



Σύγκριση απόδοσης εξελιγμένων τεχνικών κωδικοποίησης δικτύου

Διδασκάλου Γεώργιος

Διπλωματική Εργασία

Επιβλέπων: Παπαπέτρου Ευάγγελος

Ιωάννινα, Φεβρουάριος, 2025

ΤΜΗΜΑ ΜΗΧ. Η/Υ & ΠΛΗΡΟΦΟΡΙΚΗΣ

ΠΑΝΕΠΙΣΤΗΜΙΟ ΙΩΑΝΝΙΝΩΝ

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

UNIVERSITY OF IOANNINA

Ευχαριστίες

Ευχαριστώ πολύ τη μητέρα μου που με στήριξε καθ' όλη τη διάρκεια των ακαδημαϊκών σπουδών μου και τον επιβλέποντα καθηγητή μου κύριο Παπαπέτρου για την καθοδήγησή του στην διπλωματική μου εργασία.

Φεβρουάριος, 2025

Διδασκάλου Γεώργιος

Περίληψη

Η παρούσα εργασία πραγματοποιείται με αντικείμενο τη μελέτη της απόδοσης τεχνικών κωδικοποίησης δικτύου. Συγκεκριμένα συγκρίθηκε η τυχαία γραμμική κωδικοποίηση δικτύου (*Random Linear Network Coding – RLNC*) με τον Batched Sparse κώδικα (*BATS*). Χρησιμοποιήθηκε ως σημείο αναφοράς η τεχνική γραμμικής κωδικοποίησης Luby Transform (LT) με σκοπό την ανάδειξη των πλεονεκτημάτων της κωδικοποίησης δικτύου. Η υλοποίηση και η πειραματική αξιολόγηση των αλγορίθμων έγινε μέσω του εργαλείου PyErasure. Εφαρμόζοντας τα πειράματα που εξηγήθηκαν αναλυτικά, διαπιστώνεται ότι οι RLNC κώδικες πετυχαίνουν καλύτερα αποτελέσματα από τους LT κώδικες σε όλα τα σενάρια. Οι BATS κώδικες, από την άλλη, είναι ιδιαίτερα σταθεροί ακόμα και όταν η πιθανότητα απωλειών στο κανάλι αυξάνεται, μιας και η απόδοσή τους δεν μειώνεται τόσο πολύ.

Λέξεις Κλειδιά: <Κωδικοποίηση Δικτύου>, <PyErasure>, <RLNC>, <LT>, <BATS>

Abstract

The present study is carried out to study the performance of network coding techniques. Specifically, Random Linear Network Coding (*RLNC*) was compared with Batched Sparse Code (*BATS*). Luby Transform (*LT*) linear coding technique was also used as a benchmark to highlight the advantages of network coding. The implementation and experimental evaluation of the algorithms was carried out by PyErasure tool. By applying the experiments that were explained in detail, it is found that RLNC codes achieve better results than LT codes in all scenarios. BATS codes, on the other hand, are highly stable even when the probability of channel losses increases, since their performance does not decrease so much.

Keywords: <Network Coding>, <PyErasure>, <RLNC>, <LT>, <BATS>

ΠΕΡΙΕΧΟΜΕΝΑ

| | |
|---|----|
| Κεφάλαιο 1. Εισαγωγή..... | 11 |
| 1.1 Γενικές έννοιες..... | 11 |
| 1.2 Σκοπός και στόχοι εργασίας..... | 13 |
| 1.3 Δομή εργασίας..... | 13 |
| Κεφάλαιο 2. Βιβλιογραφική Ανασκόπηση..... | 14 |
| 2.1 Βασικές έννοιες κωδίκων..... | 16 |
| 2.2 Linear Network Coding (LNC)..... | 21 |
| 2.2.1 Πλεονεκτήματα LNC..... | 23 |
| 2.2.2 Βασικοί ορισμοί τυχαίας γραμμικής κωδικοποίησης δικτύου..... | 25 |
| 2.2.3 Βασικοί ορισμοί batched sparse κωδίκων..... | 27 |
| Κεφάλαιο 3. Ένα πλαίσιο για την αξιολόγηση της κωδικοποίησης δικτύου..... | 29 |
| 3.1 PyErasure..... | 29 |
| 3.2 Random Linear Network Codes (RLNC)..... | 30 |
| 3.3 Luby Transform Codes (LT)..... | 34 |
| 3.4 Batched Sparse Codes (BATS)..... | 41 |
| Κεφάλαιο 4. Πειραματική αξιολόγηση..... | 47 |
| 4.1 Τοπολογίες και χαρακτηριστικά τους..... | 47 |
| 4.2 Σύγκριση αλγορίθμων..... | 48 |
| 4.3 Μετρικές αξιολόγησης..... | 49 |
| 4.4 Βασικοί παράμετροι..... | 52 |
| 4.5 Κανάλι με ομοιόμορφες απώλειες..... | 52 |
| 4.5.1 Κωδικοποίηση χωρίς ρυθμό (rateless)..... | 52 |
| 4.5.2 Κωδικοποίηση με ρυθμό (rate-based)..... | 54 |
| 4.5.3 Η επίδραση του ρυθμού κωδικοποίησης..... | 55 |
| 4.6 Κανάλι με απώλειες σε ριπές (Gilbert-Elliot)..... | 57 |
| 4.6.1 Κωδικοποίηση χωρίς ρυθμό (rateless)..... | 57 |
| 4.6.2 Κωδικοποίηση με ρυθμό (rate-based)..... | 58 |

| | | |
|-------------------|--|----|
| 4.6.3 | Η επίδραση του ρυθμού κωδικοποίησης | 61 |
| 4.7 | Τοπολογία δύο διαδοχικών καναλιών με ομοιόμορφες απώλειες | 62 |
| 4.7.1 | Κωδικοποίηση χωρίς ρυθμό (rateless) | 62 |
| 4.7.2 | Κωδικοποίηση με ρυθμό (rate-based)..... | 63 |
| 4.7.3 | Η επίδραση του ρυθμού κωδικοποίησης | 64 |
| 4.8 | Τοπολογία δύο διαδοχικών καναλιών με απώλειες σε ριπές (Gilbert-Elliot) .. | 65 |
| 4.8.1 | Κωδικοποίηση χωρίς ρυθμό (rateless) | 65 |
| 4.8.2 | Κωδικοποίηση με ρυθμό (rate-based)..... | 67 |
| 4.8.3 | Η επίδραση του ρυθμού κωδικοποίησης | 69 |
| Κεφάλαιο 5. | Συμπεράσματα..... | 72 |
| Βιβλιογραφία..... | | 75 |

ΚΑΤΑΛΟΓΟΣ ΕΙΚΟΝΩΝ

| | |
|---|-----|
| Εικόνα 1 - Απλή κωδικοποίηση με τη χρήση XOR..... | 15 |
| Εικόνα 2 - Γραφική αναπαράσταση της διαδικασίας κωδικοποίησης και αποκωδικοποίησης..... | 19 |
| Εικόνα 3 - Λογότυπο PyErasure..... | 29 |
| Εικόνα 4 - Γράφος Tanner για τον εξωτερικό κώδικα BATS. Οι κόμβοι στη πρώτη σειρά αποτελούν τους κόμβους μεταβλητών που αντιστοιχούν στα αρχικά πακέτα. Οι κόμβοι στη δεύτερη σειρά αντιστοιχούν στους κόμβους ελέγχου που παράγονται στον εξωτερικό κώδικα | 422 |
| Εικόνα 5 - Ένας γράφος αποκωδικοποίησης. Οι κόμβοι στη πρώτη γραμμή αποτελούν τους κόμβους πληροφορίας και αναπαριστούν τα αρχικά πακέτα. Οι κόμβοι στη δεύτερη γραμμή είναι οι κόμβοι ελέγχου που αποτελούν την παρτίδα..... | 444 |
| Εικόνα 6 - Single-Hop σενάριο | 477 |
| Εικόνα 7 - Multi-Hop σενάριο..... | 488 |
| Εικόνα 8 - Αποτελέσματα τεχνικών στη rateless προσέγγιση..... | 533 |
| Εικόνα 9 - Αποτελέσματα τεχνικών στη rate-based προσέγγιση | 555 |
| Εικόνα 10 - Αποτελέσματα τεχνικών μεταβάλλοντας το code rate..... | 566 |
| Εικόνα 11 - Αποτελέσματα τεχνικών σε rateless προσέγγιση, διάρκεια ριπών ίση με 4. | 577 |
| Εικόνα 12 - Αποτελέσματα τεχνικών σε rateless προσέγγιση με διάρκεια ριπών ίση με 8. | 588 |
| Εικόνα 13 - Αποτελέσματα τεχνικών σε rate-based προσέγγιση με διάρκεια ριπών ίση με 4 | 59 |
| Εικόνα 14 - Αποτελέσματα τεχνικών σε rate based προσέγγιση με διάρκεια ριπών ίση με 8 | 600 |
| Εικόνα 15 - Αποτελέσματα τεχνικών μεταβάλλοντας το code rate..... | 611 |
| Εικόνα 16 - Αποτελέσματα τεχνικών στη rateless προσέγγιση | 622 |
| Εικόνα 17 - Αποτελέσματα τεχνικών στη rate-based προσέγγιση | 644 |
| Εικόνα 18 - Αποτελέσματα τεχνικών μεταβάλλοντας το code rate..... | 655 |
| Εικόνα 19 - Αποτελέσματα τεχνικών σε rateless προσέγγιση με διάρκεια ριπών ίση με 4. | 666 |
| Εικόνα 20 - Αποτελέσματα τεχνικών σε rateless προσέγγιση, διάρκεια ριπών ίση με 8. | 666 |
| Εικόνα 21 - Αποτελέσματα τεχνικών σε rate-based προσέγγιση, διάρκεια ριπών ίση με 4. | 677 |

| | |
|---|----|
| Εικόνα 22 - Αποτελέσματα τεχνικών σε rate-based προσέγγιση με διάρκεια ριπών ίση με 8. | 68 |
| Εικόνα 23 - Αποτελέσματα τεχνικών μεταβάλλοντας το code rate..... | 69 |

ΚΑΤΑΛΟΓΟΣ ΠΙΝΑΚΩΝ

| | |
|---|-----|
| Πίνακας 1 - Αλγόριθμος κωδικοποίησης του LT κώδικα..... | 367 |
| Πίνακας 2 - Αλγόριθμος αποκωδικοποίησης του LT κώδικα | 38 |

Κεφάλαιο 1. Εισαγωγή

1.1 Γενικές έννοιες

Η κωδικοποίηση δικτύου είναι μια τεχνική μετάδοσης δεδομένων όπου τα πακέτα δεδομένων αναμειγνύονται (κωδικοποιούνται) σε ενδιάμεσους κόμβους μέσα σε ένα δίκτυο πριν μεταδοθούν στον επόμενο προορισμό. Σε αντίθεση με την παραδοσιακή δρομολόγηση, η οποία χειρίζεται τα πακέτα δεδομένων ανεξάρτητα, η κωδικοποίηση δικτύου επιτρέπει στους κόμβους να συνδυάζουν πακέτα αλγεβρικά, προσφέροντας σημαντικές βελτιώσεις όσον αφορά τη διεκπεραιωτική ικανότητα και την αξιοπιστία του δικτύου.

Η βασική ιδέα της κωδικοποίησης δικτύου εισήχθη από τους Ahlswede et al. [AC++00] στο θεμελιώδες έργο "Network Information Flow." Οι ερευνητές αυτοί απέδειξαν ότι επιτρέποντας σε ενδιάμεσους κόμβους να κωδικοποιούν δεδομένα, η μέγιστη χωρητικότητα πολλαπλής διανομής ενός δικτύου θα μπορούσε να επιτευχθεί πιο αποτελεσματικά σε σύγκριση με τις παραδοσιακές μεθόδους αποθήκευσης και προώθησης. Αυτή η προσέγγιση αποκλίνει από τα παραδοσιακά παραδείγματα δρομολόγησης αντιμετωπίζοντας τα πακέτα δεδομένων ως στοιχεία ενός μαθηματικού πεδίου και χρησιμοποιώντας αλγεβρικές πράξεις για τον συνδυασμό και τη μετάδοσή τους. Η ιδέα υποστηρίζεται από τη γραμμική άλγεβρα σε πεπερασμένα πεδία. Όταν ένας κόμβος συνδυάζει πακέτα δεδομένων, χρησιμοποιεί ένα σύνολο συντελεστών κωδικοποίησης που καθορίζουν τον τρόπο με τον οποίο αναμειγνύονται τα πακέτα. Στην πλευρά του δέκτη, τα αρχικά πακέτα μπορούν να αποκωδικοποιηθούν χρησιμοποιώντας γραμμικούς μετασχηματισμούς, υπό την προϋπόθεση ότι λαμβάνονται αρκετοί ανεξάρτητοι συνδυασμοί.

Η κωδικοποίηση δικτύου είναι μια τεχνική με ισχυρά πλεονεκτήματα. Ένα από αυτά είναι η αυξημένη απόδοση. Συγκεκριμένα, μέσω της κωδικοποίησης μπορεί να επιτευχθεί υψηλότερη απόδοση, ειδικά σε σενάρια πολλαπλής διανομής. Για παράδειγμα, σε ένα δίκτυο όπου πολλοί δέκτες απαιτούν τα ίδια δεδομένα, η κωδικοποίηση δικτύου μειώνει τον πλεονασμό συνδυάζοντας πακέτα, επιτρέποντας σε όλους τους δέκτες να ανασυνθέσουν τα δεδομένα χρησιμοποιώντας λιγότερες μεταδόσεις. Επίσης, με την κωδικοποίηση δεδομένων, εισάγεται ένας πλεονασμός που καθιστά το σύστημα

ανθεκτικό στις απώλειες πακέτων. Αυτό είναι ιδιαίτερα ωφέλιμο σε ασύρματα και επιρρεπή σε σφάλματα δίκτυα. Ακόμη, σε σενάρια με πολλαπλές διαδρομές, η κωδικοποίηση δικτύου μπορεί να καταναίμει το φορτίο πιο ομοιόμορφα σε όλο το δίκτυο, βελτιώνοντας τη συνολική απόδοση και αποτρέποντας τα σημεία συμφόρησης. Τέλος, η κωδικοποίηση δικτύου παρέχει εγγενώς κάποιο επίπεδο ασφάλειας, καθώς τα κωδικοποιημένα πακέτα δεν αποκαλύπτουν άμεσα τα αρχικά δεδομένα, απαιτώντας αποκωδικοποίηση στον δέκτη [HM++06].

Η κωδικοποίηση δικτύου βρίσκει εφαρμογή σε μια σειρά από περιπτώσεις της καθημερινότητας μας. Ένα παράδειγμα είναι τα ασύρματα δίκτυα. Η κωδικοποίηση δικτύου ενισχύει σημαντικά τη χωρητικότητα των ασύρματων δικτύων εκμεταλλευόμενη τις ευκαιρίες μετάδοσης. Για παράδειγμα, η ευκαιριακή κωδικοποίηση δικτύου, όπως καταδεικνύεται στο σύστημα COPE [KR++08], έδειξε πως η κωδικοποίηση μπορεί να βελτιστοποιήσει την απόδοση σε ασύρματα δίκτυα πραγματικού κόσμου. Επίσης χρησιμοποιείται για τον μετριασμό της υψηλής καθυστέρησης και της επιρρεπούς σε σφάλματα φύσης των δορυφορικών συνδέσεων. Ακόμη, στα δίκτυα παράδοσης περιεχομένου, η κωδικοποίηση δικτύου επιτρέπει την αποτελεσματική διανομή περιεχομένου, μειώνοντας την καθυστέρηση και βελτιώνοντας τη διαθεσιμότητα δεδομένων. Τέλος, οι κώδικες απαλοιφής (*erasure codes*), που αποτελούν μια μορφή κωδικοποίησης δικτύου, χρησιμοποιούνται ευρέως σε συστήματα κατανομημένης αποθήκευσης όπως το Google File System και το Hadoop Distributed File System για να διασφαλιστεί η αξιοπιστία των δεδομένων.

Όμως, εκτός από τα παραπάνω, η κωδικοποίηση περιλαμβάνει διάφορους περιορισμούς και προκλήσεις. Οι διαδικασίες κωδικοποίησης και αποκωδικοποίησης απαιτούν υπολογιστικούς πόρους, οι οποίοι μπορεί να μην είναι εφικτοί για συσκευές με περιορισμούς πόρων. Επίσης, η υλοποίηση της κωδικοποίησης δικτύου σε δυναμικά και μεγάλης κλίμακας δίκτυα εμπεριέχει προκλήσεις στο σχεδιασμό και την επεκτασιμότητα του πρωτοκόλλου.

Υπάρχουν διάφορες τεχνικές κωδικοποίησης, καθεμία από τις οποίες λειτουργεί με διαφορετικό τρόπο, πετυχαίνοντας έτσι διαφορετικά αποτελέσματα. Για παράδειγμα η γραμμική κωδικοποίηση συνδυάζει δεδομένα πριν από τη μετάδοση, ενώ οι κώδικες προώθησης και διόρθωσης σφαλμάτων (*Forward Error Correction – FEC*) επιτρέπουν την ανίχνευση και διόρθωση σφαλμάτων χωρίς επανεκπομπή.

1.2 Σκοπός και στόχοι εργασίας

Η αξιόπιστη και χαμηλής καθυστέρησης μετάδοση (*ultra reliable low latency communication - urllc*) είναι ιδιαίτερα σημαντική σε εφαρμογές που απαιτούν άμεση απόκριση και υψηλή αξιοπιστία. Στην παρούσα εργασία θα μελετήσουμε την απόδοση δυο τεχνικών κωδικοποίησης δικτύου, την τυχαία γραμμική κωδικοποίηση δικτύου και τους batched sparse κώδικες. Ως σημείο αναφοράς θα χρησιμοποιήσουμε την τεχνική γραμμικής κωδικοποίησης Luby Transform, που αποτελεί βασική τεχνική γραμμικής κωδικοποίησης, αλλά όχι κωδικοποίησης δικτύου. Έτσι θα μπορέσουμε να διακρίνουμε ποια από τα αποτελέσματα των πειραμάτων οφείλονται στη γραμμική κωδικοποίηση γενικότερα και ποια στην κωδικοποίηση δικτύου που χρησιμοποιούν οι RLNC και BATS κώδικες ειδικότερα. Για την υλοποίηση των τεχνικών αυτών και την πειραματική τους αξιολόγηση θα χρησιμοποιήσουμε το εργαλείο PyErasure της εταιρείας Steinwurf.

Και ενώ η λειτουργία του RLNC αλλά και του BATS κώδικα έχει μελετηθεί από πολλούς ερευνητές, υπάρχουν κενά στη βιβλιογραφία σχετικά με τη σύγκριση των δύο αυτών τεχνικών και τα οφέλη τους σε διάφορους τύπους δικτύων. Σκοπός της παρούσας διπλωματικής εργασίας είναι να παραθέσουμε πειράματα που να δείχνουν πώς επιδρά η κάθε κωδικοποίηση σε διάφορα σενάρια δικτύων, στα πλαίσια υπερβολικά αξιόπιστης και χαμηλής καθυστέρησης μετάδοση.

1.3 Δομή εργασίας

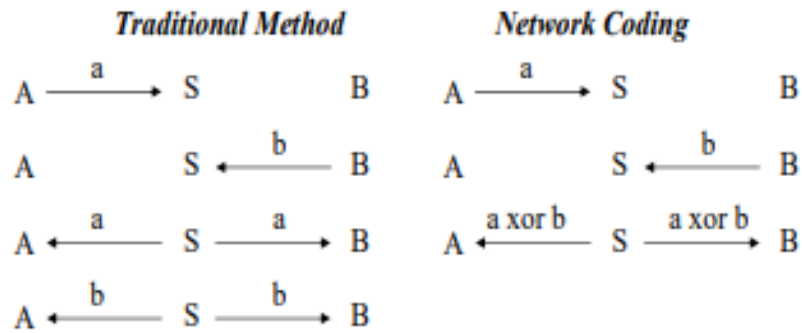
Η συνέχεια της εργασίας αυτής οργανώνεται σε τέσσερα κεφάλαια. Στο επόμενο κεφάλαιο αναλύονται οι βασικές έννοιες της κωδικοποίησης και δίνονται οι ορισμοί των κωδίκων που θα μελετήσουμε. Το τρίτο κεφάλαιο αφορά στο πλαίσιο για την αξιολόγηση της κωδικοποίησης δικτύου. Συγκεκριμένα περιλαμβάνει τη βασική δομή της βιβλιοθήκης PyErasure που χρησιμοποιήθηκε, καθώς και την περιγραφή του τρόπου λειτουργίας των κωδίκων που υλοποιήθηκαν. Το τέταρτο κεφάλαιο περιλαμβάνει την πειραματική μεθοδολογία, τα αποτελέσματα των πειραμάτων καθώς και το σχολιασμό των αποτελεσμάτων αυτών. Στο τελευταίο κεφάλαιο συνοψίζουμε τα αποτελέσματα και καταγράφουμε τα συμπεράσματα της διπλωματικής εργασίας μετάδοση.

Κεφάλαιο 2. Βιβλιογραφική

Ανασκόπηση

Η κωδικοποίηση δικτύου είναι μια προσέγγιση στη μετάδοση δεδομένων, που αποκλίνει σημαντικά από τα παραδοσιακά παραδείγματα δρομολόγησης. Στην παραδοσιακή δρομολόγηση, τα πακέτα δεδομένων αντιμετωπίζονται ανεξάρτητα, μεταδίδονται κόμβο προς κόμβο και προωθούνται χωρίς τροποποίηση. Εάν πολλοί παραλήπτες απαιτούν τα ίδια δεδομένα και χαθεί κάποιο στην πορεία για έναν από αυτούς, θα πρέπει να αποσταλεί ξανά το αρχικό πακέτο αυτούσιο, οδηγώντας σε αναποτελεσματικότητα, ειδικά σε σενάρια με κοινόχρηστους πόρους δικτύου. Η κωδικοποίηση δικτύου, αντίθετα, επιτρέπει στον κόμβο πηγής να συνδυάζει και να κωδικοποιεί πολλαπλά πακέτα δεδομένων αλγεβρικά πριν τα μεταδώσει στον επόμενο κόμβο. Αυτή η διαδικασία κωδικοποίησης μειώνει σημαντικά τον πλεονασμό, βελτιώνει την απόδοση και ενισχύει την ευρωστία έναντι της απώλειας πακέτων.

Η διαδικασία κωδικοποίησης δικτύου περιλαμβάνει τρία θεμελιώδη στάδια: κωδικοποίηση, μετάδοση και αποκωδικοποίηση. Η κωδικοποίηση λαμβάνει χώρα τόσο στον κόμβο πηγής όσο σε ενδιάμεσους κόμβους, όπου τα εισερχόμενα πακέτα συνδυάζονται χρησιμοποιώντας αλγεβρικές πράξεις, συνήθως σε πεπερασμένα πεδία [FS07]. Για παράδειγμα, υπάρχει ένας κόμβος πηγής που μεταδίδει δύο πακέτα, το P_1 και το P_2 . Ένας ενδιάμεσος κόμβος μπορεί να κωδικοποιεί αυτά τα πακέτα ως $P_1 + P_2$ (όπου η λειτουργία πρόσθεσης είναι συνήθως XOR σε δυαδικά πεδία). Τα κωδικοποιημένα πακέτα μεταδίδονται στη συνέχεια στους επόμενους κόμβους ή δέκτες, μεταφέροντας πληροφορίες για πολλαπλά πρωτότυπα πακέτα σε μία μόνο μετάδοση.



Εικόνα 1 – Απλή κωδικοποίηση με τη χρήση XOR

Στο άκρο λήψης, οι κόμβοι αποκωδικοποιούν τα λαμβανόμενα κωδικοποιημένα πακέτα για να ανασυνθέσουν τα αρχικά δεδομένα. Η αποκωδικοποίηση βασίζεται στην επίλυση συστημάτων γραμμικών εξισώσεων που προέρχονται από τα κωδικοποιημένα πακέτα [FS07]. Εάν οι δέκτες έχουν πρόσβαση σε αρκετά ανεξάρτητα κωδικοποιημένα πακέτα, μπορούν να ανακτήσουν τα αρχικά δεδομένα χρησιμοποιώντας τεχνικές από τη γραμμική άλγεβρα. Για παράδειγμα, εάν ένας δέκτης έχει το πακέτο P_1 και το κωδικοποιημένο πακέτο $P_1 + P_2$, μπορεί να υπολογίσει το P_2 αφαιρώντας το P_1 από το κωδικοποιημένο πακέτο. Αυτή η ικανότητα ανάκτησης αρχικών πακέτων από κωδικοποιημένα δεδομένα δείχνει τη δύναμη της κωδικοποίησης δικτύου στη μείωση των περιττών μεταδόσεων και στη βελτίωση της αποδοτικότητας του δικτύου.

Μία από τις κλασικές επιδείξεις της αποτελεσματικότητας της κωδικοποίησης δικτύου είναι το «δίκτυο πεταλούδων», που εισήχθη από τους Ahlswede et al. [AC++00]. Σε αυτήν τη ρύθμιση, ένας κόμβος μεμονωμένης πηγής μεταδίδει δύο πακέτα P_1 και P_2 σε δύο κόμβους προορισμού μέσω ενός κοινόχρηστου ενδιάμεσου κόμβου. Η παραδοσιακή δρομολόγηση απαιτεί τέσσερις μεταδόσεις για να διασφαλιστεί ότι και οι δύο δέκτες λαμβάνουν και τα δύο πακέτα. Ωστόσο, με την κωδικοποίηση δικτύου, ο ενδιάμεσος κόμβος κωδικοποιεί τα πακέτα ως $P_1 + P_2$, επιτρέποντας και στους δύο κόμβους προορισμού να ανασυνθέσουν τα αρχικά δεδομένα με τρεις μόνο μεταδόσεις. Αυτό το κέρδος απόδοσης υπογραμμίζει πως η κωδικοποίηση δικτύου μπορεί να βελτιστοποιήσει τη χρήση πόρων σε σενάρια πολλαπλής διανομής.

Στη συνέχεια του κεφαλαίου θα παρουσιάσουμε τις βασικές έννοιες των κωδίκων και την γραμμική κωδικοποίηση (*Linear Network Coding*) εξηγώντας και δύο βασικές τεχνικές που θα εξετάσουμε στη διπλωματική.

2.1 Βασικές έννοιες κωδίκων

Ο όρος πομπός αναφέρεται στον αρχικό κόμβο που πρόκειται να στείλει δεδομένα προς έναν άλλο κόμβο του δικτύου. Στον πομπό κωδικοποιούνται πρώτη φορά τα αρχικά δεδομένα πριν σταλούν στον επόμενο κόμβο. Ο κόμβος παραλήπτης είναι ο τελικός κόμβος στον οποίο θα γίνει η διαδικασία της αποκωδικοποίησης ώστε να πραγματοποιηθεί η ανάκτηση των αρχικών δεδομένων και ονομάζεται δέκτης.

Υπάρχουν δύο βασικοί τρόποι με τους οποίους μπορεί ο πομπός να στείλει τα δεδομένα στον δέκτη. Ένας από αυτούς είναι η κωδικοποίηση χωρίς ρυθμό (*rateless*) κατά την οποία δεν υπάρχει προκαθορισμένο πλήθος πακέτων που θα στείλει ο πομπός. Με αυτόν τον τρόπο παράγονται συνέχεια κωδικοποιημένα πακέτα τα οποία στέλνονται προς τον δέκτη. Η αποστολή κωδικοποιημένων πακέτων σταματάει όταν ο δέκτης έχει λάβει επαρκή αριθμό από αυτά για να ανακτήσει τα αρχικά δεδομένα. Μία *rateless* προσέγγιση είναι πολύ χρήσιμη σε δίκτυα που μεταβάλλονται συχνά ή δεν γνωρίζουμε πολλές πληροφορίες για τα τις απώλειες που επικρατούν στο κανάλι.

Ο δεύτερος τρόπος με τον οποίο μπορούμε να εφαρμόσουμε την κωδικοποίηση είναι με ρυθμό (*rate-based*). Σε αυτή την τεχνική καθορίζεται εξ αρχής το πλήθος των κωδικοποιημένων πακέτων που θα παράξει ο πομπός και θα στείλει. Αυτή η προσέγγιση είναι καλύτερη σε δίκτυα που δεν μεταβάλλονται πολύ και σε περιπτώσεις που έχουμε γνώση για τις απώλειες που επικρατούν στο κανάλι. Σημαντικός παράγοντας στην επίδοση μίας *rate-based* προσέγγισης είναι ο ρυθμός κωδικοποίησης (*code rate*). Αυτός καθορίζει το πλήθος των κωδικοποιημένων πακέτων που θα στείλει ο πομπός στον δέκτη και δίνεται από τη σχέση $c = \frac{K}{N}$, όπου K είναι το πλήθος των αρχικών πακέτων που θέλει να στείλει ο πομπός και N είναι το συνολικό πλήθος των πακέτων που θα σταλούν. Όσο πιο κοντά στη μονάδα είναι το *code rate* τόσο λιγότερα κωδικοποιημένα πακέτα θα στείλει ο πομπός.

Πριν ξεκινήσουμε την εξήγηση των τεχνικών κωδικοποίησης που μελετήθηκαν στη παρούσα διπλωματική, πρέπει να εισάγουμε και την έννοια του καναλιού απαλοιφής (*erasure channel*). Η επικοινωνία των υπολογιστών παραδοσιακά βασιζόταν στο πρωτόκολλο TCP (*Transmission Control Protocol*), το οποίο επιτυγχάνει μία αξιόπιστη μετάδοση δεδομένων. Το TCP χωρίζει τα δεδομένα που θέλουμε να στείλουμε σε μικρότερα πακέτα εισάγοντας σε καθένα από αυτά έναν αριθμό ακολουθίας και τα στέλνει προς τον δέκτη. Για να είναι επιτυχής η αποστολή των δεδομένων στον δέκτη, θα πρέπει αυτός να τα λάβει με τη σωστή σειρά. Για αυτόν τον λόγο, το πρωτόκολλο TCP

χρησιμοποιεί και μικρά πακέτα επιβεβαίωσης (*acknowledgment - ack*). Οπότε κάθε φορά που λαμβάνει ένα πακέτο, πρέπει να στέλνει το αντίστοιχο πακέτο επιβεβαίωσης πίσω στον πομπό. Υπάρχει ένα συγκεκριμένο χρονικό διάστημα, στο οποίο αν δεν λάβει ο πομπός αυτό το πακέτο επιβεβαίωσης, τότε θεωρεί ότι το μήνυμα αυτό έχει χαθεί και φροντίζει να το ξαναστείλει στον δέκτη. Με αυτό τον τρόπο πετυχαίνει μεγάλη αξιοπιστία στη μετάδοση δεδομένων [S03].

Ας σκεφτούμε όμως τα ασύρματα δίκτυα με τα οποία επικοινωνούν δεκάδες χιλιάδες συσκευές ταυτόχρονα. Το πρωτόκολλο TCP, απαιτεί συνεχή επικοινωνία και αναμετάδοση των πακέτων που χάνονται, τα οποία σε τέτοιου είδους δίκτυα ενδέχεται να είναι πολλά λόγω των παρεμβολών και του θορύβου που υπάρχει. Αυτό θα έχει σαν αποτέλεσμα να σπαταλάται σημαντικό μέρος του εύρους ζώνης του καναλιού και έτσι να μειώνεται η διαθέσιμη ταχύτητα που μπορούν να πετύχουν οι συσκευές. Ακόμα, επειδή θα πρέπει ο πομπός να στέλνει συνεχώς τα πακέτα επιβεβαίωσης, απαιτείται να χρησιμοποιούνται συνέχεια κανάλια διπλής κατεύθυνσης. Λόγω των προβλημάτων που προκύπτουν από τη χρήση του TCP πρωτοκόλλου, έχουν ανακαλυφθεί τα κανάλια απαλοιφής [S03;BS14].

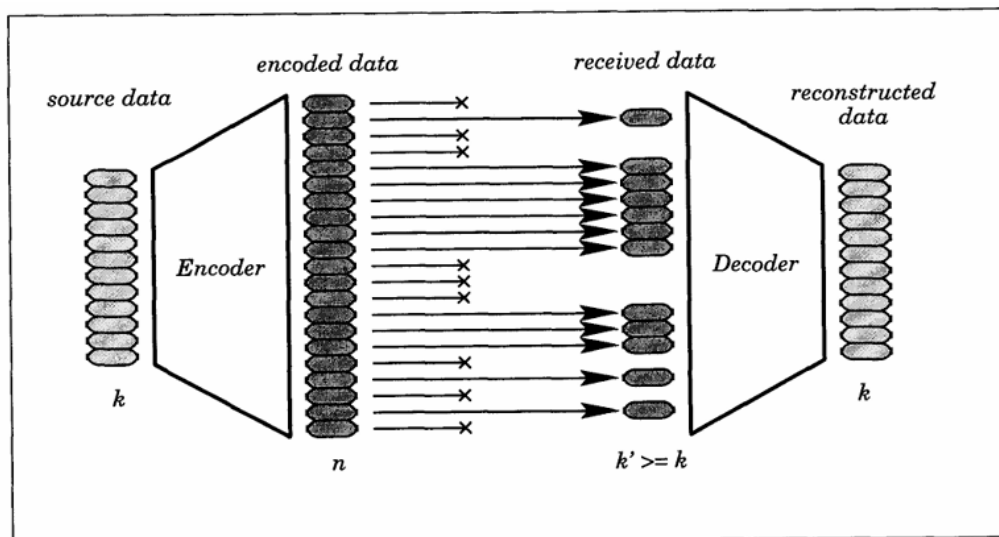
Κανάλι απαλοιφής ονομάζουμε τη σύνδεση δύο συσκευών, στην οποία χωρίζονται τα δεδομένα σε πακέτα τα οποία ο πομπός στέλνει προς τον δέκτη με μία πιθανότητα σφάλματος $p \in (0, 1)$. Αυτό σημαίνει ότι ένα πακέτο έχει πιθανότητα $1 - p$ να παραδοθεί στον δέκτη και πιθανότητα p να χαθεί. Σε αυτά τα κανάλια δεν χρειάζεται ο δέκτης να λάβει όλα τα πακέτα στη σειρά για να είναι επιτυχής η αποστολή του αρχείου, όπως συμβαίνει με το TCP πρωτόκολλο. Υπάρχει η δυνατότητα αποστολής μόνο του μέρους που λείπει από τον δέκτη, αν χρησιμοποιήσουμε ένα κανάλι ανάδρασης για να επικοινωνεί ο δέκτης με τον πομπό. Κάτι τέτοιο όπως καταλαβαίνουμε από μόνο του αυξάνει το εύρος ζώνης που μπορούν να εκμεταλλευτούν οι συσκευές [S03].

Ωστόσο η χρήση ενός καναλιού απαλοιφής μπορεί να δημιουργήσει και προβλήματα, ιδιαίτερα αν δεν υπάρχει κάποιο κανάλι ανάδρασης για την επικοινωνία δέκτη και πομπό. Κάτι τέτοιο μπορεί να εντοπιστεί κατά την ενημέρωση λογισμικού στα αυτοκίνητα η οποία γίνεται μέσω δορυφόρου. Ο δορυφόρος καλείται να αποστείλει τα ίδια δεδομένα σε ένα μεγάλο πλήθος συστημάτων. Μερικά από τα αυτοκίνητα όμως μπορεί να βρίσκονται σε κάποια περιοχή όπου δεν υπάρχει επαρκής κάλυψη και για αυτό να μην μπορέσουν να εγκαταστήσουν την ενημέρωση επειδή δεν θα έχουν λάβει κάποια κομμάτια αυτής. Λόγω της απουσίας δυνατότητας επικοινωνίας μεταξύ του δέκτη και του πομπού, ο δορυφόρος θα αναγκαστεί να ξαναστείλει ολόκληρες τις ενημερώσεις έτσι ώστε τα οχήματα που δεν μπόρεσαν να τις ολοκληρώσουν να λάβουν το κομμάτι που είχε

χαθεί. Κάτι τέτοιο όμως όπως είναι προφανές αυξάνει σε σημαντικό βαθμό τον όγκο δεδομένων που στέλνονται.

Περιπτώσεις σαν την προηγούμενη, οι οποίες οδηγούν σε μείωση της αποδοτικότητας του δικτύου είναι πολλές. Για τον λόγο είναι αναγκαία η εφαρμογή μιας κωδικοποίησης των πακέτων που αποστέλλονται σε αυτά τα κανάλια απαλοιφής. Τέτοια κωδικοποίηση γίνεται μέσα από τα σχήματα Προώθησης και Διόρθωσης Σφαλμάτων (*Forward Error Correction - FEC*), τα οποία ειδικά σε περιπτώσεις ευρείας εκπομπής (*broadcast*) προσφέρουν σημαντική βελτίωση στην απόδοση [NB96]. Αυτό που κάνουν τα σχήματα αυτά είναι να συνδυάζουν τα αρχικά πακέτα που θέλει να στείλει ο δέκτης δημιουργώντας αυτά που ονομάζουμε κωδικοποιημένα πακέτα. Έτσι αφότου σταλούν τα αρχικά πακέτα, αποστέλλονται και τα κωδικοποιημένα ώστε να μπορέσει ο δέκτης να ανακτήσει της πληροφορία για μέρη του αρχείου που πιθανόν να έχουν χαθεί κατά τη μετάδοσή τους. Αυτό γίνεται μέσα από τη διαδικασία της αποκωδικοποίησης. Δεν υπάρχει κάποιο όριο στο πλήθος των κωδικοποιημένων πακέτων που μπορούν να δημιουργηθούν. Όσο περισσότερα παραχθούν, τόσο αυξάνεται η πιθανότητα να έχουμε επιτυχή λήψη όλων των πακέτων στον δέκτη, αλλά αυξάνεται παράλληλα και ο όγκος δεδομένων που θα στείλουμε. Οπότε θα πρέπει κάθε φορά, ανάλογα με τις συνθήκες του εκάστοτε καναλιού να θέτουμε εμείς ένα όριο στα κωδικοποιημένα μηνύματα που παράγονται.

Για να πετύχουμε τον στόχο των FEC σχημάτων χρησιμοποιούμε την κωδικοποίηση απαλοιφής (*Erasure Coding*). Η βασική ιδέα αυτής της μεθόδου είναι ότι αν θέλουμε να στείλουμε k πακέτα σε έναν δέκτη, θα παραχθούν $n > k$ κωδικοποιημένα πακέτα. Αυτό επιφέρει στο δίκτυο μία επιβάρυνση της τάξης $n-k$, όσα δηλαδή θα είναι τα πλεονάζοντα πακέτα που θα παραχθούν προς αποστολή. Με αυτή τη τεχνική όμως, ο δέκτης θα μπορέσει να ανακατασκευάσει, με τη διαδικασία της αποκωδικοποίησης, τα k αρχικά πακέτα αν σε αυτόν φτάσουν τουλάχιστον k από τα n κωδικοποιημένα πακέτα. Τα πακέτα που θα φτάσουν στον δέκτη μπορεί να είναι οποιαδήποτε από τα n , καθώς πλέον δεν στέλνονται αυτά με τη σειρά. [R97]



Εικόνα 1 - Γραφική αναπαράσταση της διαδικασίας κωδικοποίησης και αποκωδικοποίησης

Επομένως αυτή η τεχνική προσφέρει λύση στο πρόβλημα που επικρατεί στα παραδοσιακά δίκτυα όπου κάθε ένας από τους παραλήπτες έπρεπε να επικοινωνεί με τον πομπό και να του στέλνει ενημερώσεις για κάθε ένα από τα πακέτα που λαμβάνει ξεχωριστά. Τώρα παράγονται πακέτα που σε πλήθος είναι περισσότερα από τα αρχικά. Και επειδή αυτά είναι κωδικοποιημένα, περιέχουν πληροφορία για περισσότερα από ένα από τα αρχικά πακέτα. Το πλήθος των αρχικών πακέτων των οποίων περιέχουν πληροφορία θα δούμε στη συνέχεια του κεφαλαίου ότι εξαρτάται από την τεχνική που θα χρησιμοποιήσουμε για να εφαρμόσουμε κωδικοποίηση. Έτσι, δεν υπάρχει πλέον το πρόβλημα της μετάδοσης πακέτων επιβεβαίωσης και επανεκπομπής των αρχικών σε περίπτωση σφάλματος ή απώλειας στο δίκτυο, αλλά λαμβάνοντας κάποιο από τα επόμενα κωδικοποιημένα πακέτα θα αντισταθμιστεί αυτή η απώλεια. Και επειδή τα κωδικοποιημένα πακέτα περιέχουν πληροφορίες για πολλά αρχικά πακέτα, ο κάθε δέκτης θα μπορέσει να ανακτήσει αυτά που χάθηκαν κατά τη μετάδοση προς αυτόν, μιας και δεν θα χάσουν όλοι τα ίδια πακέτα [BL05;WK02].

Στη διπλωματική αυτή θα μελετήσουμε τους RLNC, LT και BATS κώδικες. Ωστόσο, αυτές είναι μόνο μερικές από τις διάφορες τεχνικές που πραγματοποιούν κωδικοποίηση απαλοιφής οι οποίες μπορούν να χωριστούν σε δύο ευρύτερες ομάδες:

- Μπλοκ Κώδικες Απαλοιφής (*Block Erasure Codes*) : Στους κώδικες αυτούς, τα πακέτα που θέλουμε να στείλουμε στον δέκτη, χωρίζονται σε k μπλοκ. Μέσα από τη διαδικασία της κωδικοποίησης παράγονται n μπλοκ κωδικοποιημένων συμβόλων. Η κωδικοποίηση γίνεται με τέτοιο τρόπο, έτσι ώστε οποιοδήποτε υποσύνολο από k κωδικοποιημένα block να αρκούν για να ανακτήσουμε τα αρχικά πακέτα. Αυτοί οι κώδικες αναφέρονται συχνά και σαν (n,k) κώδικες και

επιτρέπουν στο κανάλι να έχει $n-k$ απώλειες σε ένα γκρουπ από n κωδικοποιημένα μπλοκ. Πολύ σημαντικό ρόλο παίζει το code rate που θα εφαρμόσουμε. Για παράδειγμα, αν εφαρμόσουμε ρυθμό κωδικοποίησης $c = 0.5$ και χωρίσουμε τα αρχικά πακέτα σε 10 μπλοκ, τότε με αυτή τη κωδικοποίηση θα παραχθούν 20 μπλοκ κωδικοποιημένων πακέτων. Το κανάλι αυτό θα έχει ανοχή σε 10 απώλειες πακέτων, που είναι τα μισά από τα κωδικοποιημένα πακέτα που θα σταλούν. Επομένως παρέχει 50% προστασία σε απώλειες. Ο RLNC που θα αναλύσουμε στη συνέχεια αποτελεί χαρακτηριστικό παράδειγμα μπλοκ κώδικα απαλοιφής [R97].

- **Κώδικες Πηγής (Fountain Codes)** : Το βασικότερο χαρακτηριστικό αυτών είναι ότι το πλήθος των κωδικοποιημένων πακέτων που παράγουν μπορεί να τείνει στο άπειρο θεωρητικά. Είναι δηλαδή rateless κώδικες κατά βάση. Επίσης, τα κωδικοποιημένα πακέτα μπορούν να παράγονται όσο ο πομπός στέλνει πακέτα στον δέκτη. Αυτή η διαδικασία ονομάζεται on the fly αποστολή πακέτων. Για τους λόγους αυτούς δεν χρειάζεται να καθορίσουμε από την αρχή το πλήθος των κωδικοποιημένων πακέτων που θα δημιουργηθούν. Είναι σχεδόν βέλτιστοι για κάθε κανάλι απαλοιφής και για αυτό χαρακτηρίζονται ως παγκόσμιοι (*universal*). Εφόσον είναι rateless κώδικες, μπορούν να προσαρμόζονται σε κάθε κανάλι ανάλογα με το ποσοστό απωλειών που παρατηρείται. Αυτό σημαίνει ότι αν έχουμε ένα αξιόπιστο κανάλι με λίγες απώλειες, μπορούν να παραχθούν σχετικά λίγα κωδικοποιημένα πακέτα, αφού είναι μεγάλη η πιθανότητα αυτά να φτάσουν στον δέκτη. Αντίθετα, όταν έχουμε ένα κανάλι με μεγάλες απώλειες, τότε θα παραχθούν περισσότερα κωδικοποιημένα πακέτα ώστε να αυξηθεί η πιθανότητα να λάβει ο δέκτης το πλήθος πακέτων που χρειάζεται για να ανακτήσει την αρχική πληροφορία. Το όνομά τους οφείλεται στο γεγονός ότι λειτουργούν όπως ένα σιντριβάνι. Ο πομπός που παράγει τα κωδικοποιημένα μηνύματα μπορεί να χαρακτηριστεί ως ένα σιντριβάνι που παράγει απεριόριστες σταγόνες νερού. Οι σταγόνες νερού συμβολίζουν τα κωδικοποιημένα πακέτα. Ο δέκτης μπορεί να χαρακτηριστεί ως ένα ποτήρι που είναι κάτω από τη πηγή και γεμίζει σιγά σιγά με τις σταγόνες νερού. Μπορεί όμως μερικές από αυτές να πέσουν έξω. Αυτό συμβαίνει γιατί μπορεί αρχικά να θέλουμε να στείλουμε k πακέτα στον δέκτη, αλλά θα παραχθούν $n > k$ κωδικοποιημένα πακέτα. Έτσι ο δέκτης αρκεί να λάβει k κωδικοποιημένα πακέτα, ή λίγα περισσότερα, για να μπορέσει να ανακτήσει την αρχική πληροφορία. Ωστόσο, όπως θα δούμε στη συνέχεια, αυτοί οι κώδικες βασίζονται σε κατανομές πιθανοτήτων για τη λειτουργία τους. Έτσι, όταν από τα k αρχικά πακέτα παραχθούν $n > k$ κωδικοποιημένα, αυτό δεν μας εγγυάται ότι αν

ο δέκτης λάβει αυτά τα k κωδικοποιημένα πακέτα θα μπορέσει να ανακτήσει τα αρχικά. Ενδεχομένως να χρειαστεί λίγα περισσότερα από αυτά. Ωστόσο έχει αποδειχθεί ύστερα από μελέτες ότι υπάρχουν συγκεκριμένες κατανομές που προσφέρουν πολύ υψηλή πιθανότητα να καταφέρει ο δέκτης να ανακτήσει τα αρχικά πακέτα [M08;L02]. Ο LT που θα αναλύσουμε στη συνέχεια αποτελεί χαρακτηριστικό παράδειγμα κώδικα πηγής καθώς και οι BATS κώδικες μία επέκτασή τους.

2.2 Linear Network Coding (LNC)

Η γραμμική κωδικοποίηση δικτύου (*Linear Network Coding - LNC*) είναι μία τεχνική κωδικοποίησης δικτύου που χρησιμοποιεί γραμμικές αλγεβρικές πράξεις για το συνδυασμό και τη μετάδοση πακέτων δεδομένων σε ένα δίκτυο. Επιτρέποντας στους ενδιαμέσους κόμβους να συνδυάζουν γραμμικά τα εισερχόμενα πακέτα, το LNC ενισχύει την απόδοση, την αξιοπιστία και την ευρωστία στη μετάδοση δεδομένων, καθιστώντας τον ακρογωνιαίο λίθο των σύγχρονων στρατηγικών κωδικοποίησης δικτύου. Σ' αυτή την ανάλυση διερευνούμε τα θεωρητικά θεμέλια, τους λειτουργικούς μηχανισμούς, τα πλεονεκτήματα, τους περιορισμούς και τις εφαρμογές του LNC.

Εισήχθη για πρώτη φορά το 2000 από τους Ahlswede et al, οι οποίοι πρότειναν την κωδικοποίηση δικτύου ως μέσο για τη μεγιστοποίηση της απόδοσης του δικτύου πολλαπλής διανομής [AC++00]. Οι Koetter και Médard [KM03] επέκτειναν αυτό το πλαίσιο αποδεικνύοντας ότι οι γραμμικοί συνδυασμοί πακέτων δεδομένων σε ενδιαμέσους κόμβους επαρκούν για την επίτευξη βέλτιστης απόδοσης για δίκτυα πολλαπλής διανομής. Στο LNC, κάθε εξερχόμενο πακέτο σε έναν κόμβο είναι ένας γραμμικός συνδυασμός λαμβανόμενων πακέτων, που αντιπροσωπεύεται ως:

$$Y = \sum_{i=1}^n c_i X_i$$

όπου X_i είναι πακέτα εισόδου, c_i είναι συντελεστές κωδικοποίησης και Y είναι το κωδικοποιημένο πακέτο. Αυτοί οι συντελεστές τυπικά επιλέγονται από ένα πεπερασμένο πεδίο F_q όπου το q αντιπροσωπεύει το μέγεθος του πεδίου. Ένα μεγαλύτερο μέγεθος πεδίου q παρέχει μεγαλύτερη ευελιξία στην κωδικοποίηση αλλά αυξάνει την υπολογιστική πολυπλοκότητα.

Σε μια τυπική εγκατάσταση LNC, η λειτουργία ξεκινά με τον κόμβο πηγής που μεταδίδει τα αρχικά πακέτα δεδομένων. Αυτά τα πακέτα είτε αποστέλλονται ως έχουν είτε κωδικοποιούνται σε αρχικούς γραμμικούς συνδυασμούς. Οι ενδιάμεσοι κόμβοι, οι οποίοι λειτουργούν ως relay στο δίκτυο, δεν προωθούν απλώς τα ληφθέντα πακέτα. Αντίθετα, εκτελούν γραμμικές λειτουργίες στα εισερχόμενα πακέτα, συνδυάζοντάς τα χρησιμοποιώντας προκαθορισμένους ή τυχαία επιλεγμένους συντελεστές κωδικοποίησης. Αυτοί οι συντελεστές επιλέγονται από ένα πεπερασμένο πεδίο, που συχνά υποδηλώνεται ως F_q , όπου το μέγεθος πεδίου q καθορίζει το εύρος των πιθανών συντελεστών. Η επιλογή του μεγέθους του πεδίου επηρεάζει τόσο την ευελιξία της διαδικασίας κωδικοποίησης όσο και την υπολογιστική επιβάρυνση. Τα μεγαλύτερα μεγέθη πεδίων παρέχουν πιο ισχυρές επιλογές κωδικοποίησης αλλά απαιτούν μεγαλύτερους υπολογιστικούς πόρους [HM++06].

Η διαδικασία κωδικοποίησης στους ενδιάμεσους κόμβους μπορεί να εκφραστεί μαθηματικά. Εάν ένας κόμβος λάβει πακέτα εισόδου X_1, X_2, \dots, X_n , δημιουργεί ένα εξερχόμενο κωδικοποιημένο πακέτο Y ως γραμμικό συνδυασμό αυτών των εισόδων, που αντιπροσωπεύεται ως $Y = c_1X_1 + c_2X_2 + \dots + c_nX_n$, όπου c_1, c_2, \dots, c_n είναι οι συντελεστές κωδικοποίησης. Αυτό το κωδικοποιημένο πακέτο μεταδίδεται στη συνέχεια στους επόμενους κόμβους ή προορισμούς. Δεδομένου ότι κάθε εξερχόμενο πακέτο είναι ένας συνδυασμός των εισερχόμενων πακέτων, η ποσότητα των δεδομένων που μεταδίδονται μέσω του δικτύου ελαχιστοποιείται και η χωρητικότητα του δικτύου βελτιστοποιείται.

Στους κόμβους προορισμού λαμβάνει χώρα η διαδικασία της αποκωδικοποίησης. Για την ανακατασκευή των αρχικών πακέτων, ο προορισμός πρέπει να λάβει επαρκή αριθμό γραμμικά ανεξάρτητων κωδικοποιημένων πακέτων. Κάθε κωδικοποιημένο πακέτο αντιπροσωπεύει μια γραμμική εξίσωση και η συλλογή αυτών των πακέτων σχηματίζει ένα σύστημα γραμμικών εξισώσεων. Ο κόμβος προορισμού επιλύει αυτό το σύστημα χρησιμοποιώντας τυπικές τεχνικές όπως η Gaussian elimination, υπό την προϋπόθεση ότι οι συντελεστές των λαμβανόμενων πακέτων σχηματίζουν έναν πίνακα πλήρους κατάταξης. Όταν ο πίνακας είναι πλήρους κατάταξης, ο προορισμός μπορεί να καθορίσει μοναδικά τα αρχικά πακέτα δεδομένων.

Μια σημαντική πτυχή της λειτουργίας του LNC είναι η απαίτηση για γραμμική ανεξαρτησία μεταξύ των κωδικοποιημένων πακέτων που λαμβάνονται στον προορισμό. Εάν τα πακέτα δεν είναι ανεξάρτητα, το σύστημα εξισώσεων δεν θα έχει μοναδική λύση, εμποδίζοντας τον προορισμό να αποκωδικοποιήσει πλήρως τα αρχικά δεδομένα. Για να

αντιμετωπίζεται αυτό, χρησιμοποιείται συχνά η RLNC, όπου οι συντελεστές κωδικοποίησης επιλέγονται τυχαία. Αυτή η τυχειότητα εξασφαλίζει υψηλή πιθανότητα δημιουργίας γραμμικά ανεξάρτητων κωδικοποιημένων πακέτων, ειδικά όταν το μέγεθος πεπερασμένου πεδίου q είναι αρκετά μεγάλο.

Η λειτουργία του LNC περιλαμβάνει επίσης τη μετάδοση συντελεστών κωδικοποίησης παράλληλα με τα κωδικοποιημένα πακέτα. Αυτοί οι συντελεστές πρέπει να κοινοποιούνται στους κόμβους προορισμού για να καταστεί δυνατή η διαδικασία αποκωδικοποίησης. Αν και αυτό εισάγει πρόσθετο γενικό κόστος, είναι απαραίτητο για την ακριβή ανακατασκευή των αρχικών δεδομένων.

Η αποτελεσματικότητα του LNC είναι εμφανής σε σενάρια όπως τα δίκτυα πολλαπλής διανομής, όπου τα δεδομένα πρέπει να παραδίδονται σε πολλούς δέκτες. Αντί να μεταδίδει ξεχωριστές ροές δεδομένων για κάθε δέκτη, το LNC επιτρέπει στο δίκτυο να παραδώσει ένα μόνο ρεύμα κωδικοποιημένων πακέτων, όπου όλοι οι δέκτες μπορούν να αποκωδικοποιήσουν για να ανακτήσουν τις αρχικές πληροφορίες. Αυτό μειώνει δραματικά τον αριθμό των απαιτούμενων μεταδόσεων, βελτιστοποιώντας τη χρήση των πόρων του δικτύου και ελαχιστοποιώντας τη συμφόρηση.

2.2.1 Πλεονεκτήματα LNC

Η γραμμική κωδικοποίηση δικτύου ως μια μετασχηματιστική προσέγγιση στη μετάδοση δεδομένων παρέχει πολλά πλεονεκτήματα σε σχέση με τις παραδοσιακές μεθόδους δρομολόγησης και διάδοσης δεδομένων. Αυτά τα οφέλη, που υποστηρίζονται από αναφορές υψηλού αντίκτυπου, αντιμετωπίζουν θεμελιώδεις προκλήσεις στην αποτελεσματικότητα του δικτύου, την αξιοπιστία, την επεκτασιμότητα και τη χρήση πόρων, καθιστώντας το LNC ακρογωνιαίο λίθο των σύγχρονων δικτύων επικοινωνίας.

Ένα από τα πιο σημαντικά πλεονεκτήματα του LNC είναι η ικανότητά του να μεγιστοποιεί την απόδοση του δικτύου. Σε σενάρια πολλαπλής διανομής, όπου μία μόνο πηγή μεταδίδει δεδομένα σε πολλούς προορισμούς, το LNC επιτρέπει στους ενδιάμεσους κόμβους να συνδυάζουν πακέτα πριν τα προωθήσουν. Αυτό μειώνει τον αριθμό των μεταδόσεων που απαιτούνται για την παράδοση όλων των πακέτων δεδομένων σε όλους τους παραλήπτες, επιτυγχάνοντας βέλτιστη απόδοση. Οι Ahlswede et al. [AC++00] διαπίστωσαν ότι η κωδικοποίηση δικτύου, ιδιαίτερα η γραμμική κωδικοποίηση, επιτυγχάνει την ικανότητα πολλαπλής διανομής ενός δικτύου, αποδεικνύοντας ότι ο

συνδυασμός πακέτων οδηγεί σε σημαντικά κέρδη απόδοσης σε σύγκριση με την παραδοσιακή δρομολόγηση.

Το LNC βελτιώνει εγγενώς την ευρωστία των δικτύων επικοινωνίας στην απώλεια πακέτων. Με τη μετάδοση κωδικοποιημένων πακέτων που συνδυάζουν πληροφορίες από πολλαπλά πρωτότυπα πακέτα, το LNC διασφαλίζει ότι ακόμη και αν κάποια πακέτα χαθούν κατά τη μετάδοση, οι κόμβοι προορισμού μπορούν να ανασυνθέσουν τα αρχικά δεδομένα εφόσον λαμβάνουν επαρκή αριθμό γραμμικά ανεξάρτητων κωδικοποιημένων πακέτων. Οι Fragouli και Soljanin [FS07] τόνισαν ότι το LNC εισάγει τον πλεονασμό στη διαδικασία μετάδοσης. Έτσι είναι εφικτή η ανθεκτικότητα και η ανάκτηση σφαλμάτων σε περιβάλλοντα με απώλειες ή υψηλές παρεμβολές, όπως τα ασύρματα δίκτυα.

Με τη χρήση της γραμμικής κωδικοποίησης των πακέτων, βελτιστοποιείται η χρήση των πόρων του δικτύου, όπως το εύρος ζώνης και την αποθήκευση. Σε αντίθεση με τις παραδοσιακές μεθόδους, που στέλνουν διπλά πακέτα για να διασφαλίσουν την αξιοπιστία, το LNC μειώνει τον πλεονασμό συνδυάζοντας έξυπνα πακέτα. Αυτή η προσέγγιση ελαχιστοποιεί τη συμφόρηση και μεγιστοποιεί τη χρήση του διαθέσιμου εύρους ζώνης, ιδιαίτερα σε δίκτυα με περιορισμένη χωρητικότητα [KM03]. Επιπλέον, το LNC επιτρέπει την πιο αποτελεσματική χρήση της αποθήκευσης σε κατανεμημένα συστήματα συνδυάζοντας μπλοκ δεδομένων, μειώνοντας την ανάγκη για διπλότυπη αποθήκευση [DG++10].

Η απόδοση του LNC δεν υποβαθμίζεται σημαντικά καθώς αυξάνεται το μέγεθος του δικτύου, καθιστώντας το εξαιρετικά επεκτάσιμο για μεγάλα και πολύπλοκα δίκτυα. Αυτή η επεκτασιμότητα είναι ιδιαίτερα εμφανής σε σενάρια πολλαπλής εκπομπής, όπου ο αριθμός των δεκτών μπορεί να αυξηθεί δυναμικά. Το LNC χειρίζεται αποτελεσματικά αυτή την ανάπτυξη κωδικοποιώντας πακέτα με τρόπο που εξυπηρετεί πολλούς παραλήπτες ταυτόχρονα, αποφεύγοντας την εκθετική αύξηση των μεταδόσεων που σχετίζονται με την παραδοσιακή δρομολόγηση [HM++06].

Το LNC αποδίδει εξαιρετικά σε δυναμικά δίκτυα, όπως δίκτυα ad hoc κινητών (MANETs) και δίκτυα οχημάτων, όπου η τοπολογία αλλάζει συχνά. Δεδομένου ότι οι διαδικασίες κωδικοποίησης και αποκωδικοποίησης είναι κατανεμημένες και δεν απαιτούν κεντρικό έλεγχο, το LNC προσαρμόζεται αποτελεσματικά στις μεταβαλλόμενες συνθήκες δικτύου [CWJ03]. Αυτή η αποκεντρωμένη λειτουργία το καθιστά κατάλληλο για περιβάλλοντα όπου η κεντρική υποδομή δεν είναι διαθέσιμη ή αναξιόπιστη.

Το LNC ενσωματώνει τις δυνατότητες ανίχνευσης και διόρθωσης σφαλμάτων στη διαδικασία κωδικοποίησης. Αντιμετωπίζοντας τη διαδικασία αποκωδικοποίησης ως την ανάλυση ενός συστήματος γραμμικών εξισώσεων, τα σφάλματα στα ληφθέντα πακέτα

μπορούν να εντοπιστούν και να διορθωθούν χωρίς να απαιτούνται πρόσθετοι μηχανισμοί διόρθωσης σφαλμάτων. Οι Jaggi et al. [JSC07] απέδειξαν ότι ο εγγενής πλεονασμός και η γραμμικότητα του LNC του επιτρέπουν να ανιχνεύει και να απομονώνει κατεστραμμένα πακέτα, ενισχύοντας την ακεραιότητα των δεδομένων σε αναξιόπιστα δίκτυα.

Το LNC είναι εξαιρετικά συμβατό με αναδυόμενες τεχνολογίες όπως το 5G, το Internet of Things (IoT) και το cloud computing. Για παράδειγμα, στα δίκτυα 5G, το LNC ευθυγραμμίζεται με βασικούς στόχους όπως η εξαιρετικά αξιόπιστη επικοινωνία χαμηλής καθυστέρησης (urlhc) και η βελτιωμένη ευρυζωνική σύνδεση για κινητά τηλέφωνα (eMBB). Σχετική έρευνα [SYW20] σημείωσε ότι το LNC ενισχύει την απόδοση και την αξιοπιστία στα δίκτυα 5G, αντιμετωπίζοντας τις προκλήσεις των υψηλών ρυθμών δεδομένων και των περιορισμών καθυστέρησης. Ομοίως, σε περιβάλλοντα IoT, όπου οι συσκευές έχουν συχνά περιορισμένους πόρους, η ικανότητα του LNC να βελτιστοποιεί το εύρος ζώνης και τη χρήση ενέργειας το καθιστά μια πολύτιμη λύση.

2.2.2 Βασικοί ορισμοί τυχαίας γραμμικής κωδικοποίησης δικτύου

Μία τεχνική κωδικοποίησης δικτύου η οποία μελετήθηκε στην παρούσα διπλωματική εργασία και αποτελεί κατηγορία του LNC είναι η RLNC. Στην RLNC κωδικοποίηση, κωδικοποιημένα πακέτα παράγονται τόσο στον πομπό όσο και σε ενδιάμεσους κόμβους μέσα από γραμμικούς συνδυασμούς των αρχικών πακέτων δεδομένων και τυχαίων συντελεστών στο αντίστοιχο πεπερασμένο πεδίο που δουλεύουμε. Όσο μεγαλύτερο είναι το πεδίο τόσο μικρότερη είναι η πιθανότητα να έχουμε γραμμικώς εξαρτημένους συντελεστές. Αντίστοιχα όμως αυξάνεται και η υπολογιστική πολυπλοκότητα. Έχει αποδειχθεί ότι όταν δουλεύουμε σε πεπερασμένο πεδίο F_8 , η πιθανότητα να λάβουμε γραμμικώς εξαρτημένους συντελεστές μπορεί να θεωρηθεί αμελητέα. [FWB06]

Στον κώδικα RLNC, πριν εφαρμόσουμε κωδικοποίηση, χωρίζουμε τα πακέτα του πομπού σε γενεές (*generations*) συνήθως σταθερού μεγέθους g . Το μέγεθος των γενεών προκαθορίζεται στην αρχή του αλγορίθμου και παίζει καθοριστικό ρόλο για την απόδοση του δικτύου. [FWB06]

Όταν θέλουμε να στείλουμε μεγάλο όγκο δεδομένων καταλαβαίνουμε ότι είναι εξαιρετικά περίπλοκο και χρονοβόρο να εφαρμόζουμε κωδικοποίηση αλλά και

αποκωδικοποίηση στο συνολικό πλήθος των πακέτων που θα στείλουμε. Οι γενεές έρχονται να μειώσουν αυτή την υπολογιστική πολυπλοκότητα. Έτσι αν έχουμε k αρχικά πακέτα, τα χωρίζουμε σε γενεές μεγέθους g και σχηματίζουμε τον πίνακα $M = \begin{bmatrix} m_1 & m_2 & \cdots & m_g \end{bmatrix}$ όπου κάθε πακέτο m_i αποτελεί διάνυσμα στήλη [HP++09;FWB06].

Στο RLNC υπάρχουν δύο βασικοί τρόποι με τους οποίους μπορούμε να εφαρμόσουμε τη κωδικοποίηση, οι οποίοι είναι οι εξής:

- **Συστηματική προσέγγιση (systematic)** : Σε αυτή τη τεχνική, ο πομπός χωρίζει κανονικά τα πακέτα σε γενεές όπως εξηγήσαμε αλλά δεν τα κωδικοποιεί απευθείας. Στέλνει πρώτα τα πακέτα της γενεάς αυτούσια προς τον δέκτη. Στη συνέχεια, ανάλογα με το code rate που εφαρμόζουμε παράγει τα πλεονάζοντα πακέτα τα οποία αποτελούν γραμμικούς συνδυασμούς των πακέτων μίας γενεάς. Με αυτόν τον τρόπο μειώνεται η πολυπλοκότητα τόσο της κωδικοποίησης, όσο και της αποκωδικοποίησης. Αυτό συμβαίνει γιατί στην κωδικοποίηση χρειάζεται να παραχθούν λιγότερα κωδικοποιημένα σε πλήθος πακέτα από το μέγεθος ένας γενεάς, αλλά και στην αποκωδικοποίηση ο δέκτης έχει τη δυνατότητα να αρχίσει να χρησιμοποιεί τα αρχικά δεδομένα που έλαβε επιτυχώς άμεσα. Με αυτό τον τρόπο υπάρχει και μικρότερη καθυστέρηση. Ωστόσο σε περιβάλλοντα με απώλειες ενδέχεται να μην είναι εξίσου αποτελεσματική με την επόμενη τεχνική γιατί μπορεί να απαιτείται να σταλούν περισσότερα πακέτα για την πλήρη αποκωδικοποίηση κάθε γενεάς [HP++09;TTP17].
- **Μη συστηματική προσέγγιση (non systematic ή αλλιώς full vector κωδικοποίηση)** : Σε αυτή την περίπτωση, ο πομπός στέλνει μόνο κωδικοποιημένα πακέτα στον δέκτη. Δηλαδή χωρίζει τα αρχικά πακέτα σε γενεές μεγέθους g και ξεκινάει άμεσα τη δημιουργία κωδικοποιημένων πακέτων. Και σε αυτή τη περίπτωση δεν θα στείλει μόνο g πακέτα, αλλά θα στείλει και επιπλέον πακέτα ανάλογα με τον ρυθμό κωδικοποίησης που εφαρμόζουμε. Με αυτό τον τρόπο κάθε πακέτο που μεταδίδεται στο δίκτυο περιέχει πληροφορία για όλα τα αρχικά πακέτα που περιέχονται στη γενεά. Αυτή η τεχνική λόγω του ότι στέλνουμε μόνο κωδικοποιημένα πακέτα μαζί με τα διανύσματα συντελεστών τους, έχει αυξημένη υπολογιστική πολυπλοκότητα [TTP17].

Και στις δύο προσεγγίσεις πρώτα πρέπει να σταλούν όλα τα πακέτα, κωδικοποιημένα και μη, που αφορούν τα πακέτα της γενεάς που θέλει να στείλει ο πομπός προτού αυτός προχωρήσει στην κωδικοποίηση της επόμενης γενεάς πακέτων.

Τέλος, πολύ σημαντική είναι και η έννοια του παράθυρου κωδικοποίησης (*coding window*) στο RLNC η οποία αναφέρεται στο σύνολο των αρχικών πακέτων δεδομένων

που συνδυάζονται γραμμικά για τη δημιουργία ενός κωδικοποιημένου πακέτου. Στα πειράματά μας θα χρησιμοποιήσουμε σταθερό coding window.

2.2.3 Βασικοί ορισμοί batched sparse κωδίκων

Οι BATS κώδικες είναι μία κατηγορία κωδίκων που σχεδιάστηκαν για να βελτιώσουν κυρίως την απόδοση της μετάδοσης σε δίκτυα με πολλά άλματα (*multi-hop*). Αναπτύχθηκαν από τους Shenghao Yang και Raymond W. Yeung (2014) οι οποίοι παρουσίασαν τη χρήση τους για τη βελτίωση της απόδοσης ιδιαίτερα σε περιβάλλοντα με υψηλούς ρυθμούς απώλειας [YY14].

Αποτελούν μία προέκταση των κωδίκων πηγής όπου ενσωματώνουν τυχαία γραμμική κωδικοποίηση δικτύου. Συνδυάζουν δηλαδή τις ιδιότητες των κωδίκων πηγής και του RLNC με στόχο να επωφεληθούν από τα πλεονεκτήματα και των δύο. Ένας BATS κώδικας αποτελείται από έναν εξωτερικό κώδικα και έναν εσωτερικό κώδικα που πραγματοποιούν πράξεις πάνω σε ένα πεπερασμένο πεδίο F_{2^s} . Ο εξωτερικός κώδικας, αποτελεί ουσιαστικά μία γενίκευση πίνακα ενός κώδικα πηγής και για αυτό το λόγο μπορεί να παράγει πακέτα χωρίς ρυθμό. Ομαδοποιεί τα πακέτα που θέλουμε να στείλουμε σε παρτίδες (*batches*), καθεμία από τις οποίες έχει ένα μέγεθος M . Έτσι, εφόσον δεν έχει ρυθμό, μπορεί να παράξει θεωρητικά άπειρο πλήθος τέτοιων παρτίδων προς αποστολή στον δέκτη. Ο εσωτερικός κώδικας, εφαρμόζει τυχαία γραμμική κωδικοποίηση μιας παρτίδας. Μπορεί να εφαρμοστεί στους ενδιάμεσους κόμβους του δικτύου με τον περιορισμό ότι τα πακέτα στα οποία θα εφαρμόσει γραμμική κωδικοποίηση, πρέπει να ανήκουν αυστηρά στην ίδια παρτίδα. Αυτός ο περιορισμός είναι ιδιαίτερα σημαντικός, γιατί διατηρεί τους βαθμούς των παρτίδων έτσι ώστε να μπορεί να εφαρμοστεί ένας αποδοτικός αλγόριθμος αποκωδικοποίησης για την κοινή αποκωδικοποίηση του εξωτερικού και του εσωτερικού κώδικα [YY14]. Όπως θα δούμε και στη συνέχεια, για την υλοποίηση της κωδικοποίησης του BATS κώδικα χρειάζεται μία κατανομή βαθμών η οποία καθορίζει το πλήθος των αρχικών πακέτων που θα χρησιμοποιηθούν για την δημιουργία ενός batch.

Οι κώδικες BATS είναι κατάλληλοι για κάθε δίκτυο που επιτρέπει γραμμική κωδικοποίηση δικτύου στους ενδιάμεσους κόμβους και είναι ανθεκτικοί σε δίκτυα που η τοπολογία τους και ο ρυθμός απωλειών μεταβάλλεται δυναμικά. Ακόμα, μπορούν να λειτουργήσουν και με πεπερασμένα πεδία μικρά σε μέγεθος [YY17;YY14].

Η αποκωδικοποίηση των batches που γίνεται στο δέκτη, υλοποιείται από τον αλγόριθμο αποκωδικοποίησης Διάδοσης Πιθανοτήτων (*Belief Propagation – BP*) και μπορεί να περιγραφεί εύκολα μέσω ενός διμερούς γράφου, όπως θα δούμε στο επόμενο κεφάλαιο [YY14]. Λεπτομέρειες σχετικά με τον ακριβή τρόπο με τον οποίο υλοποιούνται οι BATS κώδικες καθώς και κάποιες παραδοχές θα αναλυθούν στη συνέχεια.

Κεφάλαιο 3. Ένα πλαίσιο για την αξιολόγηση της κωδικοποίησης δικτύου

3.1 PyErasure

Η γλώσσα προγραμματισμού Python είναι δημοφιλής για πολλές εφαρμογές της, συμπεριλαμβανομένης και αυτής που αφορά τα δίκτυα υπολογιστών. Ένα από τα πλεονεκτήματα της Python είναι η απλότητά της, η οποία επιτρέπει στον χρήστη να αναπτύσσει εύκολα και γρήγορα σύνθετους αλγόριθμους. Είναι ιδανική για την ανάλυση και επεξεργασία δεδομένων, κάτι που την καθιστά ιδιαίτερα χρήσιμη στην υλοποίηση τεχνικών κωδικοποίησης δικτύου. Επίσης παρέχει πολλές βιβλιοθήκες και διαθέτει εκτεταμένη κοινότητα χρηστών και υποστηρικτών.



Εικόνα 2 - Λογότυπο PyErasure

Η PyErasure είναι μία βιβλιοθήκη που υλοποιεί τεχνικές κωδικοποίησης απαλοιφής (erasure coding). Έχει δημιουργηθεί από την δανέζικη εταιρεία Steinwurf η οποία

παρέχει λύσεις κωδικοποίησης που στοχεύουν στην αύξηση της απόδοσης του δικτύου καθώς και της μείωσης της καθυστέρησης.

Η PyErasure προσφέρει εργαλεία για τη δημιουργία και διαχείριση πεπερασμένων πεδίων F_{2^s} , διαδικασίες οι οποίες είναι ιδιαίτερα δύσκολες να πραγματοποιηθούν γιατί βασίζονται σε πράξεις πολυωνύμων. Για παράδειγμα, αν δουλεύουμε στο πεδίο F_8 , τότε οι πράξεις γίνονται με πολυώνυμα όγδοου βαθμού. Ακόμη η βιβλιοθήκη αυτή παρέχει κλάσεις με τις οποίες μπορούμε να εκτελούμε πρόσθεση αλλά και πολλαπλασιασμούς διανυσμάτων με σταθερές στα πεπερασμένα πεδία F_2 , F_4 και F_8 .

Μέσω της κλάσης RandomUniform, μας δίνει τη δυνατότητα να παράγουμε τυχαίους συντελεστές για τα διανύσματα συντελεστών που απαιτούνται για τον γραμμικό συνδυασμό των πακέτων δεδομένων τόσο στον RLNC όσο και στον BATS κώδικα ακόμη και στην περίπτωση που έχουμε κυλιόμενο παράθυρο κωδικοποίησης.

Επιπρόσθετα, υπάρχει η κλάση Encoder η οποία έχει τις απαραίτητες μεθόδους για να μπορούμε να υλοποιήσουμε τυχαία γραμμική κωδικοποίηση ανάλογα με το πεπερασμένο πεδίο στο οποίο δουλεύουμε κάθε φορά. Στα πειράματα που εκτελέσαμε ο πομπός αποτελεί αντικείμενο αυτής της κλάσης.

Για την υλοποίηση της αποκωδικοποίησης και ανάκτηση των αρχικών δεδομένων χρησιμοποιήσαμε την κλάση Decoder της βιβλιοθήκης, η οποία υλοποιεί την απαλοιφή Gauss που, όπως θα δούμε στη συνέχεια του κεφαλαίου, μέσω αυτής επιτυγχάνεται η αποκωδικοποίηση. Η κλάση αυτή παρέχει τις απαραίτητες μεθόδους για να μπορέσουμε να εκτελέσουμε RLNC είτε με systematic είτε με non systematic προσέγγιση.

Συνοπτικά η βιβλιοθήκη PyErasure προσφέρει έναν ευέλικτο και αποδοτικό τρόπο για τη μελέτη των τεχνικών σε διαφορετικά σενάρια. Στην παρούσα διπλωματική επεκτείνουμε τις κλάσεις της ώστε να υλοποιούν και τους LT και τους BATS κώδικες με όλες τις παραδοχές που θα αναλυθούν στη συνέχεια.

3.2 Random Linear Network Codes (RLNC)

Στην κωδικοποίηση δικτύου, τα δεδομένα που θέλουμε να μεταδώσουμε στους παραλήπτες, χωρίζονται σε πακέτα. Όλα αυτά τα πακέτα έχουν ίδιο μέγεθος m . Ας θεωρήσουμε ότι τα πακέτα που θέλουμε να στείλουμε είναι m_1, m_2, \dots, m_k , k σε πλήθος. Ο πομπός μέσα από τον encoder παράγει γραμμικούς συνδυασμούς των αρχικών πακέτων και δημιουργεί τα κωδικοποιημένα πακέτα x . Για να γίνει αυτή η διαδικασία

χρησιμοποιεί τη κλάση RandomUniform η οποία παράγει ομοιόμορφα τυχαίους συντελεστές στο πεπερασμένο πεδίο που δουλεύουμε. Έχει μεγάλη σημασία το πεδίο που δουλεύουμε στη παραγωγή συντελεστών γιατί η γεννήτρια αυτή παράγει διανύσματα συντελεστών g των 8 bit. Ωστόσο, αν εμείς δουλεύουμε στο πεδίο F_8 , τότε κάθε ένα διάνυσμα συντελεστών που παράγεται μπορούμε να το χρησιμοποιήσουμε με ένα από τα αρχικά πακέτα για την παραγωγή κωδικοποιημένου πακέτου. Αντίθετα, αν δουλεύουμε σε μικρότερα πεδία, όπως F_4 ή F_2 , αφού οι συντελεστές μας αποτελούνται από 8 bit ο καθένας, τότε στη πρώτη περίπτωση, ένα διάνυσμα συντελεστών μπορούμε να το χρησιμοποιήσουμε με δύο από τα αρχικά πακέτα, ενώ στη δεύτερη περίπτωση ένα διάνυσμα θα μπορέσουμε να το χρησιμοποιήσουμε με 8 αρχικά πακέτα. Κάθε κωδικοποιημένο πακέτο δίνεται από τον τύπο $x = \sum_{i=1}^k g_i m_i$, όπου g_1, g_2, \dots, g_k είναι οι συντελεστές στο πεπερασμένο πεδίο F_{2^s} [FWB06].

Για παράδειγμα, αν θέλουμε να κωδικοποιήσουμε δύο πακέτα m_1, m_2 . Θέλουμε δηλαδή από αυτά να παράξουμε δύο κωδικοποιημένα πακέτα. Θα χρειαστούμε τέσσερα διανύσματα συντελεστών $c_{11}, c_{12}, c_{21}, c_{22}$. Τα κωδικοποιημένα πακέτα που θα προκύψουν από τους γραμμικούς συνδυασμούς θα είναι τα εξής [HP++09]:

$$x_1 = c_{11} m_1 + c_{12} m_2$$

και

$$x_2 = c_{21} m_1 + c_{22} m_2$$

όπου οι συντελεστές ανήκουν στο αντίστοιχο πεπερασμένο πεδίο που δουλεύουμε. Μπορούμε ακόμα να υποθέσουμε ότι τα αρχικά μας πακέτα μεγέθους m απαρτίζουν έναν πίνακα $M = \begin{bmatrix} m_1 & m_2 & \dots & m_k \end{bmatrix}$. Αντίστοιχα και τα διανύσματα συντελεστών που παράγονται αποτελούν έναν πίνακα διαστάσεων 2×2 . Έτσι, στο παραπάνω παράδειγμα, αν υποθέσουμε ότι δουλεύουμε με τη χρήση πινάκων, τα κωδικοποιημένα πακέτα που θα παραχθούν μπορούν να γραφτούν ως εξής [HP++09]:

$$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} c_{11} & c_{12} \\ c_{21} & c_{22} \end{bmatrix} \cdot \begin{bmatrix} m_1 \\ m_2 \end{bmatrix}$$

Η διαδικασία αυτή της δημιουργίας κωδικοποιημένων πακέτων, μπορεί να εφαρμοστεί και για επανακωδικοποίηση στους ενδιάμεσους κόμβους που λαμβάνουν τα κωδικοποιημένα πακέτα. Η επανακωδικοποίηση αυτή πραγματοποιείται με την ίδια διαδικασία που περιγράψαμε προηγουμένως. Ωστόσο, εφαρμόζοντας κωδικοποίηση δικτύου γενικά, πρέπει να μεταφέρουμε μαζί με τα κωδικοποιημένα πακέτα και τους συντελεστές που χρησιμοποιήθηκαν για τον σχηματισμό τους γιατί είναι απαραίτητη αυτή η πληροφορία για τον δέκτη που θα εφαρμόσει την αποκωδικοποίηση. Η αποστολή αυτής της επιπλέον πληροφορίας προκαλεί μία επιβάρυνση στο δίκτυο. Η επιβάρυνση αυτή όμως είναι πολύ μικρή. Συγκεκριμένα, αν θεωρήσουμε ότι θέλουμε να στείλουμε 50 πακέτα, καθένα από τα οποία έχει μέγεθος 1400 bytes και δουλεύουμε στο πεπερασμένο πεδίο F_8 για κάθε κωδικοποιημένο πακέτο, θα χρειαστούν 50 συντελεστές των 8 bit (1 byte) ο καθένας λόγω του πεπερασμένου πεδίου. Επομένως θα χρειαστούν $50 \cdot 8 = 400$ bit επιπλέον πληροφορίας λόγω των συντελεστών. Τα πακέτα όπως που θα στείλουμε θα έχουν μέγεθος $1400 \cdot 8 = 11200$ bit. Επομένως αν διαιρέσουμε την επιβάρυνση που επιφέρουν οι συντελεστές με τη πληροφορία των πακέτων θα δούμε ότι είναι της τάξης του $400/11200 \approx 3\%$. Το όφελος όμως της μετάδοσης αυτής της πληροφορίας είναι πολύ μεγαλύτερο μιας και λαμβάνοντας ο δέκτης τα διανύσματα συντελεστών μαζί με τα κωδικοποιημένα πακέτα, αποκτά το πλεονέκτημα να μη χρειάζεται να γνωρίζει ούτε τις συναρτήσεις που υλοποιήθηκαν για τη κωδικοποίηση ούτε τη τοπολογία του δικτύου. Έτσι ακόμα και αν καταρρεύσουν κόμβοι και έχουμε απώλειες πακέτων σε οποιοδήποτε σημείο του δικτύου, αυτό δεν θα επηρεάσει έντονα τον δέκτη. Αυτό κάνει τη διαδικασία της αποκωδικοποίησης να είναι πιο εύρωστη [CW07].

Όλες οι λειτουργίες στο RLNC εκτελούνται στο πεπερασμένο πεδίο. Για αυτό η πιθανότητα ο πομπός να λάβει από τη γεννήτρια συντελεστές που αποτελούν γραμμικό συνδυασμό παίζει σημαντικό ρόλο. Το αν θα λάβει τέτοιους συντελεστές εξαρτάται από το μέγεθος του πεπερασμένου πεδίου πάνω στο οποίο δουλεύουμε. Ο λόγος για τον οποίο είναι τόσο σημαντική η γραμμική ανεξαρτησία των συντελεστών είναι ότι αν χρησιμοποιηθούν γραμμικά εξαρτημένοι συντελεστές, τότε και τα κωδικοποιημένα πακέτα που θα παραχθούν θα είναι γραμμικά εξαρτημένα μεταξύ τους. Αυτό σημαίνει ότι δεν θα προσφέρουν επιπρόσθετη πληροφορία για τα αρχικά πακέτα που θέλουμε να ανακτήσει ο δέκτης. Όταν αυξάνουμε το μέγεθος του πεδίου στο οποίο δουλεύουμε, τότε μειώνουμε και την πιθανότητα να λάβουμε γραμμικώς εξαρτημένους συντελεστές κωδικοποίησης. Ωστόσο ακόμα και αν οι συντελεστές κωδικοποίησης παράγονται ομοιόμορφα τυχαία στο αντίστοιχο πεπερασμένο πεδίο, έχει αποδειχθεί πως όταν δουλεύουμε στο F_8 , η πιθανότητα να λάβουμε γραμμικά εξαρτημένους συντελεστές

μειώνεται τόσο ώστε να θεωρηθεί αμελητέα. Έτσι καταφέρνει το RLNC να διασφαλίσει σε πολύ μεγάλο βαθμό γραμμική ανεξαρτησία και να παράγει τους συντελεστές με τέτοιο τρόπο που του δίνει το πλεονέκτημα να μη χρειάζεται να γνωρίζει την τοπολογία του δικτύου και το πώς λειτουργούν οι κόμβοι σε αυτό [HP++09].

Για την υλοποίηση ενός RLNC κώδικα μπορούμε να χωρίζουμε τα δεδομένα σε μπλοκ έτσι ώστε να είναι πιο αποδοτική η κωδικοποίηση και αποκωδικοποίηση, δεδομένου ότι όταν έχουμε μεγάλο όγκο δεδομένων, αν προσπαθήσουμε να εφαρμόσουμε κωδικοποίηση σε όλα τα πακέτα μαζί θα υπάρχει και μεγάλη καθυστέρηση αλλά και μεγαλύτερη ανάγκη για υπολογιστικούς πόρους. Επομένως, για τη πειραματική αξιολόγηση του RLNC, αποφασίσαμε να χωρίσουμε τα δεδομένα σε μπλοκ, των οποίων το μέγεθος θα ταυτίζεται κάθε φορά και με το μέγεθος της γενεάς. Για τις γενεές καθορίσαμε το μέγεθός τους ανάλογα με το code rate που εφαρμόσαμε σε κάθε περίπτωση όπως θα δούμε στο επόμενο κεφάλαιο.

Η διαδικασία κωδικοποίησης παραμένει ακριβώς ίδια όπως την περιγράψαμε προηγουμένως. Το μόνο που αλλάζει είναι το πλήθος των αρχικών πακέτων τα οποία χρησιμοποιούμε κάθε φορά για να παράξουμε κωδικοποιημένα πακέτα. Τα κωδικοποιημένα πακέτα με αυτόν τον τρόπο στέλνονται στον δέκτη ανά γενεά. Δηλαδή πρώτα κωδικοποιούμε τα πακέτα της πρώτης γενιάς και τα στέλνουμε κωδικοποιημένα στον δέκτη και μετά ξεκινάμε την κωδικοποίηση της επόμενης γενιάς. Όμως για να αποκωδικοποιηθεί μια γενιά, πρέπει να σταλούν $g + r$ κωδικοποιημένα πακέτα. Τα r αποτελούν την πλεονάζουσα πληροφορία που έχει σκοπό να καλύψει τις απώλειες πακέτων που μπορεί να προκύψουν στο κανάλι. Ο αριθμός των πλεοναζόντων πακέτων που θα στείλουμε εξαρτάται όπως εξηγήσαμε και στην αρχή του κεφαλαίου από τον ρυθμό κωδικοποίησης (code rate) που θα εφαρμόσουμε. Για να καταφέρει να αποκωδικοποιηθεί πλήρως μία γενιά, θα πρέπει να λάβει ο δέκτης g γραμμικώς ανεξάρτητα κωδικοποιημένα πακέτα μαζί με τα διανύσματα κωδικοποίησης. Όλα τα κωδικοποιημένα πακέτα που έλαβε ο δέκτης για μία γενιά τοποθετούνται σε ένα πίνακα $\widehat{X} = \begin{bmatrix} \widehat{x}_1 & \widehat{x}_2 & \cdots & \widehat{x}_g \end{bmatrix}$ και τα διανύσματα κωδικοποίησης που περιέχουν τους συντελεστές τοποθετούνται σε ένα πίνακα $\widehat{G} = \begin{bmatrix} \widehat{g}_1 & \widehat{g}_2 & \cdots & \widehat{g}_g \end{bmatrix}$. Έτσι, μπορεί να ανακτήσει τα M αρχικά πακέτα που περιέχονταν στη γενιά, λύνοντας το σύστημα $\widehat{M} = \widehat{X} \times \widehat{G}^{-1}$. Θα εξηγήσουμε αναλυτικά την διαδικασία της αποκωδικοποίησης στην συνέχεια του κεφαλαίου [HP++09].

Η διαδικασία της αποκωδικοποίησης στο RLNC πραγματοποιείται σε δύο στάδια με μία ελαφρώς τροποποιημένη εφαρμογή του αλγορίθμου Gauss-Jordan. Ο δέκτης αποθηκεύει τα πακέτα που λαμβάνει στον πίνακα αποκωδικοποίησης. Τα αρχικά πακέτα

της γενεάς είναι είτε αυτούσια τα αρχικά πακέτα του πομπού είτε κωδικοποιημένα πακέτα που συνοδεύονται από το αντίστοιχο διάνυσμα κωδικοποίησης, ανάλογα αν ακολουθούμε συστηματική προσέγγιση ή όχι. Για να κατανέμεται ομοιόμορφα ο φόρτος στον επεξεργαστή του συστήματος, τα πακέτα που λαμβάνει ο δέκτης αποκωδικοποιούνται συνεχώς όσο το δυνατόν περισσότερο. Στο πρώτο βήμα της αποκωδικοποίησης, για να μειώσουμε τα εισερχόμενα κωδικοποιημένα πακέτα, εφαρμόζουμε μία πρόσθια αντικατάσταση (forward substitution) των ήδη ληφθέντων πακέτων. Για να το πετύχουμε αυτό ερευνούμε τα στοιχεία του διανύσματος κωδικοποίησης. Αυτά είναι που καθόρισαν πώς έγινε ο γραμμικώς συνδυασμός των πακέτων της γενεάς και άρα μπορούμε να βρούμε ποια από τα αρχικά πακέτα έχουν συνδυαστεί για να παραχθεί το κωδικοποιημένο πακέτο. Αν ένα στοιχείο είναι ένα και έχουμε βρει ένα πακέτο με αυτό το στοιχείο σαν κύριο συντελεστή (pivot), τότε αφαιρούμε αυτό το πακέτο από το κωδικοποιημένο και συνεχίζουμε τον έλεγχο. Αν το στοιχείο είναι ένα, αλλά δεν έχουμε βρει ένα πακέτο με αυτό το στοιχείο σαν κύριο συντελεστή (pivot), τότε σε αυτή την περίπτωση το ταυτοποιούμε σαν κεντρικό πακέτο και συνεχίζουμε στο δεύτερο βήμα. Υπάρχει μία περίπτωση να μπορέσουμε να αφαιρέσουμε όλη τη πληροφορία ενός κωδικοποιημένου πακέτου ήδη από το πρώτο βήμα. Σε αυτή την περίπτωση, δεν περιέχει χρήσιμη πληροφορία για τον δέκτη οπότε το απορρίπτουμε [HP++09].

Στο δεύτερο στάδιο, εφαρμόζουμε οπισθοδρομική αντικατάσταση (backward substitution) με το κεντρικό πακέτο που εντοπίσαμε στο προηγούμενο βήμα. Από όλα τα πακέτα που έχει λάβει μέχρι εκείνη τη στιγμή ο δέκτης, κρατάμε μόνο εκείνα που είναι συνδυασμός του κεντρικού αυτού πακέτου. Την πληροφορία αυτή την παίρνουμε από το διάνυσμα συντελεστών καθενός από αυτά. Τέλος, αυτό που κάνουμε είναι να αφαιρέσουμε το κεντρικό πακέτο από όλα αυτά τα πακέτα που είναι συνδυασμός αυτού. Έτσι, μέσω αυτής της τροποποίησης του αλγόριθμου Gauss-Jordan, όταν λάβουμε επαρκή αριθμό κωδικοποιημένων πακέτων και έχει ολοκληρωθεί η αποκωδικοποίηση, θα έχουμε γνώση για όλα τα αρχικά πακέτα που στάλθηκαν από τον πομπό μέσω του πίνακα αποκωδικοποίησης. [HP++09]

3.3 Luby Transform Codes (LT)

Οι Luby Transform κώδικες αναπτύχθηκαν από τον Michael Luby το 1998. Δημοσιεύτηκαν το 2002 και είναι η πρώτη ολοκληρωμένη μέθοδος κωδικών πηγής

(fountain codes) που εφαρμόστηκε στη πράξη. Είναι κυρίως κώδικες χωρίς ρυθμό (rateless), όπου δεν χρειάζεται να έχουμε προκαθορισμένο πλήθος κωδικοποιημένων πακέτων προς αποστολή. Αυτό σημαίνει ότι το πλήθος των πακέτων που μπορούν να παράξουν θεωρητικά είναι άπειρο. Αυτό τους καθιστά ιδιαίτερα χρήσιμους, μιας και μπορούν να προσαρμόζονται σε οποιοδήποτε κανάλι απαλοιφής (erasure channel). Αν υπάρχει μεγάλος ρυθμός απωλειών στο κανάλι, τότε μπορούν απλά να επιλέξουν να στείλουν περισσότερα κωδικοποιημένα πακέτα, ενώ αν ο ρυθμός απωλειών είναι μικρός, θα σταλούν λιγότερα πακέτα. Το μόνο που χρειάζεται σε αυτή τη περίπτωση είναι μία ένδειξη από τον δέκτη ότι έχει καταφέρει να ανακτήσει με επιτυχία όλα τα αρχικά πακέτα, σε αντίθεση με τα παραδοσιακά δίκτυα όπου θα έπρεπε να στέλνονται πακέτα επιβεβαίωσης λήψης για κάθε ένα πακέτο ξεχωριστά [L02].

Οι LT κώδικες αποτελούν ειδική περίπτωση γραμμικής κωδικοποίησης και για αυτό θα αποτελέσουν σημαντικό σημείο αναφοράς στη πειραματική αξιολόγηση των τεχνικών μας. Συγκεκριμένα μας παρέχουν τη δυνατότητα να διακρίνουμε ποια από τα αποτελέσματα οφείλονται στη γραμμική κωδικοποίηση και ποια οφείλονται στην κωδικοποίηση δικτύου που εφαρμόζει το RLNC και ο BATS κώδικας.

Ένα αξιοσημείωτο πλεονέκτημα των LT κωδικών είναι το γεγονός ότι ακόμα και για πακέτα μεγάλου μεγέθους παραμένουν αποδοτικοί. Σε άλλους κώδικες, όταν επεξεργαζόμαστε μεγάλα σε μέγεθος δεδομένα, αυξάνεται σημαντικά η υπολογιστική πολυπλοκότητα. Αντίθετα, στους LT, έχει παρατηρηθεί ότι είναι μικρότερη η πολυπλοκότητα που προστίθεται όσο αυξάνουμε το μέγεθος των δεδομένων [L02].

Η διαδικασία κωδικοποίησης είναι εύκολο να περιγραφεί και να γίνει κατανοητή μιας και δεν υπάρχει μεγάλη πολυπλοκότητα στις πράξεις που εφαρμόζονται για την παραγωγή κωδικοποιημένων πακέτων.

Αρχικά για κάθε ένα από τα κωδικοποιημένα πακέτα πρέπει να επιλεγεί ένας βαθμός d , ο οποίος δεν είναι τίποτα περισσότερο από το πλήθος των αρχικών πακέτων του πομπού των οποίων η πληροφορία θα χρησιμοποιηθεί για την παραγωγή του κωδικοποιημένου πακέτου. Ο βαθμός αυτός επιλέγεται μέσα από μία συνάρτηση κατανομής πιθανοτήτων. Η επιλογή και ο σωστός σχεδιασμός της συνάρτησης κατανομής αποτελεί, ίσως, το πιο σημαντικό κομμάτι που θα καθορίσει τελικά την αποδοτικότητα της τεχνικής αυτής. Αφότου επιλεγεί ο βαθμός, διαλέγουμε ομοιόμορφα τυχαία d διακριτά πακέτα από τα αρχικά του πομπού. Τα πακέτα αυτά που επιλέχθηκαν χαρακτηρίζονται και ως γείτονες του κωδικοποιημένου πακέτου. Αντίθετα με το RLNC όπου παίρναμε τον γραμμικό συνδυασμό των αρχικών πακέτων, τώρα για να πάρουμε την τιμή του κωδικοποιημένου συμβόλου, θα εφαρμόσουμε την πράξη XOR μεταξύ

αυτών των πακέτων που επιλέχθηκαν (Πίνακας 1). Στον πομπό αποστέλλεται όχι το κωδικοποιημένο πακέτο μόνο του, αλλά μαζί με τον βαθμό του και τα πακέτα-γείτονες που επιλέχθηκαν. [L02;M03]

Η διαδικασία της κωδικοποίησης μπορεί να περιγραφεί και με τη χρήση αραιών γράφων (sparse graphs). [M08]. Τα πακέτα που θέλει να στείλει ο πομπός μπορούμε να πούμε ότι αποτελούν τους κόμβους δεδομένων ενός γράφου, ενώ τα κωδικοποιημένα πακέτα είναι οι κόμβοι ελέγχου. Όταν επιλέγουμε βαθμό d για έναν κόμβο ελέγχου μέσα από την συνάρτηση κατανομής πιθανοτήτων και στη συνέχεια τα d από τα αρχικά πακέτα, τότε ενώνουμε τον κόμβο αυτό με τους αντίστοιχους κόμβους δεδομένων με ακμές. Η τιμή του κόμβου, υπολογίζεται όπως περιγράψαμε με τη χρήση XOR.

Διαδικασία παραγωγής κωδικοποιημένου πακέτου:

- Τυχαία επιλογή βαθμού d του κωδικοποιημένου πακέτου από μία συνάρτηση κατανομής πιθανοτήτων.
- Ομοιόμορφη τυχαία επιλογή d διακριτών αρχικών πακέτων ως γείτονες του κωδικοποιημένου πακέτου.
- Εφαρμογή XOR στους d γείτονες του πακέτου για τον υπολογισμό της τιμής του.

Πίνακας 1 - Αλγόριθμος κωδικοποίησης του LT κώδικα

Είναι πολύ σημαντική η συνάρτηση κατανομής πιθανοτήτων που θα χρησιμοποιηθεί για την κωδικοποίηση του LT κώδικα. Αυτή η συνάρτηση καθορίζει πόσα πακέτα χρησιμοποιούνται κάθε φορά για την παραγωγή ενός κωδικοποιημένου πακέτου. Σε κάθε βαθμό αναθέτει ένα βάρος, το οποίο καθορίζει ποιοι βαθμοί θα εμφανιστούν περισσότερες φορές και ποιοι λιγότερες. Εκφράζουν δηλαδή την πιθανότητα ενός βαθμού να εμφανιστεί σε κάθε κωδικοποιημένο πακέτο. Ο τρόπος με τον οποίο δουλεύει η συνάρτηση κατανομής είναι αναθέτοντας μεγάλο βάρος στους μικρούς βαθμούς, αυξάνοντας έτσι τις πιθανότητες εμφάνισής τους. Στους μεγάλους βαθμούς αναθέτει μικρότερα βάρη, μειώνοντας έτσι την πιθανότητα εμφάνισής τους συγκριτικά με τους μικρούς βαθμούς. Όπως θα δούμε στη συνέχεια του κεφαλαίου, για να ξεκινήσει η διαδικασία της αποκωδικοποίησης στους LT κώδικες θα πρέπει να λάβει ο δέκτης κάποιο πακέτο που να έχει βαθμό ένα [M03;L02].

Όπως στο RLNC ήταν απαραίτητο να γνωρίζουμε για κάθε κωδικοποιημένο πακέτο και το διάνυσμα συντελεστών του, έτσι και τους LT κώδικες για κάθε κωδικοποιημένο πακέτο πρέπει να ξέρουμε τον d βαθμό του και τα d αρχικά πακέτα που επιλέχθηκαν για την κατασκευή του πακέτου. Ο αλγόριθμος αποκωδικοποίησης αρχικά ψάχνει να βρει ένα κωδικοποιημένο πακέτο που έχει βαθμό ένα. Αυτό σημαίνει ότι το πακέτο αυτό χρησιμοποίησε μόνο ένα από τα αρχικά πακέτα δεδομένων του πομπού και αποτελεί στην ουσία αντίγραφο αυτού του πακέτου. Επομένως μπορεί να το ανακτήσει άμεσα. Έπειτα εντοπίζει όλα τα κωδικοποιημένα πακέτα που έχουν αυτό το πακέτο που ανακτήθηκε ως γείτονα και εφαρμόζει σε αυτά την πράξη XOR με τη τιμή του ανακτημένου πακέτου. Παράλληλα το αφαιρεί από γείτονά τους και μειώνει και τον βαθμό τους κατά ένα. Στη συνέχεια προσπαθεί να ξαναεντοπίσει κάποιο πακέτο που έχει βαθμό ένα για να επαναλάβει την διαδικασία αυτή (Πίνακας 2). Η διαδικασία της αποκωδικοποίησης πετυχαίνει αν ο αποκωδικοποιητής ανακτήσει όλα τα αρχικά πακέτα ενώ αποτυγχάνει αν υπάρχει έστω και ένα από τα αρχικά πακέτα του πομπού που δεν καταφέρει να ανακτήσει. Εάν σε κάποιο βήμα του αλγορίθμου δεν έχουμε ανακτήσει όλα τα αρχικά πακέτα και δεν υπάρχει κάποιο κωδικοποιημένο πακέτο με βαθμό ένα τότε σημαίνει ότι η αποκωδικοποίηση απέτυχε και δεν μπορεί ο δέκτης να ανακτήσει όλα τα αρχικά δεδομένα [L02;M03].

Και η διαδικασία αποκωδικοποίησης, όπως και η διαδικασία κωδικοποίησης μπορεί να περιγραφεί με τη χρήση γράφων. Αρχικά εντοπίζουμε έναν κόμβο ελέγχου ο οποίος έχει βαθμό ίσο με ένα. Αυτό σημαίνει ότι προσπίπτει σε αυτόν μία μόνο ακμή και η τιμή του ισούται με την τιμή του κόμβου δεδομένων με τον οποίο συνδέεται. Έτσι ανακτάται η τιμή του αρχικού αυτού πακέτου. Στη συνέχεια, πηγαίνουμε στους κόμβους ελέγχους με τους οποίους συνδέεται ο κόμβος δεδομένων του οποίου την τιμή μόλις ανακτήσαμε και εφαρμόζουμε την πράξη XOR. Έτσι μπορούμε πλέον να αφαιρέσουμε τις ακμές αυτές από τον γράφο αφού πλέον αυτός ο κόμβος δεδομένων είναι γνωστός. Έπειτα βρίσκουμε τον επόμενο κόμβο ελέγχου που έχει βαθμό ίσο με ένα. Πηγαίνουμε στον κόμβο δεδομένων με τον οποίο συνδέεται με ακμή και εφαρμόζουμε την ίδια διαδικασία. Ο αλγόριθμος υλοποιείται έως ότου έχουμε αποκωδικοποιήσει όλους του κόμβους δεδομένων που έχουμε στο γράφο [M08].

1. Βρες ένα κωδικοποιημένο πακέτο t_n το οποίο έχει ακριβώς ένα γείτονα s_k
2. Θέσε $s_k = t_n$ αφού το πακέτο αποτελεί αντίγραφο του γείτονα.

3. Εφάρμοσε XOR σε όλα τα κωδικοποιημένα πακέτα t_n που έχουν γείτονα το s_k .
4. Αφαίρεσε το s_k από γείτονά τους και μείωσε τον βαθμό του καθενός κατά ένα
5. Επανάλαβε το βήμα ένα έως ότου υπολογιστούν όλα τα αρχικά πακέτα.

Πίνακας 2 - Αλγόριθμος αποκωδικοποίησης του LT κώδικα

Η διαδικασία αποκωδικοποίησης, όπως εμείς υλοποιήσαμε στην παρούσα διπλωματική εργασία, έχει κάποιες μικρές διαφορές ώστε να μπορούμε να την εφαρμόζουμε σε σενάρια δικτύων όπου εφαρμόζουμε κωδικοποίηση χωρίς ρυθμό (rateless) και παράγονται κωδικοποιημένα πακέτα καθ' όλη τη διάρκεια της αποστολής τους, έως ότου ο δέκτης μπορέσει να ανακτήσει όλα τα αρχικά πακέτα δεδομένων. Αυτή η προσέγγιση ονομάζεται on the fly. Ο τροποποιημένος LT κώδικας από εδώ και στο εξής θα αναγράφεται ως LT-M

Για να το πετύχουμε αυτό, κάθε φορά που λαμβάνει ο δέκτης ένα κωδικοποιημένο πακέτο μαζί με τους γείτονες και τον βαθμό του, ελέγχει για ποιους από τους γείτονες γνωρίζει τα δεδομένα τους. Στη συνέχεια, κρατάμε σε έναν πίνακα τις πληροφορίες για τα δεδομένα όλων των πακέτων που ανακτώνται με επιτυχία. Ένας γείτονας πρακτικά αποτελεί έναν δείκτη σε αυτόν τον πίνακα. Έτσι, όταν λαμβάνουμε νέα πακέτα, μπορούμε να δούμε ποιοι από τους γείτονές τους έχουν ήδη αποκωδικοποιηθεί από κάποιο προηγούμενο πακέτο. Παίρνουμε τα δεδομένα τους από τον πίνακα και εφαρμόζουμε XOR με το κωδικοποιημένο πακέτο μειώνοντας ταυτόχρονα και τον βαθμό του κατά ένα για κάθε γείτονα που γνωρίζουμε. Αν μετά από αυτή τη διαδικασία προκύψει ότι ο βαθμός του κωδικοποιημένου πακέτου είναι μηδέν, αυτό σημαίνει ότι όλοι οι γείτονες του, τα αρχικά πακέτα δηλαδή που χρησιμοποιήθηκαν για να το παράξουν, ήταν ήδη γνωστά. Έτσι το πακέτο αυτό θεωρείται άχρηστο και απορρίπτεται. Αν όμως ο βαθμός μετά από αυτή την επεξεργασία των γειτόνων είναι μεγαλύτερος του ένα, τότε κρατάμε το κωδικοποιημένο πακέτο και ενημερώνουμε έναν πίνακα γειτνίασης που τον χρησιμοποιούμε σαν γράφο, ώστε να ξέρουμε κάθε κωδικοποιημένο πακέτο με ποια από τα αρχικά πακέτα συνδέεται. Τέλος, αν ο βαθμός του πακέτου αφότου επεξεργαστούμε όλους τους γνωστούς του γείτονες προκύπτει να είναι ένα, τότε όπως εξηγήσαμε, αποτελεί αντίγραφο του μοναδικού του γείτονα και έτσι ανακτά άμεσα τη πληροφορία του πακέτου ο δέκτης. Και τότε υλοποιούμε πρακτικά τον αλγόριθμο που περιγράψαμε.

Η μόνη διαφορά είναι ότι ο αρχικός έλεγχος των γειτόνων γίνεται κάθε φορά που λαμβάνει ένα κωδικοποιημένο πακέτο ο δέκτης.

Παρατηρούμε ότι πολύ σημαντικό ρόλο στην απόδοση της αποκωδικοποίησης παίζει η συνάρτηση κατανομής πιθανοτήτων που εφαρμόζουμε, γιατί από αυτή κρίνεται κατά πόσο μπορεί να υπάρχει σε κάθε βήμα του αλγορίθμου ένα κωδικοποιημένο πακέτο με βαθμό ίσο με ένα. Επομένως η γενικότερη απόδοση και επιτυχία του κώδικα εξαρτάται σε μεγάλο βαθμό από την σωστή επιλογή κατανομής. Όχι μόνο για να μπορούμε να βρίσκουμε σε κάθε βήμα ένα πακέτο με βαθμό ένα, αλλά και για να περιορίζεται το πλήθος των κωδικοποιημένων πακέτων που θα χρειαστούν για την πλήρη αποκωδικοποίηση όλων των πακέτων. Ιδανικά επιδιώκουμε το πλήθος των κωδικοποιημένων πακέτων που θα χρειαστούν να είναι κοντά στο πλήθος των αρχικών πακέτων που θέλουμε να στείλουμε. Η κατανομή πιθανοτήτων που επιλέγεται στους LT-M κώδικες, ανήκει στην κατηγορία συναρτήσεων που λέγεται Κατανομή Βαθμού (Degree Distribution). Ως κατανομή βαθμού ορίζουμε $p(d)$, την πιθανότητα ένα κωδικοποιημένο πακέτο να έχει βαθμό ίσο με d , για κάθε ακέραιο αριθμό d [L02;M03].

Για την επιλογή κατάλληλης κατανομής βαθμού πρέπει να λάβουμε υπόψη δύο παραμέτρους:

- Πρέπει να χρησιμοποιήσουμε όσο το δυνατόν λιγότερα κωδικοποιημένα πακέτα, ώστε να υπάρχει μεγαλύτερη πιθανότητα πλήρους αποκωδικοποίησης στον δέκτη στέλνοντας πακέτα που να μην ξεπερνάνε κατά πολύ σε πλήθος τα αρχικά [L02].
- Πρέπει σε κάθε βήμα να υπάρχει τουλάχιστον ένα κωδικοποιημένο πακέτο με βαθμό ίσο με ένα. Αυτό σημαίνει ότι η κατανομή που θα επιλέξουμε θα πρέπει να αυξάνει το βάρος στους μικρούς βαθμούς έτσι ώστε να έχουν μεγαλύτερη πιθανότητα εμφάνισης [L02].

Μία κατανομή βαθμού που προτάθηκε και φάνηκε να έχει μερικές από τις ιδιότητες που χρειάζονται ήταν η Ιδανική Κατανομή Soliton (Ideal Soliton Distribution) η οποία φαίνεται παρακάτω:

$$\cdot \rho(1) = \frac{1}{k}$$

$$\cdot \text{Για όλα τα } i = 2, \dots, k, \rho(i) = \frac{1}{i(i-1)}$$

όπου k είναι το πλήθος των αρχικών πακέτων που θέλει να στείλει ο πομπός [L02;M08].

Ωστόσο, αποδείχθηκε αρκετά εύθραυστη αυτή η κατανομή στην πράξη χωρίς να πετυχαίνει ικανοποιητικά αποτελέσματα όσον αφορά την απόδοση του κώδικα. Αυτό

συμβαίνει γιατί όταν έχουμε μεγάλο πλήθος δεδομένων, η πιθανότητα ένα πακέτο να έχει βαθμό ένα που δίνεται από τον τύπο $1/k$, θα είναι πολύ μικρή [L02].

Έχει αποδειχθεί ότι η κατανομή Robust Soliton (Robust Soliton Distribution) έχει καλύτερα αποτελέσματα στην πράξη από την Ideal Soliton κατανομή καθώς μας εξασφαλίζει ότι σε κάθε βήμα του αλγορίθμου αποκωδικοποίησης θα υπάρχουν περισσότερα από ένα πακέτα με βαθμό ίσο με ένα ώστε να συνεχίσουν την αποκωδικοποίηση [L02;M03].

Θεωρούμε ότι δ είναι η επιτρεπόμενη πιθανότητα αποτυχίας του αλγορίθμου αποκωδικοποίησης όταν θέλουμε να ανακτήσουμε K κωδικοποιημένα πακέτα. Η κατανομή αυτή έχει σχεδιαστεί ως εξής [L02]:

$$\tau(i) = \begin{cases} \frac{R}{i \cdot k}, & i = 1, \dots, \frac{k}{R-1} \\ \frac{\ln\left(\frac{R}{\delta}\right)}{k}, & i = \frac{k}{R} \\ 0, & i = \frac{k}{R+1}, \dots, k \end{cases}$$

Όπου $R = c \cdot \ln\left(\frac{k}{\delta}\right) \cdot \sqrt{k}$, c είναι κάποια σταθερά μεγαλύτερη του μηδενός και k είναι το πλήθος των αρχικών πακέτων. Στην $\tau(i)$ πρέπει να προσθέσουμε την ιδανική κατανομή soliton $p(i)$ και έπειτα να κανονικοποιήσουμε. Έτσι η κατανομή Robust Soliton $\mu(\cdot)$ υπολογίζεται ως εξής:

$$\begin{aligned} \beta &= \sum_{i=1}^k \rho(i) + \tau(i) \\ \text{Για κάθε } i=1, \dots, k, \mu(i) &= \frac{(\rho(i) + \tau(i))}{\beta} \end{aligned}$$

Με την χρήση της Robust Soliton κατανομής βαθμού για συγκεκριμένες τιμές στις παραμέτρους c και δ σχηματίζεται ένας πίνακας όπου το i -οστό στοιχείο του δείχνει τη πιθανότητα το κωδικοποιημένο πακέτο που θα προκύψει να έχει βαθμό i . Η κατανομή αυτή πετυχαίνει ικανοποιητικότερη απόδοση συγκριτικά με την ιδανική κατανομή Soliton μιας και δίνει μεγαλύτερη πιθανότητα εμφάνισης βαθμού ίσο με ένα για μεγάλο πλήθος αρχικών πακέτων k . [L02]

Οι LT-M κώδικες λειτουργούν εξ ορισμού μόνο σε end-to-end σενάρια, χωρίς ενδιάμεσους κόμβους. Η εφαρμογή τους σε multi-hop σενάρια απαιτεί προσαρμογές και επιπλέον τεχνικές για να διατηρήσουν την αποδοτικότητά τους. Αντίθετα, για τους BATS

κώδικες απαιτείται εξ ορισμού ένα multi-hop σενάριο για να εκτελείται η διαδικασία της επανακωδικοποίησης στον ενδιάμεσο κόμβο. Ωστόσο το RLNC μπορεί να εφαρμοστεί και στις δύο περιπτώσεις.

3.4 Batched Sparse Codes (BATS)

Όπως προαναφέραμε, ένας BATS κώδικας αποτελείται από έναν εξωτερικό κώδικα και έναν εσωτερικό που πραγματοποιούν πράξεις πάνω σε ένα πεπερασμένο πεδίο F_{2^s} . Ο εξωτερικός κώδικας, αποτελεί ουσιαστικά μία γενίκευση πίνακα ενός κώδικα πηγής και για αυτό το λόγο μπορεί να παράγει πακέτα χωρίς ρυθμό. Ομαδοποιεί τα πακέτα που θέλουμε να στείλουμε σε παρτίδες (*batches*), καθεμία από τις οποίες έχει ένα μέγεθος M .

Όταν το μέγεθος των παρτίδων M είναι ίσο με ένα, τότε ο εξωτερικός κώδικας περιορίζεται σε έναν απλό κώδικα πηγής και πιο συγκεκριμένα γίνεται LT κώδικας. Όλα αυτά θα τα αναλύσουμε στη συνέχεια. [YY14]

Προτού περιγράψουμε τον τρόπο λειτουργίας του εξωτερικού κώδικα, θεωρούμε ότι θέλουμε να στείλουμε K πακέτα δεδομένων σε έναν δέκτη [YY14]. Αυτό που κάνει ο εξωτερικός κώδικας είναι να παράγει κωδικοποιημένα πακέτα σε *batches*. Έστω X_i η i -οστη παρτίδα που παράγεται από ένα υποσύνολο $B_i \subset B$ ως εξής:

$$X_i = B_i G_i$$

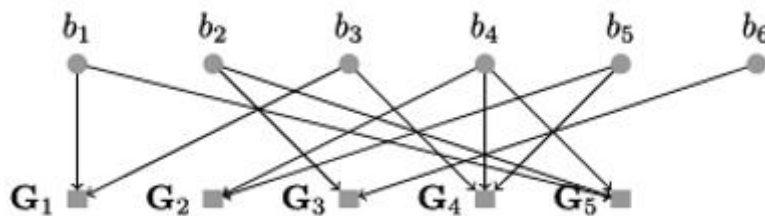
Σαν B ορίζουμε το σύνολο όλων των αρχικών πακέτων που θέλουμε να στείλουμε και το G_i είναι ένας πίνακας που ονομάζεται πίνακας γεννήτριας (*generation matrix*) της i -οστης παρτίδας [YY14]. Επειδή θέλουμε η παρτίδα να έχει M κωδικοποιημένα πακέτα, καταλαβαίνουμε ότι οι στήλες του πίνακα πρέπει να είναι M . Ο σχηματισμός του B_i καθορίζεται από μία κατανομή βαθμών $\Psi = (\Psi_0, \Psi_1, \dots, \Psi_K)$ ή οποία δουλεύει ως εξής:

1. Γίνεται δειγματοληψία της κατανομής Ψ η οποία επιστρέφει έναν βαθμό d_i με μία πιθανότητα Ψ_{d_i} , ο οποίος αποτελεί τον βαθμό αυτής της παρτίδας,
2. Επιλέγονται ομοιόμορφα τυχαία d_i από τα αρχικά πακέτα τα οποία απαρτίζουν το σύνολο B_i .

Όπως και τους LT κώδικες, είναι πολύ κρίσιμη η επιλογή και ο σχεδιασμός μίας κατάλληλης συνάρτησης κατανομής βαθμών. Ο πίνακας γεννήτριας G_i όπως προκύπτει από τα παραπάνω, θα έχει διαστάσεις $d_i \times M$. Τα στοιχεία του πίνακα μπορούν να

επιλεχθούν τυχαία και ανεξάρτητα. Κάτι τέτοιο δεν διευκολύνει μόνο την ανάλυση τους αλλά είναι και πιο εύκολο να εφαρμοστεί. Στην δική μας υλοποίηση αποφασίσαμε να χρησιμοποιήσουμε τη γεννήτρια που παρήγαγε συντελεστές στο RLNC, επειδή και εδώ τα κωδικοποιημένα πακέτα της παρτίδας προκύπτουν με γραμμικούς συνδυασμούς. Εφόσον έχουν επιλεγεί d πακέτα από τα αρχικά, τα οποία θα συνδυαστούν για την παραγωγή M κωδικοποιημένων πακέτων, μπορούμε να πούμε ότι κάθε στήλη του πίνακα γεννήτριας αποτελεί και διάνυσμα συντελεστών του αντίστοιχου κωδικοποιημένου πακέτου. Έτσι όπως και στο RLNC, μαζί με κάθε κωδικοποιημένο πακέτο, στέλνουμε και στο αντίστοιχο διάνυσμα κωδικοποίησης. Επιπλέον στέλνουμε και τα αρχικά πακέτα d που χρησιμοποιήθηκαν για την παραγωγή της παρτίδας καθώς και τον βαθμό της παρτίδας. Όπως αναφέραμε και προηγουμένως, όταν το μέγεθος παρτίδας M ισούται με ένα, τότε η διαδικασία κωδικοποίησης ταυτίζεται με τους LT κώδικες γιατί ο πίνακας μετασχηματισμού θα έχει μία στήλη η οποία θα περιέχει τα d επιλεγμένα πακέτα από την κατανομή βαθμού. Θα είναι δηλαδή σαν τα αντίστοιχα πακέτα-γείτονες. [YY17]

Η διαδικασία κωδικοποίησης μπορεί να περιγραφεί και με ένα γράφο Tanner. Ο γράφος θα περιέχει K κόμβους μεταβλητών, όπου κάθε ένας θα αντιστοιχεί σε ένα από τα αρχικά πακέτα που θέλουμε να στείλουμε, και n κόμβους ελέγχου. Οι κόμβοι ελέγχου αντιστοιχούν στα κωδικοποιημένα πακέτα μιας παρτίδας. Με κάθε κόμβο ελέγχου συνδέεται και ο αντίστοιχος πίνακας γεννήτριας της παρτίδας. (εικόνα 3) [YY14].



Εικόνα 3 - Γράφος Tanner για τον εξωτερικό κώδικα BATS. Οι κόμβοι στη πρώτη σειρά αποτελούν τους κόμβους μεταβλητών που αντιστοιχούν στα αρχικά πακέτα. Οι κόμβοι στη δεύτερη σειρά αντιστοιχούν στους κόμβους ελέγχου που παράγονται από τον εξωτερικό κώδικα.

Οι παρτίδες που παράγονται από τον εξωτερικό κώδικα, μεταδίδονται σε ένα δίκτυο που εφαρμόζει γραμμική κωδικοποίηση δικτύου με πιθανώς περισσότερους από έναν παραλήπτες. Αν θεωρήσουμε ότι έχουμε έναν κόμβο δέκτη, θα περιγράψουμε τώρα τη διαδικασία του εσωτερικού κώδικα που εφαρμόζεται στους ενδιάμεσους κόμβους.

Έστω ότι H_i είναι ο πίνακας μεταφοράς της i -οστής παρτίδας και Y_i είναι τα κωδικοποιημένα πακέτα της παρτίδας που έφτασαν στον δέκτη [YY14]. Έχουμε ότι

$$Y_i = X_i H_i = B_i G_i H_i \quad (\text{σχέση 7})$$

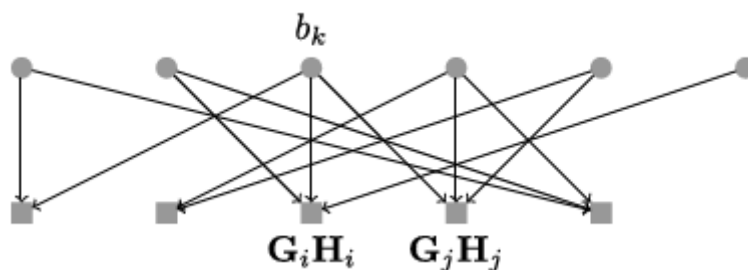
Ο πίνακας μεταφοράς έχει M γραμμές και το πλήθος των στηλών του αντιστοιχεί στον αριθμό των κωδικοποιημένων πακέτων που έλαβε για την αντίστοιχη παρτίδα. Επομένως οι στήλες είναι το πολύ M στη περίπτωση που δεν υπήρξε καμία απώλεια στο κανάλι και έφτασαν όλα τα πακέτα της παρτίδας στον δέκτη. Υποθέτουμε ότι ο πίνακας μεταφοράς είναι γνωστός για την αποκωδικοποίηση. Αυτή την πληροφορία μπορούμε να τη λάβουμε από τα διανύσματα συντελεστών τα οποία αποστέλλονται μαζί με κάθε κωδικοποιημένο πακέτο. Αυτός είναι και ο λόγος που δεν μπορεί ένα κωδικοποιημένο πακέτο να αποτελεί γραμμικό συνδυασμό πακέτων περισσότερων από μίας παρτίδας. Εφαρμόζοντας κωδικοποίηση δικτύου εντός των παρτίδων πετυχαίνουμε τα εξής πλεονεκτήματα [YY14]:

- Η πολυπλοκότητα της κωδικοποίησης σε έναν ενδιάμεσο κόμβο είναι $O(MT)$ πράξεις στο πεπερασμένο πεδίο για κάθε πακέτο. Επομένως δεν εξαρτάται από το πλήθος K των αρχικών πακέτων.
- Το πλεόνασμα των διανυσμάτων συντελεστών περιορίζεται από το M . Όταν το μέγεθος των πακέτων T είναι πολύ μεγαλύτερο από το M , τότε αυτό γίνεται αμελητέο.
- Εφόσον τα πακέτα από διαφορετικές παρτίδες δεν κωδικοποιούνται μαζί, δεν χρειάζεται να αποθηκεύουμε στους ενδιάμεσους κόμβους όλες τις παρτίδες.

Η διαδικασία της επανακωδικοποίησης στους ενδιάμεσους κόμβους μπορεί να γίνει εφαρμόζοντάς την σε κάθε ένα από τα πακέτα που λαμβάνονται ξεχωριστά έως ότου ξανασταλούν M κωδικοποιημένα πακέτα στον επόμενο κόμβο [YY14]. Αυτό όμως όταν θέλουμε να στείλουμε τα πακέτα μας και θα πρέπει να πραγματοποιηθούν πολλά άλματα στο δίκτυο, αυξάνει την πιθανότητα να λαμβάνουμε γραμμικά εξαρτημένους συντελεστές κωδικοποίησης. Επομένως εμείς υλοποιήσαμε τις συναρτήσεις μας έτσι ώστε για να εφαρμόσουμε κωδικοποίηση στους ενδιάμεσους κόμβους, θα πρέπει πρώτα να σταλούν όλα τα πακέτα της παρτίδας. Τότε εφαρμόζουμε τη κωδικοποίηση σε όλα τα πακέτα μαζί και μετά στέλνουμε τα καινούρια κωδικοποιημένα πακέτα στον επόμενο κόμβο. Αυτή η υλοποίηση προσθέτει μία επιπλέον M καθυστέρηση σε κάθε άλμα, αλλά μειώνει τον κίνδυνο να έχουμε γραμμικώς εξαρτημένα πακέτα.

Η διαδικασία αποκωδικοποίησης περιγράφεται μέσα από τον αλγόριθμο αποκωδικοποίησης Belief Propagation (BP). Ο κόμβος δέκτης προσπαθεί να αποκωδικοποιήσει τα αρχικά πακέτα χρησιμοποιώντας τα \mathbf{Y}_i και τη γνώση από τα \mathbf{G}_i και \mathbf{H}_i . Ισοδυναμεί η αποκωδικοποίηση με την επίλυση του συστήματος των γραμμικών εξισώσεων που προκύπτουν από τη σχέση 7. Η διαδικασία αποκωδικοποίησης belief

propagation, μπορεί να περιγραφεί καλύτερα με τη χρήση ενός διμερούς γράφου (Εικόνα 4) [YY14;YY17].



Εικόνα 4 - Ένας γράφος αποκωδικοποίησης. Οι κόμβοι στη πρώτη γραμμή αποτελούν τους κόμβους πληροφορίας και αναπαριστούν τα αρχικά πακέτα. Οι κόμβοι στη δεύτερη γραμμή είναι οι κόμβοι ελέγχου που αποτελούν την παρτίδα.

Ένας κόμβος ελέγχου i λέγεται αποκωδικοποιήσιμος όταν η βαθμίδα του $G_i H_i$ είναι ίση με τον βαθμό της παρτίδας. Αν ο κόμβος ελέγχου i είναι αποκωδικοποιήσιμος, τότε για να ανακτήσουμε το B_i αρκεί να λύσουμε το γραμμικό σύστημα εξισώσεων $Y_i = B_i G_i H_i$ το οποίο έχει μοναδική λύση σε αυτή τη περίπτωση. Έτσι ανακτούμε τα d_i πακέτα που απαρτίζουν το B_i και μπορούμε να αντικαταστήσουμε τις τιμές τους στις παρτίδες που δεν έχουν αποκωδικοποιηθεί ακόμα. Έστω ότι ανακτήσαμε το b_k πακέτο το οποίο θα αντιστοιχεί σε έναν κόμβο μεταβλητής στον γράφο. Αν αυτός ο κόμβος έχει μία μόνο ακμή που συνδέεται με τον κόμβο ελέγχου i , τότε αφαιρούμε τον κόμβο μεταβλητής k . Αν ο κόμβος μεταβλητής k συνδέεται και με έναν ακόμα κόμβο ελέγχου, διαφορετικό από τον i , τότε μειώνουμε τον βαθμό του κόμβου κατά ένα και αφαιρούμε τη γραμμή του πίνακα γεννήτριας του καινούριου κόμβου που αντιστοιχεί στον κόμβο k . [YY14;YY17]

Ωστόσο εμείς ακολουθήσαμε μία ελαφρώς διαφορετική προσέγγιση στην αποκωδικοποίηση η οποία εφαρμόζει απαλοιφή Gauss, όπως και στους RLNC κώδικες. Αρχικά, για να ελέγξουμε αν ένας κόμβος είναι αποκωδικοποιήσιμος, ελέγχουμε πόσα κωδικοποιημένα πακέτα για την παρτίδα αυτή έχουμε λάβει και πόσα από τα d αρχικά πακέτα που χρησιμοποιήθηκαν για την παρασκευή της έχουν αποκωδικοποιηθεί μέχρι εκείνη τη στιγμή. Αν το άθροισμα αυτών είναι μεγαλύτερο ή ίσο από τον βαθμό της παρτίδας, σημαίνει ότι έχουμε αρκετή πληροφορία ώστε να αποκωδικοποιηθεί η παρτίδα. Σε αυτή τη περίπτωση εφαρμόζουμε απαλοιφή Gauss με τον ίδιο τρόπο που εφαρμόζουμε στο RLNC. Απλά στις θέσεις των πακέτων που έχουν αποκωδικοποιηθεί προηγουμένως αντικαθιστούμε τα κωδικοποιημένα πακέτα με τα αποκωδικοποιημένα και σαν διάνυσμα συντελεστών παίρνω ένα μοναδιαίο διάνυσμα, όπου το ένα αντιστοιχεί

στη θέση που έχει το πακέτο αυτό στον πίνακα. Μόλις ένα πακέτο αποκωδικοποιηθεί πλήρως, τότε ελέγχουμε ποιες από τις παρτίδες που υπάρχουν στον δέκτη είχε κάποιο από τα πακέτα που αποκωδικοποιήσαμε στα επιλεγμένα πακέτα που χρησιμοποιήθηκαν για τη παρασκευή της και αντίστοιχα κάνουμε τον έλεγχο για να δούμε μήπως τώρα μπορεί να αποκωδικοποιηθεί. Ο τροποποιημένος BATS κώδικας από εδώ και στο εξής θα αναγράφεται ως BATS-M

Όπως και στους LT κώδικες, πολύ σημαντικό ρόλο στην απόδοση του κώδικα παίζει η κατανομή βαθμού που θα χρησιμοποιήσουμε. Για να ξεκινήσει η αποκωδικοποίηση χρειάζεται να λάβουμε μία ομάδα με βαθμό d μικρότερο ή ίσο από το μέγεθος παρτίδας M . Ωστόσο αυτό από μόνο του δεν αρκεί, γιατί τα d πακέτα που θα χρησιμοποιηθούν στην παρτίδα, επιλέγονται ομοιόμορφα τυχαία από το συνολικό πλήθος των πακέτων K που θέλουμε να στείλουμε. Για παράδειγμα, έστω ότι θέλουμε να στείλουμε 10000 πακέτα και έχουμε ορίσει σαν μέγεθος παρτίδας M ίσο με 5. Αν η κατανομή μας φέρει βαθμό μικρότερο ή ίσο του 5, αυτό σημαίνει ότι θα αποκωδικοποιηθεί η συγκεκριμένη παρτίδα. Για να αποκωδικοποιηθεί όμως μία παρτίδα με βαθμό 10, για παράδειγμα, θα πρέπει τουλάχιστον τα 5 από τα 10 πακέτα που θα έχουν επιλεγεί από την κατανομή βαθμού να έχουν αποκωδικοποιηθεί προηγουμένως. Εφόσον όμως εμείς επιλέγουμε τα πακέτα από το συνολικό πλήθος πακέτων, είναι μικρές οι πιθανότητες τα 5 πακέτα που αποκωδικοποιήσαμε αρχικά να συμπίπτουν με κάποια από τα επιλεγμένα πακέτα επόμενης παρτίδας. Ακόμα δεν είναι βέλτιστο να χρησιμοποιούμε κατά κόρον πολύ μικρούς βαθμούς γιατί δεν αξιοποιούνται στο μέγιστο τα πλεονεκτήματα του BATS-M κώδικα [YY17].

Στους BATS-M κώδικες δεν υπάρχει κάποια «βέλτιστη» κατανομή βαθμού που να θεωρείται ιδανική, όπως υπήρχε η Robust Soliton κατανομή στους LT κώδικες [YY14]. Αυτό συμβαίνει γιατί οι κώδικες αυτοί εφαρμόζονται κυρίως σε δίκτυα με πολλαπλά άλματα (multi-hop), όπου οι ενδιαμέσοι κόμβοι εφαρμόζουν επανακωδικοποίηση. Επίσης, οι συνθήκες του δικτύου, όπως ο ρυθμός απώλειας πακέτων για κάθε άλμα αλλά και η τοπολογία του δικτύου επηρεάζουν την απόδοσή τους. Μία «βέλτιστη» κατανομή βαθμού θα πρέπει να διαφέρει ανάλογα τις συνθήκες αυτές. Ακόμα, επειδή οι κώδικες αυτοί λειτουργούν με batches, η κατανομή βαθμού αφορά τους βαθμούς εντός κάθε παρτίδας. Αυτό σημαίνει ότι η λογική της κατανομής εξαρτάται και από το μέγεθος M της παρτίδας. Επομένως, θα πρέπει να σχεδιάζονται κατανομές βαθμών ανάλογα με τις συνθήκες του εκάστοτε δικτύου για να έχουμε τη μεγαλύτερη αποδοτικότητα του κώδικα αυτού. Στη διπλωματική αυτή, για την πειραματική μελέτη των κωδίκων BATS-M εφαρμόσαμε την κατανομή Robust Soliton αυτή δηλαδή που χρησιμοποιήσαμε και στους LT κώδικες.

Πολύ σημαντικό ρόλο στη λειτουργία του BATS-M κώδικα παίζει το μέγεθος των batches, το οποίο μπορεί να είναι διαφορετικό για κάθε παρτίδα. Στη πειραματική αξιολόγηση του κώδικα που υλοποιήσαμε αποφασίσαμε να διατηρήσουμε σταθερό μέγεθος για τις παρτίδες το οποίο θα είναι ίσο με τέσσερα. Η απόφαση αυτή προέκυψε έπειτα από πειράματα που έγιναν. Με τη χρήση της Robust Soliton κατανομής βαθμών, μεγαλύτερο βάρος έχουν οι μικροί βαθμοί, επομένως η χρήση μεγάλου μεγέθους στις παρτίδες είχε σαν αποτέλεσμα να παράγονται πολλά κωδικοποιημένα πακέτα για ένα batch από ένα μικρό αριθμό αρχικών πακέτων. Αυτό αύξανε κατά πολύ τη πλεονάζουσα πληροφορία, με αποτέλεσμα να καθυστερεί πολύ η παράδοση πακέτων.

Επειδή οι BATS-M κώδικες έχουν σχεδιαστεί να λειτουργούν σε multi-hop δίκτυα, δεν αποτελούν βασικό μέτρο σύγκρισης στη περίπτωση ενός single-hop δικτύου. Εμείς ωστόσο θα εξετάσουμε την απόδοση των BATS-M κωδίκων και σε single-hop σενάριο, χωρίς όμως να εφαρμόσουμε επανακωδικοποίηση. Η βασική σύγκριση του BATS-M κώδικα με τον RLNC θα αφορά σε δίκτυο με δύο διαδοχικά κανάλια όπου θα επωφεληθούμε των πλεονεκτημάτων της επανακωδικοποίησης που εφαρμόζεται στον ενδιάμεσο κόμβο.

Κεφάλαιο 4. Πειραματική αξιολόγηση

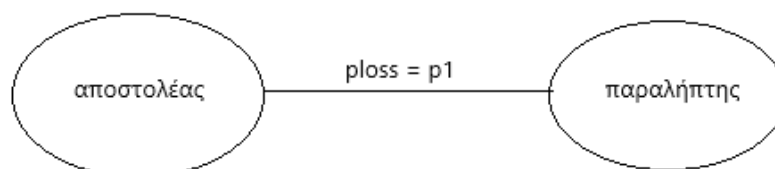
Στο προηγούμενο κεφάλαιο περιγράψαμε τρεις τεχνικές κωδικών απαλοιφής (erasure codes). Συγκεκριμένα ασχοληθήκαμε με τις μεθόδους RLNC, LT-M και BATS-M. Παρουσιάσαμε τον αλγόριθμο κωδικοποίησης και αποκωδικοποίησης κάθε μιας από αυτές και τον γενικότερο τρόπο λειτουργίας τους. Ωστόσο ο στόχος της διπλωματικής αυτής είναι να δούμε πώς επιδρά κάθε μία από αυτές σε συγκεκριμένα σενάρια επικοινωνίας και να συγκρίνουμε τις επιδόσεις τους.

4.1 Τοπολογίες και χαρακτηριστικά τους

Οι τοπολογίες που χρησιμοποιήθηκαν είναι οι εξής:

- Τοπολογία με ένα κανάλι (single-hop): Ο πομπός κωδικοποιεί τα δεδομένα και τα στέλνει στον δέκτη απευθείας, χωρίς ενδιάμεσους κόμβους, όπου και αποκωδικοποιούνται (εικόνα 5).
- Τοπολογία με δύο διαδοχικά κανάλια (multi-hop): Τα δεδομένα περνούν από έναν ενδιάμεσο κόμβο και επανακωδικοποιούνται πριν φτάσουν στον δέκτη όπου θα αποκωδικοποιηθούν (εικόνα 6).

Σημαντικό είναι να αναφέρουμε ότι στο multi-hop σενάριο τα δύο κανάλια έχουν την ίδια πιθανότητα σφαλμάτων, δηλαδή $p_1 = p_2 = p_{\text{loss}}$ (εικόνα 6).



Εικόνα 5 - Single-Hop σενάριο



Εικόνα 6 - Multi-Hop σενάριο

Για κάθε σενάριο επικοινωνίας, θα μελετήσουμε την απόδοση των κωδίκων με βάση τα χαρακτηριστικά του καναλιού αλλά και το είδος της κωδικοποίησης που εφαρμόζεται. Ως προς τα χαρακτηριστικά τους καναλιού, υλοποιήσαμε δύο ειδών κανάλια, που είναι τα εξής:

- Κανάλια με ομοιόμορφες απώλειες: Σε αυτά τα κανάλια υπάρχει μία σταθερή πιθανότητα p_{loss} να χαθεί ένα πακέτο. Για κάθε πακέτο, εμείς παίρνουμε ομοιόμορφα τυχαία έναν αριθμό από το ένα μέχρι το εκατό, και αν είναι μικρότερος από τη πιθανότητα απώλειας, τότε χάνεται αυτό το πακέτο. Αντίθετα, αν είναι μεγαλύτερος, τότε αποστέλλεται κανονικά στον δέκτη.
- Κανάλια με απώλειες σε ριπές: Για αυτό το σενάριο υλοποιήσαμε ένα Gilbert-Elliot (GE) κανάλι [G60]. Αυτό το κανάλι, αντί να θεωρεί ότι τα σφάλματα συμβαίνουν τυχαία για κάθε πακέτο ξεχωριστά, αναγνωρίζει ότι τα σφάλματα γίνονται σε ομάδες-ριπές. Υπάρχουν δύο καταστάσεις στο κανάλι αυτό: η καλή κατάσταση και η κακή κατάσταση. Όταν βρισκόμαστε στην καλή κατάσταση, τα πακέτα που στέλνουμε έχουν πολύ χαμηλή πιθανότητα να χαθούν. Στην υλοποίηση μας, θεωρούμε στη πράξη ότι στην καλή κατάσταση, τα πακέτα στέλνονται με πιθανότητα 100%. Αντίθετα, όταν βρισκόμαστε στη κακή κατάσταση, τα πακέτα χάνονται με πολύ μεγάλη πιθανότητα. Στη δική μας περίπτωση θεωρούμε ότι υπάρχει 100% πιθανότητα απώλειας στην κακή κατάσταση. Εκτός από την πιθανότητα απώλειας p_{loss} , εδώ χρειαζόμαστε και την διάρκεια της ριπής. Με αυτές τις παραμέτρους, καθορίζουμε την πιθανότητα μετάβασης από την καλή στην κακή κατάσταση s και την πιθανότητα μετάβασης από την κακή στην καλή κατάσταση q . Σε αυτά τα κανάλια, πρώτα ελέγχουμε αν θα υπάρξει αλλαγή στην κατάσταση και μετά, ανάλογα την καινούρια κατάσταση, το πακέτο στέλνεται στον δέκτη ή χάνεται.

4.2 Σύγκριση αλγορίθμων

Οι LT-M κώδικες εξ ορισμού λειτουργούν με non systematic προσέγγιση, δηλαδή ο πομπός στέλνει μόνο κωδικοποιημένα πακέτα προς τον δέκτη. Για αυτό το λόγο κατά την

πειραματική αξιολόγηση των τεχνικών οι κώδικες αυτοί θα συγκριθούν μόνο με μη συστηματική προσέγγιση του RLNC.

Στο single-hop σενάριο η βασική σύγκριση θα γίνει ανάμεσα στο LT-M και RLNC. Η απόδοση των BATS-M κωδίκων θα εξεταστεί χωρίς επανακωδικοποίηση. Αυτή η προσέγγιση μοιάζει με τους LT-M κώδικες, με τη διαφορά ότι αντί να στέλνουμε ένα κωδικοποιημένο πακέτο τη φορά, θα στέλνουμε ένα batch.

Στο multi-hop σενάριο, η βασική σύγκριση είναι ανάμεσα στο RLNC και BATS-M, όπου θα εφαρμόζουμε επανακωδικοποίηση στον ενδιάμεσο κόμβο. Ωστόσο, θα εξετάσουμε και τους LT-M κώδικες σε αυτό το σενάριο, όπου απλά θα φτάνει ένα πακέτο στον ενδιάμεσο κόμβο και από εκεί θα προωθείται προς τον δέκτη, χωρίς να γίνεται κάποια επεξεργασία του πακέτου στον ενδιάμεσο κόμβο.

Ως προς το είδος κωδικοποίησης και εκεί θα εφαρμόσουμε δύο σενάρια. Ένα από αυτά θα είναι η κωδικοποίηση χωρίς ρυθμό (rateless) που, όπως έχουμε εξηγήσει στο κεφάλαιο δύο, στέλνουμε κωδικοποιημένα πακέτα έως ότου μπορέσει ο αποκωδικοποιητής να αποκωδικοποιήσει όλα τα αρχικά πακέτα. Το δεύτερο σενάριο είναι η κωδικοποίηση με ρυθμό (rate-based) στη οποία, όπως έχουμε εξηγήσει, υπάρχει προκαθορισμένο πλήθος κωδικοποιημένων πακέτων που θα στείλουμε και εξαρτάται από τον ρυθμό κωδικοποίησης (code-rate) που εφαρμόζουμε.

4.3 Μετρικές αξιολόγησης

Στη κωδικοποίηση χωρίς ρυθμό μας ενδιαφέρει η επιτυχής μετάδοση όλων των πακέτων με όση πλεονάζουσα πληροφορία χρειαστεί. Αντίθετα στη κωδικοποίηση με ρυθμό, μας ενδιαφέρει το ποσοστό επιτυχούς παράδοσης πακέτων (*delivery ratio*) με ένα συγκεκριμένο redundancy. Όπως είχαμε αναφέρει, ο ρυθμός κωδικοποίησης γενικά ορίζεται ως $c = \frac{K}{N}$, όπου K είναι τα αρχικά πακέτα που θέλει να στείλει ο πομπός και N είναι το συνολικό πλήθος κωδικοποιημένων πακέτων που αποστέλλονται τελικά και είναι μεγαλύτερο του K. Αυτός είναι και ο τύπος με τον οποίο θα υπολογίζουμε κάθε φορά πόσα πλεονάζοντα πακέτα θα στέλνουμε στο LT-M. Στους BATS-M κώδικες, το συνολικό code rate μπορεί να μετρηθεί ως $c = \frac{N}{(M \cdot s')}$, όπου:

- N: το συνολικό πλήθος αρχικών πακέτων που θέλουμε να στείλουμε
- M: το μέγεθος κάθε παρτίδας
- s': το πλήθος των παρτίδων που θα στείλουμε συνολικά

Από αυτόν τον τύπο του code rate θα υπολογίζουμε κάθε φορά το συνολικό πλήθος των παρτίδων που θα στείλουμε στον δέκτη. Στο RLNC ωστόσο, όπως αναλύσαμε στο κεφάλαιο τρία, τα αρχικά πακέτα, χωρίζονται σε γενεές και η κωδικοποίηση εφαρμόζεται σε κάθε μία ξεχωριστά, όπως και η αποκωδικοποίηση. Επομένως, για να υπολογίζουμε το πλήθος των πλεοναζόντων πακέτων που θα στέλνουμε σε κάθε γενεά θα χρησιμοποιήσουμε τον τύπο του code rate $c = \frac{N}{N + r}$, όπου:

- N: το πλήθος των αρχικών πακέτων που απαρτίζουν τη γενεά
- r: το πλήθος των πλεοναζόντων πακέτων που θα στείλουμε για τη γενεά με μέγεθος N

Σε όλες τις περιπτώσεις, παίρνουμε την πλησιέστερη μεγαλύτερη ακέραια τιμή που θα μας δώσει ο τύπος με τη συνάρτηση ceiling της Python. Στους LT-M κώδικες, δεν θα μπορούσαμε να χωρίζουμε τα πακέτα όπως κάνουμε στο RLNC με τις γενεές γιατί θα είχαμε μεγάλη πιθανότητα να είναι γραμμικώς εξαρτημένα τα κωδικοποιημένα πακέτα που θα παράξουμε ως προς τα πακέτα-γείτονες τους.

Στις rate-based προσεγγίσεις, θέλουμε ακόμα να ελέγξουμε τη περίπτωση όπου η συνολική πλεονάζουσα πληροφορία που μεταδίδεται να είναι ίση με μία έως και τέσσερις φορές των απωλειών που προκύπτουν στο κανάλι. Δηλαδή, αν το ploss ισούται με 5%, η πλεονάζουσα πληροφορία θα πρέπει να ισούται με όσα πακέτα απαιτούνται για να καλύψουν μέχρι και 20% απώλειες στο χειρότερο σενάριο. Σε αυτά τα πειράματα όπως θα δούμε στη συνέχεια, διατηρούμε σταθερό το ploss ίσο με 5% στα κανάλια με ομοιόμορφες απώλειες και στα κανάλια με απώλειες σε ριπές το ploss ίσο με 5% και τη διάρκεια των ριπών ίση με τέσσερα πακέτα, και μεταβάλλουμε κάθε φορά το code rate μειώνοντάς το σαν τιμή, κάτι που αυξάνει τη πλεονάζουσα πληροφορία που αποστέλλεται.

Για την αξιολόγηση της απόδοσης των τεχνικών κωδικοποίησης εξετάζουμε τις ακόλουθες μετρικές:

- Μέση καθυστέρηση εν-σειρά παράδοσης των πακέτων (average in-order delay).
- Goodput = $\frac{\text{συνολικό πλήθος αρχικών πακέτων που παραδόθηκε επιτυχώς}}{\text{συνολικός χρόνος προσομοίωσης}}$ και εκφράζεται σε πακέτα / time slots.
- Ποσοστό επιτυχούς παράδοσης πακέτων (delivery ratio) στις rate-based προσεγγίσεις ή ποσοστό πλεονάζουσας πληροφορίας (packet overhead) στις rateless προσεγγίσεις.

Στην υλοποίηση των πειραμάτων μετράμε τον χρόνο σε χρονοθυρίδες (time slots), θεωρώντας ότι κατά τη διάρκεια μίας χρονοθυρίδας μπορεί να σταλεί ένα μόνο πακέτο.

Επομένως, στο single-hop σενάριο, κάθε φορά που στέλνουμε ένα πακέτο, αυξάνεται ο μετρητής των χρονοθυρίδων κατά ένα. Στο multi-hop σενάριο όμως, στο ίδιο time slot, μπορεί να μεταδίδει πακέτο και ο πομπός στον ενδιάμεσο κόμβο, αλλά και ο ενδιάμεσος κόμβος στον δέκτη. Ωστόσο στο πρώτο time slot, όπως είναι λογικό αποστέλλεται πακέτο μόνο από τον πομπό προς τον ενδιάμεσο κόμβο. Όταν ο πομπός στείλει όλα τα πακέτα προς τον ενδιάμεσο κόμβο, τότε αν υπάρχουν στον ενδιάμεσο κόμβο πακέτα που δεν έχουν σταλεί ακόμα, συνεχίζει η διαδικασία της αποστολής από τον ενδιάμεσο κόμβο προς τον δέκτη έως ότου στείλει όλα τα πακέτα που είχε. Με αυτόν τον τρόπο προσομοιώνουμε σωστά τη λειτουργία ενός multi-hop δικτύου. Στη περίπτωση του BATS-M, όπως είχαμε αναφέρει, ο ενδιάμεσος κόμβος για να ξεκινήσει την επανακωδικοποίηση στην παρτίδα και πριν ξεκινήσει να στέλνει τα πακέτα της παρτίδας στον δέκτη, υπάρχει μία έξτρα καθυστέρηση M σε κάθε άλμα.

Για τον υπολογισμό της εν-σειρά παράδοσης των πακέτων, μας ενδιαφέρει να κρατάμε τον χρόνο λήψης του κάθε πακέτου στον δέκτη. Την καθυστέρηση ενός πακέτου την υπολογίζουμε αφαιρώντας τον χρόνο αποστολής από τον χρόνο λήψης. Όλοι οι χρόνοι μετριοούνται σε time slots. Ας θεωρήσουμε για παράδειγμα ότι θέλουμε να στείλουμε τρία πακέτα, packet1, packet2 και packet3. Αφότου σταλεί και το packet3, στέλνουμε και ένα κωδικοποιημένο πακέτο coded1. Στο time slot $t1$ στέλνουμε το πακέτο packet1, αντίστοιχα στο $t2$ το packet2, στο $t3$ το packet3 και στο $t4$ το coded1. Θεωρούμε ότι το packet1 φτάνει στον δέκτη στο time slot $s1$, το packet2 χάνεται, το packet3 φτάνει στο $s3$ και το κωδικοποιημένο coded1 φτάνει στον δέκτη στο $s4$. Η καθυστέρηση για το πρώτο πακέτο υπολογίζεται κανονικά ως $s1 - t1$. Για το πακέτο 2 εφόσον έχει χαθεί δεν έχουμε πληροφορία. Για το πακέτο 3, παρόλο που έφτασε στον δέκτη τη χρονική στιγμή $s3$, δεν υπολογίζουμε την καθυστέρηση του γιατί μας ενδιαφέρει η εν-σειρά παράδοση και το πακέτο 2 αγνοείται ακόμα. Όταν λάβει ο δέκτης το κωδικοποιημένο πακέτο coded1, έχοντας και την πληροφορία του πρώτου και του τρίτου πακέτου, θα μπορέσει να ανακτήσει τα δεδομένα του δεύτερου πακέτου. Επομένως ο χρόνος παράδοσης για το πακέτο 2 είναι $s4$ και η καθυστέρηση για αυτό το πακέτο θα είναι $s4 - t2$. Και για το πακέτο 3 σαν χρόνο λήψης θα θεωρήσουμε την χρονική στιγμή $s4$, και συνεπώς η καθυστέρησή τους θα είναι $s4 - t3$.

Το goodput, εκφράζει τη ποσότητα των χρήσιμων δεδομένων που παραδίδονται στον δέκτη σε αντίθεση με τη ρυθμ απόδοση (throughput) η οποία μετρά τον συνολικό όγκο δεδομένων που μεταδίδονται στο δίκτυο.

4.4 Βασικοί παράμετροι

Για τα πειράματά μας οι παράμετροι που χρησιμοποιήσαμε είναι οι παρακάτω:

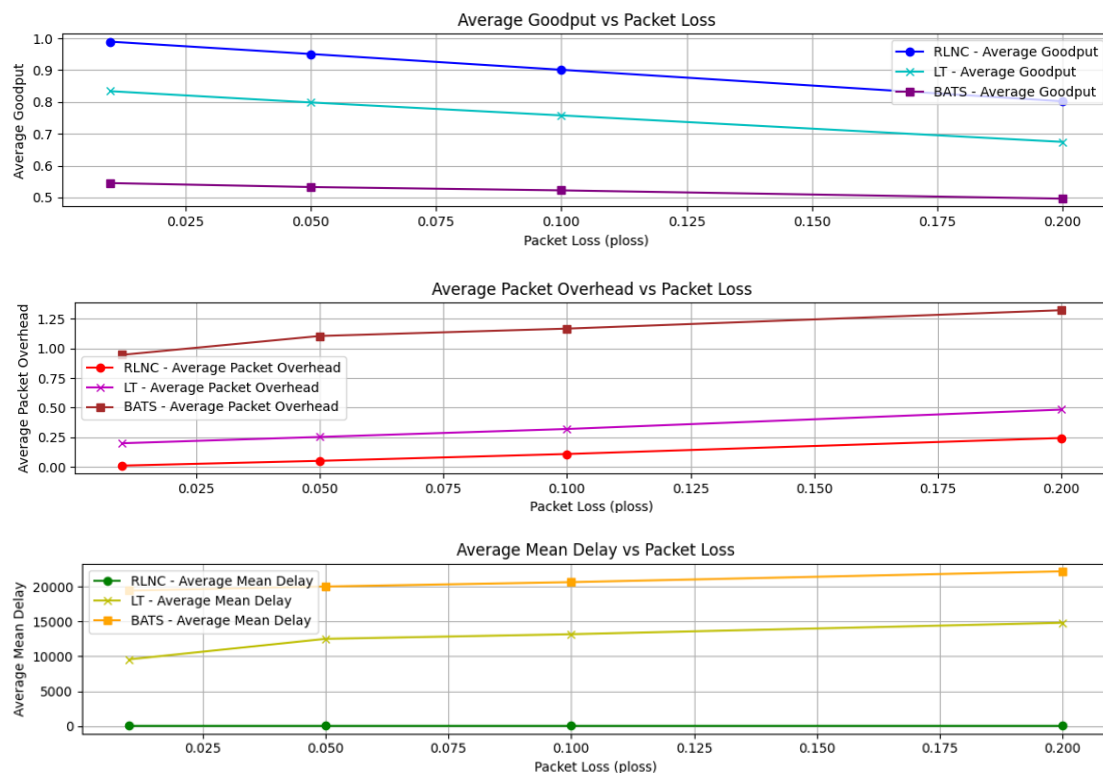
- Πλήθος πακέτων προς μετάδοση: 10000
- Μέγεθος πακέτων σε bytes: 1000
- Πεπερασμένο πεδίο: F_8
- RLNC: μη συστηματική προσέγγιση (non systematic)
- Κατανομή βαθμού στο BATS-M: Robust Soliton Distribution
- Πιθανότητα απώλειας πακέτου στο κανάλι (ploss): 1%, 5%, 10%, 20%.
- Στα Gilbert-Elliott κανάλια η διάρκεια ριπής είναι: 4 και 8 πακέτα.
- Μέγεθος παρτίδας (batch): 4
- Code rate $R1 = 1 - \text{ploss}$ και $R2 = 1 - 2 * \text{ploss}$

Για κάθε πείραμα εκτελέσαμε δέκα επαναλήψεις, εφαρμόζοντας σε κάθε επανάληψη ένα διαφορετικό seed να καθορίζει την αρχική τιμή της γεννήτριας της βιβλιοθήκης της random. Η βιβλιοθήκη αυτή χρησιμοποιείται από τη γεννήτρια της PyErasure που παράγει τυχαίους συντελεστές κωδικοποίησης.

4.5 Κανάλι με ομοιόμορφες απώλειες

4.5.1 Κωδικοποίηση χωρίς ρυθμό (rateless)

Αρχικά εφαρμόσαμε κωδικοποίηση χωρίς ρυθμό και στις τρεις τεχνικές από την οποία πήραμε την εικόνα 8.

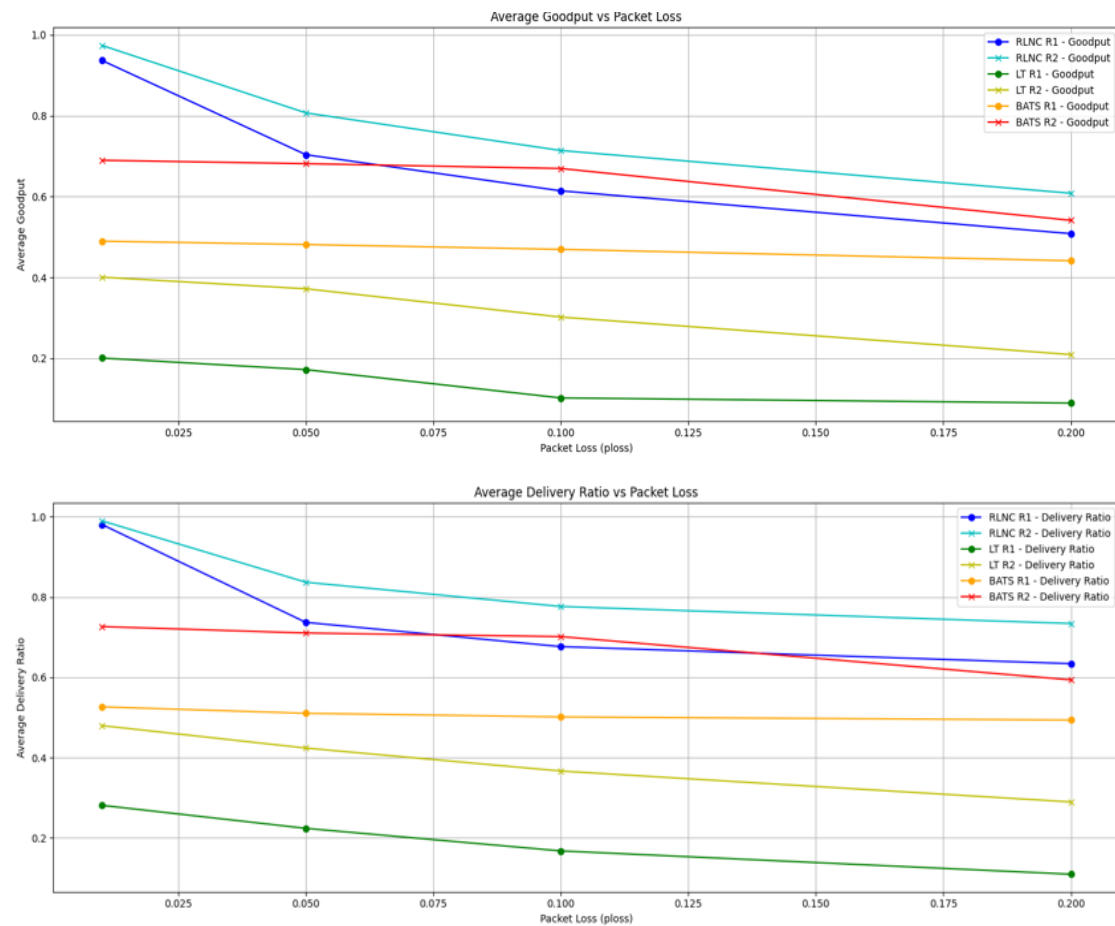


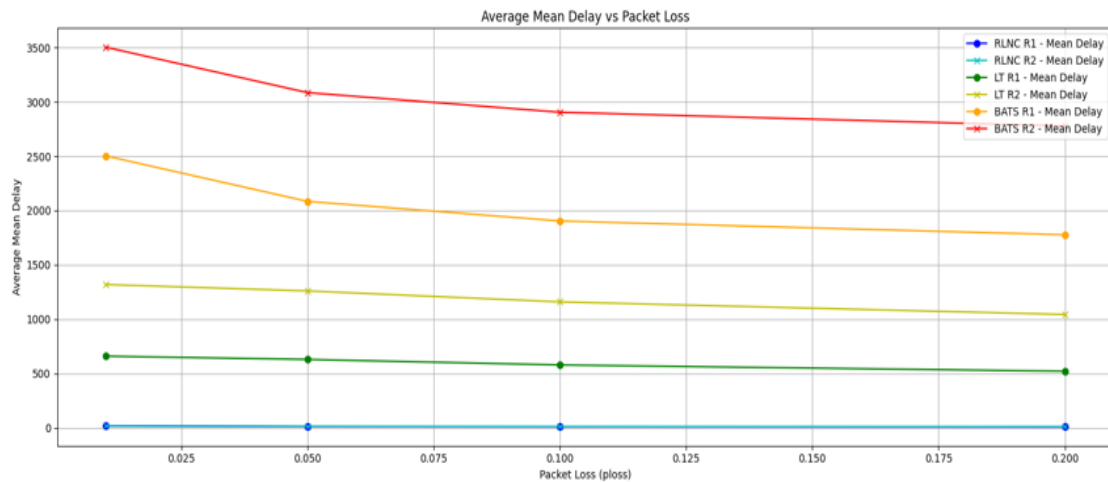
Εικόνα 7 - Αποτελέσματα τεχνικών στη *rateless* προσέγγιση

Για τις τεχνικές RLNC και LT-M παρατηρούμε ιδιαίτερα υψηλά ποσοστά ρυθμού απόδοσης των χρήσιμων δεδομένων (goodput) τα οποία μειώνονται γραμμικά όσο αυξάνονται οι απώλειες στο κανάλι. Στους BATS-M κώδικες ωστόσο παρατηρούμε μία μειωμένη απόδοση συγκριτικά με τις δύο άλλες τεχνικές, κάτι που όπως αναφέραμε οφείλεται στο γεγονός ότι η Robust Soliton κατανομή βαθμών, χωρίς μελέτη και τροποποίηση δεν μπορεί να δουλέψει βέλτιστα σε αυτούς. Ωστόσο, παρατηρούμε μικρότερη μείωση στο goodput στους BATS-M κώδικες όσο αυξάνουν οι απώλειες, κάτι που επιβεβαιώνει την ανθεκτικότητά τους στα σφάλματα. Κάτι ακόμα που φαίνεται περίεργο εκ πρώτης όψης είναι η μεγάλη διαφορά στη μέση καθυστέρηση εν-σειρά παράδοσης συγκριτικά με το RLNC και τους LT-M και BATS-M κώδικες. Στο RLNC, λόγω της κλίμακας φαίνεται σαν μία ευθεία που ταυτίζεται με τον άξονα x. Όμως στη πραγματικότητα η μέση καθυστέρηση εν-σειρά παράδοσης είναι της τάξης των 17 time slot και αυξάνεται γραμμικά όσο αυξάνονται απώλειες. Ο λόγος όμως που στους LT-M και BATS-M κώδικες, έχουμε καθυστέρηση εν-σειρά παράδοσης μεγαλύτερη από 10000 time slots είναι επειδή δεν χωρίζονται τα αρχικά πακέτα σε γενεές ή μπλοκ κωδικοποίησης. Όπως εξηγήσαμε, την καθυστέρηση παράδοσης ενός πακέτου την υπολογίζουμε αν αφαιρέσουμε τον χρόνο αποστολής του πακέτου από τον χρόνο λήψης του στον δέκτη. Ωστόσο, στην υλοποίησή μας κάναμε τον συμβιβασμό και θεωρήσαμε

σαν χρόνο αποστολής κάθε πακέτου, τον χρόνο που ξεκινάει η αποστολή μιας γενεάς. Αυτό ισχύει για το RLNC. Δηλαδή όλα τα πακέτα της πρώτης γενεάς θα έχουν χρόνο αποστολής ίσο με 0. Τα πακέτα της δεύτερης γενεάς θα έχουν σαν χρόνο αποστολής το time slot στο οποίο ολοκληρώθηκε η αποκωδικοποίηση της πρώτης και ξεκίνησαν να στέλνονται τα πακέτα της δεύτερης κ.ο.κ. Αντίθετα, στους LT-M και BATS-M κώδικες τα πακέτα δεν χωρίζονται σε γενεές και για την κωδικοποίηση επιλέγουμε πακέτα από το σύνολο των 10000 αρχικών, οπότε εφαρμόζοντας την ίδια λογική, όλα τα κωδικοποιημένα πακέτα που αποστέλλονται θα έχουν σαν χρόνο αποστολής το time slot μηδέν. Για αυτό και βλέπουμε τόσο υψηλές καθυστερήσεις.

4.5.2 Κωδικοποίηση με ρυθμό (rate-based)





Εικόνα 8 - Αποτελέσματα τεχνικών στη rate-based προσέγγιση

Στη rate-based κωδικοποίηση, αρχικά θέλαμε η πλεονάζουσα πληροφορία να καλύπτει μία φορά τη μέση πιθανότητα απωλειών στο κανάλι. Για αυτό και πήραμε ότι $\text{code rate } R1 = 1 - \text{ploss}$. Ωστόσο, παρατήσαμε ότι στους LT-M κώδικες κυρίως, με αυτό το code rate, δεν είχε αρκετή πληροφορία ο δέκτης ώστε να ανακτήσει πολλά αρχικά πακέτα. Για αυτό και εφαρμόσαμε και ένα δεύτερο code rate $R2 = 1 - 2 * \text{ploss}$ που παρατηρούμε ότι βελτιώνει λίγο τη κατάσταση μιας και στέλνουμε αρκετά περισσότερα πλεονάζοντα πακέτα, παραμένοντας όμως πάλι αρκετά χαμηλά συγκριτικά με το RLNC. Ωστόσο κάτι τέτοιο είναι και αναμενόμενο αφού και οι LT-M και κατ' επέκταση και οι BATS-M κώδικες έχουν σχεδιαστεί ώστε να λειτουργούν κυρίως σε rateless προσεγγίσεις. Στο RLNC, τα κωδικοποιημένα πακέτα που χρησιμοποιούνται παράγονται ως γραμμικοί συνδυασμοί όλων των πακέτων κάθε γενεάς, επομένως αν για παράδειγμα χαθούν δύο πακέτα σε μία γενεά, τότε αρκεί να λάβουμε δύο κωδικοποιημένα πακέτα και θα μπορέσουμε να ανακτήσουμε τη πληροφορία για τα χαμένα. Αντίθετα στους LT-M και BATS-M, για να αποκωδικοποιήσουμε δύο πακέτα θα πρέπει να έχουν επιλεγεί αυτά τα πακέτα ως γείτονες ενός ή περισσότερων κωδικοποιημένων πακέτων. Η μέση καθυστέρηση της εν-σειρά παράδοσης στο RLNC φαίνεται να ταυτίζεται για τα δύο code rates και να συμπίπτει στον άξονα x, ωστόσο κάτι τέτοιο οφείλεται στην κλίμακα που αναγκαστικά εφαρμόσαμε στη γραφική παράσταση ώστε να βάλουμε και την καθυστέρηση για τους LT-M και BATS-M κώδικες.

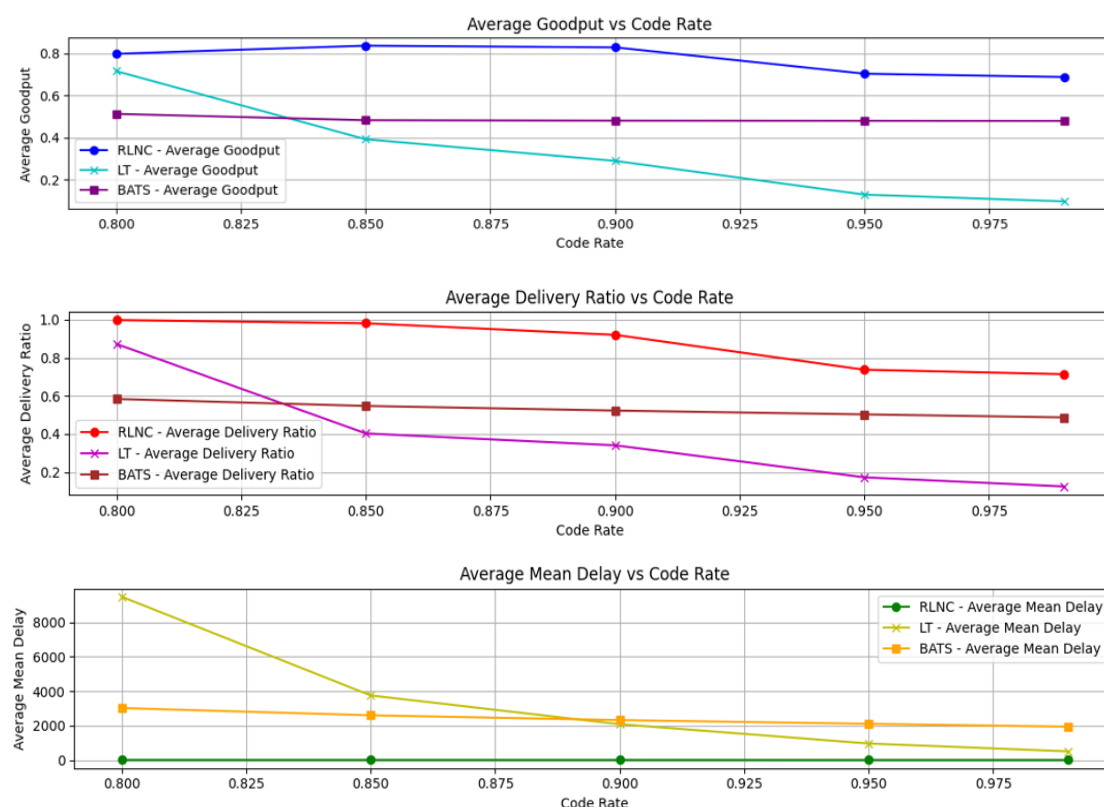
4.5.3 Η επίδραση του ρυθμού κωδικοποίησης

Στο επόμενο πείραμα διατηρήσαμε σταθερή τη πιθανότητα απωλειών ploss ίση με 5% και μεταβάλλαμε το ρυθμό κωδικοποίησης έτσι ώστε η πλεονάζουσα πληροφορία να

είναι ίση από μία έως τέσσερις φορές τις απώλειες του καναλιού. Δηλαδή για ploss ίσο με 5%, εφαρμόσαμε code rate 95%, 90%, 85% και 80% ώστε να καλύπτει μέχρι και 20% απώλειες στο χειρότερο σενάριο. Το μέγεθος των γενεών στο RLNC ορίζεται στον παρακάτω πίνακα.

| Code rate | Μέγεθος γενεάς (σε πακέτα) |
|-----------|----------------------------|
| 95% | 19 |
| 90% | 18 |
| 85% | 17 |
| 80% | 16 |

Τα αποτελέσματα αυτών των πειραμάτων παρουσιάζονται στην εικόνα 10.



Εικόνα 9 - Απόδοση τεχνικών μεταβάλλοντας το code rate

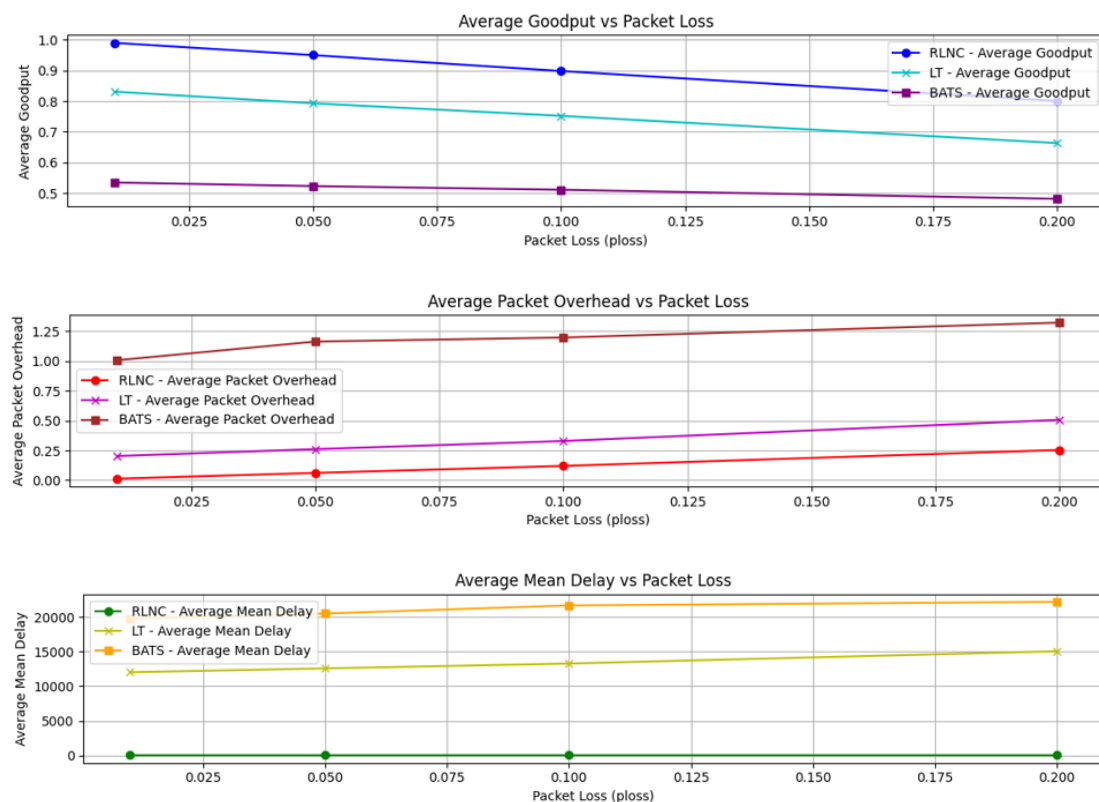
Παρατηρούμε ότι όσο αυξάνουμε τη πλεονάζουσα πληροφορία που στέλνουμε, τόσο αυξάνεται η τιμή του goodput, αλλά και το ποσοστό επιτυχών παραδόσεων. Αξίζει να επισημάνουμε ότι στους LT-M κώδικες, βλέπουμε η αύξηση αυτή να μην είναι γραμμική, και για μεγάλα code rates είναι μικρό το ποσοστό των επιτυχών παραδόσεων στον δέκτη. Ωστόσο όταν αυξάνεται πολύ η πλεονάζουσα πληροφορία για code rate ίσο με 0.8,

βλέπουμε ότι έχει πολύ υψηλά ποσοστά παράδοσης. Στο RLNC παρατηρούμε ότι ενώ αυξάνεται το goodput όσο αυξάνουμε και την πλεονάζουσα πληροφορία, όταν έχουμε code rate ίσο με 0.8, κάνει μία μικρή πτώση. Ωστόσο βλέπουμε ότι για εκείνο το code rate, το ποσοστό επιτυχών παραδόσεων πλησιάζει αρκετά το 100%, επομένως και το goodput θα πλησιάζει στο μέγιστο που μπορεί να φτάσει, που είναι το 0.8, αφού τόσο code rate έχουμε ορίσει σε αυτή την περίπτωση.

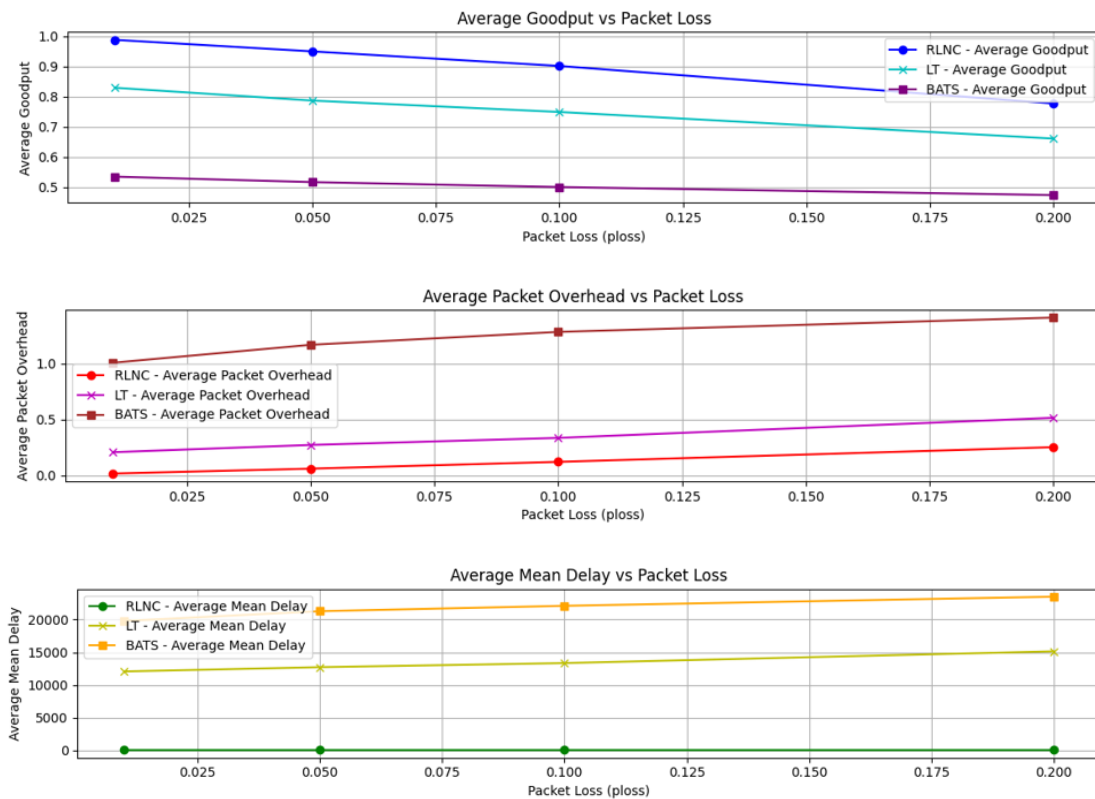
4.6 Κανάλι με απώλειες σε ριπές (Gilbert-Elliot)

4.6.1 Κωδικοποίηση χωρίς ρυθμό (rateless)

Εφαρμόζοντας κωδικοποίηση χωρίς ρυθμό σε κανάλια που έχουν απώλειες σε ριπές, πήραμε τις εικόνες 11 και 12.



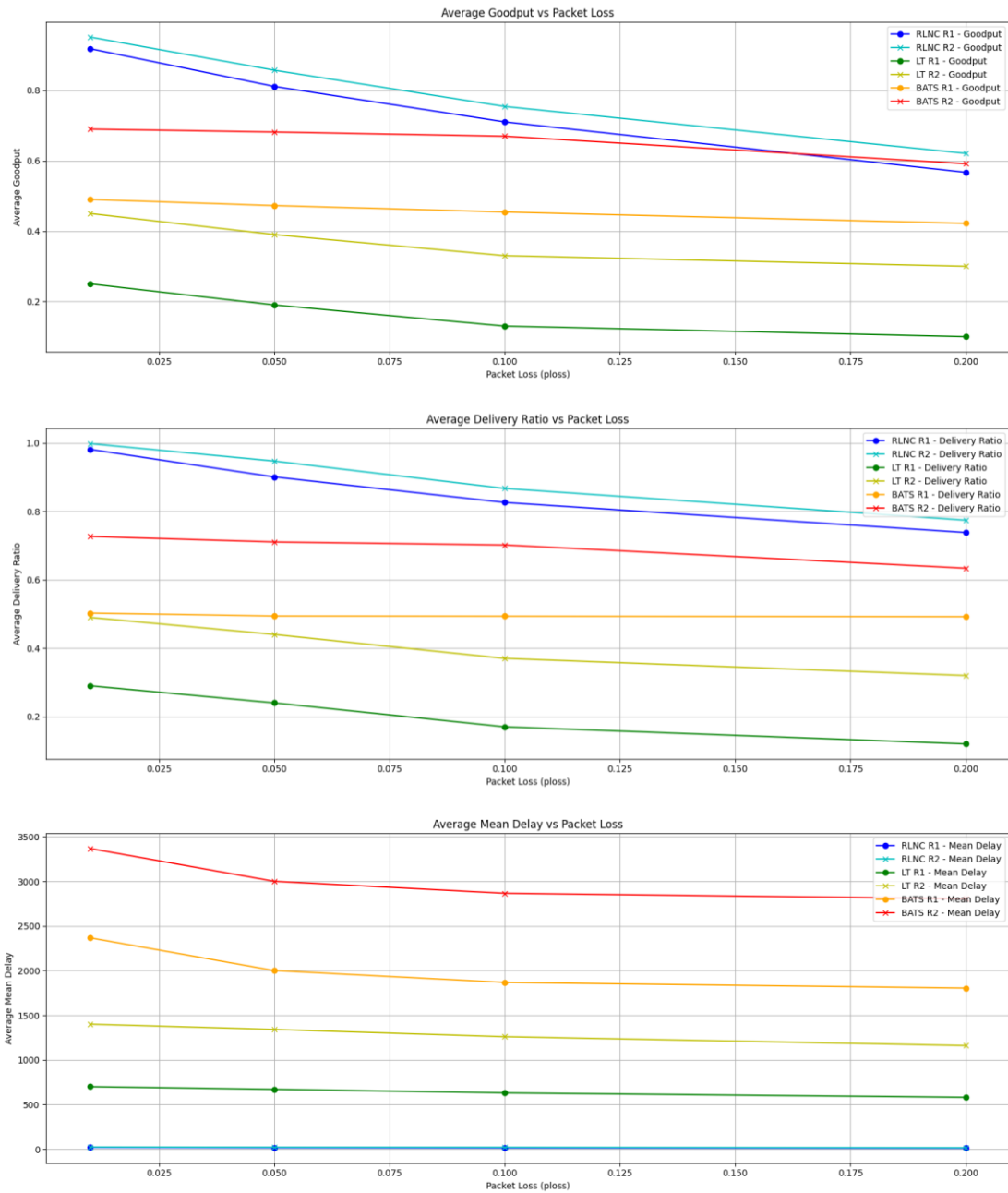
Εικόνα 10 - Αποτελέσματα τεχνικών σε rateless προσέγγιση, διάρκεια ριπών ίση με 4.



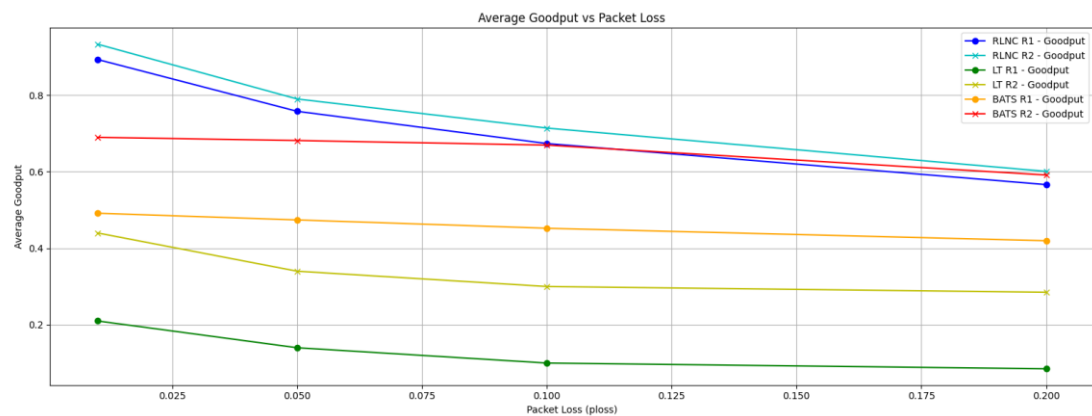
Εικόνα 11 - Αποτελέσματα τεχνικών σε rateless προσέγγιση με διάρκεια ριπών ίση με 8.

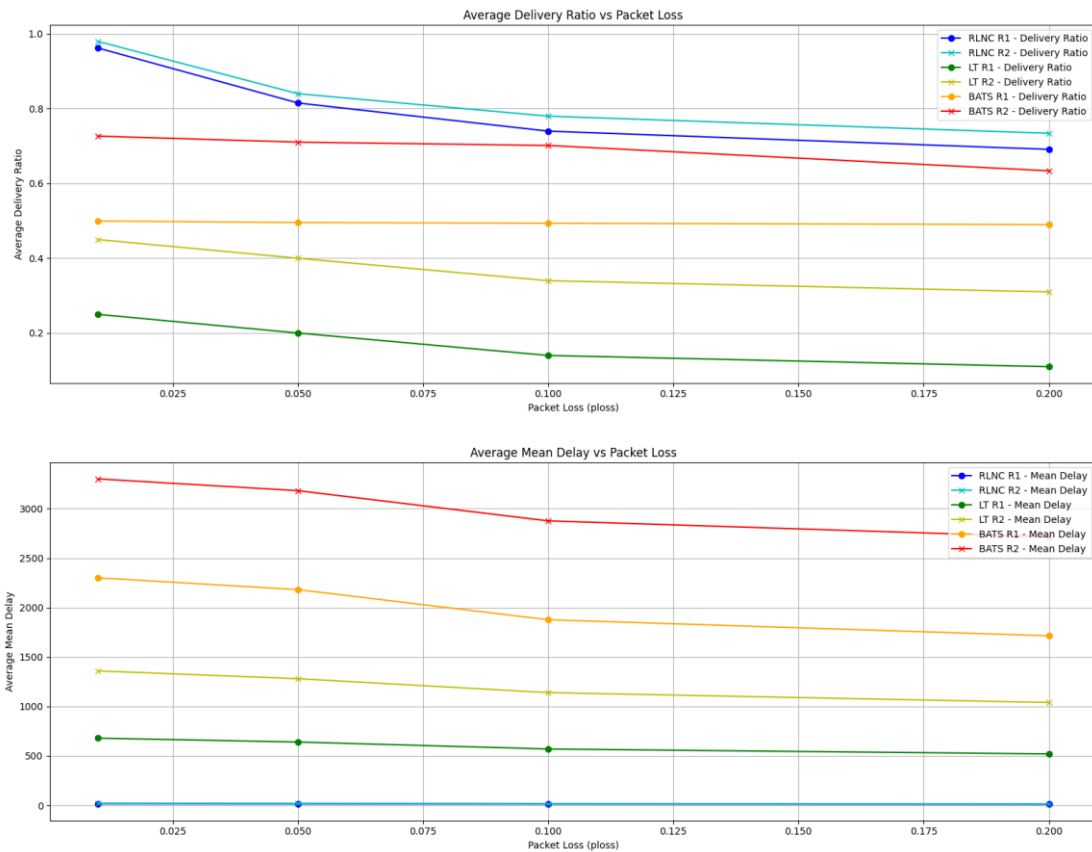
Παρατηρούμε ότι και σε κανάλι όπου οι απώλειες είναι σε ριπές, συμπεριφέρονται αρκετά καλά όλες οι τεχνικές, ακόμα και όταν οι απώλειες είναι μεγάλες. Παρατηρούμε γραμμική μείωση του goodput και αύξηση του packet overhead και στις τρεις τεχνικές. Όπως και στο κανάλι με ομοιόμορφες απώλειες, έτσι και εδώ παρατηρούμε μειωμένη απόδοση των BATS-M κωδικών στη rateless προσέγγιση συγκριτικά με τις άλλες δύο τεχνικές, κάτι που είναι αναμενόμενο αφού δεν υπάρχει κάποια βέλτιστη κατανομή βαθμού και χρησιμοποιούμε την ίδια με τον LT-M κώδικα. Ωστόσο βλέπουμε να είναι πιο ανθεκτική στις απώλειες, μιας και βλέπουμε να μειώνεται η απόδοση της με μικρότερο ρυθμό από ότι στο RLNC και στο LT-M. Όσον αφορά τους χρόνους της μέσης καθυστέρησης όπως εξηγήσαμε, υπάρχει αυτή η μεγάλη διαφορά λόγω του τρόπου υπολογισμού τους.

4.6.2 Κωδικοποίηση με ρυθμό (rate-based)



Εικόνα 12 - Αποτελέσματα τεχνικών σε rate-based προσέγγιση με διάρκεια ριπών ίση με 4



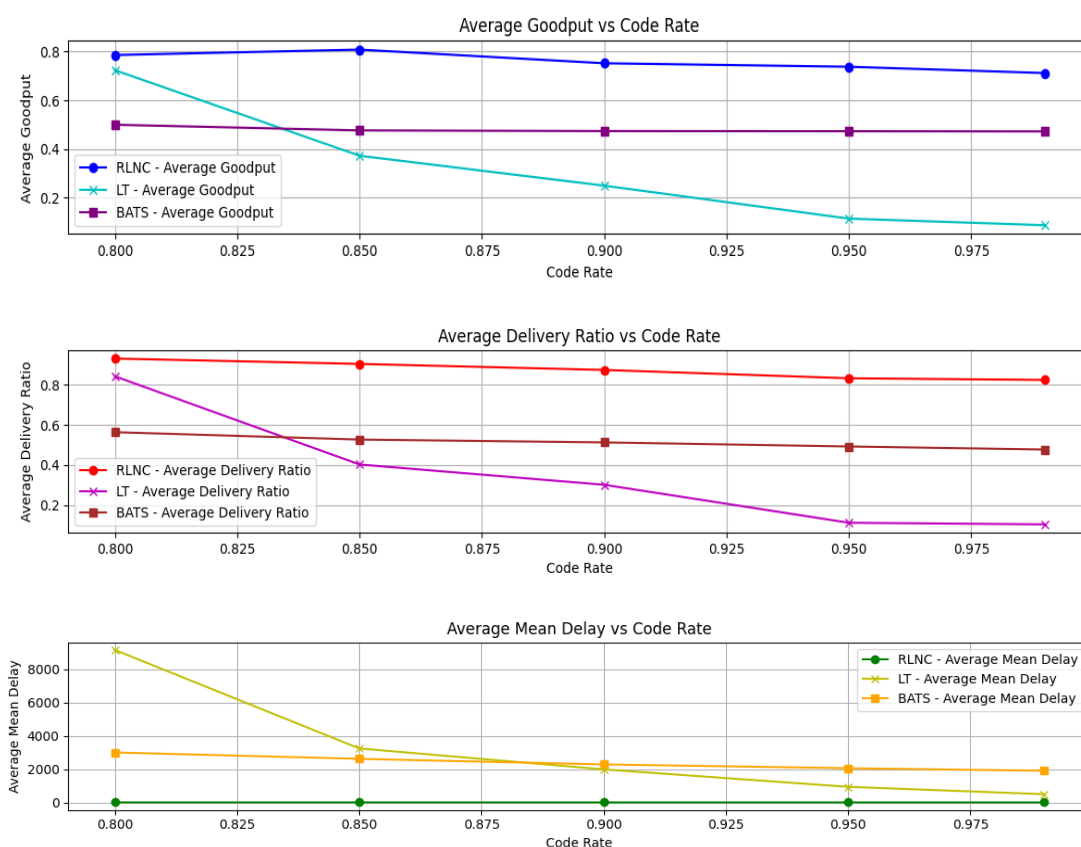


Εικόνα 13 - Αποτελέσματα τεχνικών σε rate based προσέγγιση με διάρκεια ριπών ίση με 8

Όπως και προηγουμένως, έτσι και εδώ, εφαρμόσαμε δύο διαφορετικά code rates $R1 = 1 - p_{loss}$ και $R2 = 1 - 2 * p_{loss}$ γιατί όπως παρατηρούμε, οι LT-M κώδικες, αλλά και οι BATS-M σε μικρότερο βαθμό με coded rate $R1$ δεν στέλνουν αρκετή πλεονάζουσα πληροφορία και πετυχαίνουν μικρά ποσοστά επιτυχών παραδόσεων πακέτων. Ωστόσο όπως είδαμε στο κεφάλαιο τρία και αναφέραμε προηγουμένως, αυτοί οι κώδικες έχουν σχεδιαστεί για να δουλεύουν κυρίως σε σενάρια rateless όπου στέλνουμε πακέτα στον δέκτη έως ότου αυτός καταφέρει και αποκωδικοποιήσει όλα τα αρχικά πακέτα. Για αυτό και έχουν πιο χαμηλή απόδοση συγκριτικά με τον RLNC κώδικα. Παρόλα αυτά ο BATS-M κώδικας φαίνεται να δουλεύει καλύτερα από τον LT-M, κάτι που οφείλεται στο γεγονός ότι δεν στέλνει μεμονωμένα κωδικοποιημένα πακέτα συνδυάζοντας κάποια αρχικά, αλλά στέλνει ομάδες κωδικοποιημένων πακέτων που προκύπτουν από συνδυασμό των ίδιων αρχικών πακέτων. Και πάλι όμως δεν μπορεί να πετύχει πολύ υψηλά ποσοστά μιας και η κατανομή βαθμού δεν του επιτρέπει να αξιοποιήσει στο μέγιστο όλα τα πλεονεκτήματα του.

4.6.3 Η επίδραση του ρυθμού κωδικοποίησης

Η πιθανότητα απώλειας στο κανάλι ploss παραμένει σταθερή και ίση με 5% και το code rate μεταβάλλεται από 95% έως και 80% για να καλύπτει μέχρι και 20% απώλειες στο χειρότερο σενάριο. Η διάρκεια ριπών σε αυτά τα κανάλια παραμένει σταθερή και είναι ίση με τέσσερα πακέτα. Το μέγεθος κάθε γενεάς ανάλογα με το code rate προκύπτει με τον ίδιο πίνακα που παραθέσαμε στο κανάλι με ομοιόμορφες απώλειες. Τα αποτελέσματα αυτού του πειράματος είναι στην εικόνα 15.



Εικόνα 14 - Αποτελέσματα τεχνικών μεταβάλλοντας το code rate

Όπως και στο κανάλι με ομοιόμορφες απώλειες, στο RLNC, όταν το code rate γίνεται ίσο με 0.8, παρατηρούμε μία μικρή πτώση του goodput. Αυτό γίνεται γιατί όπως είναι λογικό, δεν μπορεί να υπερβαίνει την αντίστοιχη τιμή του code rate. Αν συνέβαινε κάτι τέτοιο θα σήμαινε ότι ο δέκτης κατάφερε και αποκωδικοποίησε τελικά περισσότερα πακέτα από τα αρχικά που ήθελε να μεταδώσει ο πομπός, κάτι που είναι αδύνατο. Επομένως, για code rate ίσο με 0.8, το goodput έχει αυτή σαν μέγιστη τιμή την οποία και πλησιάζει. Ενώ παρατηρούμε μία γραμμικότητα τόσο στον RLNC όσο και στον BATS-M κώδικα. Αντίθετα στον LT-M, παρατηρείται μία απότομη αύξηση τόσο του goodput, όσο

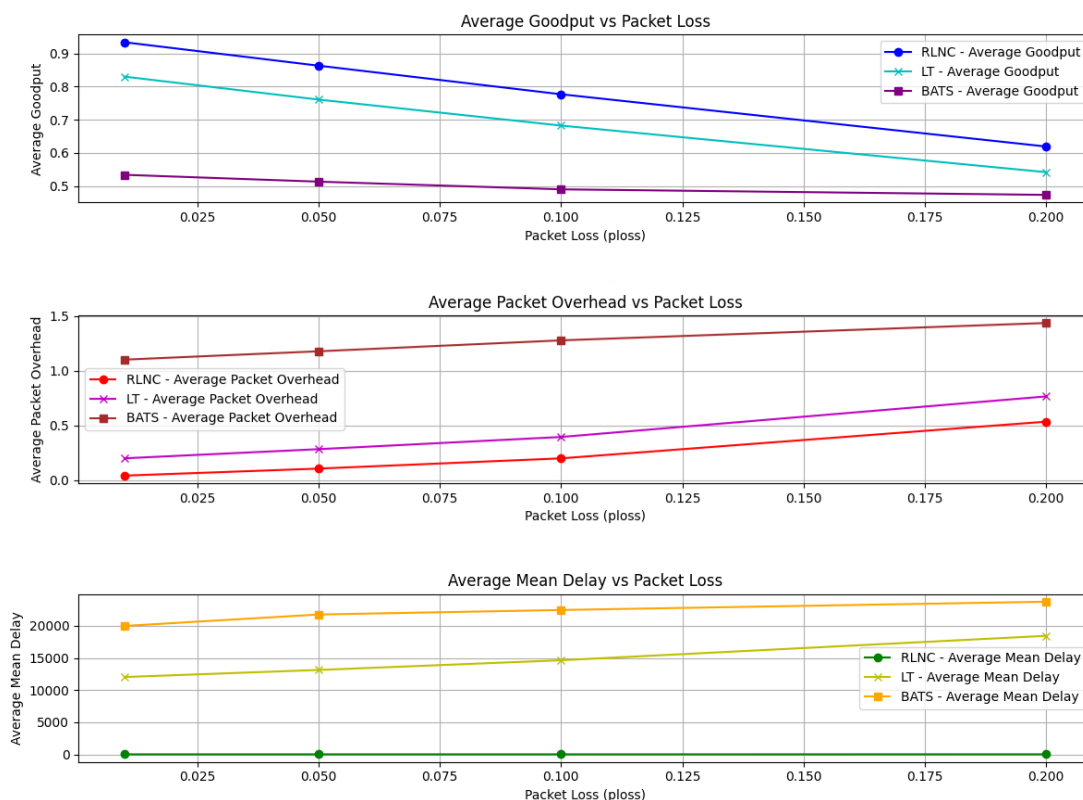
και του ποσοστού επιτυχούς παράδοσης πακέτων, κάτι που δείχνει ότι στα υπόλοιπα code rates, δεν ήταν αρκετή η πλεονάζουσα πληροφορία που στέλναμε στον δέκτη.

4.7 Τοπολογία δύο διαδοχικών καναλιών με ομοιόμορφες απώλειες

Στο multi-hop κανάλι εφαρμόζεται κανονικά ο BATS-M κώδικας, όπως τον περιγράψαμε στο κεφάλαιο τρία, υλοποιώντας επανακωδικοποίηση στον ενδιάμεσο κόμβο στα πακέτα της κάθε παρτίδας. Για τον LT-M κώδικα ωστόσο, που προορίζεται σε end-to-end σενάρια χωρίς ενδιάμεσους κόμβους, δεν εφαρμόζουμε καμία επεξεργασία στα κωδικοποιημένα πακέτα και απλά προωθούνται από τον ενδιάμεσο κόμβο προς τον δέκτη με πιθανότητα απώλειας p_{loss} .

4.7.1 Κωδικοποίηση χωρίς ρυθμό (rateless)

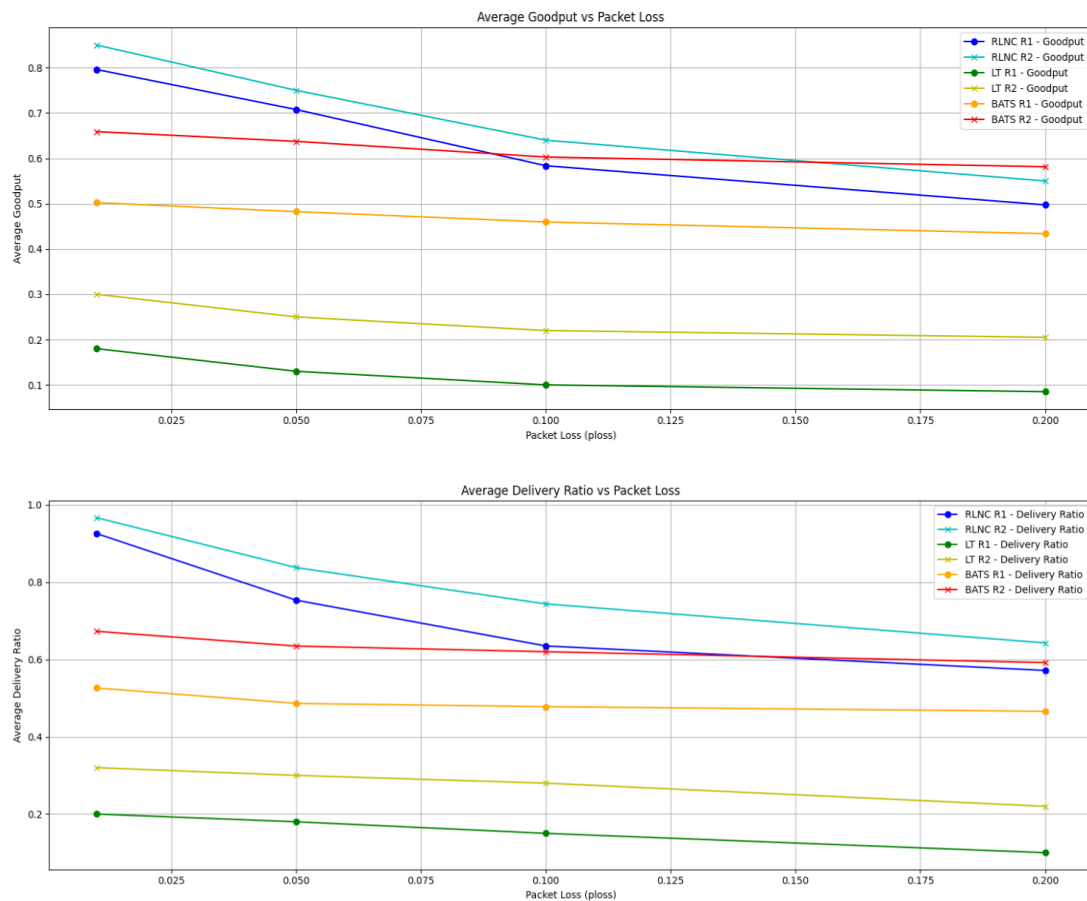
Εφαρμόζοντας κωδικοποίηση χωρίς ρυθμός σε multi-hop με ομοιόμορφες απώλειες πήραμε την εικόνα 16.

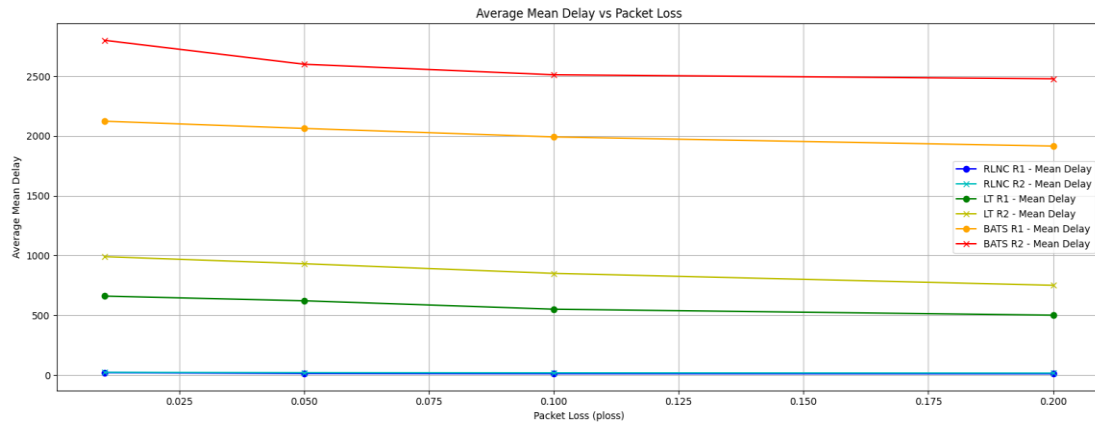


Εικόνα 15 - Αποτελέσματα τεχνικών στη rateless προσέγγιση

Παρατηρούμε παρόμοια συμπεριφορά των κωδικών με το single-hop κανάλι, όπου το goodput μειώνεται γραμμικά όσο αυξάνονται οι πιθανότητες απώλειας σε κάθε κανάλι. Όπως αναφέραμε και στα δύο κανάλια του multi-hop σεναρίου εφαρμόζουμε την ίδια πιθανότητα απώλειας πακέτων ploss. Παρατηρούμε ωστόσο ότι η απόδοση του BATS-M κώδικα μειώνεται με μικρότερο ρυθμό, άρα έχει μεγαλύτερη ανθεκτικότητα όσο αυξάνεται η πιθανότητα απώλειας. Λόγω του τρόπου με τον οποίο αποφασίσαμε να υπολογίσουμε την μέση καθυστέρηση της εν σειρά παράδοσης, όπου όλα τα πακέτα στους LT-M και BATS-M κώδικες έχουν σαν χρόνο αποστολής το time slot μηδέν, βλέπουμε τόσο μεγάλη διαφορά στη μέση καθυστέρηση συγκριτικά με το RLNC.

4.7.2 Κωδικοποίηση με ρυθμό (rate-based)



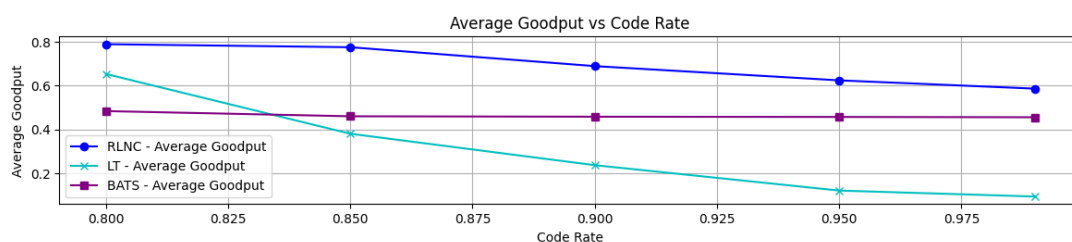


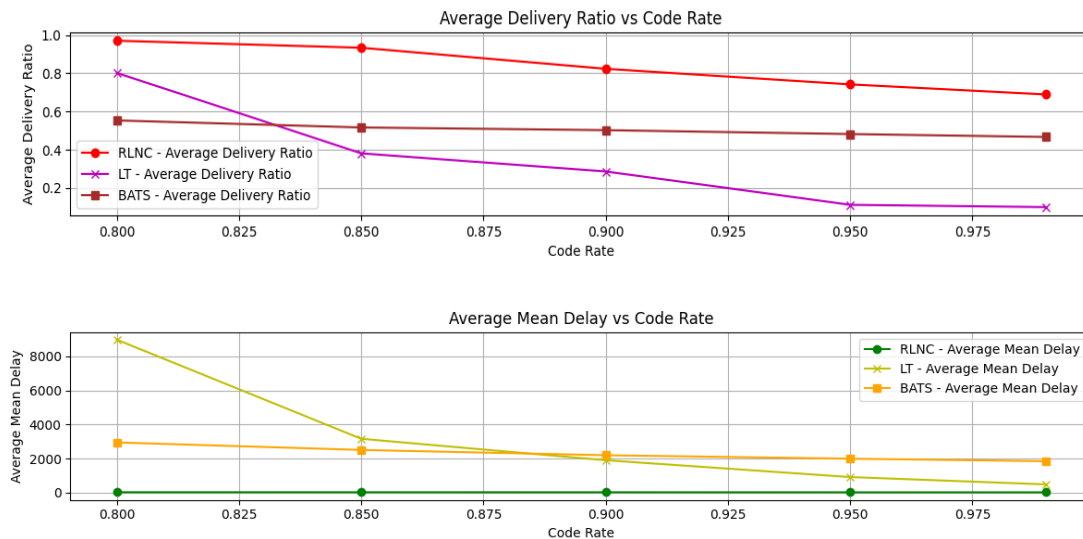
Εικόνα 16 - Αποτελέσματα τεχνικών στη rate-based προσέγγιση

Όπως και στο single-hop κανάλι, παρατηρήσαμε ότι με το code rate να είναι $R1 = 1 - ploss$, δεν στέλναμε αρκετή πλεονάζουσα πληροφορία ώστε να μπορεί ο δέκτης να ανακτήσει αρκετά από τα αρχικά πακέτα. Για αυτό εφαρμόσαμε και δεύτερο code rate $R2 = 1 - 2 * ploss$ που φαίνεται να βελτιώνει ελαφρώς τα αποτελέσματα. Ωστόσο και οι LT-M κώδικες και οι BATS-M έχουν σχεδιαστεί ώστε να είναι rateless κώδικες κυρίως και έτσι από τον τρόπο με τον οποίο δουλεύουν είναι αναμενόμενο να μην έχουν τα επιθυμητά αποτελέσματα στις rate-based προσεγγίσεις.

4.7.3 Η επίδραση του ρυθμού κωδικοποίησης

Σε αυτό το πείραμα, διατηρήσαμε πάλι σταθερή τη πιθανότητα απωλειών και στα δύο κανάλια ploss ίση με 5% και μεταβάλλαμε το code rate έτσι ώστε να καλύψουμε έως και τέσσερις φορές τις απώλειες στο κανάλι. Τα μεγέθη των γενεών στο RLNC ορίστηκαν όπως ακριβώς και στο single-hop κανάλι, δηλαδή για code rate ίσο με 95 θέσαμε μέγεθος γενεάς ίσο με 19 πακέτα, για code rate ίσο με 90 η γενεά περιέχει 18 πακέτα, για code rate ίσο με 85 η γενεά έχει 17 πακέτα και για code rate ίσο με 80 η γενεά έχει 16 πακέτα. Τα αποτελέσματα του πειράματος αποτυπώνονται στην εικόνα 18.





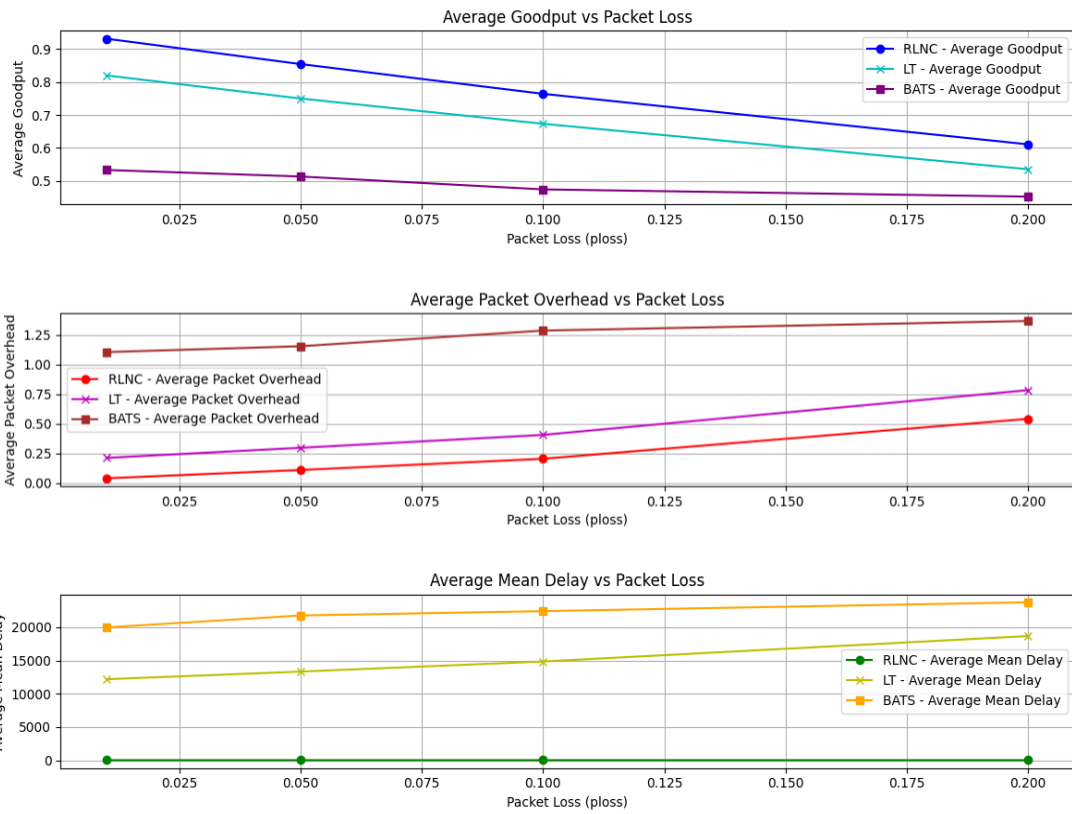
Εικόνα 17 - Αποτελέσματα τεχνικών μεταβάλλοντας το code rate

Παρατηρούμε ότι όσο αυξάνουμε τον ρυθμό κωδικοποίησης (code rate) αυξάνεται και η απόδοση των κωδίκων μαζί και με τη μέση καθυστέρηση της εν σειρά παράδοσης αφού στέλνουμε περισσότερη πλεονάζουσα πληροφορία στον δέκτη. Ωστόσο, αξίζει να παρατηρήσουμε ότι ενώ και στο RLNC και στο BATS-M κώδικα η αύξηση της απόδοσης φαίνεται να είναι γραμμική, κάτι τέτοιο δεν συμβαίνει με τον LT-M κώδικα. Παρατηρούμε όταν το code rates κοντά στη μονάδα (για παράδειγμα 0.95 και 0.9) δεν λαμβάνει ο δέκτης αρκετή πλεονάζουσα πληροφορία για να ανακτήσει μεγάλο πλήθος αρχικών πακέτων. Όταν, όμως, αυξηθεί η πλεονάζουσα πληροφορία, παρατηρούμε μεγάλη αύξηση και στην απόδοση του αλγορίθμου.

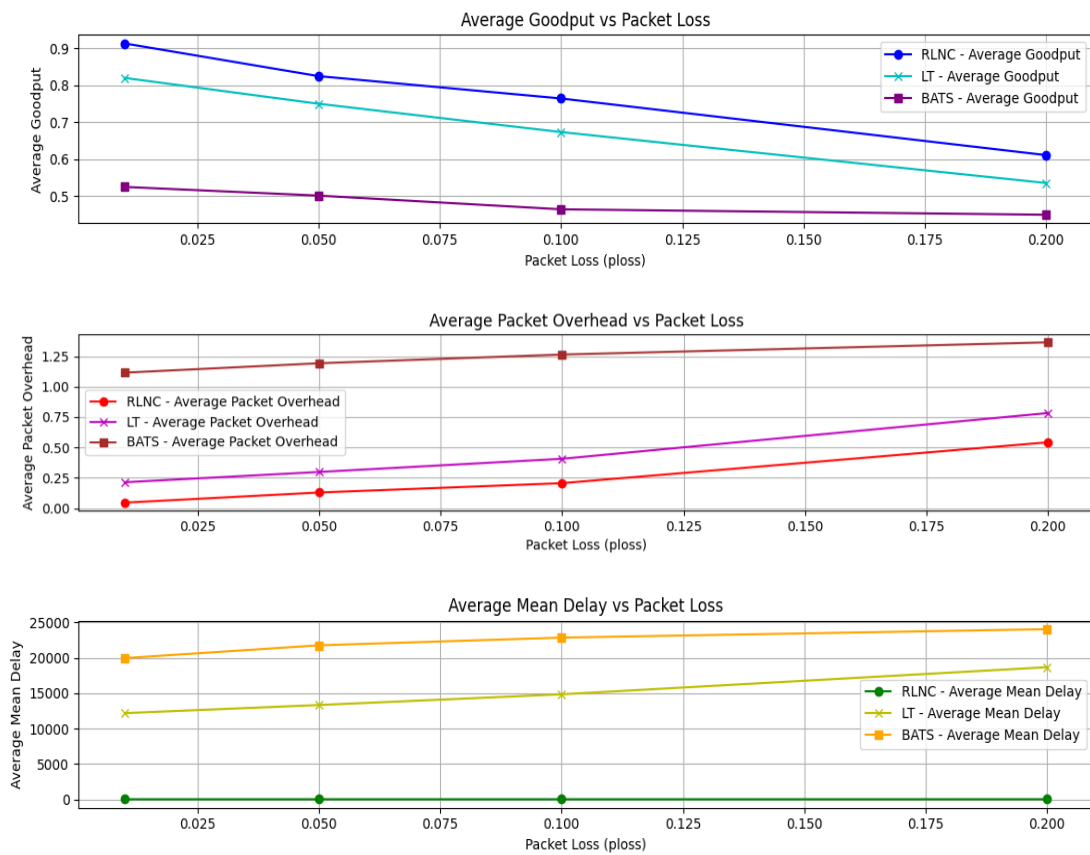
4.8 Τοπολογία δύο διαδοχικών καναλιών με απώλειες σε ριπές (Gilbert-Elliot)

4.8.1 Κωδικοποίηση χωρίς ρυθμό (rateless)

Εφαρμόζοντας κωδικοποίηση χωρίς ρυθμό σε κανάλια με απώλειες σε ριπές, πήραμε τα σχήματα 19 και 20.



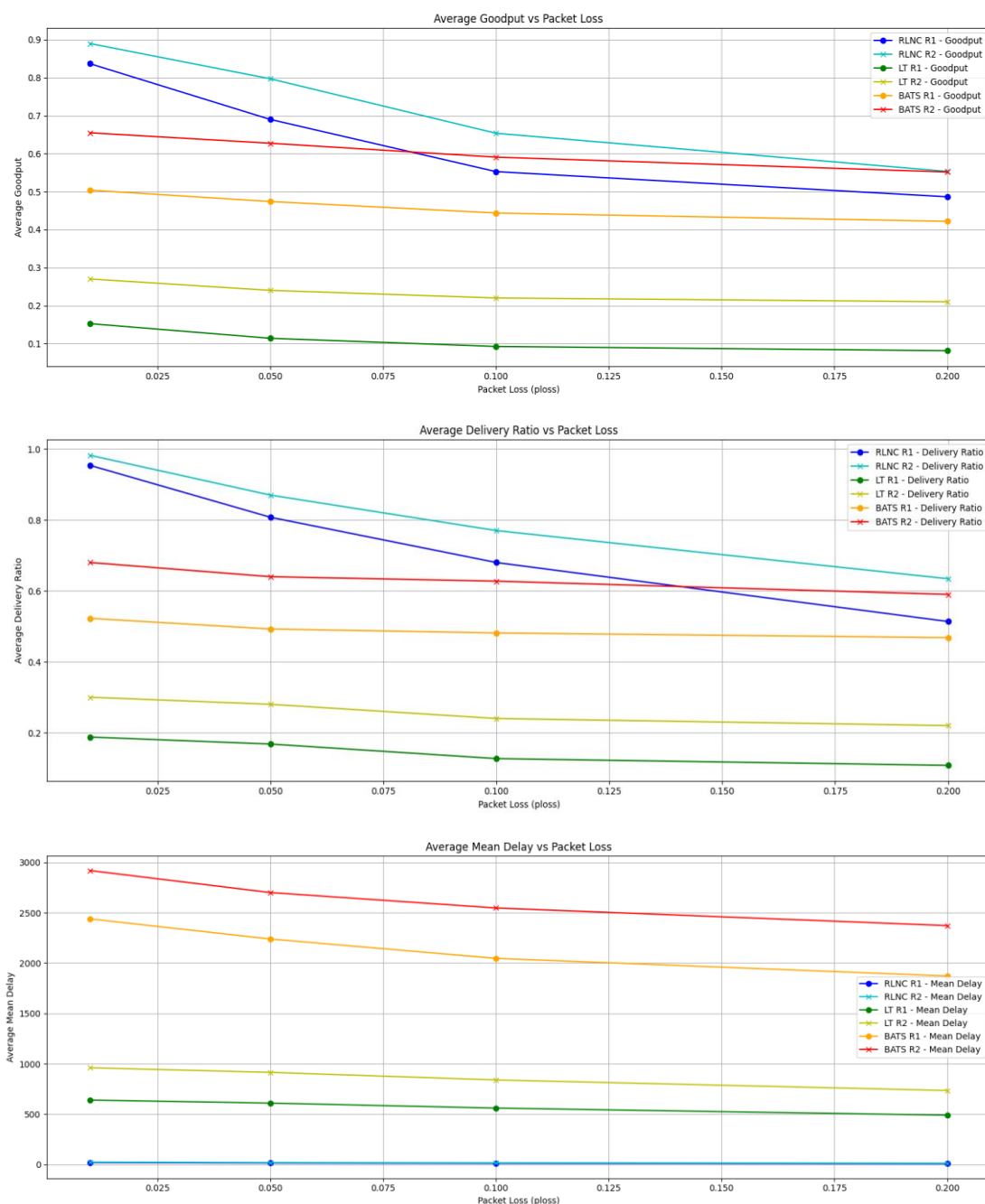
Εικόνα 18 - Αποτελέσματα τεχνικών σε rateless προσέγγιση με διάρκεια ριπών ίση με 4.



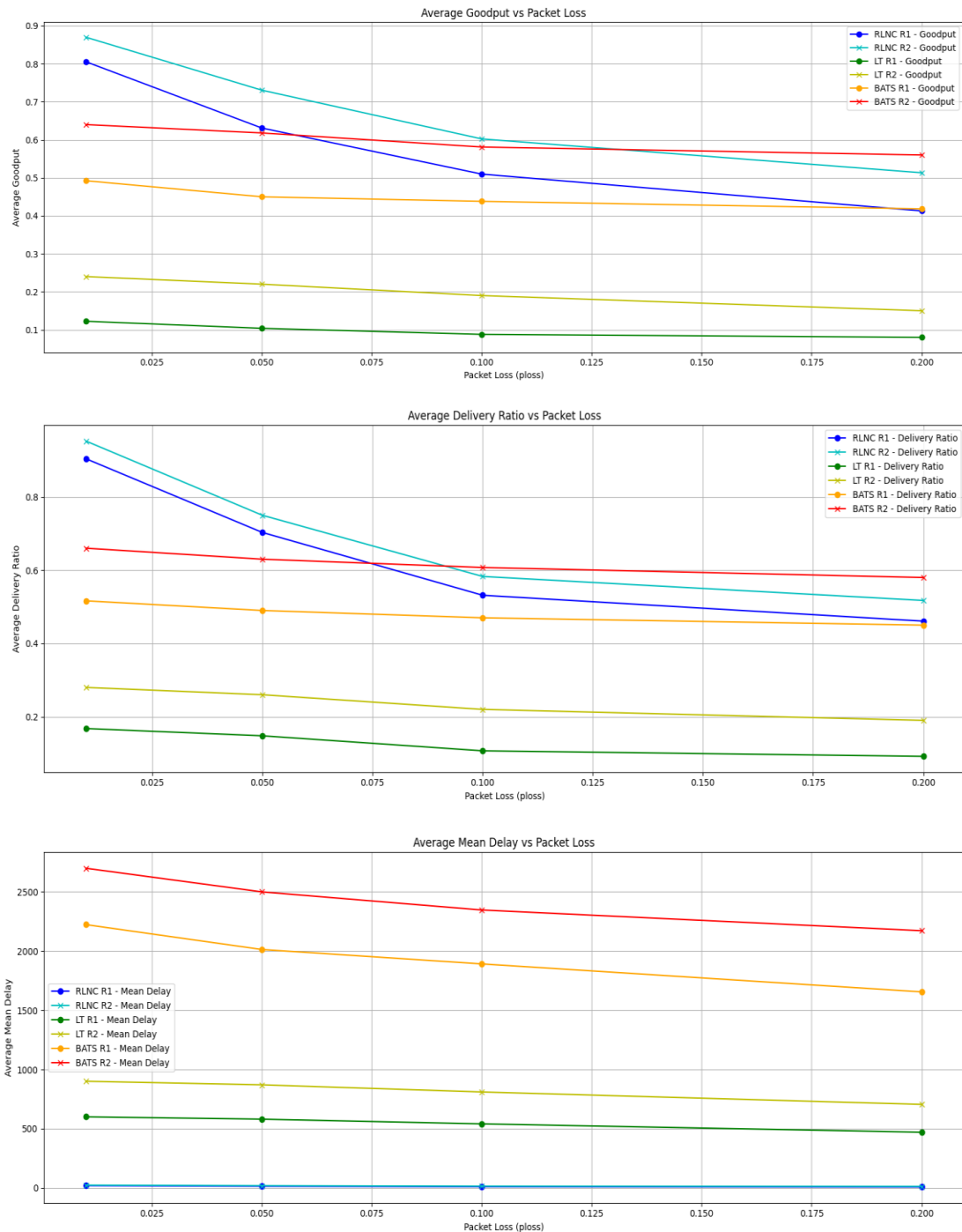
Εικόνα 19 - Αποτελέσματα τεχνικών σε rateless προσέγγιση, διάρκεια ριπών ίση με 8.

Παρατηρούμε ότι υπάρχει μία μείωση στην απόδοση των τεχνικών όσο αυξάνονται οι απώλειες κάτι που είναι αναμενόμενο. Είναι αξιοσημείωτο να σημειώσουμε και την ανθεκτικότητα που πετυχαίνουν οι BATS-M κώδικες μέσω της επανακωδικοποίησης που εφαρμόζουν στον ενδιαμέσο κόμβο και η απόδοσή τους μειώνεται με μικρότερο ρυθμό. Ωστόσο, όπως έχουμε εξηγήσει λόγω της κατανομής βαθμού που χρησιμοποιούμε δεν μπορούμε να εκμεταλλευτούμε στο μέγιστο όλα τα οφέλη του κώδικα αυτού.

4.8.2 Κωδικοποίηση με ρυθμό (rate-based)



Εικόνα 20 - Αποτελέσματα τεχνικών σε rate-based προσέγγιση, διάρκεια ριπών ίση με 4.



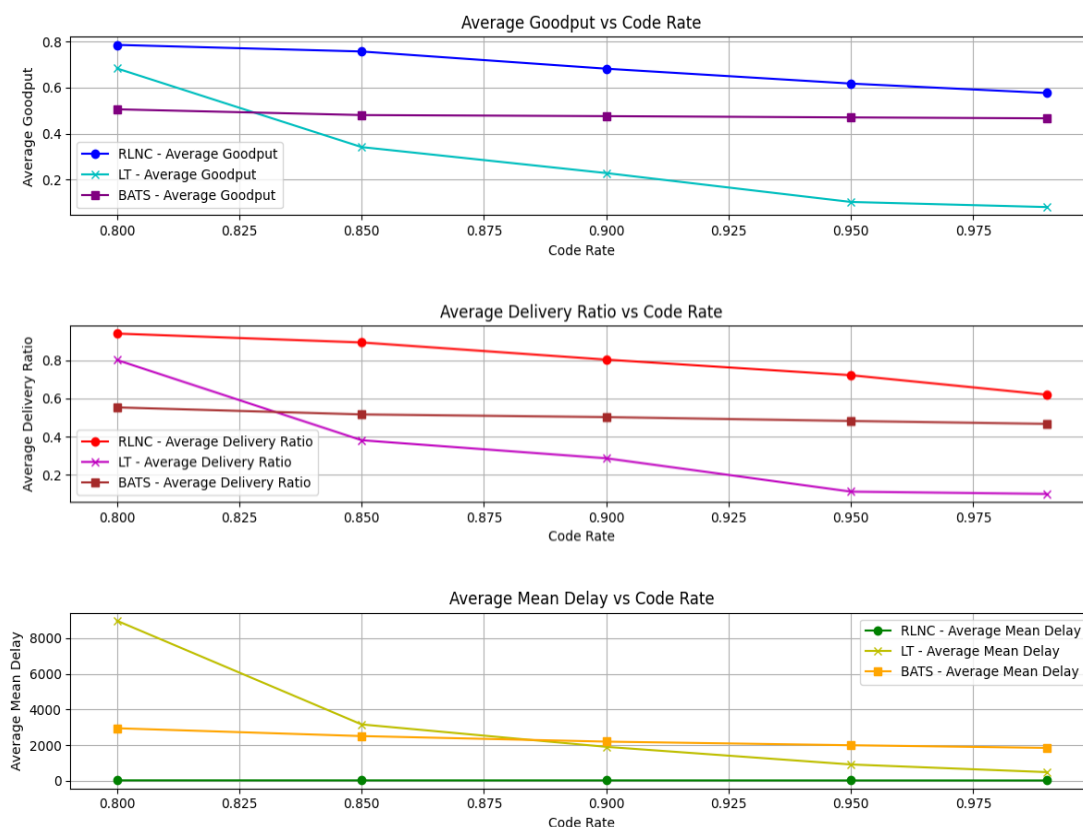
Εικόνα 21 - Αποτελέσματα τεχνικών σε rate-based προσέγγιση με διάρκεια ριπών ίση με 8.

Όπως και στις προηγούμενες επαναλήψεις σε κωδικοποίηση χωρίς ρυθμό (rate-based), εφαρμόσαμε δύο code rates R1 και R2 που έχουμε εξηγήσει έτσι ώστε να ελέγξουμε και τη συμπεριφορά των κωδίκων όταν αυξάνουμε την πλεονάζουσα πληροφορία που στέλνουμε για κάθε πιθανότητα απώλειας στα κανάλια. Είναι

σημαντικό να παρατηρήσουμε ότι και οι BATS-M κώδικες, παρότι δεν έχουν σχεδιαστεί να δουλεύουν σε τέτοια σενάρια, πετυχαίνουν σημαντικά καλύτερη απόδοση από τους LT-M κώδικες. Αυτό οφείλεται στο γεγονός ότι τα κωδικοποιημένα πακέτα δεν στέλνονται μεμονωμένα. Στους LT-M κώδικες, όταν επιλέγονται d από τα αρχικά πακέτα, τότε κατασκευάζεται ένα κωδικοποιημένο πακέτο και αποστέλλεται, ενώ στους BATS-M, από αυτά τα d πακέτα που θα επιλεγθούν, θα παραχθούν τόσα κωδικοποιημένα πακέτα όσο είναι το μέγεθος της παρτίδας. Για αυτό και έχουμε καλύτερα αποτελέσματα στους BATS-M κώδικες συγκριτικά με τους LT-M.

4.8.3 Η επίδραση του ρυθμού κωδικοποίησης

Υλοποιώντας το πείραμα ελέγχου της συνολικής πλεονάζουσας πληροφορίας να είναι ίση από μία έως τέσσερις φορές τις απώλειες των καναλιών πήραμε το σχήμα που παρατίθεται στην εικόνα 23.



Εικόνα 22 - Αποτελέσματα τεχνικών μεταβάλλοντας το *code rate*

Και σε αυτό το πείραμα παρατηρούμε την ίδια συμπεριφορά με τα πειράματα στα προηγούμενα κανάλια. Στον RLNC και στους BATS-M κώδικες η απόδοση τους καθώς και το ποσοστό παράδοσης πακέτων αυξάνεται γραμμικά όσο αυξάνεται η πλεονάζουσα πληροφορία, ενώ στον LT-M κώδικα αυξάνει απότομα όταν έχουμε πολλή πλεονάζουσα πληροφορία. Αυτό είναι αξιοσημείωτο καθώς φαίνεται η τεχνική κωδικοποίησης αυτή να

χρειάζεται περισσότερη πλεονάζουσα πληροφορία στις rate based προσεγγίσεις ώστε να λειτουργήσει αποτελεσματικά.

Στο κλείσιμο του Κεφαλαίου , είναι σημαντικό να συνοψίσουμε τις πειραματικές διαδικασίες και τα κύρια ευρήματα, τα οποία προσφέρουν πολύτιμη γνώση για την απόδοση και τη συμπεριφορά των τεχνικών που μελετήθηκαν. Η πειραματική αξιολόγηση επικεντρώθηκε στις επιδόσεις των LT-M, RLNC και BATS-M κωδίκων σε διαφορετικά σενάρια, με έμφαση στην μέση καθυστέρηση της εν-σειρά παράδοσης, το ποσοστό επιτυχούς παράδοσης πακέτων στις προσεγγίσεις με ρυθμό, το ποσοστό πλεονάζουσας πληροφορίας στις προσεγγίσεις χωρίς ρυθμό, και το goodput. Τα πειράματα αυτά διεξήχθησαν με στόχο να απαντηθούν ερωτήματα σχετικά με την ανθεκτικότητα και την αποδοτικότητα των τεχνικών σε περιβάλλοντα υψηλής απώλειας και δυναμικής τοπολογίας.

Τα πειράματα σχεδιάστηκαν με τέτοιο τρόπο ώστε να καλύψουν μια ευρεία γκάμα συνθηκών δικτύου. Οι προσομοιώσεις υλοποιήθηκαν μέσω της γλώσσας προγραμματισμού Python, όπου επιβλήθηκαν διαφορετικά ποσοστά απώλειας πακέτων και διαφορετικές συνθήκες μετάδοσης, όπως single-hop και multi-hop δίκτυα. Τα δεδομένα που χρησιμοποιήθηκαν για τη δημιουργία κωδικοποιημένων πακέτων ήταν σταθερού μεγέθους, ενώ οι ρυθμοί κωδικοποίησης προσαρμόστηκαν ανάλογα με τις ανάγκες του κάθε πειράματος. Για τους LT-M οι πράξεις κωδικοποίησης και αποκωδικοποίησης βασίστηκαν στη πράξη XOR μεταξύ των αρχικών πακέτων, ενώ για τους BATS-M κώδικες, τα πακέτα οργανώθηκαν σε παρτίδες (batches), προκειμένου να αυξηθεί η ανθεκτικότητα στις απώλειες. Στους RLNC κώδικες τα πακέτα χωρίστηκαν σε γενεές. Η μέτρηση της απόδοσης πραγματοποιήθηκε με τη χρήση των μετρικών που αναφέραμε στην αρχή του κεφαλαίου.

Τα αποτελέσματα έδειξαν ότι οι LT-M κώδικες αποδίδουν ικανοποιητικά σε περιβάλλοντα με χαμηλή απώλεια δεδομένων, όπου η Robust Soliton κατανομή βαθμών αξιοποιείται πλήρως για την αποδοτική αποκωδικοποίηση. Ωστόσο, σε συνθήκες υψηλής απώλειας, η απόδοσή τους μειώνεται αισθητά, γεγονός που υπογραμμίζει τους περιορισμούς αυτής της τεχνικής.

Αντίθετα, οι RLNC κώδικες παρουσίασαν συνολικά την καλύτερη απόδοση στα περισσότερα σενάρια. Η τυχαία γραμμική συνδυαστική τους προσέγγιση τους καθιστά ανθεκτικούς στις απώλειες, αλλά η πολυπλοκότητα της αποκωδικοποίησης αυξάνει το χρόνο επεξεργασίας, ειδικά σε περιβάλλοντα με μεγάλες απαιτήσεις υπολογιστικής ισχύος.

Οι BATS-M κώδικες ανέδειξαν τη σταθερότητά τους σε σενάρια με υψηλή απώλεια, ενώ η χρήση παρτίδων (batches) επέτρεψε τη διατήρηση υψηλών επιπέδων αποκωδικοποίησης. Παρά την αύξηση του χρόνου μετάδοσης λόγω της φύσης της μεθόδου, οι BATS-M κώδικες παραμένουν μια πολλά υποσχόμενη τεχνική για εφαρμογές όπου η ανθεκτικότητα υπερέχει της αποδοτικότητας.

Συνολικά, τα πειράματα ανέδειξαν τις διαφορετικές περιοχές στις οποίες υπερέχει κάθε τεχνική, προσφέροντας μια σαφή εικόνα για την επιλογή της κατάλληλης προσέγγισης ανάλογα με τις απαιτήσεις του δικτύου. Τα ευρήματα αυτά αποτελούν τη βάση για τις προτάσεις βελτίωσης που παρουσιάζονται στο επόμενο κεφάλαιο, εστιάζοντας στη βελτιστοποίηση των υπάρχουσών τεχνικών και την ενσωμάτωση υβριδικών μεθόδων για την αντιμετώπιση των περιορισμών που εντοπίστηκαν.

Κεφάλαιο 5. Συμπεράσματα

Η κωδικοποίηση δικτύου είναι μία επαναστατική τεχνική που υλοποιήθηκε για να λύσει μερικά από τα προβλήματα που υπήρχαν στα παραδοσιακά δίκτυα υπολογιστών. Μέχρι τώρα, όταν θέλαμε να στείλουμε πακέτα, τα στέλναμε αυτούσια και περιμέναμε μήνυμα επιβεβαίωσης για κάθε ένα από τα πακέτα ώστε να ξέρουμε ότι το πακέτο έφτασε. Σε περίπτωση που δεν λαμβάναμε μήνυμα επιβεβαίωσης λήψης, ξαναστέλναμε το ίδιο πακέτο. Κάτι τέτοιο κάνει αρκετά δύσκολη την επικοινωνία, ειδικά στις περιπτώσεις όπου θέλουμε να αποστείλουμε τα ίδια αρχικά πακέτα σε πολλούς παραλήπτες και ο καθένας μπορεί να χάσει διαφορετικό πακέτο.

Για αυτόν τον λόγο χρησιμοποιούμε τους κώδικες απαλοιφής (erasure codes), οι οποίοι δεν απαιτούν από τον δέκτη να στέλνει πακέτα επιβεβαίωσης λήψης. Ακόμα, σε περίπτωση που οι παραλήπτες χάνουν διαφορετικό πακέτο, αυτό δεν είναι πια πρόβλημα, αφού δεν στέλνουμε τα αρχικά πακέτα αυτούσια. Εφαρμόζουμε κωδικοποίηση όπου φτιάχνουμε κωδικοποιημένα πακέτα συνδυάζοντας τα αρχικά, και για αυτό σε όλους τους παραλήπτες στέλνουμε τα ίδια κωδικοποιημένα πακέτα. Το πλήθος των κωδικοποιημένων πακέτων που θα κατασκευάσουμε δεν είναι πάντα το ίδιο και εξαρτάται από την προσέγγιση που θα ακολουθήσουμε.

Μία προσέγγιση είναι αυτή στην οποία στέλνουμε πακέτα χωρίς ρυθμό. Σε αυτή την περίπτωση στέλνουμε πακέτα έως ότου ο δέκτης καταφέρει να ανακτήσει όλα τα αρχικά. Το μόνο μήνυμα που χρειάζεται να λάβουμε από τον δέκτη σε αυτήν την περίπτωση είναι όταν καταφέρει να ανακτήσει όλα τα πακέτα, ώστε να σταματήσουμε την αποστολή καινούριων. Είναι ιδιαίτερα χρήσιμη αυτή η τεχνική σε κανάλια που μεταβάλλονται δυναμικά ή δεν γνωρίζουμε τους ρυθμούς σφαλμάτων τους. Μία άλλη προσέγγιση είναι η αποστολή πακέτων με ρυθμό, στην οποία υπάρχει προκαθορισμένο πλήθος πακέτων που θα στείλουμε στον δέκτη, το οποίο εξαρτάται από τον ρυθμό κωδικοποίησης που θα εφαρμόσουμε. Αυτή η τεχνική είναι χρήσιμη στις περιπτώσεις όπου έχουμε γνώση για την ποιότητα του δικτύου και τον ρυθμό σφαλμάτων που έχει το κανάλι.

Όταν ο δέκτης λαμβάνει τα κωδικοποιημένα πακέτα, εφαρμόζει τη διαδικασία της αποκωδικοποίησης, η οποία είναι διαφορετική για την κάθε μία τεχνική που υλοποιούμε, και έτσι ανακτά τα αρχικά πακέτα που στείλαμε. Με αυτόν τον τρόπο δεν χρειάζεται η ανάδραση με τον πομπό. Η διαδικασία αυτή καθιστά την επικοινωνία πιο αποδοτική,

ιδιαίτερα σε περιβάλλοντα όπου οι συνθήκες του δικτύου αλλάζουν διαρκώς ή η επικοινωνία είναι δύσκολη, όπως σε δορυφορικά δίκτυα ή δίκτυα IoT.

Εφαρμόζοντας τα πειράματα που εξηγήσαμε στο κεφάλαιο τέσσερα, μπορούμε να διαπιστώσουμε ότι οι RLNC κώδικες πετυχαίνουν καλύτερα αποτελέσματα από τους LT κώδικες σε όλα τα σενάρια. Οι BATS κώδικες, από την άλλη, είναι ιδιαίτερα σταθεροί ακόμα και όταν η πιθανότητα απωλειών στο κανάλι αυξάνεται, μιας και η απόδοσή τους δεν μειώνεται τόσο πολύ. Βέβαια, τα ποσοστά στο goodput αλλά και το packet overhead που παρατηρήθηκαν στους BATS κώδικες δεν σημαίνουν απαραίτητα ότι είναι χειρότεροι από το RLNC, και σίγουρα δεν είναι χειρότεροι από τους LT κώδικες.

Η Robust Soliton κατανομή βαθμών που χρησιμοποιήσαμε σχεδιάστηκε αρχικά ώστε οι LT κώδικες να είναι αποτελεσματικοί. Για την αποτελεσματικότητά τους, όμως, αυτό που χρειάζονται οι κώδικες αυτοί είναι σε κάθε βήμα του αλγορίθμου αποκωδικοποίησης, που αναλύσαμε στο κεφάλαιο τρία, να υπάρχουν πάντα αρκετά κωδικοποιημένα πακέτα με βαθμό ίσο με ένα, ώστε να προχωράει η αποκωδικοποίηση. Η απευθείας εφαρμογή της στους BATS κώδικες, όμως, δεν αποδίδει τα βέλτιστα, κάτι που είναι λογικό λόγω της φύσης αυτών.

Στους BATS κώδικες συνδυάζονται στοιχεία από το RLNC στους κώδικες πηγής (fountain codes). Τα πακέτα χωρίζονται σε batches και αποστέλλονται στον δέκτη. Όταν δηλαδή επιλέγουμε κάποια αρχικά πακέτα d , με αυτά κατασκευάζουμε ένα συγκεκριμένο πλήθος κωδικοποιημένων πακέτων που θα αποτελέσουν μία παρτίδα. Αντίθετα, στους LT, με τα αρχικά d πακέτα θα κατασκευάσουμε ένα μόνο κωδικοποιημένο πακέτο που θα στείλουμε. Αυτό έχει σαν αποτέλεσμα, για τα ίδια αρχικά πακέτα, να στέλνουμε περισσότερα κωδικοποιημένα στους BATS κώδικες. Χρειάζονται δηλαδή περισσότερα time slots στο δίκτυο. Αυτό αυξάνει τον χρόνο που τρέχει το πρόγραμμα μας και, άρα, μειώνει το goodput.

Για να αντιμετωπιστεί αυτό το πρόβλημα και να βελτιστοποιηθεί η απόδοση των BATS κωδίκων, πρέπει η κατανομή βαθμών να λαμβάνει υπόψη τις ιδιαιτερότητές τους. Αυτό μπορεί να γίνει με μία νέα μελέτη, με στόχο τη βελτιστοποίηση είτε της Robust Soliton κατανομής είτε κάποιας άλλης κατανομής βαθμών. Συγκεκριμένα, για τη Robust Soliton κατανομή, μπορεί να χρειάζεται να αυξηθεί η πιθανότητα να λαμβάνουμε υψηλούς βαθμούς. Η κατανομή αυτή, όπως σχεδιάστηκε για τους LT κώδικες, εστιάζει στην εμφάνιση μικρών βαθμών. Ωστόσο, για τους BATS κώδικες, όπου δεν απαιτείται στην αποκωδικοποίηση να έχουμε σε κάθε βήμα πακέτα με βαθμό ίσο με ένα, χρειαζόμαστε να παίρνουμε πιο συχνά μεγάλους βαθμούς για να αξιοποιείται η χρήση των παρτίδων. Αντίστοιχα, μπορεί να χρειαστεί να μειωθεί η πιθανότητα εμφάνισης

μικρών βαθμών. Αυτές οι αλλαγές ενδεχομένως να συμβάλλουν στην βελτίωση της απόδοσης των BATS κωδίκων.

Μία επιπλέον πρόταση για τη βελτίωση της απόδοσης είναι η ανάπτυξη υβριδικών μεθόδων που να συνδυάζουν στοιχεία από τις υπάρχουσες τεχνικές. Για παράδειγμα, οι BATS κώδικες μπορούν να συνδυαστούν με στρατηγικές ανάδρασης σε περιβάλλοντα όπου αυτό είναι εφικτό, ώστε να μειώνεται ο αριθμός των περιττών πακέτων. Παράλληλα, η δυναμική προσαρμογή του αριθμού πακέτων ανά batch μπορεί να οδηγήσει σε καλύτερη εκμετάλλευση του διαθέσιμου εύρους ζώνης.

Τέλος, οι RLNC, LT και BATS κώδικες παρουσιάζουν διαφορετικά πλεονεκτήματα και μειονεκτήματα, ανάλογα με τις συνθήκες του δικτύου. Ενώ οι RLNC κώδικες παρέχουν τη μεγαλύτερη ευελιξία, η απλότητά τους συνοδεύεται από υψηλό υπολογιστικό κόστος. Οι LT κώδικες, από την άλλη, είναι πιο αποδοτικοί σε χαμηλές απώλειες, αλλά η απόδοσή τους μειώνεται σε δυναμικά περιβάλλοντα. Οι BATS κώδικες προσφέρουν μια ισορροπία μεταξύ αυτών των δύο προσεγγίσεων, και η περαιτέρω βελτίωσή τους μπορεί να τους καθιερώσει ως την κύρια επιλογή για εφαρμογές όπου απαιτείται υψηλή ανθεκτικότητα.

Βιβλιογραφία

- [AC++00] Ahlswede, R., Cai, N., Li, S. Y., & Yeung, R. W. (2000). Network information flow. *IEEE Transactions on Information Theory*, 46(4), 1204–1216. <https://doi.org/10.1109/18.850663>
- [BS14] Raef Bassily, Adam Smith. Casual Erasure Channels, [\[1409.3893v1\]](#) [Causal Erasure Channels](#)
- [BL05] Brauneis, W., & Linder, H. (2005). A new class of Erasure Codes and its Applications to Scalable Multicast Content Delivery in scientific computing in Salzburg, *Festschrift*
- [CLY11] Chen, Y., Li, S., & Yeung, R. W. (2011). Wireless network coding theory. *Foundations and Trends in Communications and Information Theory*, 7(3), 265-379. <https://doi.org/10.1561/01000000055>
- [CW07] Chou, P., & Wu, Y. (2007). Network Coding for the Internet and the Wireless Networks. IEE Signal Processing Magazine.
- [CWJ03] Chou, P. A., Wu, Y., & Jain, K. (2003). Practical network coding. *Proceedings of the Allerton Conference on Communication, Control, and Computing*.
- [DG++10] Dimakis, A. G., Godfrey, P. B., Wu, Y., Wainwright, M. J., & Ramchandran, K. (2010). Network coding for distributed storage systems. *IEEE Transactions on Information Theory*, 56(9), 4539-4551. <https://doi.org/10.1109/TIT.2010.2054295>
- [FS07] Fragouli, C., & Soljanin, E. (2007). Network coding applications. *Foundations and Trends in Networking*, 2(2), 135–269. <https://doi.org/10.1561/13000000002>
- [FWB06] Fragouli, C., & Widmer, J., & Boudec, J. (2006). Network Coding: An Instant Primer. ACM SIGCOMM Computer Communication Preview
- [G60] Gilbert, E. N. (1960). Capacity of a Burst-Noise Channel
- [HM++06] Ho, T., Médard, M., Koetter, R., Karger, D. R., Effros, M., Shi, J., & Leong, B. (2006). A random linear network coding approach to multicast. *IEEE Transactions on Information Theory*, 52(10), 4413–4430. <https://doi.org/10.1109/TIT.2006.881746>

- [HP++09] Janus Heide, Morten V. Pedersen, Frank H. P. Fitzek, Torben Larsen (2009). Network Coding for Mobile Devices – Systematic Binary Random Rateless Codes
- [JSC07] Jaggi, S., Sanders, P., & Chou, P. A. (2007). Polynomial time algorithms for multicast network code construction. *IEEE Transactions on Information Theory*, 51(6), 1973-1982. <https://doi.org/10.1109/TIT.2005.846423>
- [KR++08] Katti, S., Rahul, H., Hu, W., Katabi, D., Médard, M., & Crowcroft, J. (2008). XORs in the air: Practical wireless network coding. *IEEE/ACM Transactions on Networking*, 16(3), 497–510. <https://doi.org/10.1109/TNET.2008.923722>
- [KM03] Koetter, R., & Médard, M. (2003). An algebraic approach to network coding. *IEEE/ACM Transactions on Networking*, 11(5), 782-795. <https://doi.org/10.1109/TNET.2003.818197>
- [L02] M. Luby, 2002. LT-codes, in Proceedings of the ACM Symposium on Foundations of Computer Science (FOCS)
- [LY10] Li, Z., & Yeung, R. W. (2010). Network coding in wireless networks. *Proceedings of the IEEE*, 98(1), 1-28. <https://doi.org/10.1109/JPROC.2009.2035163>
- [LSL11] Liew, S. C., Zhang, S., & Lu, L. (2011). Physical-layer network coding: Tutorial, survey, and beyond. *Physical Communication*, 6(1), 4-42. <https://doi.org/10.1016/j.phycom.2011.12.002>
- [M03] David J.C. MacKay (2003). Sparse Graph Codes in Information Theory, Inference, and Learning Algorithms. Cambridge University Press.
- [M08] David J.C. MacKay (2008). "FOUNTAIN CODES". [10.1049/ip-com:20050237](https://arxiv.org/abs/10.1049/ip-com:20050237)
- [NYX17] Nguyen, D. N., Yang, X., & Xie, L. (2017). Adaptive network coding for wireless networks: Opportunities and challenges. *IEEE Communications Surveys & Tutorials*, 19(1), 4-24. <https://doi.org/10.1109/COMST.2016.2615425>
- [NB96] Jörg Nonnenmacher and Ernst W. Biersack. (1996). Reliable Multicast: Where to use FEC
- [R97] Luigi Rizzo (1997). Effective Erasure Codes for Reliable Computer Communication Protocols, <https://doi.org/10.1145/263876.263881>

- [S03] Shokrollahi Amin (2003). LDPC Codes: An Introduction, Digital Fountain, Inc.
- [SYW20] Sheng, Z., Yang, S., & Wang, Y. (2020). Network coding for 5G and beyond. *IEEE Transactions on Wireless Communications*, 19(5), 3380-3394. <https://doi.org/10.1109/TWC.2020.2975568>
- [TTP17] Evgeny Tsimbalo, Andrea Tassi and Robert J. Piechocki (2017). Reliability of Multicast under Random Linear Network Coding. <https://doi.org/10.48550/arXiv.1709.05477>
- [WCK07] Wu, Y., Chou, P. A., & Kung, S. Y. (2007). Information exchange in wireless networks with network coding and physical-layer broadcast. *IEEE Transactions on Information Theory*, 53(11), 4504-4516. <https://doi.org/10.1109/TIT.2007.908088>
- [WK02] Weatherspoon, H., & Kubiatowicz, J. (2002). Erasure Coding vs. Replication: A Quantitative Comparison.
- [YY14] Shenghao Yang, & Raymond W. Y. (2014). Batched Sparse Codes
- [YY17] Shenghao Yang, & Raymond W. Y. (2017). BATS Codes Basics in BATS Codes Theory and Practice
- [ZL06] Zhang, Q., & Liew, S. C. (2006). A framework for wireless network coding. *IEEE/ACM Transactions on Networking*, 14(4), 804-817. <https://doi.org/10.1109/TNET.2006.876217>