



Αριστοτέλειο Πανεπιστήμιο Θεσσαλονίκης  
Πολυτεχνική Σχολή

## Δίκτυα Υπολογιστών Ι

Θεόδωρος Κατζάλης  
ΑΕΜ:9282  
katzalis@auth.gr

April 18, 2021

# Περιεχόμενα

<b>1</b>	<b>Εισαγωγή</b>	<b>2</b>
1.1	Statement of originality . . . . .	2
<b>2</b>	<b>Session 1</b>	<b>2</b>
2.1	G1 . . . . .	2
2.2	G2 . . . . .	2
2.3	G3 . . . . .	2
<b>3</b>	<b>Session 2</b>	<b>3</b>
<b>4</b>	<b>Ανάλυση πηγαίου κώδικα</b>	<b>3</b>
4.1	User Interface . . . . .	4
<b>5</b>	<b>Modem</b>	<b>5</b>
5.1	Ιστορική αναδρομή . . . . .	5
5.2	Πρωτόκολλα modem . . . . .	7
5.2.1	Διαμόρφωση . . . . .	7
5.2.2	Ανίχνευση και διόρθωση σφάλματος . . . . .	7

# 1 Εισαγωγή

Η συγκεκριμένη εργασία αποσκοπεί στην εξοικείωση εννοιών σχετικά με τα δίκτυα υπολογιστών τόσο σε θεωρητικό όσο και σε πρακτικό επίπεδο. Αυτό εξασφαλίζεται με την δημιουργία δικτυακών εφαρμογών χρησιμοποιώντας την γλώσσα προγραμματισμού `java` σε συνεργασία με τον server `ithaki` του μαθήματος με IP 155.207.18.208. Αυτή λοιπόν η συνεργασία αποσκοπεί στην συλλογή δεδομένων επείτα απο συγκεκριμένα request του χρήστη στον server. Αρχικά θα σχολιάσουμε τα αποτελέσματα που πήραμε σε δύο συνόδους, στη συνέχεια θα αναλύσουμε τον πηγαίο κώδικα και στο τέλος αυτού του report θα γίνει μια σύντομη αναφορά σε μηχανισμούς και πρωτόκολλα modem.

## 1.1 Statement of originality

Αξίζει να αναφερθεί ότι η εργασία ξεκίνησε με βάση το seed code της Ιθάκης. Για λόγους πληρότητας και διαφάνειας παραθέτουμε, στο τέλος της αναφοράς, τις πηγές για την επίλυση των προβλημάτων που συναντήσαμε κατά την εκπόνηση της. Τα εργαλεία που χρησιμοποιήθηκαν ήταν ένας text editor και ένας terminal emulator σε Linux περιβάλλον, αποφεύγοντας την χρήση IDE όπως Eclipse! Αυτό μας έδωσε την δυνατότητα να έχουμε μια πιο ολοκληρωμένη επίγνωση της διαδικασίας που πολλές φορές IDE προσπαθούν, για λόγους ευκολίας, να την αποκρύψουν. Ακόμη, για να διευκολυνθεί η διαδικασία του compile, χρησιμοποιήσαμε make για build automation. Για την δημιουργία των διαγραμμάτων, χρησιμοποιήσαμε `pytho` και την βιβλιοθήκη `matplotlib`. Τέλος, το συγκεκριμένο έγγραφο έγινε με την χρήση  $\text{\LaTeX}$ !

## 2 Session 1

Σχετικά με τα διαγράμματα του session 1 έχουμε να αναφέρουμε τα εξής:

### 2.1 G1

Για να μπορούμε να καταλάβουμε τις ιδιότητες του χρόνου απόκρισης των echo packets συμπληρώσαμε στα γραφήματα ένα ιστόγραμμα και μια zoom in εκδοχή του διαγράμματος. Θα μπορούσαμε να ισχυριστούμε ότι ο χρόνος απόκρισης ακολουθεί bimodal κατανομή εξαιτίας των δύο κορυφών. Ακόμη βλέποντας την zoom in εκδοχή παρατηρούμε μια ταλάντωση γύρω απο την τιμή 30 milliseconds, στην οποία βρίσκεται και η κορύφωση του ιστογράμματος.

### 2.2 G2

Σχετικά με το ARQ παρατηρούμε spikes στο διάγραμμα της απόκρισης του χρόνου. Αυτό είναι αναμενόμενο, διότι σε αυτά τα σημεία είχαμε επαναλαμβανόμενες αποτυχίες λήψης ορθού πακέτου, με αποτέλεσμα να υπάρχει χρονοκαθυστέρηση εξαιτίας των επιπρόσθετων NACK πακέτων.

### 2.3 G3

Ωστόσο, αυτά τα spikes δεν εμφανίζονται σε πολύ μεγάλη συχνότητα με αποτέλεσμα η πλειοψηφία των πακέτων να ολοκληρώνουν την διαδικασία λήψης πληροφορίας περίπου σε 50 milliseconds. Έτσι βλέποντας το ιστόγραμμα G3, μπορούμε να ισχυριστούμε ότι ακολουθεί right skewed κατανομή.

### 3 Session 2

Σχετικά με τα διαγράμματα του session 2 έχουμε να αναφέρουμε τα εξής:

- Παρατηρούμε ότι μπορούμε να εξάγουμε παρόμοια συμπεράσματα με αυτά του session 1.

### 4 Ανάλυση πηγαιού κώδικα

Η αναγνώριση της ολοκληρωμένης λήψης **echo πακέτου** μπορεί να γίνει με διάφορες προσεγγίσεις. Αξίζει να σημειωθεί ότι στα αρχικά στάδια ανάπτυξης του κώδικα, δεν είχαμε κάποια τεχνική αναγνώρισης και χρησιμοποιούσαμε την ίδια λογική του seed code της Ιθάκης. Δηλαδή ελέγχαμε αν συναντούσαμε τιμή -1 κατά τη ροή λήψης πληροφορίας, το οποίο σηματοδοτούσε ότι η ροή δεν είναι πλέον διαθέσιμη. Στη συνέχεια, παρατηρώντας την μορφή του πακέτου, χρησιμοποιήσαμε ως λέξη κλειδί ολοκλήρωσης το "PSTOP". Συνεπώς για την υλοποίηση αυτού του εντοπισμού, δημιουργήσαμε μια ουρά FIFO η οποία δέχεται διαδοχικά όλα τα σύμβολα του μηνύματος ένα προς ένα. Μόλις συναντήσει το "PSTOP" γίνεται έξοδος της επανάληψης λήψης χαρακτήρων. Φυσικά η συγκεκριμένη αναγνώριση έπρεπε να γίνει προκειμένου να έχει νόημα και η χρονική απόκριση λήψης πακέτων που παρουσιάστηκε στα αρχεία με τα διαγράμματα.

Όσον αφορά τον τερματισμό της εικόνας, μετατρέπαμε την λαμβανόμενη πληροφορία σε δεκαεξαδική μορφή και ελέγχαμε τον delimiter 0xFFD9. Στη συνέχεια αποθηκεύαμε το αρχείο σε μορφή jpg.

Σχετικά με το GPS και τα στίγματα, αξίζει να σημειωθεί ότι για λόγους χρηστικότητας, προσπαθήσαμε να αυτοματοποιήσουμε την διαδικασία εντοπισμού στιγμάτων που απέχουν σε σημαντικό βαθμό χρονικά, έτσι ώστε να είναι ευδιάκριτη η διαφορά τους πάνω στην εικόνα. Πιο συγκεκριμένα, συλλέγοντας GPS data points σε **NMEA format**, κάνοντας parse τα δεδομένα, απομονώσαμε το latitude και το longitude και βρίσκαμε τα σημεία τα οποία διέφεραν στις δύο αυτές τιμές. Φυσικά για να ικανοποιηθεί η παραπάνω απαίτηση, τα σημεία αυτά βλέποντας στη συνέχεια και το χρονικό τους αποτύπωμα, απείχαν τουλάχιστον 4 δευτερόλεπτα μεταξύ τους.

Αρκετά σημαντική λεπτομέρεια στην διαδικασία αποσύνθεσης του NMEA format ήταν η μορφή του **GPS request code**. Απαιτείται να εισάγεται πρώτα το γεωγραφικό πλάτος και στη συνέχεια το γεωγραφικό μήκος όταν συγχωνεύουμε τις συμβολοσειρές για να δημιουργήσουμε το GPS code, το οποίο είναι ανάποδο με την σειρά την οποία έρχονται τα δεδομένα ως NMEA format! Ακόμη έπρεπε να μετατρέψουμε σε δευτερόλεπτα το δεκαδικό κομμάτι της πληροφορίας των λεπτών που λαμβάναμε, πολλαπλασιάζοντας το με 0,6.

Η διαδικασία του **ARQ** είχε αρχικά παρόμοια απαίτηση απομόνωσης της ωφέλιμης πληροφορίας με το GPS. Πιο συγκεκριμένα παίρνοντας τα δεδομένα, απομονώσαμε το FCS και την κρυπτογραφημένη συμβολοσειρά. Μετατρέψαμε την συμβολοσειρά σε έναν πίνακα απο χαρακτήρες και στη συνέχεια με διαδοχικές πράξεις XOR υπολογίζαμε αν τελικά η λήψη του πακέτου ήταν ορθή συγκρίνοντας το αποτέλεσμα του XOR με το FCS. Διαφορετικά επαναλαμβάναμε την διαδικασία μέχρι το FCS να ισούται με την κωδικοποιημένη τιμή της ακολουθίας.

Μια λίστα με τις αναφορές που χρησιμοποιήθηκαν κατά την εκπόνηση της εργασίας είναι η ακόλουθη:

- Μια γρήγορη ανακεφαλαίωση της γλώσσας προγραμματισμού java [7]
- Πως να μετατρέψεις ακέραιους σε byte array; [5]
- Πως να αποθηκεύσεις byte array σε αρχείο; [9]
- Μετατροπή byte σε hexadecimal [12]


- Αποσύνθεση συμβολοσειράς [13]
- Ημερομηνία και ώρα [10]
- XOR array [8]
- Java API [11]
- Java Best Practices [3]

Επίσης να σημειώσουμε ότι προσπαθήσαμε ως επι το πλείστον να κάνουμε καλή διαχείριση errors και **try, catch** blocks προκειμένου να εντοπίζουμε γρήγορα προβλήματα. Για debugging λόγους χρησιμοποιούσαμε print statements σε συνεργασία με command line tools όπως jdb.

## 4.1 User Interface

Φυσικά για αισθητικούς λόγους, προτιμήσαμε κατά την έναρξη της εφαρμογής, να τυπώνουμε το παρακάτω κείμενο (ASCII art)!

```
x tkatz@katzarch ~/repos/ece-networks1 master ± make build
javac -cp lib/ithakimodem.jar -d classes src/ARQ.java src/Echo.java src/GPS.java src/Image.java src/UserApplication.java
tkatz@katzarch ~/repos/ece-networks1 master ± make run
```



```
Packet No0: PSTART 11-04-2021 12:29:32 01 PSTOP
Total time: 34 (ms)

Packet No1: PSTART 11-04-2021 12:29:32 02 PSTOP
Total time: 34 (ms)

Packet No2: PSTART 11-04-2021 12:29:32 03 PSTOP
Total time: 33 (ms)

Packet No3: PSTART 11-04-2021 12:29:32 04 PSTOP
Total time: 27 (ms)

Packet No4: PSTART 11-04-2021 12:29:32 05 PSTOP
Total time: 37 (ms)

Packet No5: PSTART 11-04-2021 12:29:32 06 PSTOP
Total time: 38 (ms)
```

Figure 1: Computer Networks  
Welcome screen!

Στη συνέχεια ακολουθεί μια σύντομη βιβλιογραφική αναφορά σε μηχανισμούς και πρωτόκολλα modem.

## 5 Modem

Η αγγλική λέξη Modem προέρχεται απο την σύνθεση των λέξεων **MOD**ulation και **DE**mod-**ulation**. Το πρώτο υποδεικνύει την διαμόρφωση του ψηφιακού σήματος σε αναλογικό και το δεύτερο την αποδιαμόρφωση, η οποία είναι η αντίστροφη διαδικασία, δηλαδή την μετατροπή απο αναλογικό σε ψηφιακό. Η διαμόρφωση είναι μια απο τις βασικές έννοιες των τηλεπικοινωνιών και ο ρόλος της είναι ο μετασχηματισμός της πληροφορίας σε μια μορφή κατάλληλη η οποία μπορεί να προσαρμοστεί και να ανταπεξέλθει στις ιδιότητες του μέσου επικοινωνίας (θόρυβος) σε συνδυασμό με τις απαιτήσεις (για παράδειγμα πολυπλεξία). Το modem λοιπόν αποτελεί δομικό στοιχείο σε ένα δίκτυο υπολογιστών και θα δούμε το γιατί αναλύοντας την δομή και τις εφαρμογές του. Πολλές φορές συγχέονται οι έννοιες **router** και **modem**, διότι συνήθως, σε οικιακό επίπεδο, έχουμε στα χέρια μας συσκευές οι οποίες υλοποιούν και τις δύο λειτουργίες. Ωστόσο το router είναι μια ψηφιακή συσκευή η οποία είναι υπεύθυνη για την ασύρματη και την ενσύρματη μετάδοση της ψηφιακής πληροφορίας στο τοπικό δίκτυο μέσω Wifi και Ethernet. Αποτελεί δηλαδή μέσο διάδοσης πληροφορίας η οποία προήλθε σε αυτόν τον τοπικό κόμβο μέσω της σύνθετης υποδομής των δικτύων και μεταφράστηκε με την βοήθεια του modem [14].

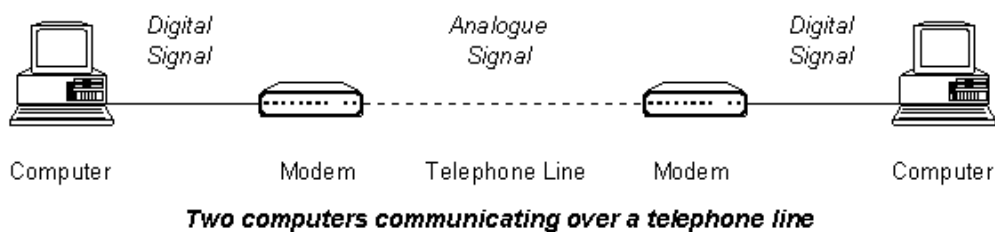


Figure 2: source: <https://ictsmart.tripod.com/ict4/online/artmoin.htm>

### 5.1 Ιστορική αναδρομή

Τις εποχές όπου δεν είχε κάθε σπίτι τουλάχιστον απο έναν προσωπικό υπολογιστή, οι τηλεφωνικές γραμμές ήταν η βασικότερη υποδομή επικοινωνίας με σκοπό την μετάδοση των αναλογικών σημάτων που προέκυπταν απο transducers και τις φωνές των ανθρώπων. Αυτό είναι το λεγόμενο Public Switched Telephone Network (PSTN). Τα modems που χρησιμοποιούν την συγκεκριμένη υποδομή ονομάζονται "voicebands" εξαιτίας του στενού εύρους ζώνης της ανθρώπινης φωνής.

Σε αυτήν την κατηγορία, εκμετάλλευσης του χαλκού του PSTN, ανήκουν επίσης τα λεγόμενα dial-up και DSL (Digital Subscriber Line) modems. Κοινό τους χαρακτηριστικό ήταν η χρήση του τηλεφωνικού δικτύου. Ωστόσο το πρώτο σε αντίθεση με το δεύτερο δεν επέτρεπε την ταυτόχρονη χρήση τηλεφώνου και περιήγησης στο διαδίκτυο. Τα dial-up modems χρησιμοποιούσαν ακριβώς τις ίδιες γραμμές μετάδοσης πληροφορίας στις ίδιες συχνότητες για δύο διαφορετικούς σκοπούς, με αποτέλεσμα οποιαδήποτε χρήση τηλεφώνου, την στιγμή μεταφοράς ψηφιακών δεδομένων, να προσθέτει θόρυβο στην επικοινωνία. Τα DSL ωστόσο δεν έχουν το ίδιο μειονέκτημα. Τα καλώδια χαλκού των τηλεφωνικών γραμμών είναι ικανά να μεταδώσουν πληροφορία μεγαλύτερων συχνοτήτων της ανθρώπινης φωνής. Οπότε το DSL εκμεταλλεύεται ακριβώς αυτο το capacity και η μετάδοση πληροφορίας με σκοπό την επικοινωνία υπολογιστών γίνεται σε διαφορετική υψηλότερη ζώνη συχνοτήτων απο αυτή του τηλεφώνου [4].

Απο την άλλη πλευρά υπάρχουν και τα **cable modems**. Η υποδομή τους βασίζεται σε fibre-coaxial κανάλια τα οποία αποσκοπούν στην μετάδοση τηλεοπτικών σημάτων με υψηλό εύρος ζώνης. Συνήθως το τηλεοπτικό σύστημα αποτελούσε ενδιαμέσο κομμάτι μεταξύ της σύνδεσης ενός modem με το υπόλοιπο διαδίκτυο (Internet), δηλαδή με την υποδομή των Internet Service Providers.

Μάλιστα μιας και αυτό το δίκτυο ταυτόχρονα χρησιμοποιούνταν και για τηλεοπτικά σήματα, οι συχνότητες θα έπρεπε να επιλεγθούν κατάλληλα για να αποφευχθούν παρεμβολές.

Ειδική κατηγορία των DSL αποτελούν τα asynchronous DSL (ADSL). Χαρακτηριστικό τους είναι η ασυμμετρία στην ταχύτητα μετάδοσης πληροφορίας ανάμεσα στο upload και το download, με το δεύτερο να υπερέχει σημαντικά σε σχέση με το πρώτο. Ο λόγος αυτής της ασυμμετρίας είναι σχεδιαστική επιλογή, διαχειρίζοντας κατάλληλα το εύρος ζώνης, εξαιτίας της συνηθισμένης απαίτησης λήψης πληροφορίας σε μεγαλύτερο όγκο σε σχέση με το ανέβασμα.

Η ανάγκη για υψηλότερους ρυθμούς μετάδοσης πληροφορίας σε συνδυασμό με την εμφάνιση των οπτικών ινών, οδήγησε στο VDSL (very high bit rate DSL). Ένα από τα βασικά μειονέκτηματά του DSL είναι η εξάρτηση της ταχύτητας από το μήκος των καλωδίων. Προκειμένου να αντιμετωπιστεί αυτό το πρόβλημα, η παρεμβολή **οπτικών ινών** στο δίκτυο **χάλκινων αγωγών**, μικραίνει το κομμάτι της χάλκινης διαδρομής, επιτυγχάνοντας έτσι υψηλότερους ρυθμούς μετάδοσης πληροφορίας. Φυσικά η εγκατάσταση οπτικής ίνας καθόλης της διαδρομής της επικοινωνίας αποτελεί μια σύγχρονη προσέγγιση με πολύ υψηλές ταχύτητες (Fiber to the home). Ωστόσο προσπαθούμε σε γενικές γραμμές να χρησιμοποιούμε ότι ήδη υπάρχει (χάλκινους αγωγούς τηλεφωνικού δικτύου). Οπότε το VDSL θα μπορούσαμε να το χαρακτηρίσουμε μια υβριδική επιλογή μεταξύ οπτικής ίνας και χαλκού. Για να επιτευχθεί κάτι τέτοιο, οι τηλεφωνικές εταιρείες προσπαθούν να αντικαταστήσουν τμήματα χαλκού με οπτικές ίνες μέχρι το κουτί παρόχου (Fiber to the Curb). Στη συνέχεια η πληροφορία διακλαδίζεται στους πελάτες με μικρές πλέον χάλκινες διαδρομές μέχρι να φτάσει στον εκάστοτε τελικό κόμβο [2].

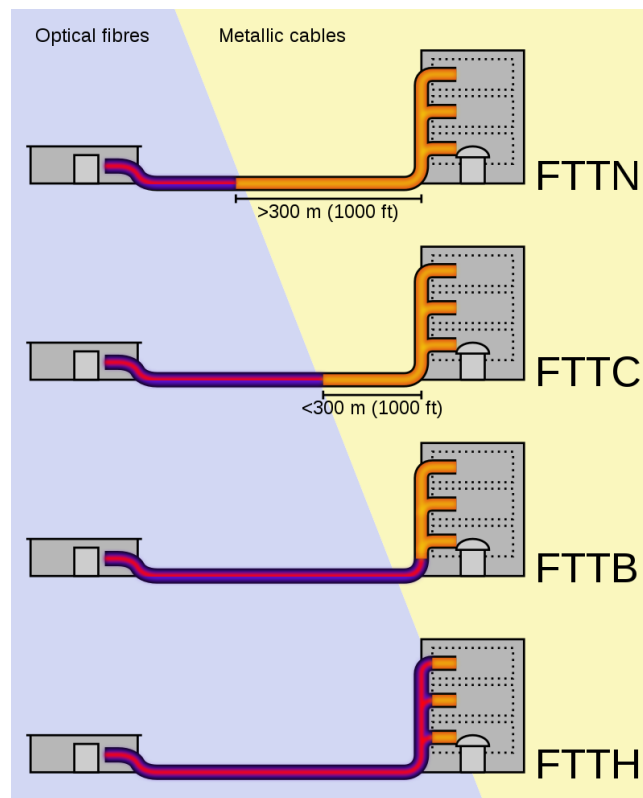


Figure 3: Fiber to the x  
source

## 5.2 Πρωτόκολλα modem

### 5.2.1 Διαμόρφωση

Μια απο τις μεθόδους διαμόρφωσης της ψηφιακής πληροφορίας σε αναλογική είναι η διαμόρφωση μετατόπισης συχνότητας (FSK). Πιο συγκεκριμένα ας υποθέσουμε ένα πολύ απλό παράδειγμα όπου ο στόχος είναι να μεταδώσουμε το γράμμα 'α' μεταξύ δύο υπολογιστών. Σύμφωνα με τον κώδικα ASCII η bit αναπαράσταση του 'α' είναι 01100001. Ορίζοντας δύο σήματα διαφορετικών συχνοτήτων (δύο τόνοι) και αντιστοιχίζοντας τα με τις λογικές τιμές 1 και 0, έχουμε πλέον διαμορφώσει την ψηφιακή πληροφορία σε αναλογική. Παρόμοιες τεχνικές διαμόρφωσης είναι και η διαμόρφωση μετατόπισης φάσης (PSK), στην οποία πλέον η διαφορά έγκειται στην φάση μεταξύ των σημάτων. Πολύ γνωστή τεχνική είναι και το QAM, του οποίου το πλεονέκτημα είναι να εμπεριέχεται μεγάλη ποσότητα πληροφορίας σε στενό εύρος ζώνης [1].

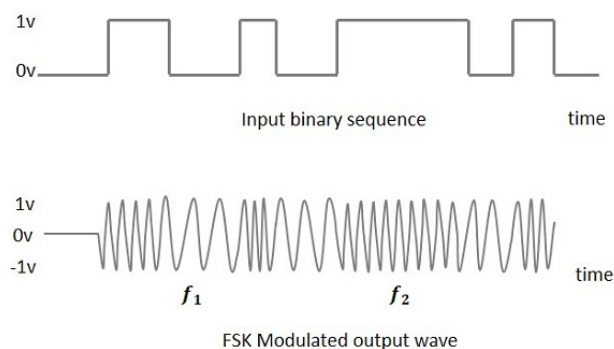


Figure 4: [source](#)

### 5.2.2 Ανίχνευση και διόρθωση σφάλματος

Ευρέως διαδεδομένη τεχνική ανίχνευσης σφαλμάτων σε μια επικοινωνία είναι τα λεγόμενα checksums και το Cyclic Redundancy Check (CRC). Η κεντρική ιδέα είναι η προσθήκη ενός check value στο τέλος του μηνύματος το οποίο υπολογίζεται αλγοριθμικά με βάση την αλληλουχία bits που πρόκειται να σταλούν. Στη συνέχεια αυτή η τιμή, απο την πλευρά του δέκτη, υπολογίζεται ξανά με βάση το λαμβανόμενο πακέτο και συγκρίνεται με το αρχικό check value. Αν οι τιμές αυτές είναι διαφορετικές σημαίνει ότι υπάρχει σφάλμα στο μήνυμα μετάδοσης, με αποτέλεσμα να ζητείται επαναποστολή του μηνύματος ή να εφαρμόζεται κάποια τεχνική διόρθωσης του.

Οι αλγόριθμοι που βασίζονται στον υπολογισμό αυτού του check value υλοποιούνται σε hardware ή σε software πραγματοποιώντας πολυωνμική διαίρεση. Πιο συγκεκριμένα θα μπορούσαμε να αναφέρουμε το εξής: Έστω ότι θέλουμε να μεταδώσουμε το ακόλουθο bit stream 1101011011. Οι ακολουθίες bits σε CRC μπορούν να αντιμετωπιστούν ως συντελεστές πολυωνύμων. Ένα πολώνυμο της μορφής  $x^4 + x + 1$  αναπαριστάται σε 10011. Έχουμε δηλαδή ένα πολώνυμο 4ης τάξης. Στη συνέχεια προσθέτουμε 4 μηδενικά, όσα δηλαδή και η τάξη του πολυωνύμου, στο αρχικό bit stream και έχουμε 11010110110000. Τέλος πραγματοποιούμε διαίρεση μεταξύ του 11010110110000 και 10011. Το υπόλοιπο της διαίρεσης είναι το check value (1110). Φυσικά η επιλογή της πολυωνμικής συνάρτησης που θα πραγματοποιηθεί η διαίρεση έχει ιδιαίτερα σημαντικό ρόλο και αποτελεί αντικείμενο βελτίωσης για την εκάστοτε εφαρμογή. Τελικά, η τιμή που θα στέλναμε θα ήταν 11010110111110 [6].



## References

- [1] How Modems Work. <https://computer.howstuffworks.com/modem2.htm>.
- [2] How VDSL Work. <https://computer.howstuffworks.com/vdsl.htm>.
- [3] A short summary of Java coding best practices. <https://rhamedy.medium.com/a-short-summary-of-java-coding-best-practices-31283d0167d3>.
- [4] Britanica: Modem. <https://www.britannica.com/technology/modem/The-second-generation>.
- [5] Convert integer into byte array (Java). <https://stackoverflow.com/questions/1936857/convert-integer-into-byte-array-java>.
- [6] Cyclic Redundancy Check | CRC | Example. <https://www.gatevidyalay.com/cyclic-redundancy-check-crc-error-detection/>.
- [7] Derek Banas Java tutorial. <https://www.youtube.com/watch?v=n-xAqcBCws4>.
- [8] Find XOR of all elements in an Array. <https://www.geeksforgeeks.org/find-xor-of-all-elements-in-an-array/>.
- [9] Java Code Geeks: Write byte array to file with FileOutputStream. <https://examples.javacodegeeks.com/core-java/io/fileoutputstream/write-byte-array-to-file-with-fileoutputstream/>.
- [10] Java Date and Time. [https://www.w3schools.com/java/java\\_date.asp](https://www.w3schools.com/java/java_date.asp).
- [11] Java™ Platform, Standard Edition 7 API Specification. <https://docs.oracle.com/javase/7/docs/api/index.html>.
- [12] Programiz: Convert Byte Array to Hexadecimal. <https://www.programiz.com/java-programming/examples/convert-byte-array-hexadecimal>.
- [13] Stackexchange: How to parse string in Java? <https://stackoverflow.com/questions/950409/how-to-parse-this-string-in-java>.
- [14] Wikipedia: Modem. <https://en.wikipedia.org/wiki/Modem>.