσεις που έχουμε δοκιμάσει, δίνει την καλύτερη απόδοση ενώ δεν υπάρχει μεγάλη διαφορά σε όσον αφορά τον επιπλέον φόρτο και καθυστέρηση.

5

Προσομοίωση

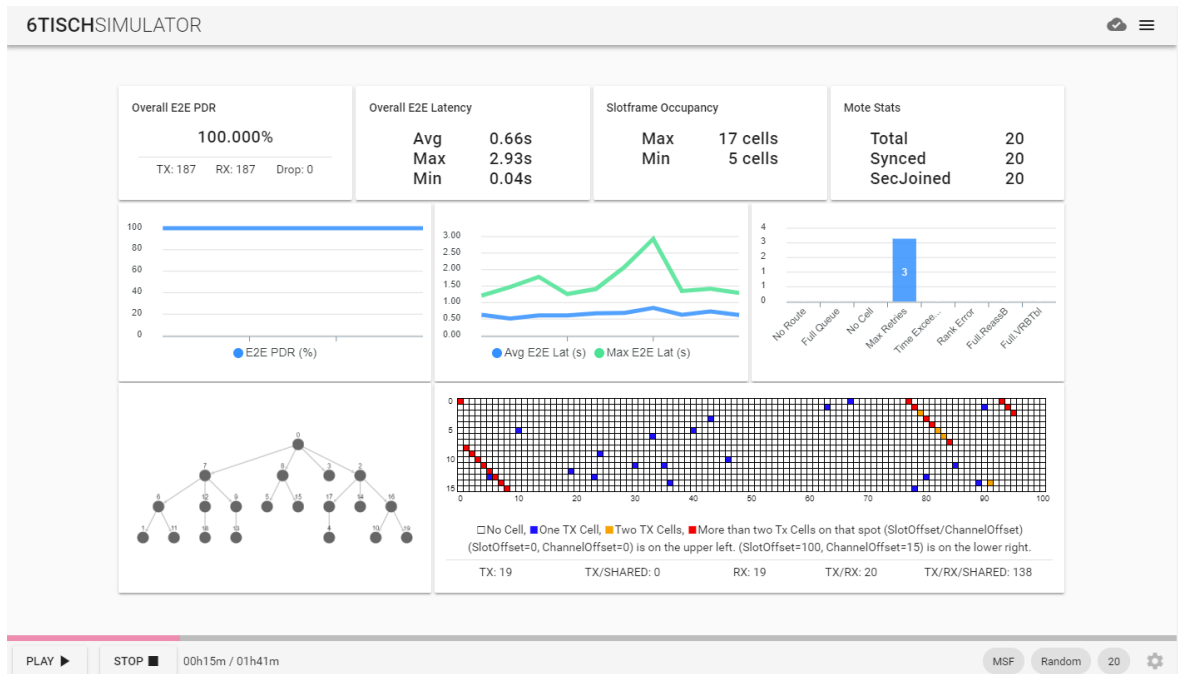
Για τη σύγκριση των μεθόδων εκτελέστηκαν προσομοιώσεις. Οι προσομοιώσεις αφορούν την λειτουργία των κόμβων σε δίκτυο και αξιολογήσαμε την απόδοση του δικτύου χρησιμοποιώντας τα αποτελέσματα των προσομοιώσεων

5.1 Εργαλείο προσομοίωσης - 6Tisch προσομοιωτή

Για την εκτέλεση των προσομοιώσεων χρησιμοποιήθηκε ο προσομοιωτής 6Tisch [19]. Αυτός ο προσομοιωτής είναι γραμμένος από γλώσσα Python και η λειτουργία του καθορίζεται από σειρά γεγονότων. Προσομοιώνει ένα δίκτυο όπως περιγράφεται στα 6Tisch πρωτόκολλα, δηλαδή τα κυριότερα χαρακτηριστικά του δικτύου είναι χαμηλή ποιότητα δικτύου και οι κόμβοι είναι περιορισμένης ικανότητας, είτε σε ενέργεια είτε σε επεξεργαστική ικανότητα.

Ένα σημαντικό χαρακτηριστικό του προσομοιωτή που πρέπει να σημειώσουμε είναι ότι δεν ασχολείται με ιδιότητες που αφορά τα χαρακτηριστικά των συσκευών όπως λειτουργίες του φυσικού επιπέδου. Παράδειγμα τέτοιας λειτουργίας είναι ο χρόνος καθυστέρησης που αφορά επεξεργασία υλικού. Ο προσομοιωτής συγκεντρώνει την προσοχή του σε λειτουργίες του δικτύου που αφορούν το επίπεδο πρόσβασης στο μέσο (Medium Access Control – MAC sublayer).

Υπάρχουν 2 τρόποι χρήσης του προσομοιωτή, μία είναι μέσω του πίνακα εντολών, ενώ η άλλη είναι μέσω WebApp που δίνει και γραφική παράσταση των προσομοιώσεων, όπως φαίνεται στην εικόνα 39. Αυτή που χρησιμοποιήθηκε από μας είναι η πρώτη, μέσω πίνακα εντολών, όπως φαίνεται στην εικόνα 40.



Εικόνα 39. Πίνακας ελέγχου στο WebApp του 6Tisch προσομοιωτή

```

chen@chen-X555LD: ~/Desktop/Test-6tisch/bin
文件(F) 编辑(E) 查看(V) 搜索(S) 终端(T) 帮助(H)
extract_config_json.py  runSim.py
__init__.py            runSim.sh
chen@chen-X555LD:~/Desktop/Test-6tisch/bin$ sh sim_config85_testver.sh
parameters 1/1, run 1/1
ALLOCATION :
('Node ', 0, ' has ', 0, ' transmission cells')
('Node ', 1, ' has ', 20, ' transmission cells')
('Node ', 2, ' has ', 20, ' transmission cells')
('Node ', 3, ' has ', 20, ' transmission cells')
('Node ', 4, ' has ', 20, ' transmission cells')
('Node ', 5, ' has ', 20, ' transmission cells')
('Node ', 6, ' has ', 20, ' transmission cells')
('Node ', 7, ' has ', 20, ' transmission cells')
('Node ', 8, ' has ', 20, ' transmission cells')
('Node ', 9, ' has ', 20, ' transmission cells')
Mote0 just woke up
Mote1 just woke up
Mote2 just woke up
Mote3 just woke up
Mote4 just woke up
Mote5 just woke up
Mote6 just woke up
Mote7 just woke up
Mote8 just woke up

```

Εικόνα 40. Γραμμή εντολών

Για την επιτυχή διεξαγωγή των προσομοιώσεων χρησιμοποιήθηκαν 2 εργαλεία, ο προσομοιωτής 6Tisch και η πλατφόρμα Grid'5000. Όπως έχουμε περιγράψει περιληπτικά προηγουμένως, ο προσομοιωτής 6Tisch είναι γραμμένος στην γλώσσα Python και προσομοιώνει τις λειτουργίες ενός δικτύου σύμφωνα με την περιγραφή στα 6Tisch πρωτόκολλα. Δεν υλοποιεί λειτουργίες που αφορούν το φυσικό επίπεδο και προσομοιώνει μόνο τις λειτουργίες του δικτύου στο MAC επίπεδο.

5.1.1 Σύγκριση προσομοιωτών

Υπάρχουν πολλές επιλογές για προσομοιωτές δικτύου, αναφέρουμε εδώ ενδεικτικά τους παρακάτω:

1. NS-3
2. OMNet++
3. TOSSIM
4. Cooja
5. Opensim

Η σύγκριση αυτών των 5 επιλογών βρίσκεται στην εικόνα 44.

Simulator	Learning Curve	Scalability	6TiSCH implementation	Standard-Compliant
ns-3	High	Medium	None	N/A
OMNet++	High	Medium	None	N/A
TOSSIM	Medium	High	None	N/A
Cooja (emulator)	High	Low	Yes (Partial)	Partially
OpenSim (emulator)	High	Low	Yes	Yes (byte-accurate)
6TiSCH Simulator	Low	High	Yes	Yes (behavioral)

Εικόνα 44. Σύγκριση προσομοιωτών

Τα 4 κριτήρια που χρησιμοποιήθηκαν για την σύγκριση είναι τα εξής:

1. Δυσκολία μάθησης
2. Δυνατότητα επέκτασης
3. Αν υπάρχει υλοποίηση του 6Tisch
4. Αν ακολουθεί τις προδιαγραφές που ορίζει τα πρωτόκολλα

Από την σύγκριση που πραγματοποιήθηκε, επιλέγουμε τελικά τον προσομοιωτή 6Tisch ως το εργαλείο που θα χρησιμοποιήσουμε για 3 λόγους:

1. Υπάρχει ολοκληρωμένη υλοποίηση του 6Tisch και ακολουθεί όλες τις υποχρεώσεις που περιγράφονται από τα πρωτόκολλα 6Tisch.
2. Είναι εύκολη η κλιμάκωση του δικτύου, δηλαδή είναι εύκολο να φτιάξουμε δίκτυα με εκατοντάδες κόμβους.
3. Είναι εύκολη η δομή του κώδικα και μπορεί με ευκολία να υλοποιήσουμε πάνω σε αυτό καινούργιες λειτουργίες και καινούργιες στρατηγικές διαχείρισης δικτύου.

Σε γενικές γραμμές, το 6Tisch προσομοιωτή είναι πολύ ιδανικό για σχεδίαση και δοκιμή καινούργιων πρωτοτύπων.

5.2 Παράμετροι προσομοίωσης

Οι προσομοιώσεις που τρέξαμε χρησιμοποιώντας τον συγκεκριμένο προσομοιωτή αφορά διάφορους μεθόδους FEC και είναι τα εξής:

1. MFF
2. XORFEC
3. RFEC
4. NCFEC
5. LoRaFEC

Ανάμεσα σε αυτά, η μέθοδος LoRaFEC είναι αυτή που προτείνουμε και είναι η τελική έκδοση που έχουμε επιλέξει, δηλαδή, ένωση σειριακό και μη σειριακό συνδυασμό. Για τις άλλες μέθοδοι, υπάρχει περιγραφή στο παράγραφο 3, που αναλυτικά παρουσιάζουμε τις άλλες λύσεις που υπάρχει.

Ο λόγος που δεν συγκρίνουμε την μέθοδο που μας έδωσε την ιδέα, δηλαδή την LoRaFFEC, είναι επειδή η ιδέα του για ανάκτηση των χαμένων fragments είναι για το σύνολο των πακέτων και είναι δύσκολη η σύγκριση του με άλλους μεθόδους που η ανάκτηση των χαμένων fragments αφορούν fragments που βρίσκονται στο ίδιο πακέτο. Οι βασικοί παράμετροι των προσομοιώσεων υπάρχουν στον παρακάτω πίνακα.

Παράμετρος	Τιμές
TSCH slotframe length	101 slots
Slot duration	10 ms
Link reliability	0.65 ~ 0.95
Maximum number of retries	3
Packet interval	54s ~ 66s
Number of fragments per packet	[1 ,15]
Number of simulation runs	10
Duration of each simulation	1000 s
NCFEC target PDR	0.99
NCFEC maximum redundancy factor	3

Πίνακας 1. Παράμετροι προσομοίωσης

Επειδή ο αριθμός των προσομοιώσεων που τρέχουμε είναι μεγάλος, χρησιμοποιούμε την πλατφόρμα Grid'5000 για την επιτάχυνση των προσομοιώσεων.

Οι προσομοιώσεις εκτελέστηκαν πάνω σε τοπολογία που αποτελείται από 10 κόμβους με γραμμική διάταξη. Ο κόμβος ρίζα είναι ο τελικός προορισμός και ο κόμβος 9 είναι ο μόνος αποστολέας. Κάθε κόμβος είναι εξοπλισμένος με 20 κελιά αποστολής για να στείλει δεδομένα προς τους γονείς του και 20 κελιά αποδοχής για να λαμβάνει δεδομένα από τα παιδιά του. Η τοπολογία είναι όπως φαίνεται στην παρακάτω εικόνα.



Εικόνα 41. Τοπολογία

5.3 Πλατφόρμα Grid'5000

Η πλατφόρμα Grid'5000 [20] είναι ένα ευέλικτο testbed που διαθέτει δυνατότητες κλιμάκωσης. Μπορεί να χρησιμοποιηθεί σε όλους τους τομείς της πληροφορικής που αφορούν εκτέλεση πειραμάτων.

Εικόνα 45. Grid5000

Τα κυριότερα χαρακτηριστικά της πλατφόρμας είναι τα εξής:

1. Μεγάλος αριθμός πόρων που μπορεί να αξιοποιηθεί, για παράδειγμα, 15000 πυρήνες, 800 κόμβοι που είναι οργανωμένοι σε ομάδες και διάφορους συνδυασμούς τεχνολογικού εξοπλισμό
2. Μεγάλη ελευθερία στην εκτέλεση πειραμάτων.
3. Αξιοπιστα παρακολούθηση στην λειτουργία των κόμβων, όποτε δίνει ακριβές αποτελέσματα όσον αφορά την λειτουργία του δικτύου ή την ενεργειακή κατανάλωση των κόμβων.
4. Υποστηρίζει την ανοικτή επιστήμη, δηλαδή είναι προσβάσιμη σε όλα τα κοινωνικά επίπεδα, ερασιτεχνικά ή επαγγελματικά.
5. Δυνατή τεχνική ομάδα που υποστηρίζει την λειτουργία της πλατφόρμας.

Οι πόροι της πλατφόρμας είναι κατανομημένα σε 8 διαφορετικά σημεία στην Γαλλία όπως φαίνεται στην εικόνα 46.



Εικόνα 46. Κατάταξη του Grid'5000

Για την χρήση της πλατφόρμας απαιτείται αίτηση και έγκριση από την τεχνική ομάδα που τη συντηρεί. Για αυτούς που επιθυμούν να χρησιμοποιήσουν την πλατφόρμα αλλά δεν ανήκουν σε ομάδα ερευνών στην Γαλλία, υπάρχει η επιλογή Open Access program.

Το συγκεκριμένο πρόγραμμα δίνει την δυνατότητα να χρησιμοποιήσουν τους πόρους υπό συγκεκριμένους περιορισμούς. Οι περιορισμοί που θέτουν είναι τα εξής:

1. Μπορεί να τρέξει μόνο ένα πείραμα σε κάθε σημείο
2. Δεν μπορεί να προγραμματίσει εκτελέσεις πειραμάτων
3. Για ένα πείραμα, αν περιμένει στην σειρά πάνω από 24 ώρες, τότε αναβάλλεται η εκτέλεση
4. Μικρότερη προτεραιότητα στην ουρά αναμονής

Η μέγιστη διάρκεια που μπορεί ένας μέλος του Open Access program είναι 2 μήνες. Ο 1^{ος} μήνας δίνεται όταν γίνει η αρχική χορήγηση του λογαριασμού και ο 2^{ος} μήνας δίνεται μετά από αίτηση στην τεχνική ομάδα πριν την λήξη της συνδρομής.

Στην λήξη της συνδρομής, ο μέλος έχει την υποχρέωση να απάντησε στο ερωτηματολόγιο που δίνεται από την τεχνική ομάδα. Οι ερωτήσεις που περιέχονται στο ερωτηματολόγιο είναι στην εικόνα 47.

```
(1) What is your Grid'5000 username ?  
<...>  
(2) What is your current situation (work status/employer/university/laboratory/project) ?  
<...>  
(3) What is your research topic, in a few lines ?  
<...>  
(4) What experiments have you been conducting on Grid'5000, in a few lines ?  
<...>  
(5) What Grid'5000 resources did you use and/or were/are you looking for (special hardware, cluster size, ...) ?  
<...>  
(6) What features of Grid'5000 did you use and/or were/are the most important to you (bare-metal deployment, network reconfiguration, ... ) ?  
<...>  
(7) Did you produce any outcome and/or publish any results (scientific papers, software,...) thanks to Grid'5000 ?  
<...>
```

Εικόνα 47. Ερωτηματολόγια μετά τη τέλος της συνδρομής

6

Συμπεράσματα

Όπως αναφέραμε προηγουμένως, συγκρίναμε την πρόταση μας με διάφορες άλλες μεθόδους που έχουμε μελετήσει. Ο σκοπός της σύγκρισης είναι να ελέγξουμε την απόδοση της πρότασης μας και να εξετάσουμε αν υπάρχει περιθώριο για περαιτέρω βελτίωση.

6.1 Παράμετροι αξιολόγησης

Για να αξιολογήσουμε την απόδοση ενός δικτύου, λαμβάνουμε υπόψη μας τους παρακάτω 3 δείκτες:

1. End - to - End ποσοστό πακέτων που έχουν παραδοθεί (Packet Delivery Ratio - PDR) για να αξιολογήσουμε την αξιοπιστία του δικτύου
2. End - to - End καθυστέρηση για τα πακέτα που λαμβάνει
3. Αριθμός των fragments για να μετρήσουμε το μέγεθος της κίνηση του δικτύου

Ανάμεσα στους 3 δείκτες, ο αριθμός των fragments, παραμένει σταθερός για τις περισσότερους μεθόδους που θα εξετάσουμε καθώς η αύξηση αυτού του μετρητή σχετίζεται μόνο και μόνο με την αύξηση του αριθμού των data fragments. Ενώ στην NCFEC και στην πρόταση μας, LoRaFEC, ο αριθμός αυτός σχετίζεται με πολλούς παράγοντες όπως ποιότητα του δικτύου, εκτιμώμενη απόδοση δικτύου και την επιθυμητή απόδοση του δικτύου που έχουμε ορίσει στην αρχή της προσομοίωσης.

6.2 Οργάνωση πειραμάτων

Οι προσομοιώσεις αφορούν τους όλες τις μεθόδους εκτός από την LoRaFEC με αριθμό των data fragments ίσο με 1 μέχρι 10 (σε κάποιες προσομοιώσεις χρησιμοποιείται και μεγαλύτερο από 10). Όλες οι προσομοιώσεις με το ίδιο αριθμό data fragments εκτελούνται 10 φορές με διαφορετικό random seed, για να πραγματοποιούνται διαφορετικοί υπολογισμοί για κάθε προσομοίωση. Οι τιμές που παίρνουμε για κάθε αριθμό data fragments είναι η μέση τιμή από τις 10 προσομοιώσεις.

Συγκρίνουμε την αξιοπιστία με την παράμετρο PDR, δηλαδή το ποσοστό των πακέτων που έχει φτάσει με επιτυχία. Την καθυστέρηση με τον χρόνο που χρειάζεται να φτάσει από τον αποστολέα προς τον παραλήπτη. Το μέγεθος του επιπλέον φόρτου με τον μέσο όρο του αριθμού των fragments που έχει σταλεί για κάθε πακέτο. Αυτή η τιμή είναι σταθερή για κάποιες από τις μεθόδους, την MFF, το XORFEC και το RFEC. Αυτό που συγκρίνουμε κυρίως με αυτόν την παράμετρο είναι το NCFEC και το LoRaFEC. Οι δύο μέθοδοι αλλάζουν τον αριθμό των redundant fragments ανάλογα με την κατάσταση που αντιμετωπίζουν. Αυτό που προσπαθούμε να πετύχουμε είναι 100% PDR με χαμηλό χρόνο καθυστέρησης και μικρό επιπλέον φόρτο στο δίκτυο.

6.3 Αποτελέσματα

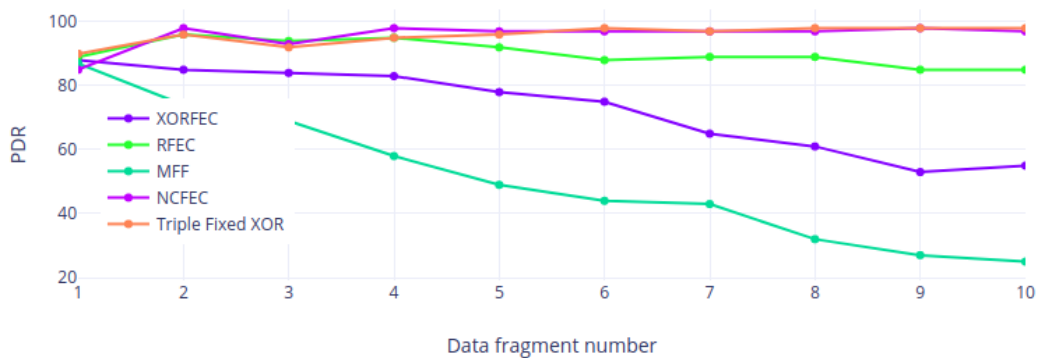
Βλέποντας τα αποτελέσματα στην εικόνα 42, το πρώτο συμπέρασμα που βγάζουμε είναι ότι η μέθοδος FEC που χρησιμοποιείται για τη βελτίωση της απόδοσης δεν μπορεί να εφαρμοστεί σε περίπτωση που το μέγεθος του πακέτου δεν έχει ξεπεράσει το MTU, δηλαδή τα 127 bytes. Ο λόγος είναι ότι σε αυτές τις περιπτώσεις δεν εφαρμόζεται αποκοπή του πακέτου, άρα δεν υπάρχει περιθώριο εφαρμογής για κωδικοποίηση των fragments. Σε όλες τις επιλογές που βλέπουμε εδώ, το αποτέλεσμα των 3 δεικτών όταν ο αριθμός του fragment ισούται με 1, είναι παρόμοιο.

Το καλύτερο αποτέλεσμα που προσπαθούμε να πετύχουμε είναι να βγάλουμε καταρχήν και για τους 3 δείκτες καλύτερο αποτέλεσμα από το MFF. Μετά θα επιχειρήσουμε να πλησιάσουμε όσο τον δυνατόν πιο κοντά στο βέλτιστο αποτέλεσμα.

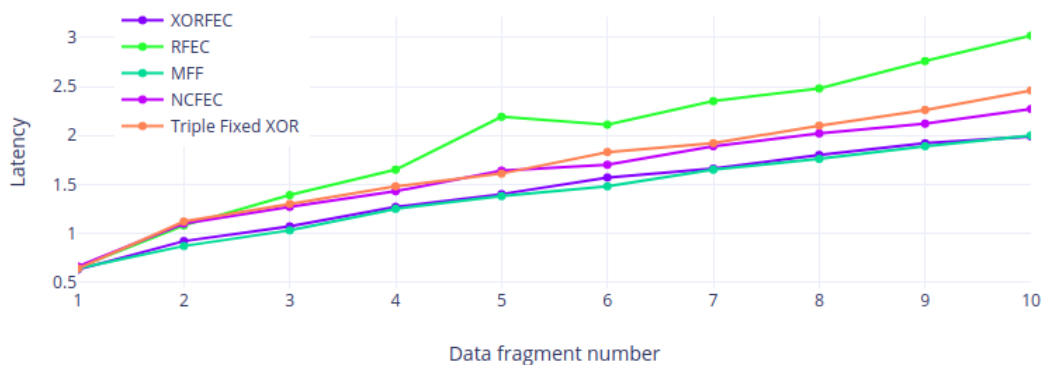
Αυτό που υποθέτουμε είναι ότι αυξάνοντας τον φόρτο μπορούμε να επιτύχουμε καλύτερη απόδοση αλλά θα έχουμε μεγαλύτερο χρόνο καθυστέρησης. Αυτό είναι επειδή η καλύτερη απόδοση επιτυγχάνεται από την αύξηση επιπλέον φόρτος στο δίκτυο, δηλαδή τα επιπλέον fragment που στέλνονται μετά από την αποστολή των data fragments, είτε αυτά είναι επανάληψη της αποστολής, είτε είναι redundant fragments που μπορεί να χρησιμοποιηθεί για την ανάκτηση των χαμένων fragments. Όπως περιμένουμε και εμείς από τη αρχή, οι μέθοδοι που χρησιμοποιούν τεχνικές FEC δίνουν καλύτερη απόδοση σε σύγκριση με την MFF.

Η αύξηση της απόδοσης είναι πολύ εμφανής στην σύγκριση, έστω και όταν ο αριθμός των fragments που αποστέλλεται είναι μικρός. Για αριθμό data fragments ίσο με 2 σε περιβάλλον

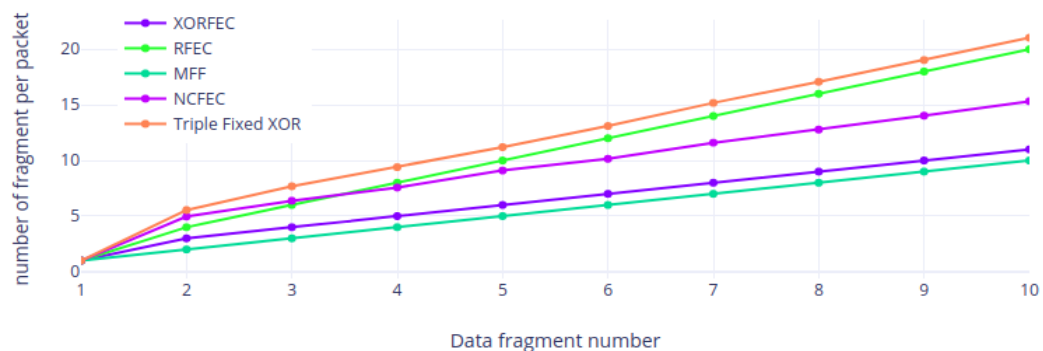
όπου η ποιότητα του δικτύου ισούται με 0.65, βλέπουμε το MFF να επιτυγχάνει 74% απόδοση, ενώ οι μέθοδοι που χρησιμοποιούν τεχνικές FEC, ακόμα και η πιο ελαφριά έκδοση που προτείνεται, δηλαδή το XORFEC, βγάζει καλύτερο αποτέλεσμα. Η απόδοση που επιτυγχάνει το XORFEC είναι 85%. Η αύξηση της απόδοσης φαίνεται πιο ξεκάθαρα και όταν αυξάνεται ο αριθμός των data fragments. Βλέπουμε στην εικόνα ότι η MFF παρουσιάζει μεγάλη κλίση άλλους τα κάτω όταν αυξάνεται ο αριθμός των data fragments, αλλάλλους αλλους μεθόδους δεν παρουσιάζεται παρόμοια κλίση, εκτός από την XORFEC. Αλλά η κλίση του XORFEC είναι μικρότερη από αυτή του MFF, επιτυγχάνει καλύτερη απόδοση και όταν ο αριθμός των data fragments είναι μεγάλο. Το XORFEC έχει απόδοση 55% ενώ η MFF έχει μόλις 25% απόδοση.



A) απόδοση του δικτύου



B) χρόνος καθυστέρησης



Γ) συνολικός αριθμός των fragments

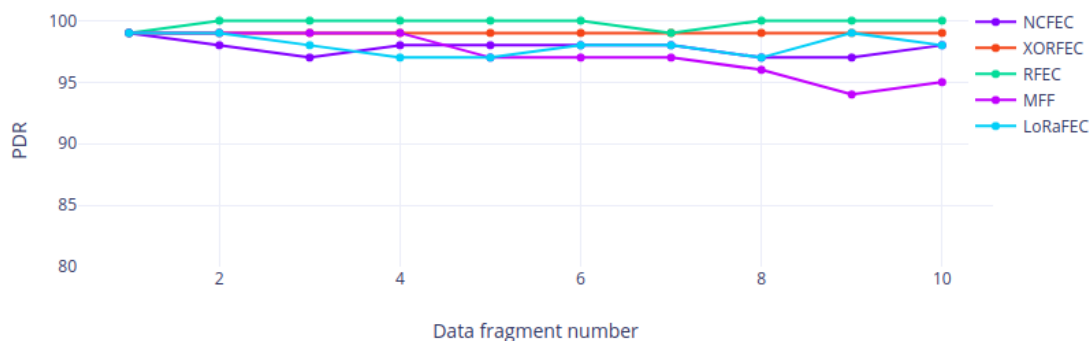
Εικόνα 42. Σύγκριση των μεθόδων για ποιότητα του δικτύου ίσον με 0.65

Η αύξηση του φόρτου δημιουργεί μεγαλύτερη καθυστέρηση αλλά αυτό είναι αποδεκτό όταν η καθυστέρηση και ο επιπλέον φόρτος δεν επιβαρύνουν σημαντικά το δίκτυο. Η επιλογή που δίνει τα καλύτερα αποτελέσματα, όπως φαίνεται από την σύγκριση είναι το NCFEC, αφού επιτυγχάνει την καλύτερη απόδοση, ενώ ο φόρτος και ο χρόνος καθυστέρησης που δημιουργεί δεν είναι ο μεγαλύτερος. Το μειονέκτημα του NCFEC είναι ότι η τεχνική κωδικοποίησης που χρησιμοποιεί είναι περίπλοκη και ίσως δεν είναι η καλύτερη λύση όταν το δίκτυο αποτελείται από κόμβους που έχουν περιορισμένη επεξεργαστική ικανότητα.

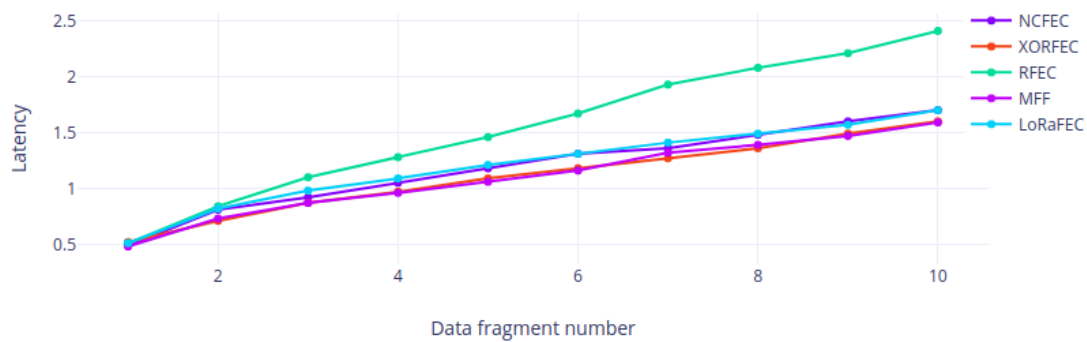
Μετά από το NCFEC, έρχονται οι μέθοδοι που χρησιμοποιούν τεχνικές FEC. Η μέθοδος που επιτυγχάνει την καλύτερη απόδοση, σε παρόμοιο επίπεδο με το NCFEC, είναι η λύση που προτείνουμε εμείς, το LoRaFEC. Αν και παρουσιάζει επιπλέον φόρτο από το NCFEC (μάλιστα ο επιπλέον φόρτος είναι ο μεγαλύτερος ανάμεσα στις μεθόδους), η αύξηση στο χρόνο καθυστέρησης είναι αποδεκτή, περίπου ίδια με το NCFEC.

Αυτό που πρέπει να σημειωθεί για το XORFEC είναι ότι η αύξηση στην απόδοση που επιτυγχάνει σε σύγκριση με την MFF είναι πολύ μεγάλη. Η μεγαλύτερη αύξηση είναι για αριθμό των fragments ίσο με 10 που φτάνει στα 30%. Η μικρότερη αύξηση που επιτυγχάνει είναι για αριθμό των fragments ίσο με 10, που είναι στα 11%.

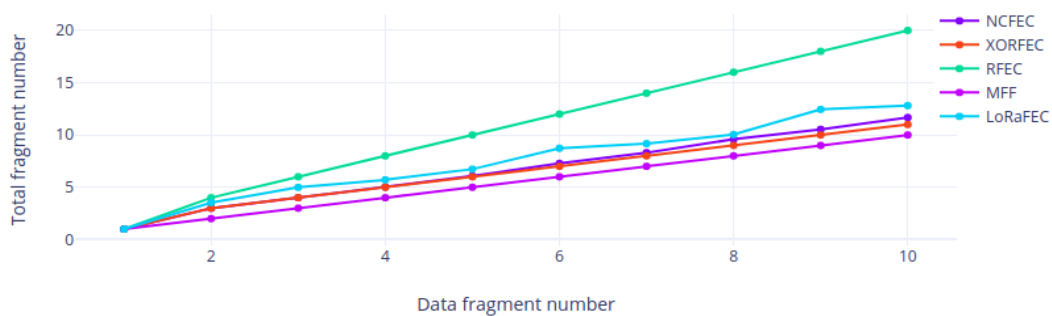
Επειδή το XORFEC είναι η πιο ελαφριά μέθοδος, το επιπλέον φόρτος που δημιουργεί είναι 1 για οποιοδήποτε αριθμό των fragments, για τον λόγο αυτό, είναι πολύ ιδανικό για δίκτυα που έχει χαμηλή ποιότητα σύνδεσης, οι κόμβοι έχει πολύ περιορισμένη επεξεργαστική ικανότητα αλλά έχει μια απαίτηση για την απόδοση. Σε τέτοιες περιπτώσεις, μπορούμε να χρησιμοποιήσουμε το XORFEC.



A) απόδοση του δικτύου



B) χρόνος καθυστέρησης



Γ) συνολικός αριθμός των fragments

Εικόνα 43. σύγκριση σε 0.85 περιβάλλον

Παρατηρώντας τη σύγκριση που έγινε σε δίκτυο με ποιότητα σύνδεσης ίση με 0.85 (εικόνα 43) παρατηρούμε ότι σε δίκτυα που η ποιότητα της σύνδεσης είναι καλή, η διαφορά στην απόδοση που επιτυγχάνουν οι μέθοδοι είναι μικρότερη. Αν η απόδοση του XORFEC είναι αποδεκτή, τότε μπορούμε να συνδυάσουμε τις μεθόδους, δηλαδή για μικρό αριθμό των fragments να χρησιμοποιείται το XORFEC που έχει την μικρότερη επιβάρυνση και για μεγάλο αριθμό των fragments να χρησιμοποιείται άλλη μέθοδος.

Ένας άλλος λόγος που προτείνουμε το συνδυασμό των μεθόδων είναι ότι ο καθορισμός του αριθμού των redundant fragments δεν επηρεάζεται από τον ίδιο παράγοντα, άρα έχουμε καλύτερη αντιμετώπιση του δικτύου. Το XORFEC στέλνει ένα επιπλέον fragment όσο και να είναι ο αριθμός των fragments και δεν επηρεάζεται από την ποιότητα της σύνδεσης. Η NCFEC στέλνει παραπάνω περίπου τέσσερα με έξι fragments όταν η ποιότητα της σύνδεσης είναι 0.65 και πάντα 2 fragments για ποιότητα της σύνδεσης 0.85. Η RFEC πάντα διπλασιάζει τον αριθμό των fragments, το ίδιο και η πρότασή μας, η LoRaFEC. Αν επιτρέπεται από τις απαιτήσεις του δικτύου μπορούμε να χρησιμοποιήσουμε την XORFEC όταν θέλουμε να ελαφρύνουμε το επιπλέον φόρτος, την NCFEC ή την LoRaFEC σε περίπτωση που απαιτείται υψηλή απόδοση. Για την RFEC, μπορούμε να το χρησιμοποιήσουμε σε περιπτώσεις που οι

κόμβοι έχουν πολύ περιορισμένη υπολογιστική ικανότητα, για τι στην μέθοδο αυτή δεν χρειάζεται η κωδικοποίηση των fragments.

6.4 Σύνοψη συμπερασμάτων αξιολόγησης

1. Οι μέθοδοι που δίνουν τα καλύτερα αποτελέσματα στην αξιοπιστία είναι το NCFEC και η LoRaFEC, με περίπου ίδιο χρόνο καθυστέρησης.
2. Η μέθοδος με το χαμηλότερο χρόνο καθυστέρησης και επιπλέον φόρτο είναι η MFF αλλά έχει την χειρότερη αξιοπιστία.
3. Το XORFEC με μικρή αύξηση στην καθυστέρηση και επιπλέον φόρτο δίνει πολύ καλύτερη αξιοπιστία.
4. Προτείνεται χρήση συνδυασμός μεθόδων ανάλογα με τις απαιτήσεις των διαφορετικών περιπτώσεων χρήσης.

7

Επίλογος

7.1 Σύνοψη και συμπεράσματα

Στην παρούσα διπλωματική αναλύσαμε για ποιους λόγους η χρήση της τεχνικής FEC είναι ένα καλό εργαλείο για την αύξηση της απόδοσης και παρουσιάσαμε 4 υπάρχουσες λύσεις. Είναι το XORFEC που προσθέτει ένα επιπλέον fragment, το περιεχόμενο του είναι η XOR όλων των data fragments, άρα σε περίπτωση που υπάρχει μόνο ένα χαμένο fragment, μπορεί να χρησιμοποιήσει για την ανάκτηση του. Η δεύτερη πρόταση που παρουσιάσαμε είναι η RFEC η οποία επαναλαμβάνει την αποστολή των data fragments. Η Τρίτη πρόταση είναι το NCFEC η οποία εφαρμόζει κωδικοποίηση στα data fragments και τα μετατρέπει σε encoded fragments. Ο αριθμός των encoded fragments είναι μεγαλύτερος από τον αριθμό των data fragments. Αν στο τελικό προορισμό φτάσει αριθμός fragments ίσος με το αριθμό των data fragments, τότε η αποστολή θεωρείται επιτυχής και μπορεί να συναρμολογήσει το πακέτο. Η τελευταία λύση που παρουσιάζουμε είναι η LoRaFFEC. Αυτή η μέθοδος κωδικοποιεί τα data fragments σε κάποια redundant fragments, όπου χρησιμοποιώντας την πληροφορία που περιέχει μπορεί να γίνει ανάκτηση των data fragments. Ένα σημαντικό χαρακτηριστικό της LoRaFFEC είναι ότι συγκρατεί σε ένα πίνακα όλα τα fragments, άρα θεωρητικά, μπορεί να γίνει ανάκτηση όποιο fragment και να χαθεί.

Κατόπιν, παρουσιάσαμε τη δική μας πρόταση με όνομα LoRaFEC (καθώς η ιδέα προέρχεται από την LoRaFFEC). Ακολουθήσαμε 3 στάδια για τον καθορισμό της τελικής έκδοσης της πρότασης μας στοχεύοντας στη βέλτιστη απόδοση της. Πρώτα, αυξήσαμε τον αριθμό των redundant fragments, γιατί συγκρίνοντας όλες τις εκδόσεις που έχουμε σκεφτεί, παρατηρούμε

ότι δεν μπορεί να δώσει καλό αποτέλεσμα εάν ο αριθμός των redundant fragments είναι κάτω από συγκεκριμένο όριο. Ο τελικός αριθμός αποφασίστηκε να είναι στα $2x-1$, δηλαδή ο αριθμός των redundant fragments να ισούται με τον αριθμό των data fragments μείον 1.

Το δεύτερο στάδιο περιλάμβανε να καθορίσουμε το περιεχόμενο των redundant fragments. Μετά από διάφορες δοκιμές, η τελική επιλογή είναι η ένωση των 2 μεθόδων τριαδικών συνδυασμών. Το τελευταίο στάδιο ήταν να καθορίσουμε αν θα χρησιμοποιηθούν οι 2 μέθοδοι βελτίωσης που σχεδιάσαμε. Το συμπέρασμα στο οποίο καταλήξαμε ήταν η τελική έκδοση η οποία είναι η ένωση των τριαδικών συνδυασμών που χρησιμοποιεί και τις 2 μεθόδους βελτίωσης.

Συγκρίνοντας τις υπάρχουσες λύσεις με την πρόταση μας, βγάζουμε το συμπέρασμα ότι η λύση μας δίνει την καλύτερη απόδοση, σε παρόμοιο επίπεδο με την NCFEC. Αν και δημιουργεί περισσότερο επιπλέον φόρτο από όλες τις άλλες επιλογές, δεν υπάρχει μεγάλη αύξηση στο χρόνο καθυστέρησης.

Συγκρίνοντας τα χαρακτηριστικά των μεθόδων, προτείνεται να χρησιμοποιείται διαφορετική μέθοδος αναλόγως την περίπτωση. Για δίκτυα με χαμηλή ποιότητα σύνδεσης του δικτύου και με κόμβους που διαθέτουν περιορισμένη επεξεργαστική ικανότητα, μπορεί να χρησιμοποιηθεί η μέθοδος με την ελαφρύτερη υλοποίηση της τεχνικής FEC, το XORFEC. Σε παρόμοια περίπτωση αλλά αν οι κόμβοι έχουν ακόμα πιο περιορισμένη επεξεργαστική ικανότητα, μπορεί να χρησιμοποιηθεί η RFEC στην οποία δεν απαιτείται κωδικοποίηση των fragments. Για δίκτυα που απαιτούν υψηλή αξιοπιστία, αν οι κόμβοι έχουν την ικανότητα κωδικοποίησης και ο σχετικά μεγάλες αναμενόμενες τιμές του χρόνου καθυστέρησης είναι αποδεκτές, τότε η καλύτερη λύση είναι είτε το NCFEC ή η LoRaFEC, αναλόγως με την επεξεργαστική ικανότητα των κόμβων.

7.2 Μελλοντικές επεκτάσεις

Η μέθοδος LoRaFEC που προτείναμε έχει ακόμα πολλές δυνατότητες και οι εκδόσεις που έχουμε δοκιμάσει μέχρι τώρα είναι απλές. Οι πιθανές επεκτάσεις της είναι οι εξής:

1. Να αυτοματοποιήσουμε την διαδικασία καθορισμού του αριθμού των redundant fragments, ειδικά σε περιπτώσεις όπου ο αριθμός των data fragment είναι πολύ μεγάλος
2. Δοκιμή νέων μεθόδων βελτίωσης
3. Βελτίωση της μεθοδολογίας με αναλογίες κατηγοριών
4. Δοκιμή νέων συνδυασμών των data fragments
5. Συνδυασμός των εκδόσεων του LoRaFEC

Ο καθορισμός του αριθμού των redundant fragments αυτή την στιγμή γίνεται με χειροποίητο τρόπο, καθορίζεται από τον χρήστη από την αρχή της λειτουργίας και δεν υπάρχει μεγάλες αλλαγές. Οι τιμές που καθορίζονται αναλόγως με την ποιότητα της σύνδεσης φαίνονται στην εικόνα 48.

```
.if self.settings.link_pdr <= 0.65:  
.....number_of_fragments = number_of_original_fragments + number_of_original_fragments - 1  
.if 0.85 > self.settings.link_pdr > 0.65:  
.....number_of_fragments = number_of_original_fragments + int(math.ceil(number_of_original_fragments / 2))  
.if self.settings.link_pdr >= 0.85:  
.....number_of_fragments = number_of_original_fragments
```

Εικόνα 48. Κώδικας καθορισμό του redundant fragments

Όπως φαίνεται από τον κώδικα, για την αυτοματοποίηση πρέπει να λαμβάνουμε υπόψη την ποιότητα της σύνδεσης, το αριθμό των data fragments, την εκτιμώμενη απόδοση που θέλουμε να επιτύχουμε και πρέπει να θέσουμε όριο για το μέγιστο αριθμό των επιπλέον φόρτος. Ένα από τα μειονεκτήματα του LoRaFEC είναι ότι δημιουργεί παραπάνω επιπλέον φόρτος σε σύγκριση με άλλους μεθόδους. Το πρόβλημα αυτό αυξάνεται καθώς αυξάνεται ο αριθμός των data fragments. Για την επίλυση του προβλήματος, μια πρόταση είναι η αυτοματοποίηση του καθορισμού, η άλλη πρόταση είναι όπως κάναμε σε αυτό το κείμενο, να δοκιμάσουμε με προσομοιώσεις τον αριθμό του redundant fragments.

Η αυτοματοποίηση μπορεί να γίνει με την δημιουργία μιας μαθηματικής πράξης που υπολογίζει την πιθανότητα της επιτυχής αποστολής του πακέτου και να αυξήσουμε ή να μειώσουμε τον αριθμό των fragments που πρέπει να αποσταλεί ανάλογα με την πιθανότητα. Η ιδέα αυτή υλοποιείται στην NCFEC.

Οι μέθοδοι βελτίωσης που δοκιμάσαμε στην παρούσα διπλωματική είναι δύο. Δύο επιπλέον παραλλαγές που μπορούμε να δοκιμάσουμε στο μέλλον. Για την πρώτη μέθοδο που χρησιμοποιεί ένα σταθερό data fragment σε όλα τα redundant fragments, το data fragment που κρατήσαμε σταθερό είναι το πρώτο. Μπορούμε να δοκιμάσουμε να κρατήσουμε σταθερό διαφορετική θέση του data fragment ή ακόμα να χρησιμοποιήσουμε μια πατέντα συνδυασμό που θα χρησιμοποιείται σε όλα τα redundant fragments. Για παράδειγμα, το 1ο και το τελευταίο data fragment να υπάρχει σε όλα τα redundant fragments.

Για την άλλη κατηγορία, δοκιμάσαμε τον συνδυασμό XOR όλων των data fragments. Μπορούμε να δοκιμάσουμε και με άλλους συνδυασμούς όπως για παράδειγμα XOR του μισού data fragments με διαδοχική ή μη διαδοχική σειρά. Μία άλλη πρόταση είναι να ορίσουμε διαφορετικούς συνδυασμοί για διαφορετικό αριθμό data fragment. Για παράδειγμα, για μικρό αριθμό data fragment να χρησιμοποιήσουμε την XOR όλων των data fragments ενώ για μεγάλο αριθμό data fragment να χρησιμοποιήσουμε XOR με μισό σε δύο redundant fragments.

Σχετικά με τη μέθοδο με αναλογίες που δοκιμάσαμε, βλέπουμε πως για τα περισσότερα σημεία όταν ο αριθμός των data fragments είναι πάνω από 10, επιτυγχάνει καλύτερη απόδοση. Η ερώτηση που έχουμε εδώ είναι αν βελτιώσουμε την αναλογία καλύτερα από τώρα, μήπως μπορούμε να έχουμε μια καλύτερη έκδοση από την τελική που έχουμε επιλέξει. Ο λόγος που έχουμε αυτή την σκέψη είναι επειδή στις αναλογίες οι προσομοιώσεις εκτελέστηκαν για αριθμό των redundant fragments ίσον με το αριθμό του data fragment μείον 2, δηλαδή χρησιμοποίησε λιγότερα επιπλέον φόρτος. Επιπλέον, η μέθοδος αυτή δεν έχει δοκιμαστεί μαζί με επιπλέον μέθοδοι βελτίωσης επειδή οι αποδόσεις που επιτυγχάνει δεν ήταν αρκετές. Μπορούμε να δοκιμάσουμε πιο αναλυτικά αυτή την μέθοδο και να το χρησιμοποιήσουμε για συνδυασμό των μεθόδων.

Σχετικά με το περιεχόμενο του συνδυασμού, δοκιμάσαμε μέχρι τριαδικό συνδυασμό. Η πρόταση είναι να δοκιμάσουμε συνδυασμούς με παραπάνω data fragment ή να ορίσουμε διαφορετικό συνδυασμό για κάθε αριθμός data fragment. Για παράδειγμα, ανάλογα με τον αριθμό του data fragment, τα redundant fragments θα χρησιμοποιεί συνδυασμούς με τα μισά ή ένα τρίτο των data fragment. Επίσης ανάλογα με τον αριθμό του data fragment είναι και η διάταξη των συνδυασμών, για τι από τα αποτελέσματα που έχουμε τώρα, βλέπουμε ότι η διαδοχική διάταξη είναι πιο ιδανικό για μικρός αριθμός data fragment και η μη διαδοχική διάταξη είναι πιο κατάλληλη για μεγάλο αριθμό data fragment.

Μια άλλη πρόταση που έχουμε για την διάταξη των συνδυασμών είναι ο τρόπος που ορίζουμε τους μη διαδοχικούς συνδυασμούς. Σε αυτό το κείμενο η διάταξη ορίζεται όπως έχουμε περιγράψει προηγουμένως. Μπορούμε να δοκιμάσουμε με διαφορετικούς συνδυασμούς όπως αυτά στην εικόνα 49.

No sequence triple χωρισμός

Data fragments	Data fragments	Data fragments
1 - 2 - 3 - 4 - 5	1 - 2 - 3 - 4 - 5	1 - 2 - 3 - 4 - 5
Redundant fragments	Redundant fragments	Redundant fragments
A : 1 XOR 2 XOR 4	A : 1 XOR 3 XOR 5	A : 1 XOR 2 XOR 5
B : 2 XOR 3 XOR 5	B : 2 XOR 4 XOR 1	B : 2 XOR 3 XOR 1

Εικόνα 49. Διαφορετικοί μη διαδοχική συνδυασμοί

Τέλος, όπως στην τελική έκδοση επιλέξαμε ένωση των τριαδικών συνδυασμών, μπορούμε να επιλέξουμε και άλλους μεθόδους στην ένωση. Για παράδειγμα, αν η μέθοδος με αναλογίες φαίνεται να επιτυγχάνει καλύτερα αποτελέσματα για μεγάλο αριθμό data fragment, τότε μπορούμε να προσθέσουμε και αυτή την μέθοδο στην ένωση.

Επιπλέον, μπορούμε να κάνουμε ένωση και με άλλους μεθόδους, όπως για παράδειγμα οι υπάρχουσες λύσεις που περιγράψαμε αναλυτικά. Για παράδειγμα, όπως έχουμε αναφέρει προηγουμένως, για δίκτυα που διαθέτουν καλή ποιότητα σύνδεσης, δεν παρατηρούμε μεγάλη διαφορά στην απόδοση που επιτυγχάνουν οι μέθοδοι, άρα μπορούμε να χρησιμοποιήσουμε την μέθοδο XORFEC για να αποφύγουμε επιπλέον φόρτο στο δίκτυο και να χρησιμοποιήσουμε την LoRaFEC όταν η ποιότητα είναι χαμηλή.

Τέλος, εκτός από τις προτάσεις επέκτασης, θέλουμε να δοκιμάσουμε προσομοιώσεις διάφορες παραμέτρους. Μερικά παραδείγματα είναι προσομοιώσεις σε διαφορετικά περιβάλλοντα, διαφορετικές τοπολογίες, με μεταβαλλόμενη ποιότητα σύνδεσης, κτλ. Επιπλέον, θέλουμε να δοκιμάσουμε τους μεθόδους όχι μόνο από προσομοιώσεις αλλά σε πραγματικό περιβάλλον [21].

Βιβλιογραφία

- [1] Wollschlaeger, M.; Sauter, T.; Jasperneite, J. The Future of Industrial Communication: Automation Networks in the Era of the Internet of Things and Industry 4.0. *IEEE Ind. Electron. Mag.* 2017, 11, 17–27.
- [2] Deering, S.; Hinden, R. Internet Protocol, Version 6 (IPv6) Specification; RFC 2460; Internet Engineering Task Force: Fremont, CA, USA, 1998.
- [3] Koutsiamanis, R.A.; Papadopoulos, G.Z.; Fafoutis, X.; Fiore, J.M.D.; Thubert, P.; Montavont, N. From Best-Effort to Deterministic Packet Delivery for Wireless Industrial IoT Networks. *IEEE Trans. Ind. Inform.* 2018, 14, 4468–4480.
- [4] "IEEE Standard for Low-Rate Wireless Networks," in *IEEE Std 802.15.4-2015 (Revision of IEEE Std 802.15.4-2011)*, vol., no., pp.1-709, 22 April 2016, doi: 10.1109/IEEESTD.2016.7460875.
- [5] G. Montenegro, N. Kushalnagar, and D. Culler, "Transmission of IPv6 Packets over IEEE 802.15.4 Networks," RFC 4944, September 2007.
- [6] Bruniaux, A.; Koutsiamanis, R.-A.; Papadopoulos, G.Z.; Montavont, N. Defragmenting the 6LoWPAN Fragmentation Landscape: A Performance Evaluation. *Sensors* 2021, 21, 1711
- [7] Hui, J.; Thubert, P. Compression Format for IPv6 Datagrams over IEEE 802.15.4-Based Networks; RFC 6282; Internet Engineering Task Force: Fremont, CA, USA, 2011.
- [8] Papadopoulos, G.Z.; Thubert, P.; Tsakalidis, S.; Montavont, N. RFC 4944: Per-Hop Fragmentation and Reassembly Issues. In *Proceedings of the 2018 IEEE Conference on Standards for Communications and Networking (CSCN)*, Paris, France, 29–31 October 2018; pp. 1–6.
- [9] Watteyne, T.; Bormann, C.; Thubert, P. On Forwarding 6LoWPAN

- Fragments over a Multi-Hop IPv6 Network; RFC 8930; Internet Engineering Task Force: Fremont, CA, USA, 2020
- [10] Tanaka, Y.; Minet, P.; Watteyne, T. 6LoWPAN Fragment Forwarding. *IEEE Commun. Stand. Mag.* 2019, 3, 35–39.
- [11] Coutaud, U.; Heusse, M.; Tourancheau, B. Fragmentation and Forward Error Correction for LoRaWAN small MTU networks. In *Proceedings of the 2020 International Conference on Embedded Wireless Systems and Networks, EWSN 2020, Lyon, France, 17–18 February 2020*; pp. 289–294.
- [12] Herrero, R. 6LoWPAN fragmentation in the context of IoT based media real time communication. *Internet Technol. Lett.* 2020, 3, e144.
- [13] Chi, K.; Zhu, Y.; Cheng, Z. Network Coding Based Mesh-Under Routing In 6LoWPAN with High End-to-End Packet Delivery Rate. In *Proceedings of the 2012 8th International Conference on Mobile Ad-hoc and Sensor Networks (MSN), Chengdu, China, 14–16 December 2012*; pp. 1–5.
- [14] Ahlswede, R.; Ning C.; Li, S.R.; Yeung, R.W. Network information flow. *IEEE Trans. Inf. Theory* 2000, 46, 1204–1216.
- [15] Yeung, R.W. Network Coding: A Historical Perspective. *Proc. IEEE* 2011, 99, 366–371.
- [16] Zhu, Y.; Chi, K.; Tian, X.; Leung, V.C.M. Network Coding-Based Reliable IPv6 Packet Delivery Over IEEE 802.15.4 Wireless Personal Area Networks. *IEEE Trans. Veh. Technol.* 2016, 65, 2219–2230.
- [17] Coutaud, U.; Heusse, M.; Tourancheau, B. Fragmentation and Forward Error Correction for LoRaWAN small MTU networks. In *Proceedings of the 2020 International Conference on Embedded Wireless Systems and Networks, EWSN 2020, Lyon, France, 17–18 February 2020*; pp. 289–294.
- [18] R. Gallager. Low-density parity-check codes. *IRE Transactions on information theory*, 8(1):21–28, 1962.
- [19] Municio, E.; Daneels, G.; Vućinić, M.; Latré, S.; Famaey, J.; Tanaka, Y.; Brun, K.; Muraoka, K.; Vilajosana, X.; Watteyne, T. Simulating 6TiSCH networks. *Trans. Emerg. Telecommun. Technol.* 2019, 30, e3494.
- [20] Balouek, D.; Carpen Amarie, A.; Charrier, G.; Desprez, F.; Jeannot, E.; Jeanvoine, E.; Lèbre, A.; Margery, D.; Niclausse, N.; Nussbaum, L.; et al. Adding Virtualization Capabilities to the Grid’5000

- Testbed. In Proceedings of the 2nd International Conference on Cloud Computing and Services Science, Porto, Portugal, 18–21 April, 2012; pp. 3–20.
- [21] Adjih, C.; Baccelli, E.; Fleury, E.; Harter, G.; Mitton, N.; Noel, T.; Pissard-Gibollet, R.; Saint-Marcel, F.; Schreiner, G.; Vandaele, J.; et al. FIT IoT-LAB: A Large Scale Open Experimental IoT Testbed. In Proceedings of the 2015 IEEE 2nd World Forum on Internet of Things (WF-IoT), Milan, Italy, 14–16 December 2015; pp. 459–464.