

**Η κωδικοποίηση δικτύου ως τεχνική ευθείας
διόρθωσης σφαλμάτων σε αναξιόπιστους
συνδέσμους**

Μαρία Κιοσέ

Διπλωματική Εργασία

Επιβλέπων: Καθηγητής Ευάγγελος Παπαπέτρου

Ιωάννινα, Μάρτιος 2024



**ΤΜΗΜΑ ΜΗΧ. Η/Υ & ΠΛΗΡΟΦΟΡΙΚΗΣ
ΠΑΝΕΠΙΣΤΗΜΙΟ ΙΩΑΝΝΙΝΩΝ**

**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
UNIVERSITY OF IOANNINA**

Ευχαριστίες

Αρχικά, θα ήθελα να ευχαριστήσω τον επιβλέποντα καθηγητή μου, κ. Ευάγγελο Παπαπέτρου, καθώς και την διδακτορικό του, Φωτεινή Καρέτση, για την πολύτιμη καθοδήγησή τους, κατά την διάρκεια της εκπόνησης διπλωματικής μου εργασίας και την άμεση ανταπόκρισή τους στις απορίες μου.

Επιπλέον, θα ήθελα να ευχαριστήσω την οικογένειά μου αλλά και τους φίλους μου, για τη στήριξή τους σε όλη τη διάρκεια της φοιτητικής μου πορείας.

Μάρτιος 2024,
Μαρία

Περίληψη

Η αποδοτική και αξιόπιστη μετάδοση δεδομένων με χαμηλή καθυστέρηση αποτελεί κρίσιμο πρόβλημα στις σύγχρονες επικοινωνίες. Η εμφάνιση εφαρμογών ευαίσθητων στην καθυστέρηση, όπως συστήματα εικονικής και επαυξημένης πραγματικότητας, μας ωθεί στην ανάπτυξη μηχανισμών για την παροχή υπηρεσιών με εξαιρετική αξιοπιστία και χαμηλή καθυστέρηση. Η τεχνική της ευθείας διόρθωσης σφαλμάτων (FEC) προσφέρει μια λύση, αλλά φαίνεται περιορισμένη όταν αντιμετωπίζει υπερβολικό αριθμό σφαλμάτων. Σε αντίθεση, η τυχαία γραμμική κωδικοποίηση δικτύου (RLNC) προτείνεται ως μια βελτιωμένη εναλλακτική, που επιτυγχάνει καλύτερη απόδοση κατά την μετάδοση δεδομένων σε αναξιόπιστα δίκτυα.

Στην παρούσα διπλωματική, μελετήσαμε μια βασική υλοποίηση της κωδικοποίησης δικτύου ως μιας τεχνικής FEC, η οποία είναι γνωστή με τον όρο systematic block-based RLNC. Στη systematic block-based RLNC, η αρχική πληροφορία που λαμβάνουμε από την πηγή, μεταδίδεται με πλεονάζουσα πληροφορία, για την προστασία της από απώλειες δεδομένων. Κατά τη διάρκεια της μελέτης, εξετάσαμε διάφορους συνδυασμούς της αρχικής πληροφορίας με την πλεονάζουσα πληροφορία, επικεντρώνοντας την προσοχή μας στην επίδραση της τοποθέτησης και της ποσότητάς της. Κυρίως, επισημάνσαμε τις διαφορές ανάμεσα σε διαφορετικούς συνδυασμούς ποσότητας και τοποθέτησης πλεονάζουσας πληροφορίας και αναλύσαμε την απόδοση αυτών των σχημάτων κωδικοποίησης κάτω από διαφορετικές συνθήκες μετάδοσης δεδομένων.

Λέξεις Κλειδιά: Κώδικες ευθείας διόρθωσης σφαλμάτων, αξιοπιστία, χαμηλή καθυστέρηση, τυχαία γραμμική κωδικοποίηση δικτύου, ρυθμός κωδικοποίησης,

Abstract

Efficient data transmission with low latency is critical in modern communications. The emergence of delay-sensitive applications, such as virtual and augmented reality systems, is pushing us to develop mechanisms for providing services with exceptional reliability and low latency. While Forward Error Correction (FEC) offers a solution, it appears limited when facing excessive errors. In contrast, Random Linear Network Coding (RLNC) is proposed as an improved alternative, enabling data transmission in networks with losses.

In this thesis, we investigated a basic implementation of network coding as a Forward Error Correction (FEC) technique, known as systematic block-based RLNC. In systematic block-based RLNC, the original information received from the source is transmitted with redundant information to protect it from data losses. During the study, we examined various combinations of the original information with the redundant information, focusing on the impact of their placement and quantity. Mainly, we highlighted the differences among different combinations of quantity and placement of redundant information and analyzed the performance of these encoding schemes under different data transmission conditions.

Keywords: Forward Error Correction codes, reliability, low latency, Random Linear Network Coding, encoding rate

Περιεχόμενα

| | |
|--|-----------|
| Κεφάλαιο 1. Εισαγωγή..... | 1 |
| 1.1 Η κωδικοποίηση δικτύου ως τεχνική διόρθωσης σφαλμάτων για αξιόπιστη μετάδοση δεδομένων..... | 1 |
| 1.2 Αντικείμενο Διπλωματικής | 4 |
| 1.3 Δομή Διπλωματικής..... | 4 |
| Κεφάλαιο 2. Θεωρητικό Υπόβαθρο και Σχετικές Εργασίες | 6 |
| 2.1 Βασικές αρχές Τυχαίας Γραμμικής Κωδικοποίησης Δικτύου(Random Linear Network Coding-RLNC) | 6 |
| 2.1.1 Κωδικοποίηση RLNC..... | 7 |
| 2.1.2 Αποκωδικοποίηση RLNC..... | 9 |
| 2.2 Κατηγοριοποίηση πρωτόκολλων RLNC | 10 |
| 2.2.1 Block-based RLNC vs sliding-window RLNC..... | 10 |
| 2.2.2 Full Vector (Non-Systematic) vs Systematic RLNC..... | 11 |
| 2.2.3 Ρυθμός κωδικοποίησης και διαχείριση πλεονάζουσας πληροφορίας σε block-based συστήματα..... | 12 |
| Κεφάλαιο 3. Υλοποίηση της συστηματικής Κωδικοποίησης Δικτύου | 13 |
| 3.1 Εισαγωγή στη βιβλιοθήκη της PyErasure | 14 |
| 3.2 Μηχανισμοί της PyErasure | 14 |
| 3.2.1 Μετάδοση πακέτων και κωδικοποίηση | 14 |
| 3.2.2 Λήψη πακέτων και αποκωδικοποίηση | 15 |
| 3.3 Έκδοση 1: Rateless block-based RLNC..... | 16 |
| 3.3.1 Μετάδοση πακέτων και κωδικοποίηση | 16 |
| 3.3.2 Λήψη πακέτων και αποκωδικοποίηση | 16 |
| 3.4 Μηχανισμοί που υλοποιήσαμε..... | 17 |
| 3.4.1 Ενημέρωση της λίστας αναμονής..... | 17 |
| 3.4.1 Μηχανισμοί διαχείρισης πακέτων..... | 17 |
| 3.4.1 Ενημέρωση χρόνου κωδικοποίησης/αποκωδικοποίησης..... | 19 |

| | | |
|--------------------|--|-----------|
| 3.4.1 | Ενημέρωση πίνακα αποκωδικοποίησης..... | 21 |
| 3.5 | Έκδοση 2: G-rated block-based RLNC..... | 21 |
| 3.6 | Έκδοση 3: k-rated block-based RLNC..... | 22 |
| 3.6.1 | Μετάδοση πακέτων και κωδικοποίηση..... | 22 |
| 3.6.1 | Λήψη πακέτων και αποκωδικοποίηση..... | 23 |
| 3.6.2 | Ενημέρωση της λίστας αναμονής..... | 24 |
| Κεφάλαιο 4. | Πειραματική Αξιολόγηση..... | 25 |
| 4.1 | Μετρικές Αξιολόγησης..... | 25 |
| 4.1.1 | <i>Average Decoding Matrix Size</i> (Μέσο μέγεθος του πίνακα αποκωδικοποίησης)..... | 25 |
| 4.1.2 | <i>Average In-Order Delay</i> (Μέση καθυστέρηση εν σειρά παράδοσης)..... | 26 |
| 4.1.3 | <i>Goodput</i> | 26 |
| 4.2 | Αξιολόγηση G-rated block-based RLNC..... | 27 |
| 4.2.1 | Πείραμα 1: Η επίδραση του <i>generation size</i> στην απόδοση της G-rated block-based RLNC..... | 27 |
| 4.2.2 | Πείραμα 2: Η επίδραση διαφορετικών επιπέδων αξιοπιστίας της G-rated block-based RLNC..... | 30 |
| 4.3 | Αξιολόγηση της k-rated block-based RLNC..... | 33 |
| 4.3.1 | Πείραμα 1: Η επίδραση του <i>code rate</i> στην απόδοση του κώδικα..... | 34 |
| 4.3.2 | Πείραμα 2: Σύγκριση των δύο στρατηγικών διαχείρισης πακέτων(drop out of order vs drop generation)..... | 39 |
| 4.3.3 | Πείραμα 3: Αξιολόγηση επίδρασης του <i>k</i> σε σχέση με το <i>n</i> στο <i>code rate</i> | 42 |
| 4.3.4 | Πείραμα 4: G-rated block based RLNC vs k-rated block-based RLNC..... | 45 |
| Κεφάλαιο 5. | Συμπεράσματα..... | 50 |
| 5.1 | Μελλοντική Έρευνα..... | 51 |

Κεφάλαιο 1. Εισαγωγή

1.1 Η κωδικοποίηση δικτύου ως τεχνική διόρθωσης σφαλμάτων για αξιόπιστη μετάδοση δεδομένων

Η αξιόπιστη μετάδοση δεδομένων με χαμηλή καθυστέρηση αντιμετωπίζεται ως διαχρονική πρόκληση στον χώρο των δικτύων. Η εμφάνιση νέων εφαρμογών που απαιτούν πολύ χαμηλή καθυστέρηση, υψηλή αξιοπιστία και υψηλούς ρυθμούς μετάδοσης δεδομένων επιβάλλει αυστηρές απαιτήσεις στα δίκτυα επικοινωνίας. Συνεπώς, η ανάγκη για ανάπτυξη εφαρμογών που είναι ικανές να επιτυγχάνουν τα εν λόγω χαρακτηριστικά αντιμετωπίζει πρόσθετους περιορισμούς, όπως η περιορισμένη χωρητικότητα και η ανάγκη για χαμηλή κατανάλωση ενέργειας. Αυτό καθιστά ακόμη πιο αυστηρό το πλαίσιο λειτουργίας των σύγχρονων δικτύων, τα οποία πρέπει να εξασφαλίζουν τη συνεχή και αποτελεσματική συνδεσιμότητα, λαμβάνοντας υπόψη τους ως εν λόγω περιορισμούς. Στα πλαίσια των δικτύων 5G και ειδικότερα στην Ultra-Reliable Low-Latency Communication (URLLC)[3][13], είναι αναγκαία η αξιόπιστη και γρήγορη μεταφορά δεδομένων, σε εφαρμογές που απαιτούν ασφαλή σύνδεση και χαμηλή καθυστέρηση, με στόχο την επίτευξη ενός χρόνου καθυστέρησης των 1 ms ανά 100 χιλιόμετρα. Αυτή η τεχνολογία εφαρμόζεται σε πεδία όπως ο αυτόνομη πλοήγηση και τα απομακρυσμένα χειρουργεία, προσφέροντας αξιόπιστες λύσεις σε απαιτητικές εφαρμογές.

Κατά τη μετάδοση των δεδομένων, ενδέχεται να συμβούν σφάλματα ή απώλειες πακέτων λόγω διαφόρων παραγόντων, όπως προβλήματα δικτύου ή συμφόρηση. Όταν ο παραλήπτης ανιχνεύει την απώλεια ενός πακέτου ή την ύπαρξη σφάλματος, ειδοποιεί τον αποστολέα για να επαναλάβει τη μετάδοση του πακέτου που χάθηκε ή παρουσίασε

σφάλμα. Αυτή η διαδικασία ονομάζεται επαναμετάδοση. Το δίκτυο εγγυάται την ασφαλή μετάδοση δεδομένων με τεχνικές επαναμετάδοσης. Παρόλα αυτά αυτές οι τεχνικές αυξάνουν πολύ την καθυστέρηση του δικτύου σε περιπτώσεις πολλών σφαλμάτων.

Έχουν προταθεί μηχανισμοί για τη μείωση της καθυστέρησης που προκαλείται από επαναμεταδόσεις. Ένας από αυτούς τους παραδοσιακούς μηχανισμούς είναι η τεχνική της ευθείας διόρθωσης σφαλμάτων (Forward Error Correction-FEC)[1]. Η ευθεία διόρθωση σφαλμάτων εκπροσωπεί μια τεχνική που αναγνωρίζει και διορθώνει έναν περιορισμένο αριθμό σφαλμάτων στα δεδομένα που μεταδίδονται, χωρίς την αναγκαιότητα επαναμετάδοσης. Στο πλαίσιο αυτής της μεθόδου, ο αποστολέας προσθέτει επιπλέον bits στο τέλος του πακέτου και στην συνέχεια ο παραλήπτης χρησιμοποιεί αυτά τα επιπλέον bits για τον εντοπισμό ή την διόρθωση των σφαλμάτων. Παρόλο που η τεχνική FEC μπορεί να βοηθήσει στην αντιμετώπιση σφαλμάτων χωρίς την ανάγκη επαναμετάδοσης δεδομένων, η πληροφορία που προστίθεται για τη διόρθωση σφαλμάτων αφορά μόνο ένα μεμονωμένο πακέτο. Αυτό σημαίνει ότι σε περιπτώσεις υπερβολικού αριθμού σφαλμάτων, η πληροφορία που απαιτείται για την διόρθωσή τους θα είναι μεγάλη και οδηγεί στην μείωση της ρυθμικόδοσης του συστήματος. Κατά συνέπεια, ο μηχανισμός δεν είναι αποτελεσματικός για μεγάλες ή συνεχόμενες απώλειες και δεν είναι επαρκής για τις αυξημένες απαιτήσεις της URLLC, η οποία απαιτεί ταυτόχρονα υπερβολικά υψηλή αξιοπιστία και χαμηλή καθυστέρηση. Ως αποτέλεσμα, υπάρχει ανάγκη για καινοτόμες προσεγγίσεις κωδικοποίησης πέραν των παραδοσιακών μεθόδων διόρθωσης σφαλμάτων.

Σε αυτό το πλαίσιο, προτείνεται η εφαρμογή της Τυχαίας Γραμμικής Κωδικοποίησης Δικτύου (Random Linear Network Coding-RLNC), ως ελπιδοφόρα προσέγγιση για να ικανοποιήσει τις απαιτήσεις της URLLC. Η Τυχαία Γραμμική Κωδικοποίηση Δικτύου (RLNC) είναι μια τεχνική κωδικοποίησης δεδομένων που βασίζεται στην θεωρία των πεπερασμένων πεδίων. Τα πακέτα που δημιουργούνται από την πηγή, τα οποία δεν έχουν υποστεί κάποια επεξεργασία, γνωστά ως native ή πηγαία πακέτα, σχηματίζουν ομάδες και στην συνέχεια, με την χρήση τυχαίων γραμμικών συνδυασμών, δημιουργούν τα κωδικοποιημένα πακέτα. Κάθε κωδικοποιημένο πακέτο στέλνεται για την προστασία των πηγαίων πακέτων. Συγκεκριμένα, το κωδικοποιημένο πακέτο στοχεύει στην ανάκτηση των πηγαίων πακέτων σε περίπτωση απώλειας. Υπάρχουν δύο κύρια οφέλη της RLNC τα οποία είναι η καλύτερη αξιοποίηση του διαθέσιμου εύρους ζώνης (δηλαδή αφήνει διαθέσιμο περισσότερο εύρος ζώνης για

καινούρια πληροφορία με την αφαίρεση των επανεκπομπών) και ανθεκτικότητα στην απώλεια πακέτων.

Ένας σημαντικός παράγοντας που επηρεάζει την αποδοτικότητα ενός πρωτοκόλλου RLNC είναι ο τρόπος καθορισμού των ομάδων πηγαίων πακέτων που θα εμπλακούν στη κωδικοποίηση. Με βάση αυτόν τον παράγοντα, οι μηχανισμοί κωδικοποίησης διακρίνονται σε δύο υλοποιήσεις, την block-based RLNC και την sliding-window RLNC. Η block-based RLNC, σχηματίζει ομάδες από πηγαία πακέτα. Κάθε ομάδα από αυτές, ονομάζεται γενιά (generation) ή block και το πλήθος των πακέτων από το οποίο αποτελείται ονομάζεται generation size (G). Η κωδικοποίηση και αποκωδικοποίηση αφορούν μόνο τα πακέτα που περιέχονται στη ίδια γενιά. Τα κωδικοποιημένα πακέτα δημιουργούνται από τυχαίους γραμμικούς συνδυασμούς των G πηγαίων πακέτων. Αυτή η προσέγγιση επιτυγχάνει χαμηλή υπολογιστική πολυπλοκότητα σε σύγκριση με την επεξεργασία ολόκληρου του μηνύματος.

Η προσέγγιση του sliding-window RLNC αφορά την κωδικοποίηση μέσω ενός δυναμικά μεταβαλλόμενου παραθύρου από πηγαία πακέτα, το οποίο κινείται κατά μήκος της ακολουθίας των πηγαίων πακέτων. Σε αυτή την προσέγγιση, υπάρχει επικάλυψη, δηλαδή ένα πακέτο μπορεί να χρησιμοποιείται σε περισσότερα υποσύνολα κατά την κωδικοποίηση.

Δύο σημαντικές υλοποιήσεις που αφορούν την μορφή των πακέτων που θα σταλούν κατά τη μετάδοση είναι η full vector RLNC και η systematic RLNC. Στην μέθοδο της full vector RLNC, τα πηγαία πακέτα, μεταδίδονται με κωδικοποιημένη μορφή. Τα κωδικοποιημένα πακέτα είναι γραμμικοί συνδυασμοί των πηγαίων πακέτων. Η διαδικασία της αποκωδικοποίησης ξεκινά όταν ληφθούν όλα τα κωδικοποιημένα πακέτα, με αποτέλεσμα αυτή η προσέγγιση να εισάγει μεγάλες καθυστερήσεις αναμονής.

Από την άλλη πλευρά, η systematic RLNC μεταδίδει τα πακέτα πηγής, αυτούσια, ακολουθούμενα από πλεονάζουσα πληροφορία (κωδικοποιημένα πακέτα). Σε περίπτωση που δεν χαθούν πακέτα πηγής, τα ληφθέντα μη κωδικοποιημένα πακέτα μπορούν να παραδοθούν αμέσως στον παραλήπτη. Αντίθετα, αν κάποια πηγαία πακέτα χαθούν, χρειάζεται να λάβουμε τόσα κωδικοποιημένα πακέτα όσα τα χαμένα πηγαία, για την αποκωδικοποίησή τους. Αυτή η υλοποίηση εισάγει μεγάλες καθυστερήσεις σε πολλές απώλειες πακέτων.

Τέλος, ένα ακόμη σημαντικό κριτήριο για την κατηγοριοποίηση των πρωτοκόλλων RLNC αναφέρεται στη συχνότητα και την ποσότητα των πλεονάζοντων κωδικοποιημένων πακέτων που εισάγονται στο πλαίσιο ενός block ή μιας γενιάς.

Σύμφωνα με το block-based σχήμα, τα πλεονάζοντα πακέτα αποστέλλονται στο τέλος του τμήματος, δηλαδή μετά τη μετάδοση των πηγαίων πακέτων. Ωστόσο, αυτό μπορεί να τροποποιηθεί με τη διάσπαση των κωδικοποιημένων πακέτων κατά μήκος της γενιάς, δημιουργώντας έτσι μια μορφή υπογενιάς. Με τη διάσπαση των κωδικοποιημένων πακέτων εντός της γενιάς, η αποκωδικοποίηση είναι ταχύτερη σε σύγκριση με την αποκωδικοποίηση όταν τα κωδικοποιημένα πακέτα λαμβάνονται στο τέλος της γενιάς.

1.2 Αντικείμενο Διπλωματικής

Σε αυτή την διπλωματική εργασία, ασχοληθήκαμε με systematic block-based RLNC, καθώς παρέχει ορισμένα πλεονεκτήματα σε σύγκριση με την full vector block-based RLNC. Τα πλεονεκτήματα αυτά αφορούν την μεγάλη ευκολία στην ανάκτηση των πηγαίων πακέτων, καθώς αυτά μεταδίδονται απευθείας στον παραλήπτη χωρίς κωδικοποίηση. Αυτό συνεπάγεται χαμηλότερη καθυστέρηση στην αποκωδικοποίηση και κάνει πιο απλή την υλοποίηση, αφού δεν απαιτείται πλήρης κωδικοποίηση και αποκωδικοποίηση των πηγαίων πακέτων.

Στα πλαίσια της παρούσας διπλωματικής εργασίας, αναπτύξαμε μια τροποποίηση της systematic block-based RLNC, χρησιμοποιώντας τη βιβλιοθήκη PyErasure. Σύμφωνα με την κλασική υλοποίηση της systematic block-based RLNC, τα πλεονάζοντα πακέτα στέλνονται στο τέλος κάθε γενιάς. Η τροποποίηση που πραγματοποιήσαμε επικεντρώνεται στο πότε θα εισάγουμε πλεονάζουσα πληροφορία κατά τη διάρκεια μιας γενιάς, καθώς και στον προσδιορισμό της ποσότητας αυτής. Τέλος, κατά την πειραματική αξιολόγηση, αποδεικνύουμε ότι η προτεινόμενη τροποποίηση επιτυγχάνει βελτιωμένη απόδοση τόσο ως προς την καθυστέρηση όσο και στη μείωση της πολυπλοκότητας της κωδικοποίησης, σε σύγκριση με την κλασική υλοποίηση της systematic block-based RLNC.

1.3 Δομή Διπλωματικής

Το υπόλοιπο της διπλωματικής οργανώνεται ως εξής. Το Κεφάλαιο 2 περιλαμβάνει τις βασικές αρχές της Τυχαίας Γραμμικής Κωδικοποίησης Δικτύου

(Random Linear Network Coding), καθώς και μια κατηγοριοποίηση των block-based τεχνικών. Το Κεφάλαιο 3 περιλαμβάνει το κύριο μέρος της διπλωματικής μας καθώς και λεπτομερή ανάλυση κάθε μηχανισμού που περιλαμβάνει. Το Κεφάλαιο 4 περιλαμβάνει τα πειράματα αξιολόγησης που χρησιμοποιήθηκαν, ακολουθούμενα από τα αποτελέσματα των προσομοιώσεων και παρατηρήσεις. Το Κεφάλαιο 5 παρέχει τη σύνοψη των ευρημάτων μας και μια μελλοντική επέκταση αυτής της εργασίας.

Κεφάλαιο 2.

Θεωρητικό

Υπόβαθρο και

Σχετικές

Εργασίες

Σε αυτό το κεφάλαιο, περιγράφουμε την τεχνική RLNC, αναφερόμαστε λεπτομερώς στην κωδικοποίηση και αποκωδικοποίηση RLNC. Στην συνέχεια αναφερόμαστε στις κατηγοριοποιήσεις του RLNC και παρέχουμε λεπτομέρειες σχετικά με τη λειτουργία τους, τα οφέλη και τα μειονεκτήματά τους.

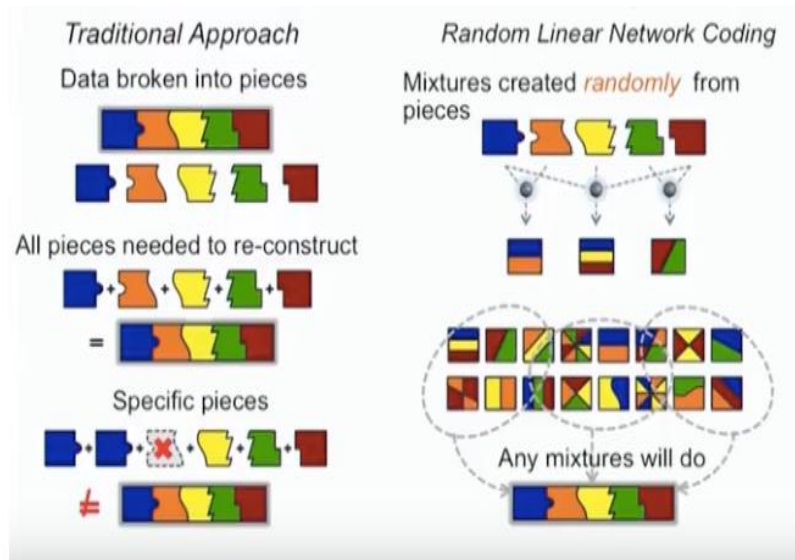
2.1 Βασικές αρχές Τυχαίας Γραμμικής

Κωδικοποίησης Δικτύου(Random Linear Network Coding-RLNC)

Η Κωδικοποίηση Δικτύου (Network Coding) και η πιο διαδεδομένη Τυχαία Γραμμική Κωδικοποίηση Δικτύου (Random Linear Network Coding-RLNC) ([10],[11]), αποτελούν βασικά εργαλεία για αποτελεσματικά δίκτυα χαμηλής καθυστέρησης.

Η RLNC, βασίζεται στην έννοια του γραμμικού συνδυασμού πολλαπλών πακέτων, χρησιμοποιώντας τη θεωρία των πεπερασμένων πεδίων ($\text{Galois Field}(2^s)$). Στην πράξη, αυτή η μέθοδος σχηματίζει ομάδες από πακέτα που δεν έχουν υποστεί κάποια

επεξεργασία, γνωστά ως native ή πηγαία πακέτα, τα οποία στη συνέχεια χρησιμοποιούνται για την παραγωγή κωδικοποιημένων πακέτων, μέσω της αριθμητικής των πεπερασμένων αριθμών.



Εικόνα 2.1 : Παραδοσιακό Κανάλι vs Κωδικοποίηση Δικτύου με RLNC. Η πηγή της εικόνας παρέχεται στο [14].

Στην Εικόνα 2.1 παρουσιάζεται ο τρόπος λειτουργίας ενός παραδοσιακού μηχανισμού, όπου τα δεδομένα διαιρούνται σε πακέτα και μεταδίδονται, με πλεονάζουσα πληροφορία να προστατεύει ένα μόνο πακέτο. Αντίθετα, στην κωδικοποίηση δικτύου, τα κωδικοποιημένα πακέτα δημιουργούνται με τη χρήση τυχαίων γραμμικών συνδυασμών των αρχικών πακέτων. Αυτό έχει ως αποτέλεσμα τα κωδικοποιημένα πακέτα να προστατεύουν τα πηγαία από τα οποία αποτελούνται.

2.1.1 Κωδικοποίηση RLNC

Υποθέτουμε ότι ένας αριθμός πακέτων M^1, M^2, \dots, M^n [10], παράγονται από μία ή περισσότερες πηγές (τα οποία ονομάζονται πηγαία πακέτα). Στην τυχαία γραμμική κωδικοποίηση δικτύου, κάθε πακέτο που διασχίζει το δίκτυο συσχετίζεται με μία ακολουθία συντελεστών $g = (g_1, g_2, \dots, g_n)$ οι οποίοι διαλέγονται τυχαία από το πεπερασμένο πεδίο Galois Field(2^s). Τα κωδικοποιημένα πακέτα υπολογίζονται ως το άθροισμα των n πηγαίων πακέτων $M^i (i=1, 2, \dots, n)$ πολλαπλασιασμένα με τους συντελεστές g_i . Κάθε κωδικοποιημένο πακέτο είναι της μορφής :

$$X = \sum_{i=1}^n g_i M^i$$

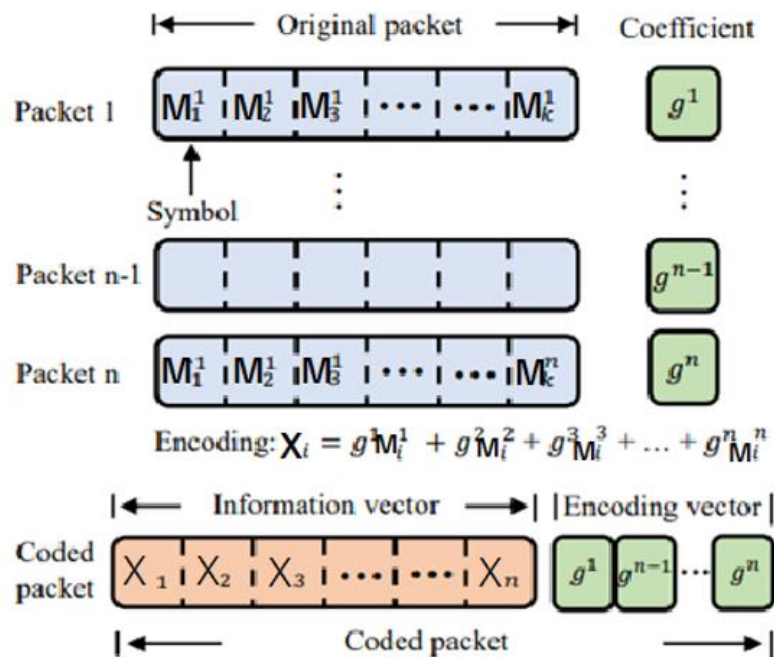
Όπου οι συντελεστές $g=(g_1, g_2, \dots, g_n)$ ονομάζονται διάνυσμα κωδικοποίησης (encoding vector) και το X ονομάζεται διάνυσμα πληροφοριών (information vector). Η κωδικοποίηση μπορεί να πραγματοποιηθεί αναδρομικά, σε ήδη κωδικοποιημένα πακέτα. Αν υποθέσουμε έναν κόμβο που έχει λάβει και αποθηκεύσει ένα σύνολο $(g_1, X^1), (g_2, X^2), \dots, (g_m, X^m)$ κωδικοποιημένων πακέτων. Αυτός ο κόμβος μπορεί να δημιουργήσει ένα νέο κωδικοποιημένο πακέτο (g', X') επιλέγοντας ένα τυχαίο σύνολο συντελεστών $h=(h_1, h_2, \dots, h_m)$ και υπολογίζοντας τον γραμμικό συνδυασμό:

$$X' = \sum_{j=1}^m h_j X^j$$

Το αντίστοιχο διάνυσμα κωδικοποίησης g' δίνεται από την σχέση:

$$g_i' = \sum_{j=1}^m h_j g_i^j$$

Στην εικόνα 2.1.1 απεικονίζεται αναλυτικά η διαδικασία της κωδικοποίησης καθώς και τα περιεχόμενα του κωδικοποιημένου πακέτου.



Εικόνα 2.1.1 Μια εικονογράφηση της διαδικασίας κωδικοποίησης στον κόμβο αποστολέα. Η πηγή αυτής της εικόνας είναι [7].

2.1.2 Αποκωδικοποίηση RLNC

Η αποκωδικοποίηση δικτύου με RLNC [10], είναι η διαδικασία ανάκτησης των πηγαίων πακέτων, από τα κωδικοποιημένα που έχουν μεταδοθεί.

Προϋπόθεση για την αποκωδικοποίηση είναι έστω ένας κόμβος να λάβει τα πακέτα $(g_1, X^1), (g_2, X^2), \dots, (g_m, X^m)$. Προκειμένου να ανακτήσει τα αρχικά πακέτα, πρέπει να λύσει το σύστημα:

$$X^j = \sum_{i=1}^n g_i^j M^j$$

Σε αυτό το σύστημα οι άγνωστοι είναι τα M^i (τα πηγαία πακέτα). Σε αυτό το γραμμικό σύστημα με m εξισώσεις και n αγνώστους. Χρειάζεται να ισχύει $m \geq n$, για να υπάρχει πιθανότητα ανάκτησης όλων των δεδομένων. Απαραίτητη προϋπόθεση για την αποκωδικοποίηση είναι ο αριθμός των ληφθέντων κωδικοποιημένων πακέτων να είναι τουλάχιστον τόσο μεγάλος όσο ο αριθμός των αρχικών πακέτων (όπως Εικόνα 2.1.2). Στην RLNC η πιθανότητα αποκωδικοποίησης εξαρτάται από το s (Galois Field(2^s)). Σε ένα μεγάλο πεπερασμένο πεδίο, η πιθανότητα παραγωγής γραμμικώς ανεξάρτητων πακέτων είναι αμελητέα[10].



Εικόνα 2.1.2 : Η αποκωδικοποίηση των κωδικοποιημένων πακέτων σε πηγαία πακέτα. Η πηγή της εικόνας είναι το [15].

Για να πραγματοποιηθεί η αποκωδικοποίηση, πρέπει να αποθηκευτούν οι ήδη γνωστές εξισώσεις σε μια δομή που ονομάζεται πίνακας αποκωδικοποίησης (decoding matrix). Ο πίνακας αποκωδικοποίησης περιέχει τα πηγαία πακέτα, με τα αντίστοιχα διανύσματα κωδικοποίησης και τα ληφθέντα κωδικοποιημένα πακέτα. Συγκεκριμένα οι γραμμές του, αντιστοιχούν στα ληφθέντα πακέτα, κωδικοποιημένα ή πηγαία. Οι στήλες του πίνακα αντανακλούν τα πηγαία πακέτα που εμπλέκονται στα ληφθέντα πακέτα και κρατούν τους αντίστοιχους συντελεστές που κάθε ληφθέν πακέτο περιέχει στο διάνυσμα κωδικοποίησής του. Η αποκωδικοποίηση πραγματοποιείται

χρησιμοποιώντας την μέθοδο απαλοιφής Gauss-Jordan όταν ο πίνακας φτάσει σε πλήρη βαθμό. Είναι σημαντικό να σημειώσουμε ότι η απόκωδικοποίηση πρέπει να πραγματοποιείται μόνο στον παραλήπτη.

Ένα κωδικοποιημένο πακέτο είναι χρήσιμο στην αποκωδικοποίηση όταν είναι καινοτόμο. Καινοτόμο ονομάζεται ένα πακέτο το οποίο αυξάνει τον βαθμό του πίνακα αποκωδικοποίησης δηλαδή είναι γραμμικώς ανεξάρτητο. Αν το υποψήφιο πακέτο δεν είναι καινοτόμο, μετατρέπεται σε μια σειρά μηδενικών και αγνοείται.

2.2 Κατηγοριοποίηση πρωτόκολλων RLNC

2.2.1 Block-based RLNC vs sliding-window RLNC

Η απόδοση του RLNC επηρεάζεται σημαντικά από την πολυπλοκότητα των λειτουργιών κωδικοποίησης. Καθώς η πολυπλοκότητα αυξάνεται, αυξάνονται και οι καθυστερήσεις, επομένως ένα πεπερασμένο σύνολο πακέτων πρέπει να συμμετέχει στη διαδικασία κωδικοποίησης. Εκτός από το πλήθος των πακέτων, σημαντικός παραγοντας για την αποτελεσματικότητα του πρωτόκολλου, είναι ο τρόπος με τον οποίο επιλέγονται οι ομάδες που θα εμπλακούν στην κωδικοποίηση, δηλαδή πόσα και ποια πηγαία πακέτα θα χρησιμοποιηθούν για τη δημιουργία ενός κωδικοποιημένου πακέτου. Με βάση αυτόν το παράγοντα, τα σχήματα RLNC μπορούν να κατηγοριοποιηθούν σε block-based RLNC και sliding-window RLNC.

Τα sliding window πρωτόκολλα ([2],[3],[7]), χρησιμοποιούν ένα δυναμικό πλήθος πηγαίων πακέτων για την κωδικοποίηση (το οποίο έχει ένα άνω όριο ίσο με το μέγεθος του παραθύρου (W)). Το παράθυρο καθορίζει την ακολουθία των πηγαίων πακέτων που βρίσκονται επί του παρόντος υπό μετάδοση. Το μέγεθος του παραθύρου ολίσθησης εξαρτάται από το γινόμενο εύρους ζώνης-καθυστέρησης του καναλιού, το οποίο περιγράφει το μέγιστο όγκο δεδομένων που μπορεί να βρίσκεται υπό μετάδοση στο κανάλι ταυτόχρονα. Σε αυτά τα σχήματα υπάρχει επικάλυψη, δηλαδή ένα πακέτο μπορεί να χρησιμοποιείται σε περισσότερα υποσύνολα κατά την κωδικοποίηση.

Στην block-based RLNC ([3],[16],[17]) η οποία είναι γνωστή και ως generation-based, ένα generation αναφέρεται σε μια ομάδα ή ένα block πηγαίων πακέτων που

κωδικοποιούνται μαζί. Κάθε γενιά αποτελείται συνήθως από έναν σταθερό αριθμό πηγαίων πακέτων, ο οποίος καθορίζεται από την παράμετρο generation size (G). Τα κωδικοποιημένα πακέτα δημιουργούνται από τυχαίους γραμμικούς συνδυασμούς των G πηγαίων πακέτων. Αυτή η προσέγγιση επιτυγχάνει χαμηλή υπολογιστική πολυπλοκότητα σε σύγκριση με την επεξεργασία ολόκληρου του μηνύματος και ευκολότερη υλοποίηση.

2.2.2 Full Vector (Non-Systematic) vs Systematic RLNC

Ένας ακόμη σημαντικός παράγοντας για την αποτελεσματικότητα του πρωτοκόλλου RLNC είναι η μορφή των πακέτων που θα σταλούν κατά τη μετάδοση. Με βάση αυτό το κριτήριο, μπορούμε να διακρίνουμε τη systematic και τη full-vector (ή non-systematic) προσέγγιση. Στην systematic προσέγγιση, τα πηγαία πακέτα μεταδίδονται αυτούσια, ενώ στην full-vector προσέγγιση κάθε πηγαίο πακέτο έχει κωδικοποιημένη μορφή.

Στην full vector κωδικοποίηση, γνωστή και ως non-systematic ([4],[8]), κάθε πακέτο που μεταδίδεται δημιουργείται μέσω τυχαίων γραμμικών συνδυασμών των πηγαίων πακέτων. Αυτό σημαίνει ότι ο κωδικοποιητής πρέπει να συγκεντρώσει όλα τα πηγαία πακέτα πριν ξεκινήσει την κωδικοποίηση. Κατά την αποκωδικοποίηση, είτε όλα τα πακέτα αποκωδικοποιούνται είτε κανένα. Επιπλέον, όλα τα πηγαία πακέτα έχουν την ίδια καθυστέρηση καθώς αποκωδικοποιούνται ταυτόχρονα, όταν ληφθεί το τελευταίο πακέτο μετάδοσης. Αυτή η μέθοδος επιβάλλει μεγάλες καθυστερήσεις αναμονής λόγω της ανάγκης πλήρους λήψης της γενιάς πριν την έναρξη της αποκωδικοποίησης. Αυτό επηρεάζει την απόδοση του συστήματος, καθιστώντας το λιγότερο αποτελεσματικό όσον αφορά την αντιμετώπιση καθυστερήσεων.

Από την άλλη πλευρά στη systematic RLNC ([5],[4],[8]), τα πηγαία πακέτα μεταδίδονται αυτούσια, ακολουθούμενα από πλεονάζοντα κωδικοποιημένα πακέτα. Τα κωδικοποιημένα πακέτα, δημιουργούνται από τυχαίους γραμμικούς συνδυασμούς των πηγαίων πακέτων και αποστέλλονται για την εξασφάλιση προστασίας σε περίπτωση που κάποιο από τα πηγαία χαθεί. Στο systematic RLNC, ο παραλήπτης μπορεί να μεταβιβάσει αμέσως το ληφθέν πηγαίο πακέτο στο ανώτερο επίπεδο, εφόσον έχουν ληφθεί και παραδοθεί σωστά όλα τα προηγούμενα πακέτα πηγής. Σε περίπτωση που

κάποιο πηγαίο πακέτο χαθεί, πρέπει να περιμένει τα κωδικοποιημένα πακέτα για να ανακτήσει τα δεδομένα.

2.2.3 Ρυθμός κωδικοποίησης και διαχείριση πλεονάζουσας πληροφορίας σε block-based συστήματα

Συνήθως, τα πλεονάζοντα πακέτα σε ένα block-based σχήμα μεταδίδονται στο τέλος του block, δηλαδή μετά την αποστολή των πηγαίων πακέτων. Ωστόσο, αυτό μπορεί να αλλάξει ανάλογα με τον ορισμό του ρυθμού κωδικοποίησης (R), που καθορίζει πόσα κωδικοποιημένα πακέτα μπορούν να μεταδοθούν μεταξύ των πηγαίων του block. Η τιμή του R επηρεάζει τον αριθμό των κωδικοποιημένων πακέτων που μεταδίδονται τελικά, καθώς αυτό εξαρτάται από το κλάσμα που ορίζεται ως ρυθμός και δίνεται από την σχέση:

$$R = \frac{k}{k+n} , R \leq 1$$

Όπου k είναι τα πηγαία πακέτα. Επομένως κάθε k πηγαία πακέτα μεταδίδονται n κωδικοποιημένα. Συνοψίζοντας, στις περιπτώσεις χωρίς ή με πολύ σπάνια απώλεια πακέτων στο κανάλι, η systematic κωδικοποίηση RLNC επιτυγχάνει χαμηλούς χρόνους καθυστέρησης. Ωστόσο, σε περιπτώσεις σημαντικών απωλειών πακέτων, η καθυστέρηση της systematic RLNC κωδικοποίησης συγκλίνει στην καθυστέρηση της full vector RLNC.

Στην περίπτωση που δεν έχει καθοριστεί κανένας ρυθμός, το πρωτόκολλο λειτουργεί ως rateless, δηλαδή μπορεί να στέλνει απεριόριστα κωδικοποιημένα πακέτα μέχρις ότου να είναι δυνατή η αποκωδικοποίησή τους.

Στην παρούσα διπλωματική θα ασχοληθούμε με systematic block-based σχήματα.

Κεφάλαιο 3. Υλοποίηση

της

συστηματικής

Κωδικοποίησης

ης Δικτύου

Σε αυτό το κεφάλαιο, παρουσιάζουμε αρχικά τη βιβλιοθήκη PyErasure [12]. Στη συνέχεια, αναλύουμε τα διάφορα σχήματα κωδικοποίησης με τα οποία ασχοληθήκαμε. Αυτά περιλαμβάνουν 1) την Rateless block-based RLNC, η οποία αποτελεί την πρώτη έκδοση του κώδικα της PyErasure, μια systematic block-based προσέγγιση με τη χρήση του RLNC. Σε αυτήν την προσέγγιση, η μετάδοση των δεδομένων γίνεται σε ένα μεγάλο block και αποστέλλονται αδιάκοπα κωδικοποιημένα πακέτα μέχρι την αποκωδικοποίηση όλων των χαμένων, 2) η G-rated block-based RLNC, η οποία αποτελεί την έκδοση με την πρώτη τροποποίηση και διαχωρίζει τα δεδομένα σε πολλά μικρότερα block, περιορίζοντας τα κωδικοποιημένα πακέτα στο τέλος κάθε γενιάς 3) η k-rated block-based RLNC, η οποία αποτελεί τη δεύτερη τροποποίηση του αρχικού κώδικα και μεταδίδει τα κωδικοποιημένα πακέτα ενδιάμεσα από τα πηγαία.

3.1 Εισαγωγή στη βιβλιοθήκη της PyErasure

Η PyErasure[12] είναι μια βιβλιοθήκη προγραμματισμού που παρέχει λειτουργίες σχετικές με την κωδικοποίηση και αποκωδικοποίηση δεδομένων με χρήση της τεχνικής erasure coding, μέσω της γλώσσας προγραμματισμού Python. Ένα από τα βασικά χαρακτηριστικά της είναι η απλότητα χρήσης, προκειμένου να είναι εύκολο να χρησιμοποιηθεί για εκπαιδευτικούς και ερευνητικούς σκοπούς. Η κωδικοποίηση δικτύου ως ευθείας διόρθωσης σφαλμάτων, είναι μια τεχνική που χρησιμοποιείται για την προστασία δεδομένων από την απώλεια πακέτων .

Στο πλαίσιο της παρούσας διπλωματικής εργασίας, χρησιμοποιήσαμε ένα από τα παραδείγματα της βιβλιοθήκης PyErasure που παρουσιάζει τη διαδικασία κωδικοποίησης και αποκωδικοποίησης, χρησιμοποιώντας έναν κώδικα Τυχαίας Γραμμικής Κωδικοποίησης Δικτύου (Random Linear Network Coding – RLNC. Ο στόχος είναι να εξασφαλίσουμε την αξιόπιστη μετάδοση των αρχικών πακέτων μέσω ενός καναλιού που μπορεί να προκαλεί απώλειες δεδομένων. Αυτό επιτυγχάνεται με την αποστολή πλεονάζουσας πληροφορίας, με την χρήση της κωδικοποίησης RLNC. Στην ενότητα 3.3 περιγράφεται αναλυτικά το παράδειγμα που χρησιμοποιήσαμε.

3.2 Μηχανισμοί της PyErasure

Πίνακας 3.2 : Οι κυριότερες συμβολισμοί όλων των εκδόσεων

| G | Πλήθος πακέτων σε μία γενιά (generation size) |
|--------------|---|
| ploss | Η πιθανότητα απώλειας πακέτων |
| n | Πλήθος κωδικοποιημένων πακέτων |

3.2.1 Μετάδοση πακέτων και κωδικοποίηση

Ο αποστολέας δρα ως ένας παραδοσιακός αποστολέας βασισμένος σε block-based προσέγγιση. Κάθε πηγαίο πακέτο σε αυτό το πλαίσιο περιλαμβάνει έναν

μοναδικό αριθμό ακολουθίας i , ο οποίος βρίσκεται εντός του εύρους $[0, G]$, όπου G αντιπροσωπεύει το μέγεθος της γενιάς.

Κάθε block μεταδίδεται και κωδικοποιείται ξεχωριστά. Σε περιπτώσεις όπου το πλήθος των πακέτων δεν διαιρείται ακέραια με το μέγεθος γενιάς, η τελευταία ομάδα θα περιέχει λιγότερα πακέτα από τις υπόλοιπες, και το μέγεθος αυτής της ομάδας θα εξαρτάται από το υπόλοιπο της διαίρεσης των συνολικών πακέτων με το μέγεθος της γενιάς.

Μια ακόμη λειτουργία που παρέχει η προσέγγιση systematic block-based RLNC, είναι η μετάδοση των πηγαίων πακέτων, ακολουθούμενα από πλεονάζουσα πληροφορία για την προστασία τους από απώλειες. Αυτή η πλεονάζουσα πληροφορία, ή αλλιώς τα κωδικοποιημένα πακέτα, σχηματίζονται από τυχαίους γραμμικούς συνδυασμούς των πηγαίων πακέτων του block:

$$X = \sum_{i=0}^G g_i M^i$$

Όπου οι συντελεστές (διάνυσμα κωδικοποίησης) επιλέγονται τυχαία από ένα πεπερασμένο πεδίο $GF(2^s)$ όπως αναφέραμε στην 2.1.1 ενότητα. Τα κωδικοποιημένα πακέτα στέλνονται στο τέλος κάθε γενιάς.

3.2.2 Λήψη πακέτων και αποκωδικοποίηση

Στον παραλήπτη, εκ νέου, εφαρμόζεται η βασική λειτουργικότητα του systematic block-based RLNC. Δηλαδή, τα πηγαία πακέτα που λαμβάνονται σε σειρά παραδίδονται στο ανώτερο επίπεδο, ενώ τα πακέτα που δεν είναι σε σειρά αποθηκεύονται μέχρι να επαναφερθεί η σειρά. Όμοια με όλα τα πρωτόκολλα βασισμένα σε RLNC, ο παραλήπτης αποτελείται από έναν πίνακα αποκωδικοποίησης. Ο πίνακας αποκωδικοποίησης, όταν φτάσει σε πλήρη βαθμό, μπορεί να χρησιμοποιηθεί για την αποκωδικοποίηση των χαμένων πηγαίων πακέτων με βάση τα ληφθέντα πηγαία και κωδικοποιημένα πακέτα όπως αναφέραμε στην ενότητα 2.1.2. Η αποκωδικοποίηση μπορεί να γίνει με τη χρήση της μεθόδου Gauss-Jordan απαλοιφή.

3.3 Έκδοση 1: Rateless block-based RLNC

Η τεχνική Rateless Systematic Block-Based RLNC ειδική υποπερίπτωση της systematic block-based, όπου όλα τα αρχικά πηγαία πακέτα ανήκουν σε ένα block. Το κύριο χαρακτηριστικό της Rateless τεχνικής είναι η αποστολή ενός απεριόριστου αριθμού κωδικοποιημένων πακέτων, με σκοπό την ανάκτηση όλων των πηγαίων πακέτων που μπορεί να έχουν χαθεί κατά τη διάρκεια της μετάδοσης. Ο Πίνακας 3.2 περιλαμβάνει τις κύριες σημειώσεις που έχουν χρησιμοποιηθεί.

3.3.1 Μετάδοση πακέτων και κωδικοποίηση

Η κωδικοποίηση γίνεται με τον ίδιο τρόπο που αναφέραμε στην ενότητα 3.2, κάποιες διαφοροποιήσεις, μία από αυτές είναι ότι τα πηγαία πακέτα ανήκουν σε ένα block, που στην προσέγγιση μας αποτελεί το σύνολο των πακέτων. Μία άλλη διαφοροποίηση είναι ότι το πλήθος της πλεονάζουσας πληροφορίας δεν έχει κάποιο καθορισμένο ρυθμό μετάδοσης, καθώς τα κωδικοποιημένα πακέτα που στέλνονται είναι απεριόριστα, με σκοπό να εξασφαλίσουν την ανάκτηση όλων των χαμένων πακέτων στον παραλήπτη.

3.3.2 Λήψη πακέτων και αποκωδικοποίηση

Η αποκωδικοποίηση μπορεί να εκτελεστεί με τον τρόπο που περιγράψαμε στην ενότητα 3.3. Ωστόσο, η Rateless τεχνική, λόγω της έλλειψης καθορισμένου ρυθμού μετάδοσης, σε περιπτώσεις όπου υπάρχει μεγάλος αριθμός πακέτων με υψηλή πιθανότητα απώλειας, το σύστημα παρόλο που μπορεί να ανακτήσει όλα τα πακέτα, θα έχει μεγάλη χρονική καθυστέρηση και κατανάλωση χρήσιμου εύρους ζώνης. Αυτό μπορεί να επηρεάσει αρνητικά την απόδοση του συστήματος.

3.4 Μηχανισμοί που υλοποιήσαμε

3.4.1 Ενημέρωση της λίστας αναμονής

Η λίστα με τα εκκρεμή πακέτα (`pending_symbol`) περιέχει τα πακέτα που φτάνουν στον παραλήπτη εκτός σειράς, δηλαδή αμέσως μετά από ένα χαμένο μη κωδικοποιημένο πακέτο στην ίδια γενιά. Όταν ένα πηγαίο πακέτο χάνεται κατά τη μετάδοση, ο αποστολέας συνεχίζει να στέλνει τα επόμενα πακέτα. Αυτά τα επόμενα πακέτα που αποστέλλονται προστίθενται στη `pending_symbol`. Συγκεκριμένα, στη λίστα αποθηκεύονται μόνο πηγαία μη κωδικοποιημένα πακέτα. Η λίστα αυτή χρησιμοποιείται για να παρακολουθηθούν τα πακέτα που πρέπει να παραδοθούν και βρίσκονται σε εκκρεμότητα λόγω της αναμονής για την ανάκτηση των χαμένων πακέτων. Αφού γίνει η αποκατάσταση των χαμένων πακέτων, η λίστα αδειάζει. Σε περίπτωση που δεν γίνει αποκατάσταση των χαμένων πακέτων, η `pending symbols` δεν αδειάζει μέχρι το τέλος της γενιάς. Σε κάθε έναρξη νέας γενιάς, γίνεται εκκαθάριση των εκκρεμών πακέτων της προηγούμενης .

3.4.1 Μηχανισμοί διαχείρισης πακέτων

Σε περιβάλλοντα με ασύρματης επικοινωνία ή σε κανάλια με θόρυβο, συχνά παρατηρούνται απώλειες πακέτων. Για να διαχειριστούμε αυτές τις απώλειες, έχουμε υλοποιήσει δύο μηχανισμούς διαχείρισης πακέτων: α) `drop out of order` και β) την `drop generation`. Στην G-rated τεχνική, που θα δούμε στην συνέχεια, έχουμε υλοποιήσει μόνο τον μηχανισμό `drop out of order`, ενώ στην k-rated και τους δύο μηχανισμούς

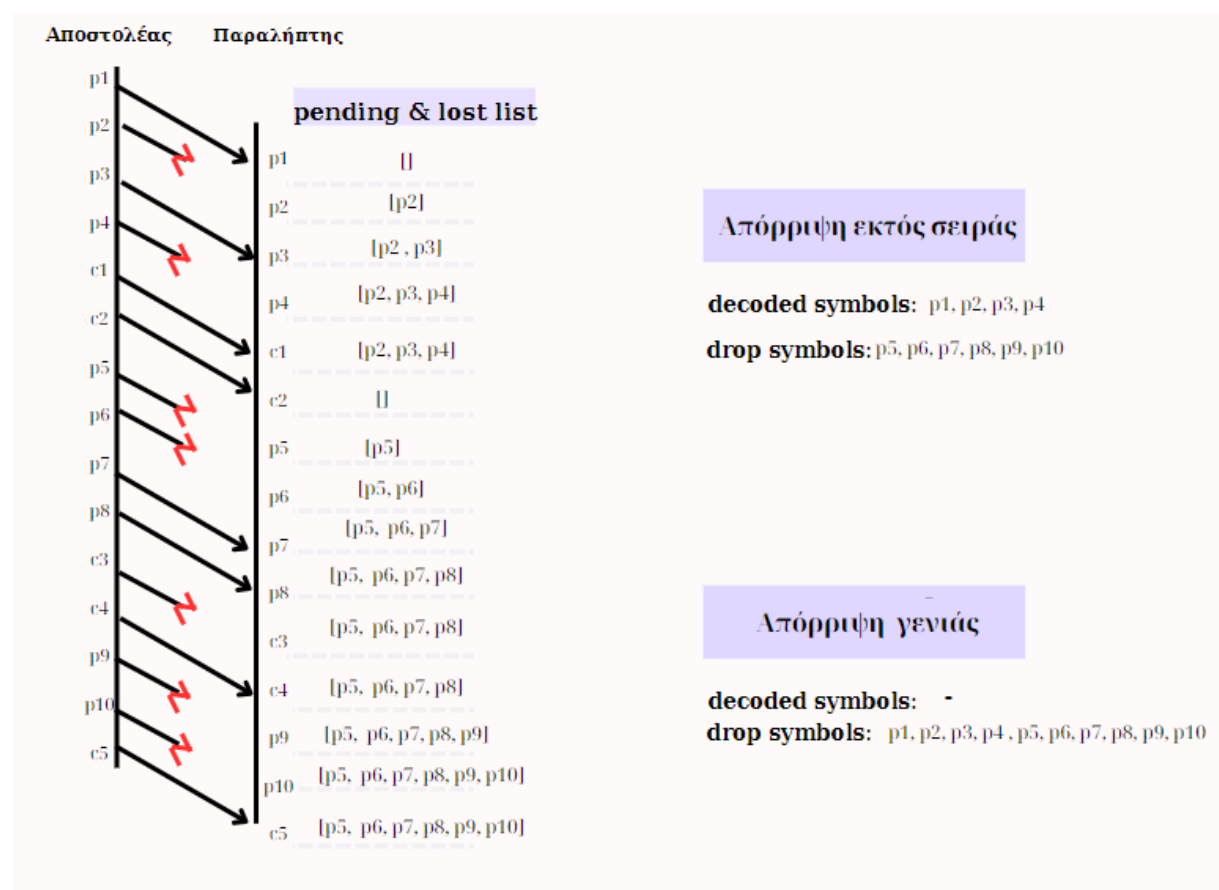
3.4.1.1 Drop out of order(Απόρριψη πακέτων εκτός σειράς)

Ο συγκεκριμένος μηχανισμός διαχείρισης πακέτων επικεντρώνεται στην απόρριψη των πακέτων ένα προς ένα. Κατά τη διαδικασία αυτή, όταν ένα πηγαίο πακέτο χάνεται, όλα τα επόμενα πακέτα της ίδιας γενιάς προστίθενται σε μια λίστα αναμονής, ανεξαρτήτως του εάν ληφθούν σωστά ή όχι. Εάν το χαμένο πακέτο

ανακτηθεί μέσω των κωδικοποιημένων πακέτων, τότε τα πακέτα που είχαν τεθεί σε αναμονή απελευθερώνονται. Αντίθετα, εάν δεν επιτευχθεί η ανάκτηση, τότε απορρίπτονται όλα τα επόμενα εν σειρά πηγαία πακέτα από το χαμένο πακέτο μέχρι το τελευταίο πακέτο αυτής της γενιάς.

3.4.1.2 Drop generation (Απόρριψη ολόκληρης της γενιάς)

Σε αντίθεση με τον μηχανισμό drop out of order, ο εν λόγω μηχανισμός αναφέρεται στη μαζική απόρριψη ομάδων πακέτων που ανήκουν σε μία συγκεκριμένη γενιά. Σε αυτήν την περίπτωση, η μόνη διαφοροποίηση της μεθόδου από την drop out of order είναι ότι προβαίνουμε στην απόρριψη ολόκληρης της γενιάς που περιλαμβάνει το χαμένο πηγαίο πακέτο, και όχι μόνο των πακέτων που βρίσκονται στην αναμονή μετά το εν λόγω πακέτο.



Εικόνα 3.4.4 Απόρριψη εκτός σειράς vs Απόρριψη γενιάς

Στην εικόνα 3.4.4 παρουσιάζεται ένα στιγμιότυπο μίας γενιάς με $G = 10$ και ρυθμό $R = 4/6$ (δηλαδή κάθε 4 πηγαία πακέτα ακολουθούνται από 2 κωδικοποιημένα), όπου τα πηγαία πακέτα συμβολίζονται με το πακέτο p_i και τα κωδικοποιημένα με το πακέτο c_i . Εκτελείται σύγκριση μεταξύ της drop out of order σειράς και της drop generation, αναδεικνύοντας τα πακέτα που μπορούν να ανακτηθούν και αυτά που χάνονται, ανάλογα με τον μηχανισμό διαχείρισης πακέτων που χρησιμοποιείται. Επιπλέον, παρουσιάζει λεπτομερώς τον τρόπο διαχείρισης των λιστών με τα χαμένα και τα εκρεμμή πακέτα, καθώς και τον τρόπο εκκαθάρισης αυτών κατά τη διάρκεια της αποκωδικοποίησης, η μόνη αποκωδικοποίηση στο συγκεκριμένο παράδειγμα γίνεται με την άφιξη του c_2 , τα πηγαία πακέτα από p_5 μέχρι p_{10} απορρίπτονται καθώς δε λαμβάνουμε τόσα κωδικοποιημένα πακέτα όσα το πλήθος των πηγαίων που χάθηκαν.

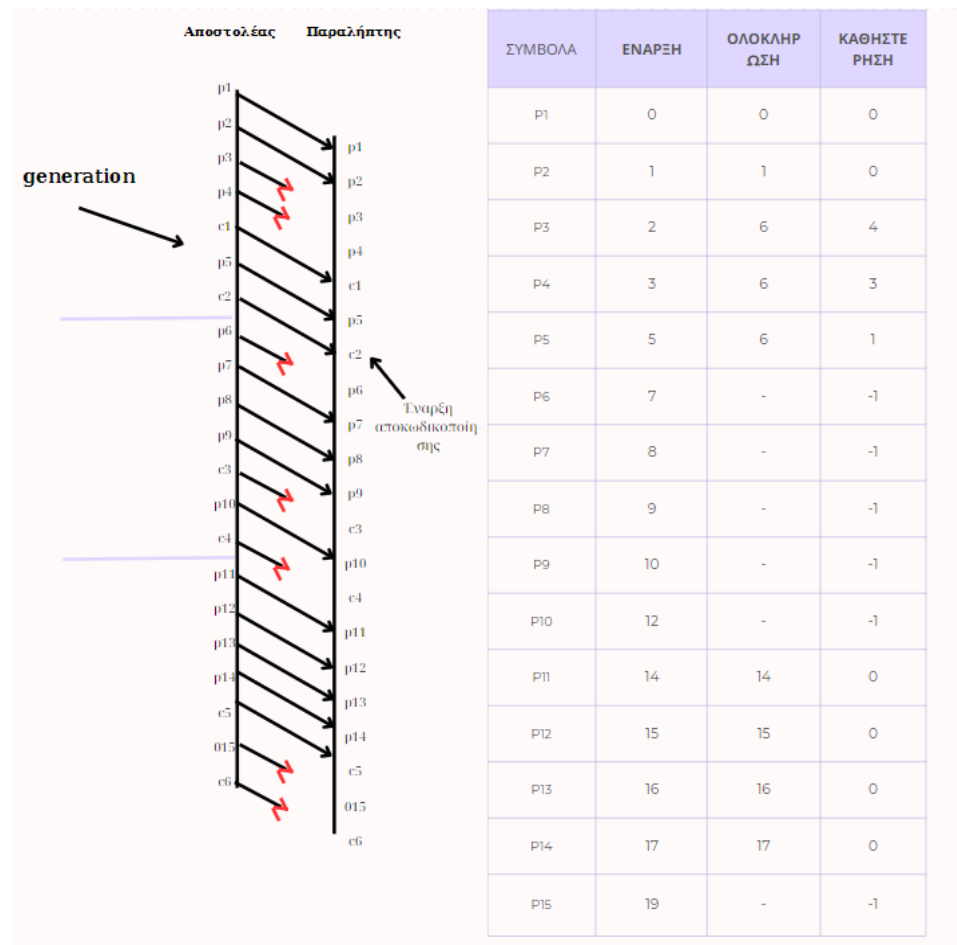
3.4.1 Ενημέρωση χρόνου κωδικοποίησης/αποκωδικοποίησης

Η παράδοση των πακέτων είναι σειριακή, επομένως και ο χρόνος καθυστέρησης αφορά την εν σειρά παράδοση τους. Ορίζουμε την εν σειρά καθυστέρηση πακέτου ως το χρονικό διάστημα από τη στιγμή που ένα πακέτο πηγής είναι έτοιμο για μετάδοση στον αποστολέα έως τη στιγμή που παραδίδεται με επιτυχία σειριακά στον παραλήπτη. Υπολογίζουμε τον μέσο όρο καθυστέρησης προσθέτοντας όλες τις καθυστερήσεις πακέτων σε σειρά και στη συνέχεια διαιρούμε με τον συνολικό αριθμό πακέτων που παραδίδονται επιτυχώς. Ο μέσος όρος καθυστέρησης λαμβάνει υπόψη μόνο τα πακέτα πηγής που παραδίδονται επιτυχώς, όχι τα ληφθέντα κωδικοποιημένα που συνδράμουν στη διαδικασία αποκωδικοποίησης.

Χρησιμοποιούμε ένα σύστημα επικοινωνίας με χρονοθυρίδες και φυσικό επίπεδο που μεταδίδει ένα κωδικοποιημένο πακέτο ή ένα πηγαίο πακέτο σε μια χρονοθυρίδα από την έξοδο του κωδικοποιητή στην είσοδο του αποκωδικοποιητή. Οι χρόνοι υπολογισμού της κωδικοποίησης και αποκωδικοποίησης θεωρούνται αμελητέοι σε σύγκριση με το χρονικό διάστημα μετάδοσης του πακέτου.

Για κάθε πακέτο, καταγράφουμε τον χρόνο έναρξης και ολοκλήρωσης της μετάδοσής του. Αυτός ο χρόνος ξεκινά όταν το πακέτο αποστέλλεται και τελειώνει όταν είτε ληφθεί επιτυχώς είτε απορριφθεί λόγω απώλειας. Πιο συγκεκριμένα, κατά την

αποστολή ενός πακέτου, η χρονική στιγμή έναρξής του καταγράφεται με το αντίστοιχο αναγνωριστικό του. Κατά την λήψη του πακέτου, ο χρόνος ολοκλήρωσής του καταγράφεται επίσης. Σε περίπτωση απόρριψης ενός πακέτου λόγω απώλειας, η αντίστοιχη εγγραφή του μπορεί να συμπληρωθεί με έναν ειδικό δείκτη, όπως έναν αρνητικό αριθμό ή ένα ειδικό πακέτο, όπως -1 ή None, για να υποδείξει ότι το πακέτο απορρίφθηκε λόγω απώλειας, ώστε κατά την διάρκεια του υπολογισμού της μέσης εν σειρά καθυστέρησης να μη ληφθεί υπόψιν καθώς δεν συνεισφέρει.



Εικόνα 3.4.5 Παράδειγμα του τρόπου ενημέρωσης των χρόνων

Στην εικόνα 3.4.5 παρουσιάζεται ένα παράδειγμα για το πως διαμορφώνεται η ενημέρωση χρόνου. Συγκεκριμένα απεικονίζει 15 πακέτα, όπου χωρίζονται σε 3 ομάδες των 5 πακέτων με ρυθμό κωδικοποίησης $R=4/5$, τα p_i αναπαριστούν τα πηγαία πακέτα και τα c_i τα κωδικοποιημένα. Στο συγκεκριμένο παράδειγμα με την drop out of order των πακέτων γίνεται αποκωδικοποίηση στο πρώτο group, το δεύτερο group γίνεται drop και στο τρίτο γίνεται αποκωδικοποίηση μέχρι το p_{14} και το p_{15} γίνεται drop. Με

την drop generation στο συγκεκριμένο παράδειγμα μόνο τα στοιχεία του πρώτου group θα γίνουν decode τα άλλα δύο group κάνουν drop όλα τα στοιχεία τους.

3.4.1 Ενημέρωση πίνακα αποκωδικοποίησης

Όταν ο παραλήπτης απορρίπτει ένα πακέτο πηγής, αυτό σημαίνει ότι το πακέτο δεν μπορεί να χρησιμοποιηθεί για την ανάκτηση των δεδομένων. Ως εκ τούτου, όλα τα κωδικοποιημένα πακέτα που περιλαμβάνουν αυτό το απορριπτόμενο πακέτο πρέπει να αφαιρεθούν από τον πίνακα αποκωδικοποίησης. Αυτό εξασφαλίζει ότι ο παραλήπτης λαμβάνει μόνο τα πακέτα που μπορούν να χρησιμοποιηθούν για την αποκωδικοποίηση.

3.5 Έκδοση 2: G-rated block-based RLNC

Η G-rated block based RLNC αποτελεί παραλλαγή της Rateless block-based RLNC. Μια από τις πρώτες αλλαγές που τροποποιήσαμε στον αρχικό κώδικα ήταν η υποστήριξη πολλών μικρών block πακέτων σε σχέση με την αρχική έκδοση Rateless block-based RLNC. Συγκεκριμένα, αντί να χειριζόμαστε ένα συνεχές ρεύμα πακέτων, τονίσαμε τη σημασία του χωρισμού σε γενιές ή blocks, τα οποία αποτελούν μικρότερες ομάδες πακέτων με το ίδιο μέγεθος και περιορίσαμε τον αριθμό των κωδικοποιημένων πακέτων στο τέλος κάθε γενιάς. Επιπλέον, περιορίσαμε τα κωδικοποιημένα πακέτα που στέλνονται στο τέλος κάθε γενιάς. Εφόσον τα κωδικοποιημένα πακέτα είναι περιορισμένα σε κάθε block, υπάρχουν πακέτα τα οποία ενδέχεται να χαθούν και δε θα μπορέσουμε να τα ανακτήσουμε. Τα πακέτα τα οποία χάθηκαν και δε μπορέσαμε να τα ανακτήσουμε τα απορρίπτουμε με ένα μηχανισμό διαχείρισης πακέτων. Με την αλλαγή αυτή, επεκτείναμε την λειτουργικότητα της Rateless block-based RLNC, μειώνοντας τις καθυστερήσεις και την πολυπλοκότητα κωδικοποίησης/αποκωδικοποίησης, της συνεχούς μετάδοσης δεδομένων που είχε η αρχική έκδοση. Στον Πίνακα 3.2 και 3.5 περιέχονται οι κύριες σημειώσεις που χρησιμοποιήθηκαν.

Πίνακας 3.5 : Οι επιπλέον ευμβολισμοί της G-rated block based RLNC

| R | Ρυθμός κωδικοποίησης, $R = \frac{G}{G+n}$ |
|----------|---|

3.6 Έκδοση 3: k-rated block-based RLNC

Η k-rated block based RLNC αποτελεί μία παραλλαγή της G-rated block-based RLNC, με τη βασική διαφοροποίηση να αφορά τον ρυθμό με τον οποίο στέλνουμε τα κωδικοποιημένα πακέτα. κωδικοποίησης/αποκωδικοποίησης. Η k-rated block based RLNC είναι ελπιδοφόρα για την αντιμετώπιση του προβλήματος αναμονής των κωδικοποιημένων πακέτων στο τέλος κάθε ομάδας που αντιμετωπίζει η G-rated block-based RLNC. Για τον λόγο αυτό, προσπαθούμε να εισαγάγουμε ένα ρυθμό κωδικοποίησης ο οποίος θα χωρίζει νοητά κάθε γενιά σε μικρότερες υπογενιές μεγέθους k πηγαίων πακέτων ακολουθούμενα από n κωδικοποιημένα πακέτα, ώστε να πραγματοποιεί σε συντομότερο χρονικό διάστημα αποκωδικοποίηση. Στον Πίνακα 3.3 περιέχονται οι κύριες σημειώσεις που χρησιμοποιήθηκαν.

Πίνακας 3.6 : Οι επιπλέον συμβολισμοί της προσέγγισης μας με k-rated.

| k | Πλήθος μη κωδικοποιημένων πακέτων που στέλνονται πριν τα κωδικοποιημένα |
|-------------|---|
| R | Ο ρυθμός κωδικοποίησης, $R = \frac{k}{k+n}$ |
| c | Αριθμός κωδικοποιημένων πακέτων που έχουν σταλεί, $c = \lfloor \frac{G}{k} \rfloor \cdot n$ |
| extra_coded | Κωδικοποιημένα πακέτα για ημιτελή Group, $extra_coded = \lfloor \frac{G}{R} - G - c \rfloor$ |

3.6.1 Μετάδοση πακέτων και κωδικοποίηση

Ο αποστολέας έχει την δυνατότητα να στείλει σε κάθε block, ένα πλήθος πηγαίων (μη κωδικοποιημένων) πακέτων και ένα πλήθος κωδικοποιημένων πακέτων. Όπως και στα προηγούμενα σχήματα, Rateless block-based RLNC και G-rated block-based RLNC.

Ο νέος ρυθμός κωδικοποίησης (R), που υλοποιήσαμε, στέλνει ομοιόμορφα κωδικοποιημένα πακέτα κατά την διάρκεια του block και όχι μόνο στο τέλος του. Για παράδειγμα, αν έχουμε k μη κωδικοποιημένα πακέτα και n κωδικοποιημένα πακέτα, ο ρυθμός κωδικοποίησης R (όπου $R \leq 1$) είναι ο λόγος των αρχικών πακέτων προς το άθροισμα των αρχικών πακέτων με τα κωδικοποιημένα πακέτα και ορίζεται ως:

$$R = \frac{k}{k + n}$$

Κάθε κωδικοποιημένο πακέτο μπορεί να χρησιμοποιηθεί για αποκωδικοποίηση μόνο των πηγαίων πακέτων πριν από αυτό. Η διαδικασία της κωδικοποίησης γίνεται με τον τρόπο που αναφέραμε στην 3.2.1 ενότητα.

Το πλήθος των κωδικοποιημένων πακέτων καθορίζεται από τον ρυθμό R που αναφέραμε παραπάνω, δηλαδή για κάθε k μη κωδικοποιημένα πηγαία πακέτα, μεταδίδονται n κωδικοποιημένα. Ωστόσο στην περίπτωση που βρισκόμαστε στο τελευταίο block και το πλήθος των μη κωδικοποιημένων πακέτων δε συμπληρώνουν το πλήθος k πρέπει να στείλουμε κάποια κωδικοποιημένα πακέτα για να προστατέψουν τα τελευταία πηγαία. Τα κωδικοποιημένα αυτά πακέτα προκύπτουν από τον τύπο:

$$extra_coded = \lceil \frac{G}{R} - G - c \rceil$$

όπου το c δίνεται από το γινόμενο των πλήρη blocks k πακέτων που υπάρχουν σε ένα generation size, επί των n κωδικοποιημένων για κάθε group k πακέτων, δηλαδή από τον τύπο :

$$c = \lfloor \frac{G}{k} \rfloor \cdot n$$

Το c αφορά το πλήθος των κωδικοποιημένων πακέτων που έχουν μεταδοθεί μέχρι στιγμής με βάση τον ορισμό του σχήματος κωδικοποίησης που υποθέτουμε.

3.6.1 Λήψη πακέτων και αποκωδικοποίηση

Η αποκωδικοποίηση πραγματοποιείται με τον ίδιο τρόπο που περιγράψαμε στην ενότητα 3.2.2, με μοναδική διαφορά ότι τα κωδικοποιημένα πακέτα μπορούν να ανακτήσουν στην διαδικασία της αποκωδικοποίησης, μόνο τα πηγαία πακέτα που προηγούνται αυτών στην ίδια γενιά.

3.6.2 Ενημέρωση της λίστας αναμονής

Η λίστα εκκρεμών πακέτων (`pending_symbol`) ενημερώνεται με τον ίδιο τρόπο που αναφέραμε στην ενότητα 3.4.1. Η μόνη διαφορά από την αντίστοιχη λίστα αναμονής στην G-rated block-based RLNC είναι ότι στην περίπτωση αυτή, η λίστα μπορεί να αδειάσει τα εκκρεμή πακέτα πριν το τέλος του block.

Κεφάλαιο 4. Πειραματική Αξιολόγηση

Στο παρόν κεφάλαιο, θα διεξαγάγουμε μια εκτενή αξιολόγηση της προσέγγισης G-rated block-based RLNC και της k-rated block-based RLNC. Στη συνέχεια, θα προβούμε σε σύγκριση των δύο μεθόδων. Επιπλέον, θα εξετάσουμε ποιος τρόπος διαχείρισης πακέτων υπερτερεί για κάθε πιθανότητα απώλειας δεδομένων.

4.1 Μετρικές Αξιολόγησης

Οι μετρικές που χρησιμοποιήθηκαν για την αξιολόγηση της υλοποίησης περιλαμβάνουν το μέσο μέγεθος του πίνακα αποκωδικοποίησης, τη μέση καθυστέρηση σε σειρά και το Goodput.

4.1.1 Average Decoding Matrix Size (Μέσο μέγεθος του πίνακα αποκωδικοποίησης)

Το μέγεθος του πίνακα αποκωδικοποίησης αποτυπώνει τον βαθμό του πίνακα κατά την έναρξη της αποκωδικοποίησης, δηλαδή τον αριθμό των γραμμών που είναι γραμμικά ανεξάρτητες στον πίνακα αποκωδικοποίησης.

Ορίζουμε ως μέσο μέγεθος του πίνακα αποκωδικοποίησης, τον λόγο του συνόλου των γραμμών που είναι γραμμικά ανεξάρτητες κατά τη διάρκεια της

αποκωδικοποίησης προς τον συνολικό αριθμό αποκωδικοποιήσεων που πραγματοποιήθηκαν.

4.1.2 Average In-Order Delay (Μέση καθυστέρηση εν σειρά παράδοσης)

Η καθυστέρηση πακέτου σε σειρά αποτελεί το χρονικό διάστημα από τη στιγμή που ένα πακέτο πηγής είναι έτοιμο για μετάδοση από τον αποστολέα μέχρι τη στιγμή που το πακέτο πηγής παραδίδεται με επιτυχία σε σειρά στο στον παραλήπτη. Η καθυστέρηση κωδικοποίησης και αποκωδικοποίησης RLNC δεν λαμβάνεται υπόψη. Ο μέσος χρόνος εν σειρά καθυστέρησης προκύπτει από το άθροισμα των καθυστερήσεων όλων των πακέτων πηγής που παραδίδονται με επιτυχία, διαιρούμενη με τον συνολικό αριθμό πακέτων που παραδόθηκαν με επιτυχία. Η μέση καθυστέρηση λαμβάνει υπόψη μόνο τα πακέτα πηγής που παραδίδονται με επιτυχία, αγνοώντας τα ληφθέντα κωδικοποιημένα πακέτα που συμβάλλουν στη διαδικασία αποκωδικοποίησης. Στο σύστημα επικοινωνίας, χρησιμοποιούμε χρονοθυρίδες (slots). Κάθε κωδικοποιημένο ή πηγαίο πακέτο μεταφέρεται από την έξοδο του κωδικοποιητή προς την είσοδο του αποκωδικοποιητή μέσα σε μία χρονική περίοδο.

4.1.3 Goodput

Το Goodput αναφέρεται στο πραγματικό ρυθμό μετάδοσης δεδομένων που είναι χρήσιμα και αξιοποιούνται από τον τελικό χρήστη, αφού ληφθούν υπόψη πιθανές απώλειες δεδομένων και άλλα πρωτόκολλα επικοινωνίας που επηρεάζουν την πραγματική απόδοση του δικτύου. Ουσιαστικά, αποτελεί μια μέτρηση της χρήσιμης πληροφορίας δεδομένων που φτάνουν στον τελικό προορισμό. Προκύπτει από τον λόγο του συνολικού αριθμού πακέτων που παραδόθηκαν με επιτυχία, διαιρούμενα με τον συνολικό χρόνο μετάδοσης που μετριέται σε χρονοθυρίδες (slots).

4.2 Αξιολόγηση G-rated block-based RLNC

Στο πλαίσιο αυτής της ενότητας, θα διερευνήσουμε την απόδοση του κώδικα G-rated block-based RLNC. Εξετάζουμε την απόδοση της G-rated block-based RLNC με περιορισμό στην πλεονάζουσα πληροφορία στα πλαίσια δύο διακριτών πειραμάτων όπου μελετούμε τα εξής: α) την επίδραση του generation size και β) τα διαφορετικά επίπεδα αξιοπιστίας του καναλιού, μέσα από τη μεταβολή της πιθανότητας σφαλμάτων στο κανάλι. Τα κωδικοποιημένα συμβολα (N_{coded}) προκύπτουν $N_{\text{coded}} = p_{\text{loss}} * G$

Πίνακας 4.2. Κοινές ρυθμίσεις προσομοιώσεων της G-rated.

| Παράμετρος | Τιμή |
|-------------|-------|
| πακέτα | 10000 |
| επαναλήψεις | 15 |

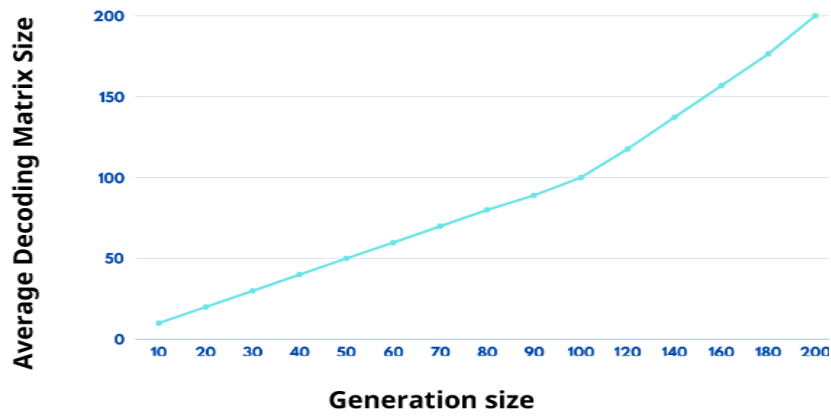
4.2.1 Πείραμα 1: Η επίδραση του generation size στην απόδοση της G-rated block-based RLNC

Στο πλαίσιο του πρώτου πειράματος, στοχεύουμε στην αξιολόγηση της απόδοσης της G-rated block-based RLNC με τη μεταβολή του μεγέθους της γενιάς (G). Οι παράμετροι του καναλιού καθώς και ο αριθμός των επαναλήψεων περιγράφονται στον πίνακα 4.2 με $p_{\text{loss}} = 10\%$. Με την παραλλαγή του G, επιδιώκουμε να διερευνήσουμε τον τρόπο με τον οποίο διαφορετικά μεγέθη block επηρεάζουν τις μετρήσεις που πραγματοποιούμε.

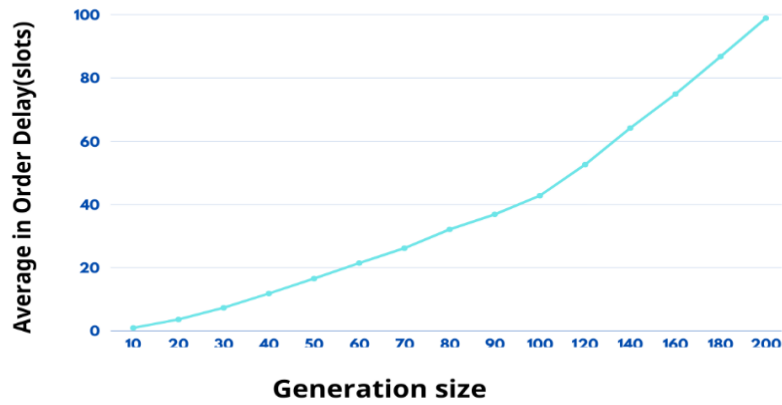
Τα εύρη τιμών που χρησιμοποιήσαμε για την επίδραση του G στην προσέγγιση με G-rated block-based RLNC καταγράφονται στον πίνακα 4.2.1.1 καθώς και το πλήθος των κωδικοποιημένων πακέτων που επιλέξαμε να στείλουμε, το οποίο έχει καθοριστεί με βάση το μέγεθος του generation και το ποσοστό σφαλμάτων. Οι ακριβείς τιμές καταγράφονται στον πίνακα 4.2.1.1

Πίνακας 4.2.1.1 Πλήθος κωδικοποιημένων πακέτων ανάλογα το μέγεθος της γενιάς με ploss=10%.

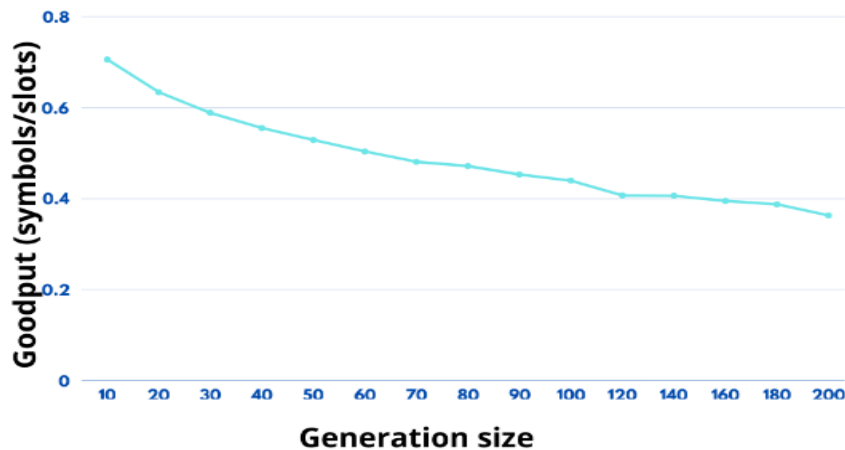
| G | 10 | 20 | 30 | 40 | 50 | 60 | 70 | 80 | 90 | 100 | 120 | 140 | 160 | 180 | 200 |
|--------------------|----|----|----|----|----|----|----|----|----|-----|-----|-----|-----|-----|-----|
| N _{coded} | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 12 | 14 | 16 | 18 | 20 |



(α)



(β)



(γ)

Εικόνα 4.2.1: Πείραμα 1: Απόδοση σχήματος κωδικοποίησης G-rated block-based RLNC , με μεταβαλλόμενο generation size. Οι γραφικές παρουσιάζουν τη σύγκριση της αύξησης του μεγέθους της γενιάς με τα αποτελέσματα στο μέσο μέγεθος του πίνακα αποκωδικοποίησης (4.2.1(α)), τη μέση εν σειρά καθυστέρηση (4.2.1(β)) και το Goodput (4.2.1(γ)).

4.2.1.1 Απόδοση στο average decoding matrix size

Όπως παρατηρούμε στην εικόνα 4.2.1(α), με την αύξηση του generation size, παρατηρείται τάση αύξησης του μέσου μεγέθους του πίνακα αποκωδικοποίησης. Αυτό συμβαίνει, γιατί με την αύξηση του μεγέθους της γενιάς, αυξάνεται και η πολυπλοκότητα της κωδικοποίησης και αποκωδικοποίησης .

4.2.1.2 Απόδοση στην average in order delay

Όπως παρατηρούμε στην εικόνα 4.2.1(β), με την αύξηση του generation size, η μετρική της average in order delay τείνει να αυξάνεται. Καθώς αυξάνεται το μέγεθος της γενιάς, αυξάνεται η απόσταση των κωδικοποιημένων πακέτων, επομένως τα χαμένα πηγαία πακέτα χρειάζονται μεγαλύτερο χρόνο αναμονής για την ανάκτηση

τους. Αυτό οδηγεί σε μεγαλύτερο συνολικά χρόνο εν σειρά καθυστέρησης, καθώς η μετάδοση των πακέτων και η αποκωδικοποίηση των πακέτων απαιτεί περισσότερο χρόνο.

4.2.1.3 Απόδοση στο Goodput

Όπως φαίνεται στην εικόνα 4.2.1(γ), η αύξηση του μεγέθους της γενιάς μπορεί να οδηγήσει σε μείωση του Goodput. Αυτό συμβαίνει διότι οι μεγαλύτερες γενιές συνεπάγονται μεγαλύτερο χρόνο μετάδοσης των δεδομένων. Καθώς αυξάνεται ο χρόνος μετάδοσης, η δυνατότητα του συστήματος να μεταδώσει δεδομένα με υψηλό goodput μειώνεται.

Η σύνοψη των αποτελεσμάτων αποδεικνύει ότι η αύξηση του μεγέθους της γενιάς οδηγεί σε αυξημένο average decoding matrix size . Αυτό δημιουργεί περισσότερη πολυπλοκότητα στην αποκωδικοποίηση. Παράλληλα, παρατηρείται αύξηση της average in order delay λόγω της επιλέον καθυστέρησης λήψης των κωδικοποιημένων πακέτων. Τέλος, η αύξηση του μεγέθους της γενιάς μπορεί να επιφέρει μείωση του Goodput λόγω του επιπλέον χρόνου μετάδοσης δεδομένων.

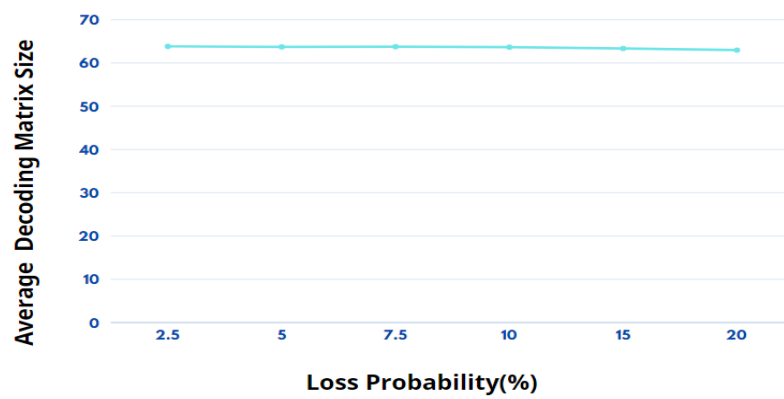
4.2.2 Πείραμα 2: Η επίδραση διαφορετικών επιπέδων αξιοπιστίας της G-rated block-based RLNC

Στο πλαίσιο του δεύτερου πειράματος, αποσκοπούμε στην αξιολόγηση της απόδοσης της G-rated block-based RLNC με τη μεταβολή της πιθανότητας απώλειας (ploss). Οι παράμετροι του καναλιού, καθώς και ο αριθμός των επαναλήψεων, περιγράφονται στον πίνακα 4.2 με $G=64$.

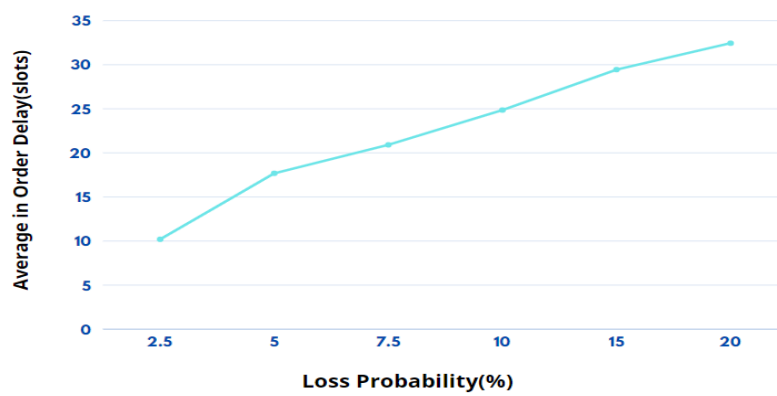
Τα εύρη τιμών που χρησιμοποιήσαμε για την επίδραση του ploss στην G-rated block-based RLNC καταγράφονται στον πίνακα 4.2.2.1 καθώς και το πλήθος των κωδικοποιημένων πακέτων που επιλέξαμε να στείλουμε, το οποίο έχει καθοριστεί με βάση το μέγεθος του generation και το ποσοστό σφαλμάτων. Οι ακριβείς τιμές καταγράφονται στον πίνακα 4.2.2.1.

Πίνακας 4.2.2.1 Πλήθος κωδικοποιημένων πακέτων ανάλογα το μέγεθος του ploss με $G=64$.

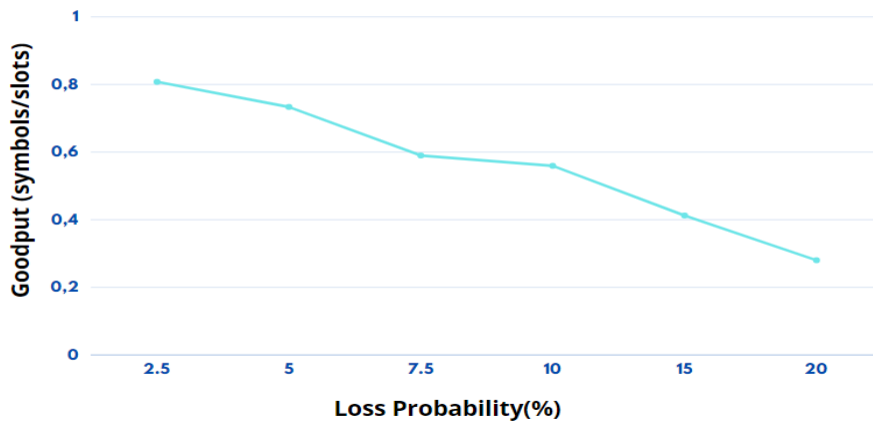
| ploss | 2.5 | 5 | 7.5 | 10 | 15 | 20 |
|--------------------|-----|---|-----|----|----|----|
| N _{coded} | 2 | 4 | 5 | 7 | 10 | 13 |



(α)



(β)



(γ)

Εικόνα 4.2.2: Πείραμα 2: Απόδοση της G-rated block-based RLNC, με μεταβαλλόμενο ploss. Οι γραφικές παρουσιάζουν τη σύγκριση της αύξησης του μεγέθους απώλειας με τα αποτελέσματα στο μέσο μέγεθος του πίνακα αποκωδικοποίησης (4.2.2(α)), τη μέση εν σειρά καθυστέρηση (4.2.2(β)) και το goodput (4.2.2(γ)).

4.2.2.1 Απόδοση στο average decoding matrix size

Όπως παρατηρούμε στην εικόνα 4.2.2(α), ενώ η πιθανότητα απώλειας αυξάνεται, η μετρική του μέσου μεγέθους του πίνακα αποκωδικοποίησης παραμένει σταθερή. Αυτό υποδηλώνει ότι η απώλεια δεδομένων δεν επηρεάζει σημαντικά την πολυπλοκότητα της διαδικασίας αποκωδικοποίησης, καθώς το μέγεθος του πίνακα αποκωδικοποίησης εξαρτάται κυρίως από το generation size, δηλαδή από το πλήθος των πακέτων που εμπλέκονται στην κωδικοποίηση και όχι τόσο από το ποσοστό απώλειας πακέτων.

4.2.2.2 Απόδοση στην average in order delay

Όπως φαίνεται στην εικόνα 4.2.2(β), η αύξηση της πιθανότητας απώλειας οδηγεί σε αύξηση της μετρικής average in order delay. Σε αυτό το πλαίσιο, τα πακέτα

που χάνονται δεν μπορούν να αποκωδικοποιηθούν μέχρι να έρθουν κωδικοποιημένα πακέτα. Επομένως, αυξημένη πιθανότητα απώλειας σημαίνει περισσότερα πακέτα που πρέπει να περιμένουν για να ολοκληρωθεί η διαδικασία αποκωδικοποίησης, οδηγώντας έτσι σε μεγαλύτερες καθυστερήσεις στην παράδοση των πακέτων.

4.2.2.3 Απόδοση στο Goodput

Όπως παρατηρούμε στην εικόνα 4.2.2(γ), ενώ η πιθανότητα απώλειας αυξάνεται, το Goodput μειώνεται. Όσο αυξάνεται η πιθανότητα απώλειας, ο αριθμός των πακέτων που χάνονται αυξάνεται, με αποτέλεσμα λιγότερα δεδομένα να φτάνουν επιτυχώς στον προορισμό τους. Άρα, η αύξηση της πιθανότητας απώλειας μπορεί να οδηγήσει σε μείωση του Goodput.

Η σύνοψη των αποτελεσμάτων δείχνει ότι με την αύξηση της πιθανότητας απώλειας δεδομένων, παρατηρούμε αύξηση της average in order delay και μείωση του Goodput του συστήματος. Παράλληλα, το average decoding matrix size παραμένει σταθερό παρά την αύξηση της πιθανότητας απώλειας, υποδεικνύοντας ότι η πολυπλοκότητα της αποκωδικοποίησης δεν επηρεάζεται σημαντικά από την απώλεια δεδομένων.

4.3 Αξιολόγηση της k-rated block-based RLNC

Σε αυτή την ενότητα θα εξετάσουμε την επίδραση της k-rated block-based RLNC στην απόδοση του κώδικα μας. Συγκεκριμένα, εξετάζουμε την απόδοση του k-rated block-based RLNC στα πλαίσια τεσσάρων διακριτών πειραμάτων όπου μελετούμε τα εξής: α) την επίδραση του code rate στην απόδοση του κώδικα μέσα από τη μεταβολή της πιθανότητας σφαλμάτων στο κανάλι, β) σύγκριση των δύο στρατηγικών διαχείρισης πακέτων μέσα από τη μεταβολή της πιθανότητας σφαλμάτων στο κανάλι, γ) την επίδραση του k σε σχέση με το n σε ισοδύναμα rate και δ) σύγκριση απόδοσης της G-rated block-based RLNC με τη k-rated block-based RLNC, μέσα από τη μεταβολή της πιθανότητας σφαλμάτων στο κανάλι.

Πίνακας 4.3 Κοινές ρυθμίσεις προσομοιώσεων της k-rated.

| Παράμετρος | Τιμή |
|-------------|-------|
| πακέτα | 10000 |
| επαναλήψεις | 10 |
| G | 100 |

4.3.1 Πείραμα 1: Η επίδραση του code rate στην απόδοση του κώδικα

Στο πρώτο πείραμα, τροποποιούμε το R, προκαλώντας διαφορετικό επίπεδο προστασίας έναντι των απωλειών πακέτων. Για τον σκοπό αυτό, ελέγχουμε τις τιμές του R από την τιμή $R = 1 - p_{\text{loss}}$ και τις μειώνουμε βήμα προς βήμα μέχρι την τιμή $R = 1 - 2 \cdot p_{\text{loss}}$, επιλέγοντας την τιμή που προσφέρει την καλύτερη απόδοση. Τα εύρη τιμών που χρησιμοποιήσαμε για την επίδραση του p_{loss} στην k-rated block-based RLNC καταγράφονται στον πίνακα 4.3.1.2 καθώς και το συνολικό πλήθος των R που δοκιμάσαμε. Τα βέλτιστα R βρίσκονται στον πίνακα 4.3.1.2. Οι ρυθμίσεις του καναλιού που χρησιμοποιήθηκαν σε αυτό το πείραμα περιγράφονται στον πίνακα 4.3, η στρατηγική διαχείρισης των καέτων που χρησιμοποιήσαμε είναι η Απόρριψη εκτός σειράς. Για την γραφική αναπαράσταση επιλέξαμε την τιμή που προσφέρει την καλύτερη απόδοση.

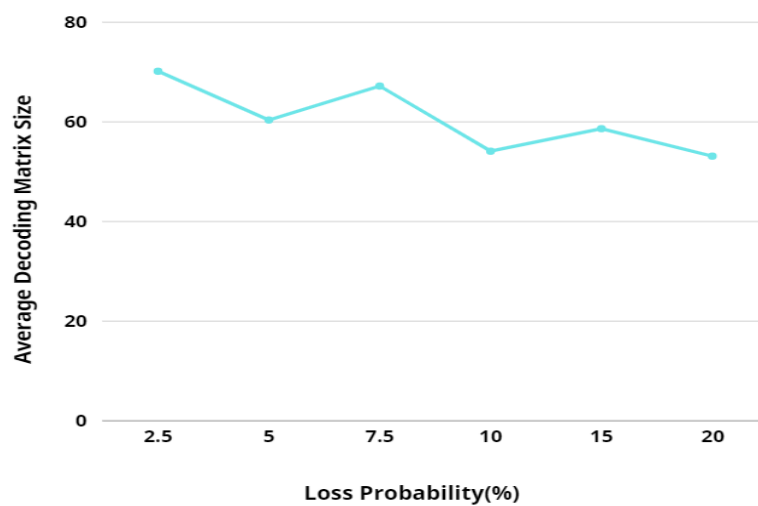
Πίνακας 4.3.1.1 Όλα τα R που χρησιμοποιήσαμε για την προσομοίωση.

| p_{loss} | R |
|-------------------|---|
| 2.5 | 39/40, 38/40, 77/80, 76/80 |
| 5 | 19/20, 18/20, 37/40, 36/40, 55/60, 54/60 |
| 7.5 | 37/40, 36/40, 35/40, 34/40, 70/80, 69/80, |

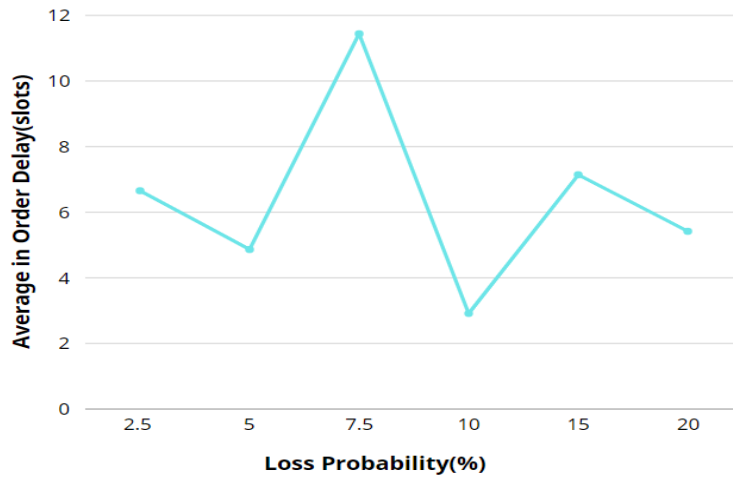
| | |
|----|---|
| | 68/80 |
| 10 | 9/10, 8/10, 17/20, 16/20, 25/30, 24/30 |
| 15 | 17/20, 16/20, 15/20, 14/20, 32/40, 31/40, 30/40, 43/60, 42/60 |
| 20 | 8/10, 7/10, 6/10, 14/20, 13/20, 21/30, 20/30 |

Πίνακας 4.3.1.2 Τα R που έχουν καλύτερη απόδοση ανάλογα το μέγεθος του ploss με G= 100.

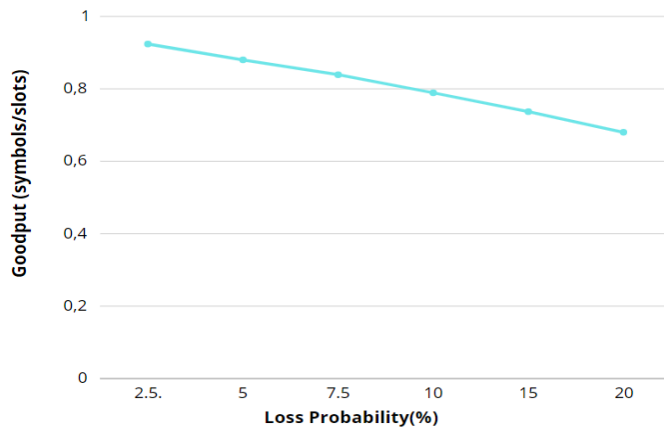
| ploss | 2.5 | 5 | 7.5 | 10 | 15 | 20 |
|-------|-------|-------|-------|------|-------|------|
| R | 38/40 | 18/20 | 34/40 | 8/10 | 15/20 | 7/10 |



(α)



(β)



(γ)

Εικόνα 4.3.1: Πείραμα1: Απόδοση της k-rated block-based RLNC με μεταβαλλόμενο ploss. Οι γραφικές παρουσιάζουν τη σύγκριση της αύξησης του μεγέθους απώλειας (με τα R με την καλύτερη απόδοση για κάθε ploss), με τα αποτελέσματα στο μέσο μέγεθος του πίνακα αποκωδικοποίησης (4.3.1(α)), τη μέση εν σειρά καθυστέρηση (4.3.1(β)) και το Goodput (4.3.1(γ)).

4.3.1.1 Απόδοση στο average decoding matrix size

Όπως παρατηρούμε στην εικόνα 4.3.1(α) με τα βέλτιστα R που επιλέξαμε, η

συμπεριφορά του average decoding matrix size δεν είναι σταθερή, αλλά εξαρτάται από την τιμή του R που επιλέγουμε. Συγκεκριμένα, όσο μικρότερος είναι ο αριθμητής του κλάσματος, τόσο λιγότερα πηγαία πακέτα θα συμμετέχουν στη διαδικασία κωδικοποίησης/αποκωδικοποίησης. Επομένως, το average decoding matrix size θα είναι μικρότερο.

Με μια πιο λεπτομερή παράθεση των αποτελεσμάτων από τα πειράματα που εκτελέσαμε, για $ploss = 7.5\%$ και διάφορες τιμές του ρυθμού R, παρατηρήσαμε ότι για τον ρυθμό 37/40 (δηλαδή 0.925), το μικρότερο average decoding matrix size που καταγράφηκε ήταν 61.53090403. Ωστόσο, αποφασίσαμε να επιλέξουμε την τιμή 34/40 (δηλαδή 0.875), καθώς ο ρυθμός 34/40 εμφάνισε τη μικρότερη καθυστέρηση και μεγαλύτερη τιμή `goodput`. Επίσης, παρατηρήθηκε ότι ο ρυθμός 69/80 (δηλαδή 0.8625), που αντιστοιχεί σε περισσότερη πλεονάζουσα πληροφορία, είχε αρκετά μεγαλύτερο average decoding matrix size ίσο με 83.44396389. Επομένως, συμπεραίνουμε ότι η επίδραση στο average decoding matrix size φαίνεται να εξαρτάται περισσότερο από το πότε στέλνουμε την πλεονάζουσα πληροφορία, παρά από την ποσότητα της πληροφορίας αυτής.

4.3.1.2 Απόδοση στην average in order delay

Στην εικόνα 4.3.1(β)) με τα βέλτιστα R που επιλέξαμε, παρατηρούμε μια αντίστοιχη συμπεριφορά με το average decoding matrix size. Δηλαδή, όσο μικρότερος είναι ο αριθμητής του κλάσματος, τόσο πιο κοντά βρίσκονται τα κωδικοποιημένα πακέτα, με αποτέλεσμα την ταχύτερη αποκωδικοποίηση. Επομένως, με μικρότερο αριθμητή παρατηρούμε μικρότερη average in-order delay.

Με βάση τις πειραματικές αξιολογήσεις, παρατηρούμε ότι όσο μειώνουμε την πλεονάζουσα πληροφορία, δηλαδή αυξάνουμε το R, τόσο αυξάνεται η μετρική της average in-order delay. Για παράδειγμα, για ένα ποσοστό απώλειας πακέτων ίσο με 15%, ο ρυθμός 17/20 έχει μέση καθυστέρηση ίση με 14,00246, ενώ ο ρυθμός 14/20 έχει καθυστέρηση ίση με 5,496745. Αυτό αντιστοιχεί σε μια μείωση της καθυστέρησης κοντά στο 60%. Παρόλο που ο ρυθμός 14/20 είχε την μικρότερη καθυστέρηση, σε αυτή την πιθανότητα απώλειας, επιλέξαμε ως βέλτιστο τον ρυθμό 15/20 με καθυστέρηση 7,142223 καθώς είχε αρκετά καλύτερη τιμή στο `goodput`.

Επομένως, παρατηρούμε ότι με την αύξηση της πλεονάζουσας πληροφορίας, γίνεται ταχύτερη η ανάκτηση των πηγαίων πακέτων. Αυτό συνεπάγεται σε μικρότερο

average in-order delay, καθώς τα πηγαία πακέτα μπορούν να ανακτώνται πιο γρήγορα σε περίπτωση απώλειας.

4.3.1.3 Απόδοση στο Goodput

Στην εικόνα 4.3.1(γ)) με τα βέλτιστα R που επιλέξαμε, καταγράφεται μια μείωση του Goodput καθώς η πιθανότητα απώλειας αυξάνεται. Με την αύξηση της πιθανότητας απώλειας, ο αριθμός των χαμένων πακέτων αυξάνεται επίσης, με αποτέλεσμα να φτάνουν λιγότερα δεδομένα με επιτυχία στον προορισμό τους.

Καθώς αυξάνουμε την πλεονάζουσα πληροφορία, δηλαδή μειώνουμε τον ρυθμό, παρατηρούμε αρχικά αύξηση στη μετρική του Goodput. Ωστόσο, μετά από ένα σημείο, αυτή η αύξηση της πλεονάζουσας πληροφορίας δεν συνεχίζει να επιφέρει περαιτέρω αύξηση του Goodput. Για παράδειγμα, σε πιθανότητα απώλειας 20%, με $R=8/10$, το Goodput είναι 0,471904, ενώ με $R=7/10$ είναι 0,680084 και με $R=6/10$ είναι 0,597198. Σε αυτήν την περίπτωση, επιλέξαμε για βέλτιστο ρυθμό το 7/10 λόγω της μεγαλύτερης απόδοσης στο goodput. Με περισσότερα κωδικοποιημένα πακέτα, η k-rated είναι πιο ανθεκτική στα σφάλματα. Αυτό σημαίνει ότι οι απώλειες πακέτων έχουν μικρότερο αντίκτυπο, καθώς περισσότερα κωδικοποιημένα πακέτα είναι διαθέσιμα για την ανάκτησή τους. Παρόλα αυτά, η υπερβολική πλεονάζουσα πληροφορία επιδρά αρνητικά στη συνολική απόδοση του αλγορίθμου. Επομένως, υπάρχει ένα σημείο όπου η πρόσθετη πλεονάζουσα πληροφορία δεν συνεισφέρει περαιτέρω στη βελτίωση του Goodput και μπορεί ακόμη να το μειώσει.

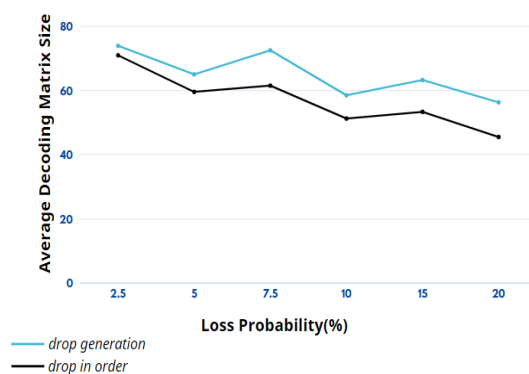
Με βάση τα πειράματα που πραγματοποιήσαμε, καταλήξαμε στο συμπέρασμα ότι η αύξηση της πλεονάζουσας πληροφορίας οδηγεί σε μείωση της μετρικής της average in order delay. Μέχρι ένα σημείο, η αύξηση της πλεονάζουσας πληροφορίας οδηγεί σε αύξηση του Goodput, αλλά υπερβολική πλεονάζουσα πληροφορία μπορεί να επηρεάσει αρνητικά την απόδοση του συστήματος. Τέλος, παρατηρήσαμε ότι η μετρική του average decoding matrix size επηρεάζεται περισσότερο από το πότε στέλνουμε την πλεονάζουσα πληροφορία, παρά από το πλήθος αυτής.

4.3.2 Πείραμα 2: Σύγκριση των δύο στρατηγικών διαχείρισης πακέτων(drop out of order vs drop generation)

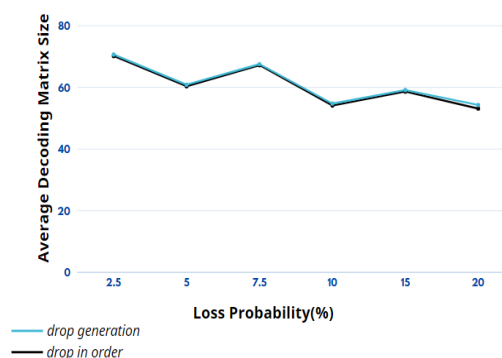
Στα πλαίσια του πειράματος, στοχεύουμε στην σύγκριση των επιδράσεων δύο στρατηγικών διαχείρισης πακέτων, καθώς αλλάζει η πιθανότητα απώλειας. Οι δύο στρατηγικές είναι η drop generation και drop out of order. Για τον σκοπό αυτό, επιλέγουμε τιμές του R που καλύπτουν τόσο τις καλύτερες αποδόσεις όσο και εκείνες που είχαν τις χειρότερες αποδόσεις που εκτιμήσαμε στο προηγούμενο πείραμα, προκειμένου να πραγματοποιήσουμε μια πλήρη σύγκριση. Τα R με τις καλύτερες αποδόσεις παρουσιάζονται στον πίνακα 4.3.1.2, ενώ τα R με τις χειρότερες αποδόσεις στον πίνακα 4.3.2.1. Οι παράμετροι του καναλιού, καθώς και ο αριθμός των επαναλήψεων, περιγράφονται στον πίνακα 4.3.

Πίνακας 4.3.2.1 Τα R με την χειρότερη απόδοση.

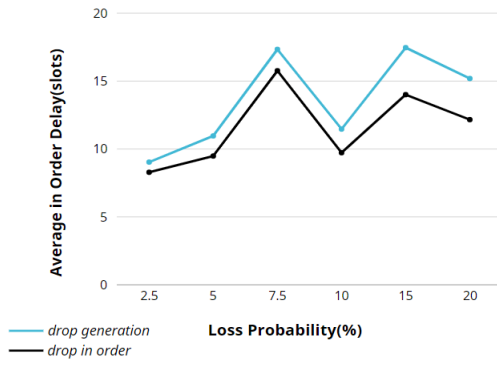
| ploss | 2.5 | 5 | 7.5 | 10 | 15 | 20 |
|-------|-------|-------|-------|------|-------|------|
| R | 39/40 | 19/20 | 37/40 | 9/10 | 17/20 | 8/10 |



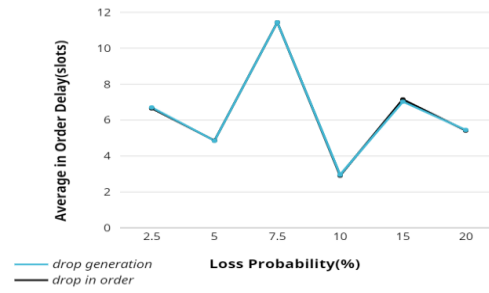
(α)



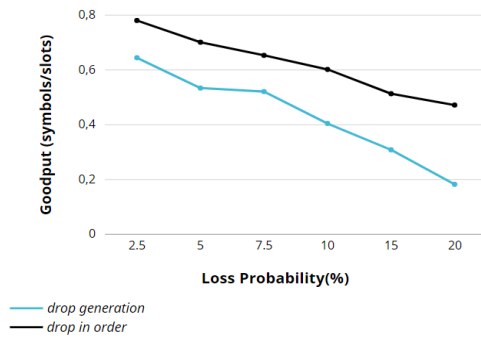
(β)



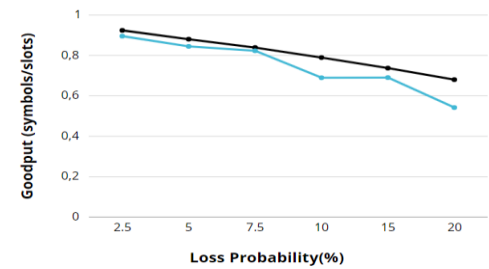
(γ)



(δ)



(ε)



(ζ)

Εικόνα 4.3.2 Πείραμα 2 : Απόδοση της k-rated block-based RLNC με μεταβαλλόμενο loss probability. Οι γραφικές παρουσιάζουν τη σύγκριση των επιδράσεων δύο στρατηγικών διαχείρισης πακέτων καθώς αυξάνεται το μέγεθος απώλειας. Στα διαγράμματα (4.3.2(β), (δ), (στ)), παρουσιάζονται τα αποτελέσματα για το μέσο μέγεθος του πίνακα αποκωδικοποίησης, τη μέση εν σειρά καθυστέρηση και το Goodput με περισσότερη πλεονάζουσα πληροφορία, ενώ στα διαγράμματα (4.3.2(α), (γ), (ε)) παρουσιάζονται οι ίδιες μετρικές με λιγότερη πλεονάζουσα πληροφορία.

4.3.2.1 Απόδοση στο average decoding matrix size

Στην εικόνα 4.3.2 (β), όπου παρουσιάζεται η μέθοδος με την περισσότερη πλεονάζουσα πληροφορία (Πίνακας 4.3.1.2) στην μετρική του average decoding matrix

sizee, οι μέθοδοι διαχείρισης πακέτων σχεδόν συγκλίνουν, μόνο στο loss probability=20% η drop generation έχει λίγο μεγαλύτερη τιμή από την drop out of order. Αυτό συμβαίνει διότι όσο πιο ισχυρός είναι ένας ρυθμός κωδικοποίησης (δηλαδή όσο μικρότερη τιμή έχει), τόσο πιο πιθανό είναι να μπορεί να ανακτήσει πλήρως ένα generation.

Από την άλλη πλευρά στην εικόνα 4.3.2 (α), όπου η πλεονάζουσα πληροφορία είναι λιγότερη (Πίνακας 4.3.2.1), αυξάνονται οι πιθανότητες να μην μπορεί να γίνει αποκωδικοποίηση, με αποτέλεσμα η drop out of order symbols να έχει εμφανώς μικρότερες τιμές στο average decoding matrix size, σε όλα τα ploss σε σύγκριση με την drop generation. Αυτό δείχνει ότι, σε περιπτώσεις όπου η πλεονάζουσα πληροφορία είναι πιο περιορισμένη, η στρατηγική drop out of order επιτυγχάνει καλύτερο average decoding matrix size από την drop generation.

4.3.2.2 Απόδοση στην average in order delay

Αντίστοιχα, με το average decoding matrix size, παρατηρούμε στην εικόνα 4.3.2 (δ) ότι και το average in order delay, με περισσότερη περιττή πληροφορία, παρουσιάζει μια ανάλογη συμπεριφορά. Συγκεκριμένα, οι δύο μέθοδοι σχεδόν συγκλίνουν εκτός από την πιθανότητα απώλειας ίση με 15%, όπου η drop out of order έχει ελάχιστα περισσότερη καθυστέρηση.

Αντίθετα, στην περίπτωση με την λιγότερη πλεονάζουσα πληροφορία (Εικόνα 4.3.2(γ)), φαίνεται ότι η μέθοδος drop out of order παρουσιάζει σημαντικά μικρότερες καθυστερήσεις σε όλα τα ploss. Αυτό υποδεικνύει ότι η στρατηγική drop out of order είναι σημαντικά πιο αποδοτική.

4.3.2.3 Απόδοση στο Goodput

Στην περίπτωση του Goodput, παρατηρούμε ότι η μέθοδος drop out of order εμφανίζει σημαντικά βελτιωμένη απόδοση τόσο στις περιπτώσεις με περισσότερη πλεονάζουσα πληροφορία (Εικόνα 4.3.2(στ)) όσο και σε αυτές με λιγότερη (Εικόνα 4.3.2(ε)). Μόνο στην περίπτωση της λιγότερης πλεονάζουσας πληροφορίας και στο ploss = 7,5%, παρατηρούμε μια τάση προς σύγκλιση. Αντίθετα, στην περίπτωση της

περισσότερης πλεονάζουσας πληροφορίας, οι αποκλίσεις είναι ιδιαίτερα έντονες, με ξεκάθαρη υπεροχή της μεθόδου drop out of order.

Τα συμπεράσματα από αυτό το πείραμα επιβεβαιώνουν ότι, στο average in order delay και στο average decoding matrix size, όσο αυξάνεται η πλεονάζουσα πληροφορία που αποστέλλεται, η προσέγγιση drop generation προσεγγίζει σταδιακά την απόδοση της drop out of order, αυτό συμβαίνει επειδή όσο αυξάνεται η ποσότητα της πλεονάζουσας πληροφορίας, τόσο μεγαλύτερη είναι η πιθανότητα ανάκτησης των πηγαίων πακέτων. Παρόλα αυτά, με λιγότερη πλεονάζουσα πληροφορία, η προσέγγιση drop out of order αποδίδει πολύ καλύτερα. Όσον αφορά το Goodput, φαίνεται ότι η προσέγγιση drop out of order παρουσιάζει βελτιωμένη απόδοση, είτε με λιγότερη είτε με περισσότερη πλεονάζουσα πληροφορία.

4.3.3 Πείραμα 3: Αξιολόγηση επίδρασης του k σε σχέση με το n στο code rate

Στα πλαίσια του πειράματος, σκοπός μας είναι να συγκρίνουμε την απόδοση διαφορετικών ισοδύναμων κλασμάτων R , τα οποία έχουν διαφορετικό αριθμητή και παρονομαστή (δηλαδή διαφορετικά k και n), κρατώντας τον κλάσμα R σταθερό. Συγκεκριμένα, εξετάζουμε πώς επηρεάζει η μεταβολή του k την απόδοση του κώδικα, με το k να είναι πάντα μικρότερο από το μέγεθος του generation, διατηρώντας σταθερές τις συνθήκες του καναλιού.

Τα πειράματα έγιναν για τις πιθανότητες απώλειας 2.5% και 20% και στον πίνακα 4.3.3.1 και 4.3.3.2 αντίστοιχα, εξετάζουμε το εύρος ισοδύναμων R που χρησιμοποιήσαμε. Αναζητούμε τις βέλτιστες επιδόσεις code rate, λαμβάνοντας υπόψη τις τιμές που προκύπτουν από τον πίνακα 4.3.1.2. Οι παράμετροι του καναλιού και ο αριθμός των επαναλήψεων περιγράφονται στον πίνακα 4.3 με την διαχείριση πακέτων εκτός σειράς.

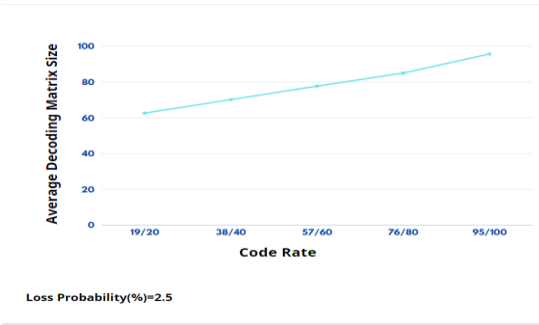
Πίνακας 4.3.3.1 Τα R για το ploss = 2.5%

| Ploss=2.5% | | | | | |
|------------|-------|-------|-------|-------|--------|
| R | 19/20 | 38/40 | 57/60 | 76/80 | 95/100 |

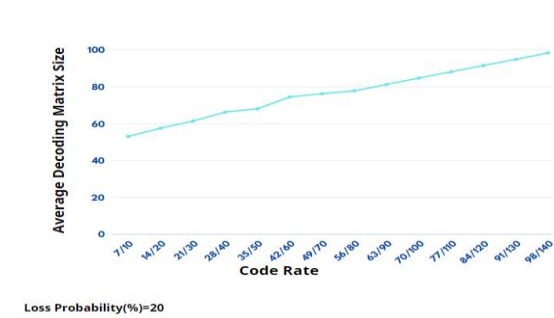
Πίνακας 4.3.3.2 Τα R για το ploss= 20%

| Ploss=20% | | | | | | | | | | | | | | |
|-----------|------|-------|-------|-------|-------|-------|-------|-------|-------|--------|--------|--------|--------|--------|
| R | 7/10 | 14/20 | 21/30 | 28/40 | 35/50 | 42/60 | 49/70 | 56/80 | 63/90 | 70/100 | 77/110 | 84/120 | 91/130 | 98/140 |

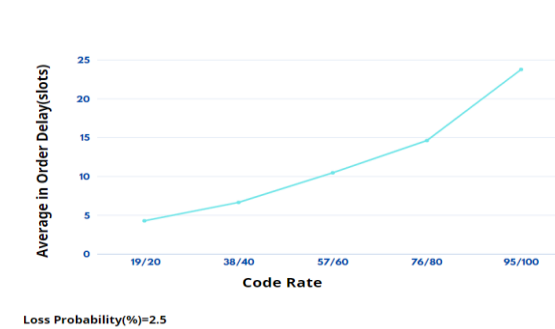
(α)



(β)



(γ)

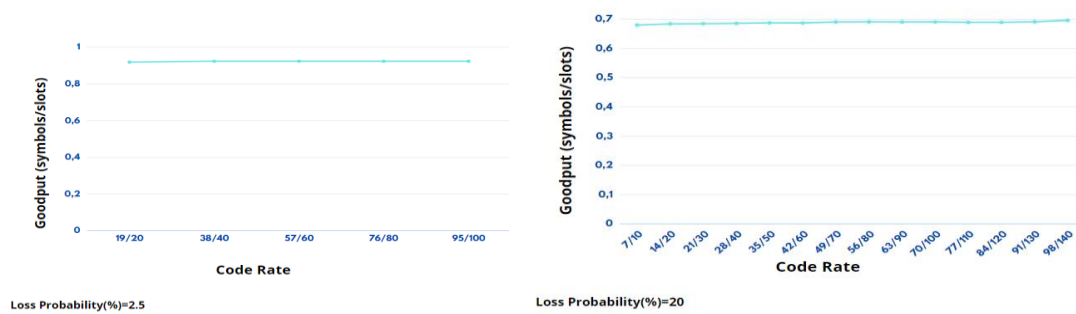


(δ)



(ε)

(ζ)



Εικόνα 4.3.3 Πείραμα 3: Οι γραφικές παρουσιάζουν την επίδραση ισοδύναμων κλασμάτων στην απόδοση όταν η πιθανότητα απώλειας είναι 2.5% και 20%, με βέλτιστο R που προέκυψε από το πρώτο πείραμα. Στα διαγράμματα (4.3.3(β), (δ), (στ)), παρουσιάζονται τα αποτελέσματα για το μέσο μέγεθος του πίνακα αποκωδικοποίησης, τη μέση εν σειρά καθυστέρηση και το Goodput με πιθανότητα απώλειας 2.5%, ενώ στα διαγράμματα (4.3.3(α), (γ), (ε)) παρουσιάζονται τα ίδια μετρήσιμα με πιθανότητα απώλειας 20%.

4.3.3.1 Απόδοση στο average decoding matrix size

Με βάση την εικόνα 4.3.3(α), παρατηρούμε ότι με $p_{loss} = 2.5\%$, το average decoding matrix size αυξάνεται, καθώς αυξάνουμε την απόσταση των κωδικοποιημένων πακέτων μεταξύ τους, δηλαδή καθώς αυξάνονται οι αριθμητές και παρανομαστές των διαφορετικών ισοδύναμων κλασμάτων. Αυτό είναι αναμενόμενο, καθώς όσο μικρότερο είναι το k , τόσο μικρότερες είναι οι υποομάδες που προκύπτουν και χρησιμοποιούνται, ως αποτέλεσμα να απλοποιείται η διαδικασία της κωδικοποίησης και αποκωδικοποίησης.

Η παρατήρηση από την εικόνα 4.3.3(β), με $p_{loss} = 20\%$, είναι αντίστοιχη. Σε αυτή την περίπτωση, παρατηρούμε ότι υπάρχουν περισσότερα ισοδύναμα κλάσματα, τα οποία καταγράφουν ανοδική πορεία. Αυτό συμβαίνει για τον ίδιο λόγο που προαναφέραμε για την πιθανότητα απώλειας του 2.5%.

4.3.3.2 Απόδοση στην average in order delay

To average in order delay, όπως φαίνεται στις εικόνες 4.3.3(γ) και 4.3.3(δ) για πιθανότητες απώλειας 2.5% και 20%, έχει ανοδική πορεία. Όσο μεγαλύτεροι είναι οι αριθμητές και οι παρανομαστές στα ισοδύναμα κλάσματα, τόσο μεγαλύτερη είναι απόσταση μεταξύ οποιουδήποτε πηγαίου πακέτου και του επόμενου κωδικοποιημένου πακέτου, με αποτέλεσμα να αυξηθεί ο χρόνος ανάκτησης των χαμένων πηγαίων πακέτων και, επομένως, αυξηθεί η καθυστέρηση.

4.3.3.3 Απόδοση στο Goodput

Η παρατήρησή μας σχετικά με το Goodput από τις εικόνες 4.3.3(ε) και 4.3.3(στ) είναι πολύ ενδιαφέρουσα. Παρατηρούμε ότι παραμένει σχεδόν σταθερό (και για τις δύο πιθανότητες απώλειας), με μια μικρή ανοδική τάση κατά την αύξηση των αριθμητών και παρανομαστών των ισοδύναμων κλασμάτων. Αυτή η σταθερότητα μπορεί να οφείλεται στο γεγονός ότι τα κλάσματα είναι ισοδύναμα, επομένως μπορούν να αποκαταστήσουν περίπου τα ίδια χαμένα πακέτα.

Από αυτό το πείραμα, παρατηρούμε ότι τα ισοδύναμα κλάσματα με μικρότερο k και n έχουν σημαντικά μικρότερο average decoding matrix size και το average in order delay, ενώ διατηρούν περίπου τις ίδιες τιμές Goodput. Επομένως, προτιμάμε τα ισοδύναμα κλάσματα με τα μικρότερα k , διότι παρουσιάζουν καλύτερη απόδοση στην καθυστέρηση και στον πίνακα αποκωδικοποίησης.

4.3.4 Πείραμα 4: G-rated block based RLNC vs k-rated block-based RLNC

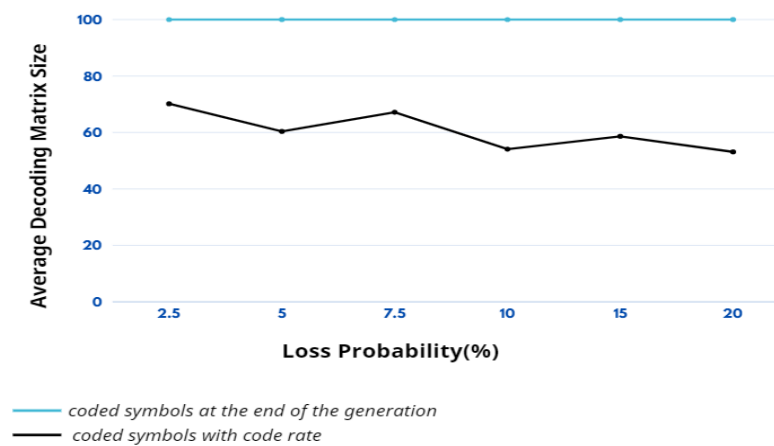
Κατά τη διάρκεια του πειράματος, στοχεύουμε στη σύγκριση των δύο διαφορετικών προσεγγίσεων. Η πρώτη προσέγγιση είναι η G-rated block based RLNC, ενώ η δεύτερη προσέγγιση αφορά την k-rated block-based RLNC. Εξετάζουμε τον τρόπο με τον οποίο αυτές οι προσεγγίσεις επηρεάζουν διάφορες μετρικές, καθώς αυξάνεται η πιθανότητα απώλειας. Επιδιώκουμε να προσδιορίσουμε τις βέλτιστες επιδόσεις του

κώδικα, λαμβάνοντας υπόψη τις τιμές που προκύπτουν από τον πίνακα 4.3.4.2. Οι παράμετροι του καναλιού και ο αριθμός των επαναλήψεων περιγράφονται στον πίνακα 4.3 με μηχανισμό απόρριψης πακέτων την απόρριψη εκτός σειράς.

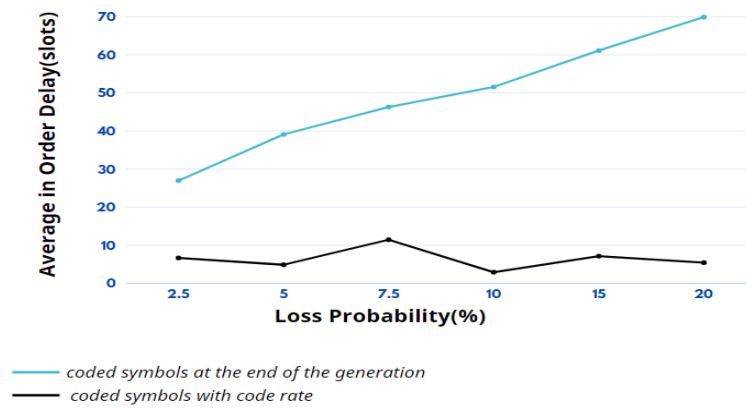
Πίνακας 4.3.4.2 Τα βέλτιστα code rate για την αποστολή κωδικοποιημένων πακέτων καθώς και το πλήθος πακέτων για το τέλος κάθε γενιάς ανάλογα με την πιθανότητα απώλειας.

| ploss | 2.5 | 5 | 7.5 | 10 | 15 | 20 |
|-------|-------|-------|-------|------|-------|------|
| n | 6 | 12 | 18 | 25 | 34 | 43 |
| R | 38/40 | 18/20 | 34/40 | 8/10 | 15/20 | 7/10 |

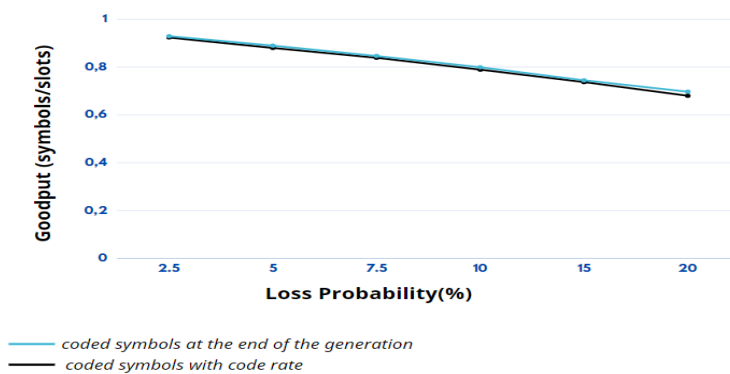
(α)



(β)



(γ)



Εικόνα 4.3.4 Πείραμα 4: Οι γραφικές παρουσιάζουν την σύγκριση δύο διαφορετικών προσεγγίσεων, όταν η πιθανότητα απώλειας αυξάνεται με βέλτιστο code rate. Στα διαγράμματα παρουσιάζονται τα αποτελέσματα στο μέσο μέγεθος του πίνακα αποκωδικοποίησης (4.3.4(α)), τη μέση εν σειρά καθυστέρηση (4.3.4(β)) και το Goodput (4.3.4(γ)).

4.3.4.1 Απόδοση στο average decoding matrix size

Όπως παρατηρούμε στην εικόνα 4.3.4(α), η k-rated block-based RLNC, όπου τα κωδικοποιημένα πακέτα τοποθετούνται ενδιάμεσα των πηγαίων πακέτων, φαίνεται να

απλοποιεί σημαντικά το average decoding matrix size σε σύγκριση με τη G-rated block-based RLNC η οποία στέλνει τα κωδικοποιημένα πακέτα στο τέλος κάθε γενιάς. Αυτό συμβαίνει διότι οι υποομάδες που παράγουν τα κωδικοποιημένα πακέτα της k-rated block-based έχουν μικρότερη πολυπλοκότητα, από τις ομάδες της G-rated block-based RLNC.

4.3.4.2 Απόδοση στην average in order delay

Η σύγκριση μεταξύ των δύο μεθόδων όσον αφορά average in order delay αποκαλύπτει σημαντική διακύμανση (Εικόνα 4.3.4(β)), ιδίως με την πιθανότητας απώλειας στο 20% όπου η k-rated block-based RLNC, έχει καθυστέρηση 5,4177, σε σύγκριση με την G-rated block-based RLNC, όπου η καθυστέρηση είναι 69,91701. Είναι λογικό ότι η διαφορά στην καθυστέρηση είναι τόσο μεγάλη, καθώς με την k-rated block-based RLNC μπορούμε να ανακτήσουμε χαμένα πακέτα πολύ πιο γρήγορα έφθον τα κωδικοποιημένα πακέτα μεταδίδονται ομοιομορφά κατά την διάρκεια του block, ενώ στην άλλη περίπτωση πρέπει να περιμένουμε μέχρι το τέλος της γενιάς για να τα ανακτήσουμε.

4.3.4.3 Απόδοση στο Goodput

Στην εικόνα 4.3.4(γ), παρατηρούμε ότι και οι δύο προσεγγίσεις σχεδόν ταυτίζονται όσον αφορά το Goodput. Τόσο η k-rated block-based RLNC, όσο και G-rated block-based RLNC προσαρμόζονται έτσι ώστε να στέλνουν κατά μέσο όρο ίδια πλεονάζουσα πληροφορία. Αυτό σημαίνει ότι και οι δύο προσεγγίσεις επιδιώκουν να επιτύχουν το μέγιστο Goodput.

Από το πείραμα, προκύπτει ότι η k-rated block-based RLNC βελτιώνει σημαντικά την απόδοση στο average decoding matrix size και στο average in order delay. Αυτό συμβαίνει διότι η διαδικασία αποκωδικοποίησης μπορεί να εκτελείται πιο αποτελεσματικά όταν τα κωδικοποιημένα πακέτα είναι διαθέσιμα ενδιάμεσα στη διαδικασία, επιτρέποντας έτσι μία γρηγορότερη αποκωδικοποίηση.

Ωστόσο, αν και η k-rated block-based RLNC βελτιώνει την απόδοση στο average decoding matrix size και στο average in order delay, οδηγεί σε παρόμοιο Goodput με την άλλη προσέγγιση. Αυτό οφείλεται στο γεγονός ότι και οι δύο προσεγγίσεις προσαρμόζονται ώστε να μεγιστοποιήσουν το Goodput, λαμβάνοντας υπόψη τις συνθήκες κάθε φάσης με βάση την πιθανότητα απώλειας.

Κεφάλαιο 5.

Συμπεράσματ

α

Η αποτελεσματική μεταφορά δεδομένων με χαμηλή καθυστέρηση αποτελεί προτεραιότητα στη σύγχρονη επικοινωνία. Ενώ η προσέγγιση της FEC μπορεί να προσφέρει λύσεις, αυτές είναι περιορισμένες σε περιπτώσεις υψηλού αριθμού σφαλμάτων. Συνεπώς, η RLNC προτείνεται ως πιο αποτελεσματική εναλλακτική, προσφέροντας ευελιξία και αποδοτική διαχείριση σφαλμάτων σε πιο απαιτητικές συνθήκες.

Η κωδικοποίηση δικτύου με τη χρήση της τυχαίας γραμμικής κωδικοποίησης δικτύου (RLNC) επιτρέπει τη μετάδοση δεδομένων σε δίκτυα όπου οι κόμβοι μπορεί να υποστούν απώλεια πακέτων. Με τη χρήση της RLNC, οι κόμβοι μπορούν να συνδυάσουν τα δεδομένα που λαμβάνουν και να δημιουργήσουν κωδικοποιημένα πακέτα, ενισχύοντας έτσι την ανθεκτικότητα στις απώλειες πακέτων. Η systematic κωδικοποίηση RLNC είναι μια προσέγγιση όπου τα αρχικά δεδομένα μεταδίδονται πρώτα χωρίς κωδικοποίηση και στη συνέχεια μεταδίδονται πλεονάζοντα κωδικοποιημένα, εξασφαλίζοντας χαμηλούς χρόνους καθυστέρησης και απλότητα στη διαδικασία μετάδοσης και ανάκτησης πληροφοριών σε περιβάλλοντα με σπάνια απώλεια πακέτων, ενώ συγκλίνει στην απόδοση της πλήρους κωδικοποίησης RLNC σε περιβάλλοντα με σημαντική απώλεια πακέτων.

Σε αυτή την διπλωματική εργασία, επιλέξαμε τη βιβλιοθήκη PyErasure, καθώς περιείχε μια απλή υλοποίηση της systematic block-based RLNC. Ξεκινήσαμε εξετάζοντας την αλλαγή της αρχικής προσέγγισης (Rateless block-based RLNC) που έστειλε τα δεδομένα σαν ένα μεγάλο block, με την αποστολή των πηγαίων πακέτων ακολουθούμενα από απεριόριστο αριθμό κωδικοποιημένων πακέτων, επισημαίνοντας τους περιορισμούς της, ως προς την απόδοση. Στη συνέχεια, υλοποιήσαμε τη G-rated

block-based RLNC, όπου αντί να χειριζόμαστε ένα συνεχές ρεύμα πακέτων (όπως στην Rateless), τονίσαμε τη σημασία του χωρισμού σε γενιές ή blocks, που απλοποιεί την διαδικασία της αποκωδικοποίησης και περιορίσαμε τα κωδικοποιημένα πακέτα που στέλνουμε στο τέλος κάθε γενιάς. Τέλος υλοποιήσαμε την k-rated block-based RLNC η οποία αποτελείται από ένα ρυθμό κωδικοποίησης ο οποίος θα χωρίζει νοητά κάθε γενιά σε μικρότερες υπογενιές μεγέθους k πηγαίων πακέτων ακολουθούμενα από n κωδικοποιημένα πακέτα. Με αυτή την ομοιόμορφη μετάδοση των κωδικοποιημένων πακέτων ανάμεσα από τα πηγαία προσπαθήσαμε να μειώσουμε τη καθυστέρηση και την πολυπλοκότητα της αποκωδικοποίησης.

Μέσω εκτενών αξιολογήσεων που πραγματοποιήθηκαν καταλήξαμε σε σημαντικά συμπεράσματα σχετικά με την επίδραση του R στην απόδοση του κώδικα και τις στρατηγικές διαχείρισης πακέτων. Επισημάναμε την σημασία της ισορροπίας στην πλεονάζουσα πληροφορία, καθώς υπερβολική πλεονάζουσα πληροφορία μπορεί να επηρεάσει αρνητικά την απόδοση του συστήματος, αντίστοιχα και η ελάχιστη επιδρά αρνητικά στην απόδοση του. Επιπλέον, παρατηρήσαμε ότι όσο πιο συχνά στέλνουμε πλεονάζουσα πληροφορία, τόσο καλύτερη είναι η απόδοση στην εν σειρά καθυστέρηση και τον πίνακα αποκωδικοποίησης.

Συγκρίνοντας τις δύο μεθόδους διαχείρισης πακέτων, προέκυψε ότι η drop out of order υπερτερεί σημαντικά έναντι της drop generation, παρέχοντας βελτιωμένη απόδοση, ακόμη και σε περιπτώσεις με λιγότερη πλεονάζουσα πληροφορία.

Τέλος, συγκρίνοντας την k-rated block-based RLNC με την G-rated block-based RLNC, παρατηρούμε σημαντική βελτίωση στο μέσο μέγεθος του πίνακα αποκωδικοποίησης και στη μέση εν σειρά καθυστέρηση από την k-rated, που στέλνει τα κωδικοποιημένα πακέτα ενδιάμεσα από τα πηγαία, σε σύγκριση με την G-rated. Τέλος, και οι 2 προσεγγίσεις έχουν το ίδιο Goodput.

5.1 Μελλοντική Έρευνα

Μια ενδιαφέρουσα κατεύθυνση για μελλοντική έρευνα θα μπορούσε να εστιάσει στην ενσωμάτωση της πλήρους διανυσματικής (full vector) κωδικοποίησης στην k-rated block-based RLNC προσέγγισή μας. Η ενσωμάτωση αυτής της τεχνικής θα περιλαμβάνει τη σύγκριση των προσεγγίσεων με την k-rated block-based RLNC ώστε να διαπιστώσουμε σε ποιες περιπτώσεις κάθε μία από αυτές επιδεικνύει καλύτερη

απόδοση. Μια άλλη ενδιαφέρουσα κατεύθυνση θα μπορούσε να επικεντρωθεί στην ανάπτυξη και αξιολόγηση της k -rated block-based RLNC σε περιβάλλοντα με μεταβαλλόμενες συνθήκες όπως η εμφάνιση ριπών απώλειας (burst of errors). Τέλος, θα μπορούσε να γίνει αξιολόγηση της απόδοσης της με την ενσωμάτωση μιας τεχνικής recoding σε ενδιάμεσους κόμβους δικτύου.

Βιβλιογραφία

- [1] Forward Error Correction. Error detection and correction — Wikipedia, the free encyclopedia, February 2024 .
- [2] S. Wunderlich, F. Gabriel, S. Pandi, Frank H.P. Fitzek, and Martin Reisslein. Caterpillar RLNC (CRLNC): A practical finite sliding window RLNC approach. IEEE Access, 2017.
- [3] F. Karetsi and E. Papapetrou, “A low complexity network-coded ARQ protocol for ultra-reliable low latency communication,” , June 2021.
- [4] S. Pandi, F. Gabriel, J. A. Cabrera, S. Wunderlich, M. Reisslein, and F. H. Fitzek, “Pace: Redundancy engineering in RLNC for low-latency communication,” IEEE Access, 2017.
- [5] J. Heide, Frank H.P. Fitzek and T. Larsen “Network Coding for Mobile-Devices-Systematic Binary Random Rateless Codes” , June 2009
- [6] Evangelos Papapetrou. Lecture notes in computer networks, reliable communication over a link, December 2023.
- [7] Mr. Shahzad, Rashid Ali, Amir Haider Malik, Hyung Seok Kim RS-RLNC: A Reinforcement Learning-Based Selective Random Linear Network Coding Framework for Tactile Internet, January 2023
- [8] Ioannis Chatzigeorgiou and Andrea Tassi. Decoding delay performance of random linear network coding for broadcast. IEEE Transactions on Vehicular Technology, August 2017.

- [9] Simon Wunderlich, Frank H. P. Fitzek and Martin Reisslein. "Progressive Multicore RLNC Decoding with Online DAG Scheduling of Parallel Matrix Block Operations, November 2019.
- [10] Christina Fragouli, Jean-Yves Le Boudec, and Jörg Widmer. Network coding: an instant primer. ACM SIGCOMM Computer Communication Review, 2006.
- [11] Tracey Ho, Muriel Medard, Ralf Koetter and David Ron Karger "A random linear network coding approach to multicast," IEEE Transactions on Information Theory, 2006.
- [12] PyErasure Documentation [Online]. Available: <https://pyerasure.steinwurf.com/1.3.0/>, July 2023
- [13] M. Bennis, M. Debbah, and H. V. Poor, "Ultrareliable and low-latency wireless communication: Tail, risk, and scale", Proceedings of the IEEE, October 2018.

- [14] Elena Neira. Network Coding for 5G Visualization [Online]. Available: <http://elenaneira.com/cloud/5g-virtualization-mit-network-coding/>, February 2024.
- [15] Frank H.P. Fitzek, Shye-Chen Li, Martin Reisslein. Tactile Internet, Communication and Control, March 6 2021
- [16] R. Bassoli, H. Marques, J. Rodriguez, K. W. Shum, and R. Tafazolli, “Network coding theory: A survey,” IEEE Commun. Surveys Tuts., 2013.
- [17] M. Z. Farooqi, S. M. Tabassum, M. H. Rehmani, and Y. Saleem, “A survey on network coding: From traditional wireless networks to emerging cognitive radio networks,” J. Netw. Comput, November 2014.