

---

## Εισαγωγή

---

Η παρούσα εργασία πραγματεύεται την προσομοίωση μιας σειριακής επικοινωνίας με το εικονικό εργαστήριο του μαθήματος (Ithaki virtual lab) και την διεξαγωγή πειραμάτων υλοποιώντας κώδικα σε Java. Η εργασία, πέρα από την παραλαβή πακέτων, εικόνων, στιγμάτων gprs, προσομοιώνει και σφάλματα στην επικοινωνία μεταξύ του virtual modem και του server του εικονικού εργαστηρίου. Παρακάτω ακολουθεί μία σύντομη βιβλιογραφική αναφορά τόσο σε μηχανισμούς διόρθωσης σφαλμάτων σε ληφθέντα πακέτα αλλά και στη λειτουργία των modem.

---

## Μηχανισμοί διόρθωσης σφαλμάτων

---

Οι μηχανισμοί ανίχνευσης και διόρθωσης σφαλμάτων είναι τεχνικές που επιτρέπουν την αξιόπιστη μεταφορά δεδομένων μέσα από θορυβώδη κανάλια επικοινωνίας και επιτρέπουν την ανακατασκευή των αρχικών δεδομένων χωρίς σφάλματα.

---

## Automatic Repeat Request (ARQ)

---

Το ARQ είναι ένας μηχανισμός διαχείρισης σφαλμάτων, στην μεταφορά δεδομένων (πακέτων), χρησιμοποιώντας κωδικούς ανίχνευσης acknowledgement(ACK) και negative acknowledgement(NACK), timeouts για αξιόπιστη επικοινωνία. Ο παραλήπτης χρησιμοποιεί τον κωδικό ACK για να δείξει ότι το πακέτο παραλήφθηκε σωστά και ότι είναι έτοιμος να παραλάβει το επόμενο πακέτο.

Ο αποστολέας αν δεν λάβει τον κωδικό ACK μέσα σε ένα συγκεκριμένο χρονικό διάστημα (timeout), στέλνει ξανά το πακέτο έως ότου λάβει πίσω ACK κωδικό ή μέχρι να φτάσει κάποιο μέγιστο αριθμό επανεκπομπών. Το ARQ μπορεί να χωριστεί σε τρία είδη:

- Stop-and-wait ARQ: Ο αποστολέας στέλνει το πακέτο και περιμένει να λάβει πίσω ACK κωδικό για να συνεχίσει την αποστολή των υπόλοιπων πακέτων.
- Go-Back-N ARQ: Ο αποστολέας στέλνει πακέτα (N στον αριθμό), τα οποία καθορίζονται από ένα παράθυρο. Ο παραλήπτης κρατά τον αριθμό του κάθε πακέτου που λαμβάνει και αν βρεθεί λάθος γνωρίζοντας σε ποιο ακριβώς πακέτο του παραθύρου ήταν ζητά την επανεκπομπή του.
- Selective Repeat ARQ: Είναι σαν το Go-Back-N ARQ με τη διαφορά ότι ο παραλήπτης μπορεί να παραλάβει πακέτα με οποιαδήποτε σειρά και ο αποστολέας μπορεί να στέλνει ξανά πακέτα που έχουν περάσει το timeout.

Το ARQ είναι απαραίτητο για κανάλια των οποίων η χωρητικότητα είναι είτε άγνωστη είτε εναλλάσσεται, όπως για παράδειγμα γίνεται με το Internet. Επιπλέον, χρειάζεται κανάλι επιστροφής έτσι ώστε ο παραλήπτης να στέλνει τους κωδικούς ACK/NACK. Παράλληλα, έχει και αυξημένο latency λόγω των επανεκπομπών.

---

## Forward Error Correction (FEC)

---

Το FEC είναι μια διαδικασία προσθήκης επιπλέον δεδομένων (redundant data) σε ένα πακέτο που αποστέλλεται, έτσι ώστε ο παραλήπτης να μπορεί να ανακατασκευάσει το αρχικό μήνυμα ακόμα και αν αυτό έχει υποστεί έναν συγκεκριμένο αριθμό από errors. Το FEC δεν χρειάζεται back channel γι' αυτό είναι καλύτερο για broadcasting. Συνήθως χωρίζονται σε δύο κατηγορίες:

- Convolutional codes: Ανάλυση του πακέτου bit-by-bit, κυρίως χρήσιμο για υλοποίηση σε hardware.
- Block codes: Ανάλυση ανά πακέτο, όπως επαναληπτικοί κώδικες, Hamming και parity check codes.

Το θεώρημα του Shannon είναι ιδιαίτερα σημαντικό για το FEC καθώς περιγράφει την μέγιστη πληροφορία που μπορεί να μεταφερθεί μέσω ενός καναλιού με συγκεκριμένο SNR.

## Hybrid Schemes

---

Το Hybrid ARQ είναι ένας συνδυασμός του ARQ και του FEC. Υπάρχουν δύο διαφορετικές προσεγγίσεις:

- Τα πακέτα πάντα στέλνονται με FEC (περιέχουν δηλαδή parity δεδομένα). Ο αποδέκτης αποκωδικοποιεί το μήνυμα χρησιμοποιώντας τα parity δεδομένα και ελέγχει αν το πακέτο λήφθηκε σωστά. Αν δεν έχει ληφθεί σωστά τότε χρησιμοποιεί ARQ για την επανάληψη της αποστολής του πακέτου.
- Τα δεδομένα αποστέλλονται χωρίς FEC. Ο παραλήπτης ελέγχει το πακέτο που έλαβε και αν υπάρχει σφάλμα ζητά τις parity δεδομένα FEC μέσω ARQ και τα χρησιμοποιεί για να αναδομήσει το αρχικό πακέτο.

### Συμπεράσματα:

- Αν στόχος είναι το μικρό latency τότε προτιμάται το FEC αντί του ARQ. Για παράδειγμα για τηλεφωνία ή για διάδοση υλικού από κάμερες τηλεόρασης.
- Το ARQ χρειάζεται κανάλι επιστροφής για να λειτουργήσει.
- Εφαρμογές που απαιτούν εξαιρετικά χαμηλό αριθμό σφαλμάτων, όπως για παράδειγμα χρηματικές συναλλαγές, χρησιμοποιούν ARQ.

---

## Μηχανισμοί ανίχνευσης σφαλμάτων

---

Οι μηχανισμοί ανίχνευσης βοηθούν στον εντοπισμό των σφαλμάτων.

## Repetition Codes (Επαναληπτικοί κώδικες)

---

Το πακέτο χωρίζεται σε blocks από bits. Στη συνέχεια κάθε block επαναλαμβάνεται πάνω στο κανάλι για ένα συγκεκριμένο αριθμό φορών. Στη συνέχεια, ο παραλήπτης εξετάζει το μήνυμα και ανιχνεύει τα λάθη στα επαναλαμβανόμενα bit. Οι επαναληπτικοί κώδικες είναι αναποτελεσματικοί αν συμβεί το ίδιο ακριβώς σφάλμα στα ίδια bits του block που επαναλαμβάνεται. Ο λόγος για τον οποίο προτιμώνται είναι διότι είναι εξαιρετικά απλοί στην υλοποίηση τους.

## Parity bit

---

Το parity bit είναι ένα επιπλέον bit το οποίο εισάγεται σε ένα ήδη υπάρχον bit stream και έχει ως στόχο την δημιουργία άρτιων (ή περιττών ανάλογα τη σύμβαση) αριθμό από 1 στο stream. Αν για παράδειγμα έχει συμφωνηθεί περιττός αριθμός από 1, και ο παραλήπτης βρει άρτιο αριθμό τότε ανιχνεύεται σφάλμα στη μεταφορά του πακέτου. Αξίζει να σημειωθεί ότι με το parity bit ανιχνεύεται μόνο περιττός αριθμός σφαλμάτων.

Παρόμοια λογική με περισσότερα parity bits μπορεί να ακολουθηθεί για την ανίχνευση της θέσης του σφάλματος στο bit stream, σε περίπτωση που τα σφάλματα δεν ξεπερνούν το ένα.

## Checksum

---

Το Checksum είναι ένα modular αριθμητικό άθροισμα από κωδικές λέξεις ενός μηνύματος. Το μήνυμα περνάει μέσα από μία συνάρτηση κατακερματισμού (hash function). Ο παραλήπτης γνωρίζει τη συνάρτηση αυτή, υπολογίζει το checksum και το συγκρίνει με αυτό που παρέλαβε. Αν τα δύο checksum είναι ίδια τότε το μήνυμα παραλήφθηκε σωστά.

## Cyclic redundancy check (CRC)

Χαρακτηρίζεται από ένα πολυώνυμο το οποίο χρησιμοποιείται ως διαιρέτης σε μία πολυωνυμική διαίρεση και τα δεδομένα εισόδου (bits προς μετάδοση) χρησιμοποιούνται ως διαιρετέος. Το υπόλοιπο της διαίρεσης αυτής (FCS) μεταδίδεται μαζί με το υπόλοιπο μήνυμα στον παραλήπτη. Έπειτα, ο παραλήπτης ελέγχει το υπόλοιπο της διαίρεσης  $REM[\frac{message_{bits}-FCS}{Q}]$ , αν είναι 0 τότε δεν υπήρξε error στη μεταφορά.

Το CRC είναι φτιαγμένο για την ανίχνευση burst errors και υλοποιείται εύκολα σε hardware γι' αυτό χρησιμοποιείται στα δίκτυα, αλλά και σε συσκευές αποθήκευσης.

Τέλος, υπάρχουν και άλλες τεχνικές αναγνώρισης σφαλμάτων σε δεδομένα όπως με Cryptographic hash function, με παρόμοια λειτουργία με το checksum τα data περνάν μέσα από ένα hash function και αλλαγές σε αυτά εσκεμμένα ή μη ανιχνεύονται (βοηθά στο data integrity). Επίσης, υπάρχουν και κώδικες, όπως ο κώδικας Hamming που ανιχνεύουν και διορθώνουν σφάλματα.

## Modem

Το modem (modulator-demodulator) είναι μια συσκευή η οποία μετατρέπει τα δεδομένα που προέρχονται από ψηφιακή μορφή από τον υπολογιστή, σε αναλογικά σήματα ώστε να μπορούν να διαδοθούν μέσω του τηλεφωνικού δικτύου. Επιπλέον, διαθέτει hardware και για την αντίστροφη διαδικασία, δηλαδή να λαμβάνει αναλογικά σήματα και να τα μετατρέπει σε ψηφιακά δεδομένα για την επεξεργασία τους από υπολογιστή.

Με την ανάπτυξη του Digital Subscriber Line (DSL) δόθηκε η δυνατότητα για αποστολή δεδομένων με τις ήδη υπάρχουσες γραμμές τηλεφωνίας, καθώς το DSL χρησιμοποιεί μεγαλύτερες συχνότητες από ότι η φωνή στο τηλεφωνικό δίκτυο. Τα σήματα αυτά συγκεντρώνονται στο καφάο από μία συσκευή (το Digital Subscriber Line Access Multiplexer- DSLAM) η οποία πολυπλέκει τα διάφορα σήματα που καταφθάνουν στο καφάο έτσι ώστε να είναι δυνατή η μεταφορά τους από ένα καλώδιο. Στη γραμμή που καταφθάνει στο χρήστη, για τις συσκευές που δεν χρησιμοποιούν το DSL (τηλέφωνα) είναι απαραίτητη η χρήση χαμηλοπερατού φίλτρου. Τα είδη το DSL περιγράφονται παρακάτω:

- Symmetric DSL (SDSL): Στο συμμετρικό DSL οι ταχύτητες download είναι ίδιες με τις ταχύτητες upload (έτσι και το Symmetric).
- Asymmetric DSL (ADSL): Στο ασυμμετρικό DSL οι ταχύτητες download είναι υψηλότερες από τις ταχύτητες upload (έτσι και το Asymmetric).
- Very high-speed DSL (VDSL): Προσφέρει μεγαλύτερες ταχύτητες που φτάνουν τα 52 Mbit/s download και τα 16 Mbit/s upload.
- VDSL2: Χρησιμοποιεί συχνότητες μέχρι και τα 30 Mhz και ταχύτητες που ξεπερνούν τα 100 Mbit/s σε download/upload.

Τα ίδια τα modem χρησιμοποιούν πρωτόκολλα για την διαχείριση σφαλμάτων, όπως αυτά που αναφέρθηκαν παραπάνω. Στο παρακάτω πίνακα φαίνονται κάποια από αυτά.

Πρωτόκολλο	Μέγεθος block	Συχνότητα (ACK/NACK)	Τύπος ελέγχου
<b>Xmodem</b>	128 bytes	Σε κάθε block	Checksum
<b>Xmodem CRC</b>	128 bytes	Σε κάθε block	CRC
<b>Xmodem-1K</b>	1024 bytes	Σε κάθε block	CRC
<b>Wxmodem</b>	128 bytes	Σε κάθε block αλλά δεν κρατάει τον αποστολέα πριν δεχτεί το επόμενο block	Checksum
<b>Ymodem</b>	1024 bytes	Σε κάθε block	CRC

<b>Ymodem-g</b>	1024 bytes	Only when an error is detected to blocks sent as a stream and all must arrive successfully	CRC
<b>Zmodem</b>	512 bytes	When a block with an error is detected	CRC
<b>Kermit</b>	Προσαρμόζεται ανά υπολογιστή	Όταν ανιχνευτεί σφάλμα σε κάποιο block	Checksum
<b>V.42 (LAPM)</b>	128 bytes	Μπορούν να αποσταλούν μέχρι και 15 block πρώτου να σταλεί απάντηση	CRC
<b>V.42 (MNP4)</b>	Ποικίλει	Όταν ένα block έχει σφάλμα	CRC

### Αποτελέσματα/Μετρήσεις

Στο πρώτο session η ταχύτητα του modem τέθηκε στο 1 Mbps τόσο για το πείραμα με τα echo packet όσο και για το automatic repeat request. Όπως είναι γνωστό από την εκφώνηση, το modem παρουσιάζει τυχαίες μεταβολές στη ταχύτητα, γεγονός που φαίνεται και από το διάγραμμα G1 με το χρόνο απόκρισης. Ο χρόνος απόκρισης για τα περισσότερα πακέτα είναι γύρω στα 300ms, ενώ σε περιπτώσεις όπου η ταχύτητα μεταβάλλεται τυχαία ο χρόνος απόκρισης μπορεί να φτάσει μέχρι και τα 850ms. Όμοια συμπεριφορά παρατηρείται και στο πείραμα ARQ(G2), όπου ο χρόνος απόκρισης κυμαίνεται γύρω στα 500ms, με τη διαφορά όμως τώρα ότι πέρα από τις τυχαίες μεταβολές στη ταχύτητα υπάρχουν και τυχαία σφάλματα στα πακέτα τα οποία επηρεάζουν και αυτά το χρόνο απόκρισης.

Στο δεύτερο session η ταχύτητα του modem τέθηκε στα 2 Mbps για τα echo και ARQ πειράματα. Από το αντίστοιχο διάγραμμα G1 οι χρόνοι απόκρισης για το πείραμα των echo packet είναι γύρω στα 200ms με μέγιστες τιμές κοντά στα 300ms, ενώ για το πείραμα ARQ(G2) 400ms μέχρι και 2000ms αντίστοιχα. Όπως είναι αναμενόμενο στο δεύτερο session λαμβάνονται περισσότερα πακέτα λόγω της μεγαλύτερης ταχύτητας του modem.

Στον παρακάτω πίνακα φαίνονται οι μέσοι χρόνοι απόκρισης για το εκάστοτε πείραμα και για τα δύο sessions καθώς και τα BER.

Experiment	Session 1	Session 2
<b>Echo Packet mean Response Time(ms)</b>	385.2472	227.49
<b>ARQ mean Response Time(ms)</b>	723.0452	481.02
<b>BER</b>	0.0014	0.0031

Αναφορικά με την κατανομή πιθανότητας του αριθμού επανεκπομπών, δημιουργήθηκε το bar chart(διάγραμμα G3) με τη σχετική συχνότητα κάθε σφάλματος, παρατηρήθηκε ότι η κατανομή της πιθανότητας είναι εκθετική η οποία ξεκινάει από το 1 και όχι από το 0 (ή ακριβέστερα ακολουθεί γεωμετρική κατανομή αφού μιλάμε για διακριτό αριθμό επαναλήψεων). Στη συνέχεια, η παρατήρηση αυτή επιβεβαιώνεται με χρήση της συνάρτησης fitdist του matlab, η οποία δημιουργεί την pdf η οποία φαίνεται πάνω στο διάγραμμα.

**Σημείωση:** Ως χρόνος απόκρισης του server lthaki ορίστηκε ως το χρονικό διάστημα από την αποστολή του αιτήματος του πακέτου μέχρι και τη στιγμή που κατέφθασε ο τελευταίος χαρακτήρας του πακέτου (το P από το PSTOP).

---

## Βιβλιογραφία

---

- [1] <https://en.wikipedia.org/wiki/Modem>
- [2] [https://en.wikipedia.org/wiki/Error\\_detection\\_and\\_correction](https://en.wikipedia.org/wiki/Error_detection_and_correction)
- [3] <https://searchnetworking.techtarget.com/definition/modem-error-correcting-protocols>
- [4] <https://www.javatpoint.com>
  - [i] <https://www.javatpoint.com/exception-handling-in-java>
  - [ii] <https://www.javatpoint.com/java-fileoutputstream-class>
- [5] [https://www.tutorialspoint.com/java/java\\_documentation.htm](https://www.tutorialspoint.com/java/java_documentation.htm)
- [6] <http://ithaki.eng.auth.gr/netlab/index.html>