



## ΔΙΚΤΥΑ ΥΠΟΛΟΓΙΣΤΩΝ 1

( 6<sup>ο</sup> εξάμηνο )

Όνομ/νυμο: Κυπαρίσσης Κυπαρίσσης

A.E.M.:

Εργασία δικτυακού προγραμματισμού  
JAVA SERIAL COMMUNICATIONS PROGRAMMING

# REPORT

Το αποθετήριο της εργασίας βρίσκεται στο GitHub, κάνοντας *click* στο παρακάτω κουμπί:



[ <https://github.com/Kyparissis/Networks1-2022-Assignment> ]

Για οποιαδήποτε διευκρίνιση ή απορία για την εργασία αυτή ή εντοπίσετε κάποιο λάθος,  
παρακαλώ επικοινωνήστε μαζί μου!

⇒ Ακαδημαϊκό e-mail: [kyparkypar@ece.auth.gr](mailto:kyparkypar@ece.auth.gr)

# ΔΙΚΤΥΑ ΥΠΟΛΟΓΙΣΤΩΝ 1

Όνομ/νυμο: Κυπαρίσσης Κυπαρίσσης

Α.Ε.Μ.:

[ 2 ]

Η αναφορά αυτή περιέχει ΣΥΝΤΟΜΑ σχόλια και παρατηρήσεις στις μετρήσεις και τα διαγράμματα που παρουσιάσαμε, στα *session1.pdf* και *session2.pdf*, αλλά και μία ΜΙΚΡΗ βιβλιογραφική τεχνική αναφορά σε μηχανισμούς και πρωτόκολλα λειτουργίας των διατάξεων modem που χρησιμοποιούνται σε παρόμοιες εφαρμογές με αυτή που δημιουργήσαμε.

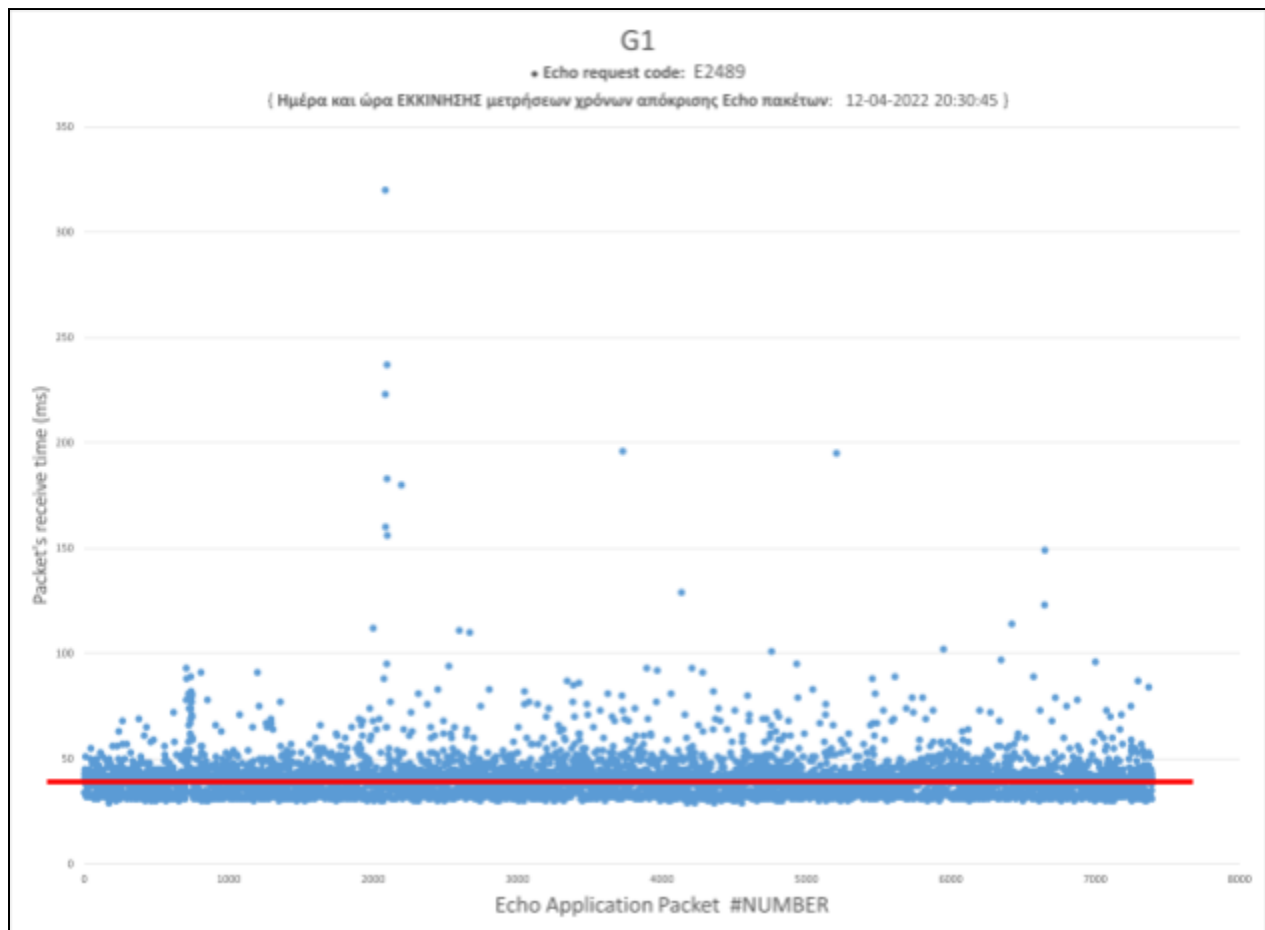
( Σχολία σχετικά με τον τρόπο και λογική λειτουργίας του πηγαιου κωδικα βρισκονται με την μορφη comments πανω στον ιδιο τον κωδικα! )

## ΠΕΡΙΕΧΟΜΕΝΑ

<b>ΣΧΟΛΙΑ ΚΑΙ ΠΑΡΑΤΗΡΗΣΕΙΣ</b>	<b>3</b>
<b>SESSION #1</b>	<b>3</b>
G1	3
G2	4
G3	5
E1	6
E2	6
M1	6
BER	7
<b>SESSION #2</b>	<b>8</b>
G1	8
G2	9
G3	10
E1	11
E2	11
M1	11
BER	12
<b>ΣΥΜΠΕΡΑΣΜΑΤΙΚΑ</b>	<b>12</b>
<b>ΒΙΒΛΙΟΓΡΑΦΙΚΗ ΤΕΧΝΙΚΗ ΑΝΑΦΟΡΑ</b>	<b>13</b>
Τί είναι το Modem;	13
Η λογική λειτουργίας του Modem	14
ΜΕΘΟΔΟΙ ΕΠΙΚΟΙΝΩΝΙΑΣ	14
ΣΕΙΡΙΑΚΗ ΕΠΙΚΟΙΝΩΝΙΑ ΚΑΙ ΈΛΕΓΧΟΣ ΡΟΗΣ	16
ΛΕΙΤΟΥΡΓΙΕΣ ΕΝΤΟΛΩΝ ΚΑΙ ΔΕΔΟΜΕΝΩΝ	16
ΑΝΪΧΝΕΥΣΗ ΚΑΙ ΔΪΟΡΘΩΣΗ ΣΦΆΛΜΑΤΟΣ	17
Τελευταία σχόλια και συμπεράσματα	20

**ΣΧΟΛΙΑ ΚΑΙ ΠΑΡΑΤΗΡΗΣΕΙΣ****SESSION #1****G1**

Παρατηρώντας το ιστογράμμα του χρόνου αποκρισης των Echo πακετων, για την χρονικη διαρκεια που τρεξαμε το *EchoApplication()*, παρατηρουμε οτι ο χρονος αποκρισης μενει σχετικα **ΣΤΑΘΕΡΟΣ**.



Αυτο μπορεί να φανει και απο την **κόκκινη γραμμή** (ευθεια γραμμικης παλινδρομησης) που εχουμε σχεδιασει και παρατηρουμε οτι οι χρονοι αποκρισης κυμαινονται κοντα σε αυτην και τα σημεια ειναι πολυ πυκνα γυρω απο την ευθεια

αυτη, δηλαδη εχουμε, κοντα στην μέση τιμή των 39 (ms) , σταθερή τιμή χρόνου λήψης πακέτου.

Παρολαυτα, παρατηρουμε και τιμες που βρισκονται σχετικα μακρια απο την γραμμη αυτη, δηλαδη χρονους  $\gg 39$  (ms).

Αυτο συμβαινει καθως μπορεί, κατα την χρονικη στιγμη ληψης του Echo πακετου εκεινου, να υπηρξε καποιο προβλημα με την σύνδεση μας στον server Ithaki και να δημιουργηθηκε ετσι η καθυστερηση αυτη στην λήψη του αντιστοιχου πακέτου. (Noise μετρήσεων).

Ομως, επειδη οι ακραιες τιμες αυτες ειναι λιγες σχετικα με τις αλλες που βρισκονται πολυ κοντα στη μέση τιμή των 39 (ms), μπορουμε να ισχυριστούμε οτι ο χρονος αποκρισης παραμενει σχετικα σταθερος!!!

# ΔΙΚΤΥΑ ΥΠΟΛΟΓΙΣΤΩΝ 1

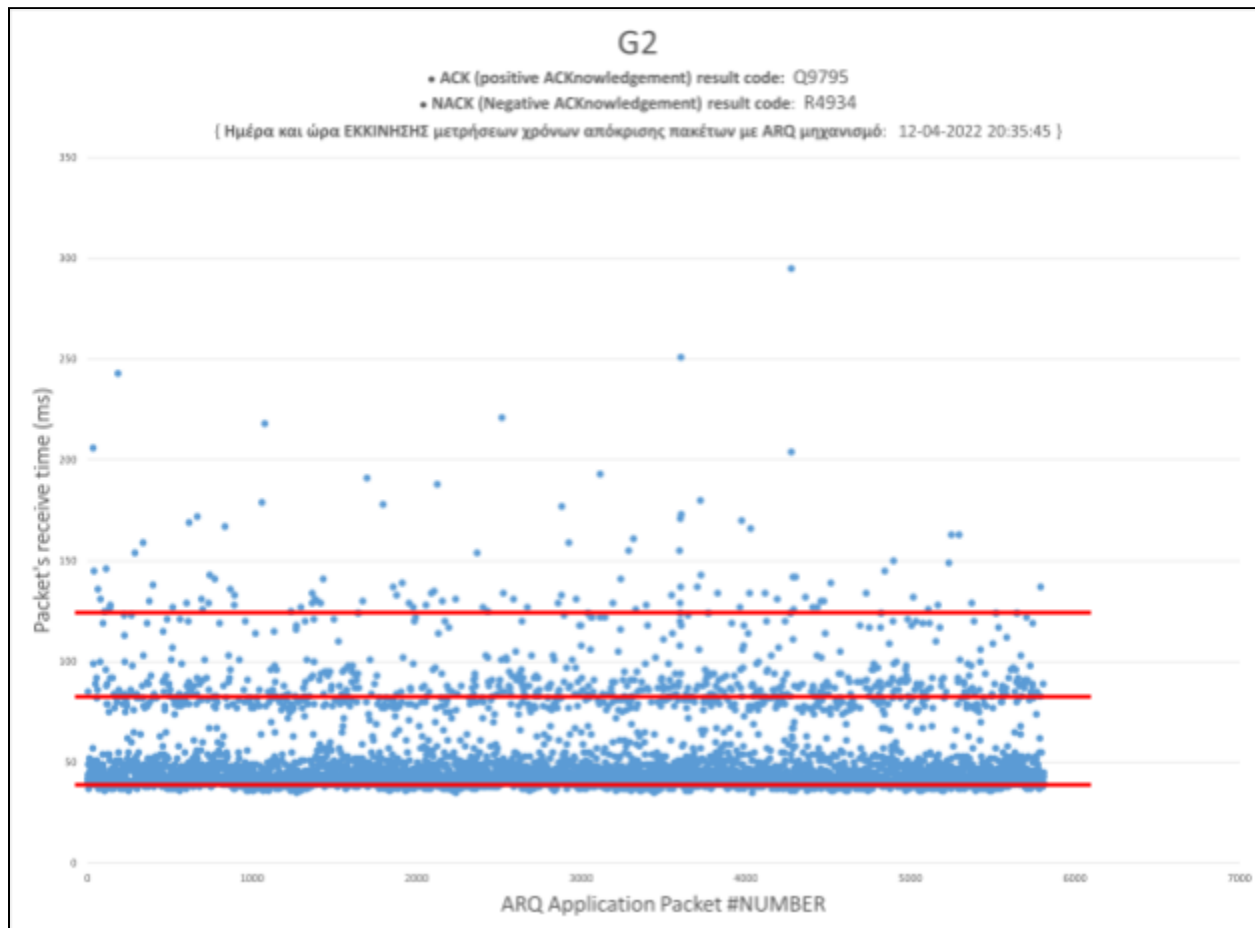
Όνομ/νυμο: Κυπαρίσσης Κυπαρίσσης

Α.Ε.Μ.:

[ 4 ]

**G2**

Παρατηρώντας το ιστογράμμα του χρόνου αποκρίσης των ARQ πακέτων, για την χρονική διάρκεια που τρέξαμε το `ARQApplication()`, παρατηρούμε ότι ο χρόνος αποκρίσης **ΑΥΞΑΝΕΤΑΙ ΓΡΑΜΜΙΚΑ ΜΕ ΤΟΝ ΑΡΙΘΜΟ ΕΠΑΝΕΚΠΟΜΠΩΝ ΤΟΥ ΠΑΚΕΤΟΥ**.



Αυτο μπορεί να φανει και απο τις **κοκκινες γραμμες** (ευθειες γραμμικης παλινδρομησης) που εχουμε σχεδιασει και παρατηrouμε οτι οι χρονοι αποκρισης κυμαινονται κοντα σε αυτες και τα σημεια ειναι πολυ πυκνα γυρω απο τις ευθειες αυτες (Πυκνοτερα στους χαμηλοτερους χρονους, αραιωτερα στους μεγαλυτερους χρονους λογω της μικρης πιθανοτητας να χρειαστει να λαβουμε ενα πακετο πολλες φορες).

Παρατηrouμε λοιπον οτι εχουμε πολλαπλα επιπεδα (σχετικα) σταθερων χρονων λήψης πακέτων. Αυτη η χρονοκαθυστέρηση συμβαινει καθως ετρεχε και ο ARQ μηχανισμος που ελεγχε για τυχόν λαθη στην ληψη του πακετου και αναλογα με το ποσες φορες ετρεξε για το αντιστοιχο πακετο, το πακέτο βρισκεται κοντα στην αντιστοιχη ευθεια/επιπεδο. Δηλαδη :

- Αν ενα πακετο χρειαστηκε να σταλθει 1 φορα, βρισκεται στη 1η ευθεια, δηλαδη **~45 (ms)**
- Αν ενα πακετο χρειαστηκε να σταλθει 2 φορες καθως στην πρωτη λήφθηκε με errors, βρισκεται στη 2η ευθεια, δηλαδη **~80 (ms)**
- Αν ενα πακετο χρειαστηκε να σταλθει 3 φορες καθως στην πρωτη λήφθηκε με errors, βρισκεται στη 3η ευθεια, δηλαδη **~120 (ms)**
- Κ.Ο.Κ...

(Δηλαδη περιπου 40 (ms) διαρκει η ληψη παλι και ο επανελεγχος του ιδιου πακετου για errors)

Τελος παρατηrouμε οτι και εδω εχουμε ακραιες τιμες χρονων που δεν βρισκονται κοντα στις ευθειες (Noise μετρήσεων λογω errors στην συνδεση με την Ιthaki)

# ΔΙΚΤΥΑ ΥΠΟΛΟΓΙΣΤΩΝ 1

Όνομ/νυμο: Κυπαρίσσης Κυπαρίσσης

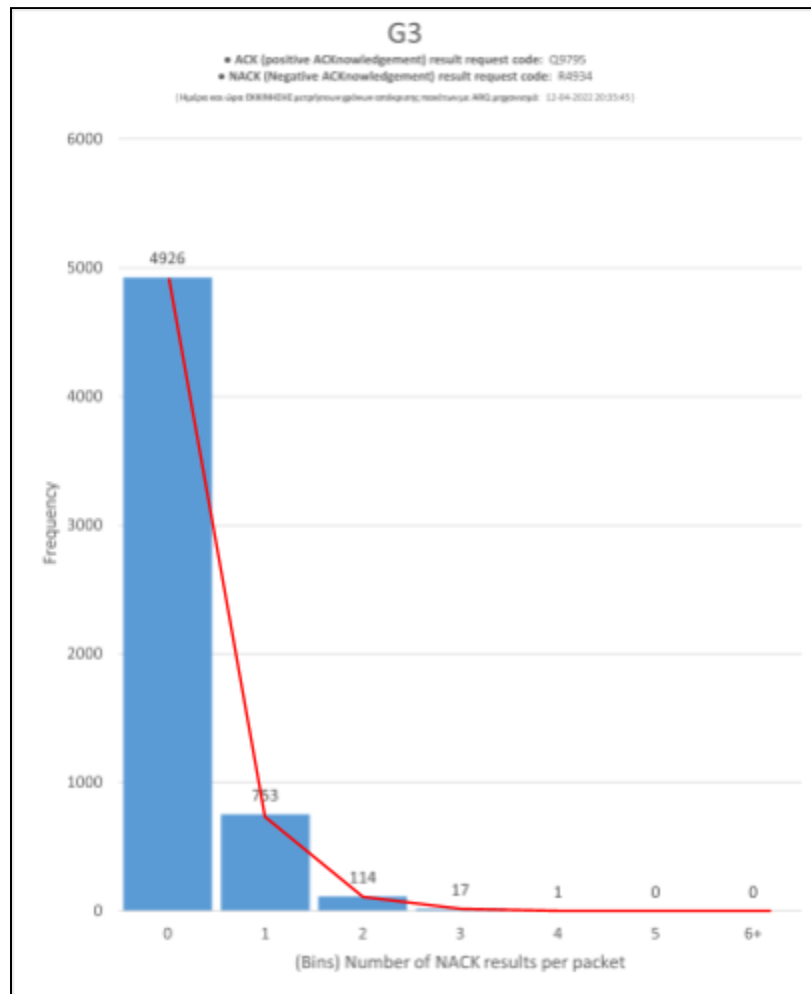
Α.Ε.Μ.:

[ 5 ]

**G3**

Παρατηρώντας το ιστόγραμμα συχνοτήτων του αριθμού επανεκπομπών (Negative acknowledgement result) των πακέτων που καταγράψαμε (G3), μπορούμε να έχουμε μια εκτίμησή για την κατανομή πιθανότητας του αριθμού αυτού για κάθε πακέτο, εφόσον έχουμε λάβει και έναν αρκετά μεγάλο αριθμό δειγμάτων / error-free πακέτων (5811 error-free πακέτα / positive acknowledgement results) .

Παρατηρούμε ότι η **[1]** κορυφή της κατανομής βρίσκεται εκτός κεντρου πλησιάζοντας το αριστερο ακρο (εδω η κορυφη ειναι το αριστερο ακρο) και **[2]** η ουρα του τεντωνεται μακρια απο την κορυφη αυτη, προς τα δεξια. (Δείτε την **κόκκινη γραμμή** στο γράφημα G3).



Η ασυμμετρία αυτή υπάρχει λογο υπαρξης ενός φυσικου ορίου που δεν μπορεί να μας δώσει αποτελεσματα απο την μια μερια (αφου δεν μπορούμε να έχουμε αριθμο επανεκπομπής, σε ενα πακετο, μικροτερο του μηδενος (ΦΥΣΙΚΟΣ ΑΡΙΘΜΟΣ) ).

Ετσι συμπερενουμε οτι εχουμε:



**ΛΟΞΗ ΚΑΤΑΝΟΜΗ ΜΕ ΘΕΤΙΚΗ ΑΣΥΜΜΕΤΡΙΑ (RIGHT SKEWED DISTRIBUTION)**

# ΔΙΚΤΥΑ ΥΠΟΛΟΓΙΣΤΩΝ 1

Όνομ/νυμο: Κυπαρίσσης Κυπαρίσσης

Α.Ε.Μ.:

[ 6 ]

## E1

Κανένα σχόλιο ή παρατήρηση.

## E2

Κανένα σχόλιο ή παρατήρηση.

## M1

Κανένα σχόλιο ή παρατήρηση.

( Γίνεται σύγκριση των δύο M1, στην αναλυση της session #2 αργότερα )

# ΔΙΚΤΥΑ ΥΠΟΛΟΓΙΣΤΩΝ 1

Όνομ/νυμο: Κυπαρίσσης Κυπαρίσσης

A.E.M.:

[ 7 ]

**BER**

Μεσω της συναρτησης `getBitErrorRate()`, η οποία χρησιμοποιει την παρακατω εξισωση:

$$P1 = (1 - Pb)^F \Rightarrow \dots \Rightarrow Pb = 1 - P1^{\frac{1}{F}}$$

,ΟΠΟΥ >  **$P_b$** : η πιθανότητα ενα bit να φτάσει εσφαλμένο (αλλιως γνωστο ως **Bit Error Rate - BER**).

>  **$P_1$** : η πιθανότητα ενα frame/sequence να φτάσει εσφαλμένο

, δηλαδή είναι: 
$$P1 = \frac{NACKResultsCounter}{NACKResultsCounter + ACKResultsCounter}$$

, ΟΠΟΥ: *NACKResultsCounter* : Το συνολικο αθροισμα των λανθασμενων πακετων που λαβαμε κατα τη διαρκεια της *ARQApplication()*

*ACKResultsCounter* : Το συνολικο αθροισμα των error-free πακετων που λαβαμε κατα τη διαρκεια της *ARQApplication()*

>  **$F$** : ο αριθμός των bits σε κάθε frame/sequence

, δηλαδη  $F = (\text{Αριθμος των bytes ανα χαρακτήρα}) * (\text{Αριθμός των bits σε καθε byte}) * (\text{Αριθμός χαρακτήρων ανα frame})$   
 $= 2 * 8 * 16 = 256 \text{ bits σε κάθε frame/sequence}$

, αφού: Στην *Java* κάθε χαρακτήρας είναι 2 bytes (ΟΧΙ 1 BYTE ΟΠΩΣ ΣΕ ΑΛΛΕΣ ΓΛΩΣΣΕΣ) (Λογω: Unicode)

(<https://docs.oracle.com/javase/1.5.0/docs/api/java/lang/Character.html#unicode>)

([https://www.w3schools.com/java/java\\_data\\_types.asp](https://www.w3schools.com/java/java_data_types.asp))

Το 1 byte ειναι ισο με 8 bits.

Το κάθε frame/sequence εχει 16 χαρακτήρες εδω.

, μεσω του *ARQApplication()*, βρήκαμε για την διαρκεια της συνοδου αυτης οτι:

<b>BER (Bit Error Rate) Value = 0.0006406459762015038</b>
---

ή

**0.06406459762015038%** η πιθανότητα ενα bit της ακολουθίας 16 χαρακτήρων να φτάσει εσφαλμένο,

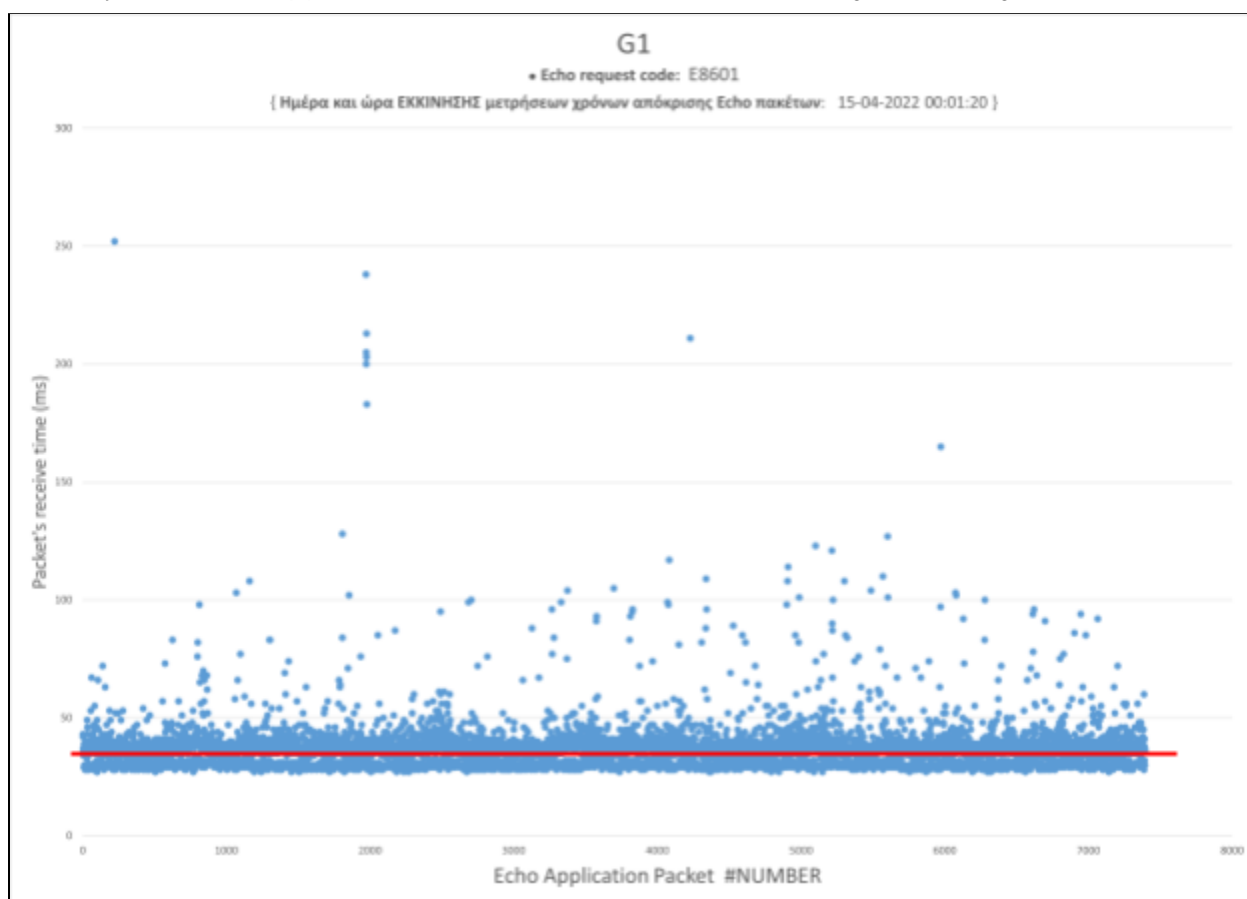
,αφου ειδαμε οτι λαβαμε 1036 εσφαλμενα πακετα (*NACKResultsCounter*) σε ενα συνολο 6847 αιτηματων που καναμε για πακέτα (άθροισμα *NACKResultsCounter* + *ACKResultsCounter*).

**ΤΕΛΙΚΑ:**

Παρατηρουμε οτι η πιθανοτητα αυτη ειναι αρκετα μικρη οπως δεν μπορεί να θεωρηθει αμελητα!!  
Για την βελτίωση της πιθανότητας αυτής , μπορούν να χρησιμοποιηθούν, επιπροσθετα, συστήματα κωδικοποίησης καναλιών καθώς και κώδικες διόρθωσης σφαλμάτων.

## G1

Ομοίως με το session #1, παρατηρώντας το ιστογράμμο του χρόνου αποκρίσης των Echo πακέτων, για την χρονική διάρκεια που τρέξαμε το *EchoApplication()*, παρατηρούμε ότι ο χρόνος αποκρίσης μένει σχετικά **ΣΤΑΘΕΡΟΣ**.



Αυτο μπορεί να φανει και απο την **κόκκινη γραμμή** (ευθεια γραμμικης παλινδρομησης) που εχουμε σχεδιασει και παρατηrouμε ότι οι χρονοι αποκρίσης κυμαινoνται κοντα σε αυτην και τα σημεια ειναι πολυ πυκνα γυρω απο την ευθεια αυτη, δηλαδη εχουμε, κοντα **στην μέση τιμή των 37 (ms), σταθερή τιμή χρόνου λήψης πακέτου**.

Παρολαυτα, παρατηrouμε και τιμες που βρισκονται σχετικα μακρια απο την γραμμη αυτη, δηλαδη χρονους  $\gg 37$  (ms).

Αυτο συμβαινει καθως μπορεί, κατα την χρονικη στιγμη ληψης του Echo πακετου εκεινου, να υπηρξε καποιο προβλημα με την σύνδεση μας στον server Ithaki και να δημιουργηθηκε ετσι η καθυστερηση αυτη στην λήψη του αντιστοιχου πακέτου. (Noise μετρήσεων).

Ομως, επειδη οι ακραιες τιμες αυτες ειναι λιγες σχετικα με τις αλλες που βρισκονται πολυ κοντα στη μέση τιμή των 37 (ms), μπορούμε να ισχυριστούμε ότι ο χρόνος αποκρίσης παραμενει σχετικά σταθερος!!!



# ΔΙΚΤΥΑ ΥΠΟΛΟΓΙΣΤΩΝ 1

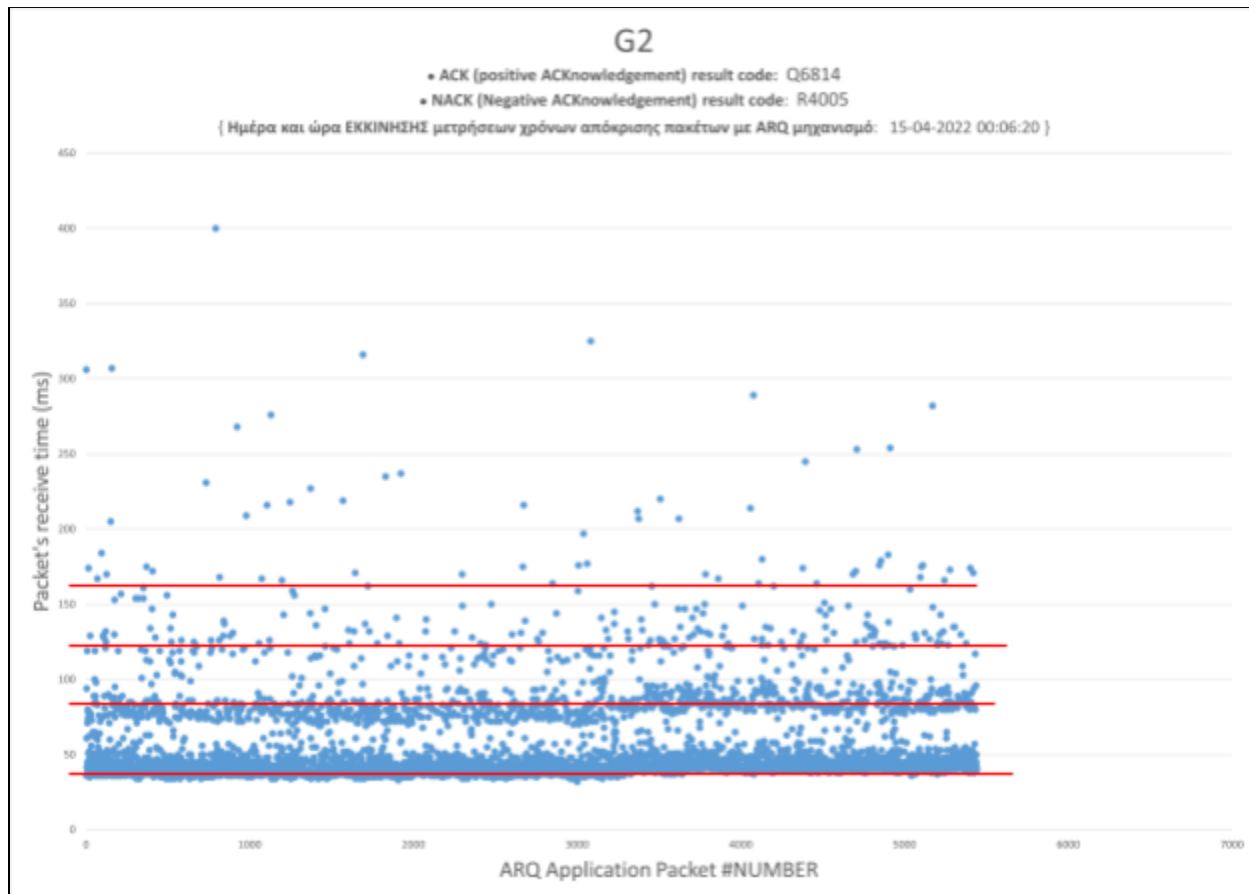
Όνομ/νυμο: Κυπαρίσσης Κυπαρίσσης

Α.Ε.Μ.:

[ 9 ]

**G2**

Ομοίως με το session #1, παρατηρώντας το ιστογράμμο του χρόνου αποκρίσης των ARQ πακέτων, για την χρονική διάρκεια που τρέξαμε το *ARQApplication()*, παρατηρούμε ότι ο χρόνος αποκρίσης **ΑΥΞΑΝΕΤΑΙ ΓΡΑΜΜΙΚΑ ΜΕ ΤΟΝ ΑΡΙΘΜΟ ΕΠΑΝΕΚΠΟΜΠΩΝ ΤΟΥ ΠΑΚΕΤΟΥ**.



Αυτο μπορεί να φανει και απο τις **κοκκινες γραμμες** (ευθειες γραμμικης παλινδρομησης) που εχουμε σχεδιασει και παρατηrouμε οτι οι χρονοι αποκρισης κυμαινονται κοντα σε αυτες και τα σημεια ειναι πολυ πυκνα γυρω απο τις ευθειες αυτες (Πυκνοτερα στους χαμηλοτερους χρονους, αραιωτερα στους μεγαλυτερους χρονους λογω της μικρης πιθανοτητας να χρειαστει να λαβουμε ενα πακετο πολλες φορες).

Παρατηrouμε λοιπον οτι εχουμε πολλαπλα επιπεδα (σχετικα) σταθερων χρονων λήψης πακέτων. Αυτη η χρονοκαθυστέρηση συμβαινει καθως ετρεχε και ο ARQ μηχανισμος που ελεγχε για τυχόν λαθη στην ληψη του πακετου και αναλογα με το ποσες φορες ετρεξε για το αντιστοιχο πακετο, το πακέτο βρισκεται κοντα στην αντιστοιχη ευθεια/επιπεδο. Δηλαδη :

- Αν ενα πακετο χρειαστηκε να σταλθει 1 φορα, βρισκεται στη 1η ευθεια, δηλαδη **~40 (ms)**
- Αν ενα πακετο χρειαστηκε να σταλθει 2 φορες καθως στην πρωτη λήφθηκε με errors, βρισκεται στη 2η ευθεια, δηλαδη **~80 (ms)**
- Αν ενα πακετο χρειαστηκε να σταλθει 3 φορες καθως στην πρωτη λήφθηκε με errors, βρισκεται στη 3η ευθεια, δηλαδη **~120 (ms)**
- Αν ενα πακετο χρειαστηκε να σταλθει 3 φορες καθως στην πρωτη λήφθηκε με errors, βρισκεται στη 3η ευθεια, δηλαδη **~160 (ms)**
- Κ.Ο.Κ...

(Δηλαδη περιπου 40 (ms) διαρκει η ληψη παλι και ο επανελεγχος του ιδιου πακετου για errors)

Τελος παρατηrouμε οτι και εδω εχουμε ακραιες τιμες χρονων που δεν βρισκονται κοντα στις ευθειες (*Noise* μετρήσεων λογω errors στην συνδεση με την *Ithaki*)

# ΔΙΚΤΥΑ ΥΠΟΛΟΓΙΣΤΩΝ 1

Όνομ/νυμο: Κυπαρίσσης Κυπαρίσσης

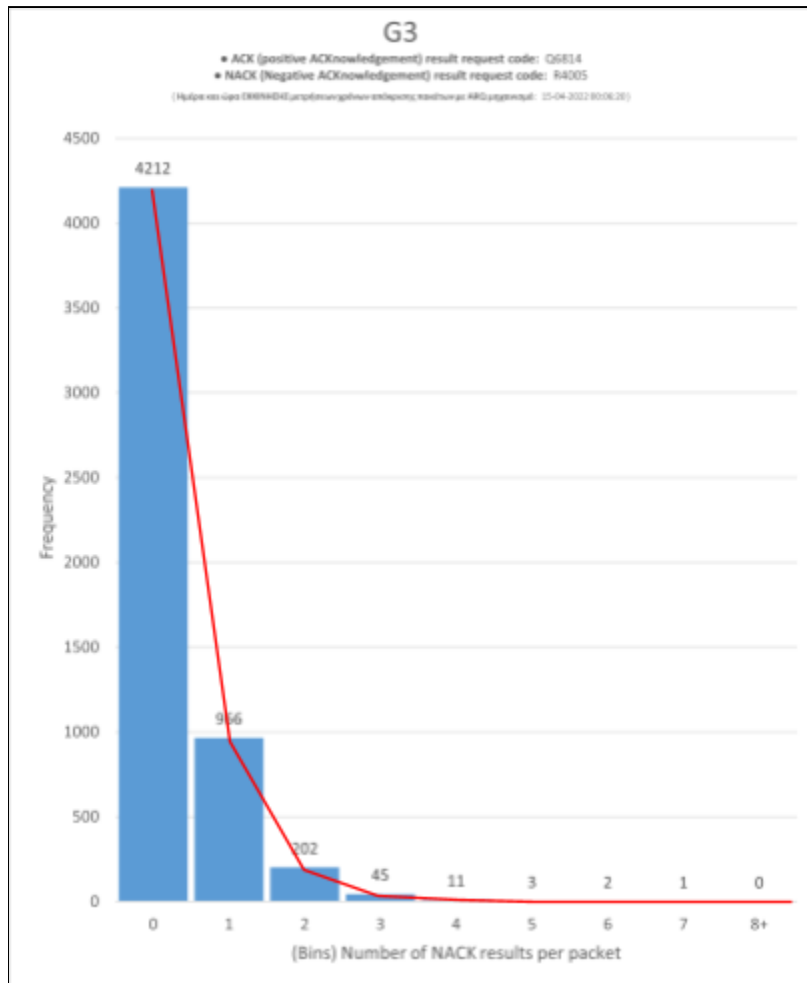
Α.Ε.Μ.:

[ 10 ]

**G3**

Ομοίως με το session #1, παρατηρώντας το ιστόγραμμα συχνοτήτων του αριθμού επανεκπομπών (Negative acknowledgement result) των πακέτων που καταγράψαμε (G3), μπορούμε να έχουμε μια εκτίμησή για την κατανομή πιθανότητας του αριθμού αυτού για κάθε πακέτο, εφόσον έχουμε λάβει και έναν αρκετά μεγάλο αριθμό δειγμάτων / error-free πακέτων (5442 error-free πακέτα / *positive acknowledgement results*) .

Παρατηρούμε ότι η **[1]** κορυφή της κατανομής βρίσκεται εκτός κεντρου πλησιάζοντας το αριστερο ακρο (εδω η κορυφη ειναι το αριστερο ακρο) και **[2]** η ουρα του τεντωνεται μακρια απο την κορυφη αυτη, προς τα δεξια. (Δείτε την **κόκκινη γραμμή** στο γράφημα G3).



Η ασυμμετρία αυτή υπάρχει λογο υπαρξης ενός φυσικου ορίου που δεν μπορεί να μας δώσει αποτελεσματα απο την μια μερια (αφου δεν μπορούμε να έχουμε αριθμο επανεκπομπής, σε ένα πακετο, μικροτερο του μηδενος (ΦΥΣΙΚΟΣ ΑΡΙΘΜΟΣ) ).

Ετσι συμπερενουμε οτι εχουμε:



**ΛΟΞΗ ΚΑΤΑΝΟΜΗ ΜΕ ΘΕΤΙΚΗ ΑΣΥΜΜΕΤΡΙΑ (RIGHT SKEWED DISTRIBUTION)**

# ΔΙΚΤΥΑ ΥΠΟΛΟΓΙΣΤΩΝ 1

Όνομ/νυμο: Κυπαρίσσης Κυπαρίσσης

A.E.M.:

[ 11 ]

## E1

Κανένα σχόλιο ή παρατήρηση.

## E2

Κανένα σχόλιο ή παρατήρηση.

## M1

Παρατηρούμε ότι τόσο στο SESSION #1 όσο και στο SESSION #2, στην εικόνα εμφανίζονται τα ίδια ίχνη GPS, αφού τα ίχνη της διαδρομής  $X=1$  είναι προαποθηκευμένα (εχουμε και ίδιο αρχικό ίχνος PPPP και λαβαμε ίδιο αριθμο ιχνων LL) και ο αλγοριθμος επιλογης των σημειων που πρεπει να εχουν αποσταση μεταξυ τους τουλαχιστον 4", **ΔΕΝ** αλλαξε στα δυο sessions!!

# ΔΙΚΤΥΑ ΥΠΟΛΟΓΙΣΤΩΝ 1

Όνομ/νυμο: Κυπαρίσσης Κυπαρίσσης

A.E.M.:

[ 12 ]

**BER**

Ομοίως με το session #1:

Μεσω της συναρτησης `getBitErrorRate()`, η οποία χρησιμοποιει την παρακατω εξίσωση:

$$P1 = (1 - Pb)^F \Rightarrow \dots \Rightarrow Pb = 1 - P1^{\frac{1}{F}}$$

,ΟΠΟΥ >  $P_b$ : η πιθανότητα ενα bit να φτάσει εσφαλμένο (αλλιως γνωστο ως *Bit Error Rate - BER*).

>  $P_1$ : η πιθανότητα ενα frame/sequence να φτάσει εσφαλμένο

, δηλαδή είναι:  $P1 = \frac{NACKResultsCounter}{NACKResultsCounter + ACKResultsCounter}$

, ΟΠΟΥ: *NACKResultsCounter* : Το συνολικο αθροισμα των λανθασμενων πακετων που λαβαμε κατα τη διαρκεια της *ARQApplication()*

*ACKResultsCounter* : Το συνολικο αθροισμα των error-free πακετων που λαβαμε κατα τη διαρκεια της *ARQApplication()*

>  $F$ : ο αριθμός των bits σε κάθε frame/sequence

, δηλαδή  $F = (\text{Αριθμος των bytes ανα χαρακτήρα}) * (\text{Αριθμός των bits σε καθε byte}) * (\text{Αριθμός χαρακτήρων ανα frame})$   
 $= 2 * 8 * 16 = 256 \text{ bits σε κάθε frame/sequence}$

, αφού: Στην *Java* κάθε χαρακτήρας είναι 2 bytes (ΟΧΙ 1 BYTE ΟΠΩΣ ΣΕ ΑΛΛΕΣ ΓΛΩΣΣΕΣ) (Λογω: Unicode)

(<https://docs.oracle.com/javase/1.5.0/docs/api/java/lang/Character.html#unicode>)

([https://www.w3schools.com/java/java\\_data\\_types.asp](https://www.w3schools.com/java/java_data_types.asp))

To 1 byte ειναι ισο με 8 bits.

To κάθε frame/sequence εχει 16 χαρακτήρες εδω.

, μεσω του *ARQApplication()*, βρήκαμε για την διαρκεια της συνοδου αυτης οτι:

**BER (Bit Error Rate) Value = 0.0009968800301745073**

ή

**0.09968800301745073%** η πιθανότητα ενα bit της ακολουθίας 16 χαρακτήρων να φτάσει εσφαλμένο,

,αφου ειδαμε οτι λαβαμε 1583 εσφαλμενα πακετα (*NACKResultsCounter*) σε ενα συνολο 7025 αιτηματων που καναμε για πακέτα (αθροισμα *NACKResultsCounter* + *ACKResultsCounter*).

ΤΕΛΙΚΑ:

Παρατηρουμε οτι η πιθανοτητα αυτη ειναι αρκετα μικρη οπως δεν μπορει να θεωρηθει αμελητα!!  
Για την βελτίωση της πιθανότητας αυτής, μπορούν να χρησιμοποιηθούν, επιπροσθετα, συστήματα κωδικοποίησης καναλιών καθώς και κώδικες διόρθωσης σφαλμάτων.

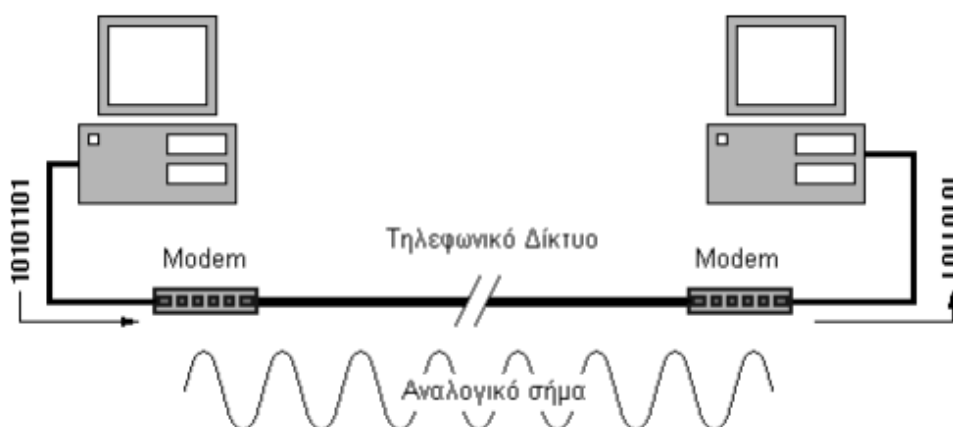
## ΣΥΜΠΕΡΑΣΜΑΤΙΚΑ

Παρατηρουμε οτι στο session #1 και session #2 βγαλαμε ΟΜΟΙΑ ΑΠΟΤΕΛΕΣΜΑΤΑ και ΔΕΝ βρεθηκαν σημαντικες διαφορες κατα τις αναλύσεις των αποτελεσμάτων και των μετρησεων αυτων!!

## ΒΙΒΛΙΟΓΡΑΦΙΚΗ ΤΕΧΝΙΚΗ ΑΝΑΦΟΡΑ

### Τι είναι το Modem;

Το Modem (Modulator-Demodulator) είναι μια συσκευή που επιτρέπει σε υπολογιστές να επικοινωνούν μεταξύ τους μέσω τηλεφωνικών γραμμών, δίνοντας έτσι την ευκαιρία στους χρήστες να έχουν άμεση και εύκολη πρόσβαση σε πολλές υπηρεσίες. Το modem παίρνει την πληροφορία από τον υπολογιστή και την μετατρέπει σε ένα σήμα που μπορεί να μεταφερθεί μέσω τηλεφωνικών γραμμών. Η πληροφορία στο εσωτερικό του υπολογιστή είναι αποθηκευμένη σε ψηφιακή μορφή ενώ κατά μήκος των τηλεφωνικών γραμμών μεταδίδεται με τη μορφή αναλογικών σημάτων. Το modem μετατρέπει μεταξύ των δύο αυτών μορφών:



Για την σωστή λειτουργία του modem απαιτείται η ύπαρξη συγκεκριμένου λογισμικού το οποίο διευθύνει την επικοινωνία με τα άλλα modem.

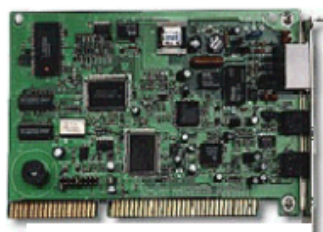
Τα modem συχνά ταξινομούνται με βάση τη μέγιστη ποσότητα δεδομένων που μπορούν να στείλουν σε μια δεδομένη μονάδα χρόνου, συνήθως εκφρασμένη σε bit ανά δευτερόλεπτο (σύμβολο bit/s, μερικές φορές με συντομογραφία "bps") ή σπάνια σε byte ανά δευτερόλεπτο (σύμβολο B/s).

Τα σύγχρονα ευρυζωνικά modem περιγράφονται συνήθως σε megabit με ταχύτητες που κυμαίνονται στα 28.800 και στα 33.600 bps. Υπάρχουν βέβαια και modem στα 57.600 bps.

Στη πραγματικότητα οι ταχύτητες που επιτυγχάνονται είναι υψηλότερες γιατί υπάρχει συμπίεση δεδομένων!

Υπάρχουν τρία είδη modem:

- Το εσωτερικό ενσωματώνεται στο εσωτερικό του υπολογιστή σε ειδική υποδοχή και είναι το πιο φθηνό.



- Το εξωτερικό συνδέεται με καλώδιο στο port που υπάρχει για modem και μερικά από αυτά παρέχουν και δυνατότητα χρήσης Fax.

# ΔΙΚΤΥΑ ΥΠΟΛΟΓΙΣΤΩΝ 1

Όνομ/νυμο: Κυπαρίσσης Κυπαρίσσης

A.E.M.:

[ 14 ]



- Τέλος υπάρχει και το ISDN modem το οποίο χρησιμοποιεί μια ειδική τηλεφωνική γραμμή, η οποία καλείται γραμμή ISDN, πετυχαίνοντας πολύ υψηλούς ρυθμούς μετάδοσης.



## Η λογική λειτουργίας του Modem

Για να κατανοήσουμε όμως τη δομή και τον τρόπο λειτουργίας των Modem, η καλύτερη μέθοδος είναι να σκιαγραφήσουμε τη λογική ενός παλιού και λιγότερο σύνθετου μηχανήματος. Ένα Modem 300 bps, λειτουργεί χρησιμοποιώντας τη λεγόμενη κωδικοποίηση μετάθεσης συχνότητας (Frequency Shift Keying) προκειμένου να μεταδώσει τα ψηφιακά δεδομένα μέσω των αναλογικών τηλεφωνικών γραμμών.

## ΜΕΘΟΔΟΙ ΕΠΙΚΟΙΝΩΝΙΑΣ

Στην κωδικοποίηση μετάθεσης συχνότητας, διαφορετικός τόνος αποδίδει τα διαφορετικά bits. Όταν ένα Modem τερματικού καλεί το Modem ενός υπολογιστή, μεταδίδει έναν τόνο 1,070 hertz για κάθε 0 και έναν 1,270 hertz για κάθε 1.

Το Modem του κεντρικού υπολογιστή αντίστοιχα χρησιμοποιεί έναν τόνο 2,025 hertz για το 0 και έναν τόνο 2,225 hertz για το 1.

Επειδή λοιπόν τα δύο Modem, μεταδίδουν διαφορετικούς τόνους, μπορούν και χρησιμοποιούν τη γραμμή ταυτόχρονα. Ο τρόπος αυτός επικοινωνίας, ονομάζεται full - duplex, δηλαδή πλήρως αμφίδρομος.

Σπανίως, συναντάμε Modem με δυνατότητα μόνο να λαμβάνουν ή να μεταδίδουν ανά φορά τα οποία αποκαλούνται half - duplex δηλαδή ημιαμφίδρομα.

Ας υποθέσουμε τώρα ότι δύο Modem 300 bps συνδέονται μεταξύ τους και ότι ο χρήστης του τερματικού πληκτρολογεί το χαρακτήρα "a".

- Ο χαρακτήρας αυτός, στο δυαδικό σύστημα αναγνωρίζεται ως 01100001 σύμφωνα με τον κώδικα ASCII.
- Το τερματικό μεταβιβάζει τα bits του χαρακτήρα που πληκτρολογήθηκε στο Modem μέσω της σειριακής θύρας.
- Το Modem δέχεται τα ψηφιακά δεδομένα και αναλαμβάνει την αποστολή τους στον κεντρικό υπολογιστή χρησιμοποιώντας την απλή γραμμή του τηλεφωνικού δικτύου της περιοχής και μεταδίδοντας την κατάλληλη σειρά τόνων.
- Το Modem του υπολογιστή δέχεται τα ηχητικά διαμορφωμένα δεδομένα και τα μεταφράζει σε ψηφιακά παρέχοντας στο σύστημα τα αποτελέσματα.
- Ο κεντρικός υπολογιστής επεξεργάζεται τις ψηφιακές αυτές πληροφορίες και επιστρέφει τα αποτελέσματα μέσω του δικού του Modem με την αντίστροφη διαδικασία.

# ΔΙΚΤΥΑ ΥΠΟΛΟΓΙΣΤΩΝ 1

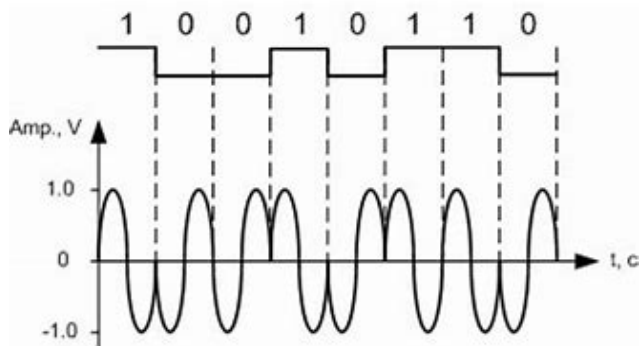
Όνομ/νυμο: Κυπαρίσσης Κυπαρίσσης

A.E.M.:

[ 15 ]

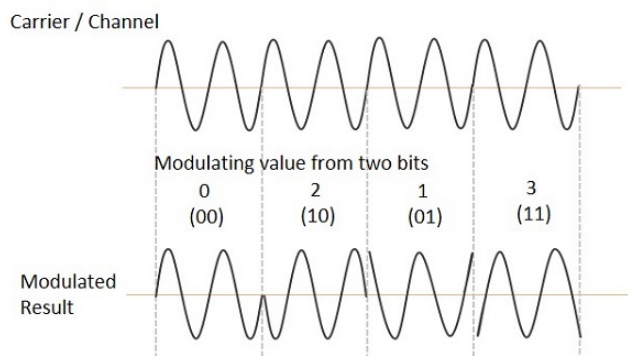
Προκειμένου πάντως τα Modem να αποκτήσουν τη δυνατότητα αποστολής και λήψης σε πολύ μεγαλύτερες ταχύτητες από αυτή των 300 bps, απαιτήθηκαν τεχνικές σαφώς πιο εξελιγμένες και πιο σύγχρονες από την κωδικοποίηση μετάθεσης συχνότητας.

Βήματα προς την κατεύθυνση αυτή, αποτέλεσαν η κωδικοποίηση μετάθεσης φάσης (Phase Shift Keying) αρχικά



Phase shift keying (PSK)

και η διαμόρφωση ορθογωνικής τάσης (Quadrature Amplitude Modulation) στη συνέχεια, του οποίου το πλεονέκτημα είναι να εμπεριέχεται μεγάλη ποσότητα πληροφορίας σε στενό εύρος ζώνης .



Αυτές οι τεχνικές επιτρέπουν τη μετάδοση μιας μεγάλης ποσότητας πληροφοριών δεδομένου του εύρους των 3000 hertz που διατίθεται σε μια απλή γραμμή τηλεφώνου. Τα σύγχρονα Modem των 56Kbps θεωρούνται και είναι τουλάχιστον μέχρι αποδείξεως του αντιθέτου η απόλυτη κορυφή στις δυνατότητες των σημερινών τεχνικών. Επιπλέον τα νέα Modem, υποστηρίζουν μία πληθώρα από παλαιά πρωτόκολλα και έτσι έχουν τη δυνατότητα να συνδέονται δίχως πρόβλημα με παλαιότερα μοντέλα και βέβαια να ρίχνουν τη σύνδεση σε πιο ανεκτικά και σταθερά πρωτόκολλα όταν οι συνθήκες της γραμμής το απαιτούν.

Οποιαδήποτε τεχνική από τις διάφορες μεθόδους διαμόρφωσης ή ακόμη και οποιοσδήποτε συνδυασμός των μεθόδων (μέθοδος ολοκληρωμένης διαμόρφωσης) μπορεί να χρησιμοποιηθεί για μεταφορά δεδομένων. Για παράδειγμα δείτε τον παρακάτω πίνακα:

relative amplitude	phase shift	bit meaning	symbol value
1	45	0 0 0	"0"
1	135	0 0 1	"1"
1	225	0 1 0	"2"
1	315	0 1 1	"3"
2	45	1 0 0	"4"
2	135	1 0 1	"5"
2	225	1 1 0	"6"
2	315	1 1 1	"7"

## ΣΕΙΡΙΑΚΉ ΕΠΙΚΟΙΝΩΝΊΑ ΚΑΙ ΈΛΕΓΧΟΣ ΡΟΉΣ

---

Η σειριακή επικοινωνία είναι μια μέθοδος επικοινωνίας που χρησιμοποιεί μία ή δύο γραμμές μετάδοσης για την αποστολή και λήψη δεδομένων και τα δεδομένα στέλνονται και λαμβάνονται συνεχώς ένα bit τη φορά. Στους υπολογιστές, μια σειριακή θύρα είναι μια σειριακή διεπαφή επικοινωνίας μέσω της οποίας οι πληροφορίες μεταφέρονται μέσα ή έξω διαδοχικά ένα bit τη φορά.

Η σειριακή θύρα UART (ή απλώς "σειριακή θύρα για συντομία" είναι μια συσκευή I/O (Είσοδος/Έξοδος). Οι περισσότεροι υπολογιστές έχουν μία ή δύο σειριακές θύρες. Κάθε ένα έχει μια υποδοχή 9 ακίδων (μερικές φορές 25 ακίδων) στο πίσω μέρος του υπολογιστή. Τα προγράμματα υπολογιστών μπορούν να στέλνουν δεδομένα (bytes) στον ακροδέκτη εκπομπής (έξοδος) και να λαμβάνουν byte από τον ακροδέκτη λήψης (είσοδος). Οι άλλες ακίδες είναι για λόγους ελέγχου και γείωσης.

Η σειριακή θύρα είναι πολλά περισσότερα από μια απλή υποδοχή. Μετατρέπει τα δεδομένα από παράλληλα σε σειριακά και αλλάζει την ηλεκτρική αναπαράσταση των δεδομένων. Μέσα στον υπολογιστή, τα bits δεδομένων ρέουν παράλληλα (χρησιμοποιώντας πολλά καλώδια ταυτόχρονα). Η σειριακή ροή είναι μια ροή bit πάνω από ένα μόνο καλώδιο (όπως στον ακροδέκτη εκπομπής ή λήψης του σειριακού βύσματος). Για να δημιουργήσει μια τέτοια ροή η σειριακή θύρα, πρέπει να μετατρέψει δεδομένα από παράλληλη (μέσα στον υπολογιστή) σε σειριακή στον ακροδέκτη εκπομπής (και αντίστροφα).

## ΛΕΙΤΟΥΡΓΊΕΣ ΕΝΤΟΛΩΝ ΚΑΙ ΔΕΔΟΜΈΝΩΝ

---

Οι λειτουργίες Command και Data αναφέρονται στις δύο λειτουργίες στις οποίες μπορεί να λειτουργεί ένα μόντεμ υπολογιστή. Αυτές οι λειτουργίες ορίζονται στο σύνολο εντολών Hayes, το οποίο είναι το de facto πρότυπο για όλα τα μόντεμ. Αυτές οι λειτουργίες υπάρχουν επειδή υπάρχει μόνο ένα κανάλι επικοινωνίας μεταξύ του μόντεμ και του υπολογιστή, το οποίο πρέπει να μεταφέρει τόσο τις εντολές του υπολογιστή στο μόντεμ, όσο και τα δεδομένα που το μόντεμ έχει δεσμευτεί να μεταδώσει στο απομακρυσμένο άτομο μέσω της τηλεφωνικής γραμμής.

- Όταν ένα μόντεμ βρίσκεται σε λειτουργία εντολών, τυχόν χαρακτήρες που αποστέλλονται σε αυτό ερμηνεύονται ως εντολές για εκτέλεση του μόντεμ, σύμφωνα με το σύνολο εντολών Hayes. Πριν από μια εντολή υπάρχουν τα γράμματα «AT», που σημαίνει «Προσοχή/Attention». Για παράδειγμα, εάν ένα μόντεμ λάβει «ATDT5551212» ενώ βρίσκεται στη λειτουργία εντολών, το ερμηνεύει ως οδηγία για να καλέσετε τους αριθμούς 5551212 στο τηλέφωνο, χρησιμοποιώντας τονική κλήση. Ενώ βρίσκεται σε λειτουργία εντολών, το μόντεμ μπορεί να στείλει απαντήσεις πίσω στον υπολογιστή που υποδεικνύουν το αποτέλεσμα της εντολής. Για παράδειγμα, το μόντεμ μπορεί να απαντήσει με τη λέξη "BUSY" ως απόκριση στην εντολή ATDT, εάν ακούσει ένα σήμα κατειλημμένου μετά την κλήση και έχει ρυθμιστεί να ακούει για σήματα κατειλημμένου.

Οποιαδήποτε επικοινωνία σε λειτουργία εντολών (και προς τις δύο κατευθύνσεις) τερματίζεται με επιστροφή μεταφοράς.

- Όταν ένα μόντεμ βρίσκεται σε λειτουργία δεδομένων, τυχόν χαρακτήρες που αποστέλλονται στο μόντεμ προορίζονται να μεταδοθούν στο απομακρυσμένο μέρος. Το μόντεμ εισέρχεται σε λειτουργία δεδομένων αμέσως μετά τη σύνδεση. Για παράδειγμα, εάν το ATDT5551212 είχε ως αποτέλεσμα μια τηλεφωνική κλήση που απαντήθηκε από άλλο μόντεμ υπολογιστή, το μόντεμ θα ανέφερε τη λέξη "CONNECT" και στη συνέχεια θα μεταβεί σε λειτουργία δεδομένων. Τυχόν



περαιτέρω χαρακτήρες που λαμβάνονται μέσω του σειριακού συνδέσμου θεωρείται ότι προέρχονται από το απομακρυσμένο μέρος και τυχόν χαρακτήρες που αποστέλλονται μεταδίδονται στο απομακρυσμένο μέρος.

## ΑΝΊΧΝΕΥΣΗ ΚΑΙ ΔΙΉΡΘΩΣΗ ΣΦΆΛΜΑΤΟΣ

---

Τα σύγχρονα Modem επίσης έχουν να αντιμετωπίσουν και ένα άλλο πολύ βασικό ζήτημα.

Αυτό είναι η ΑΝΊΧΝΕΥΣΗ ΚΑΙ ΔΙΉΡΘΩΣΗ ΣΦΆΛΜΑΤΩΝ.

Κατά τη μεταφορά των δεδομένων μέσω της γραμμής του τηλεφώνου, διάφορα προβλήματα μπορεί να προκύψουν τα οποία επιφέρουν την κακή λήψη σήματος από το Modem. Το σήμα αυτό πρέπει φυσικά να επαναληφθεί εφόσον δεν είναι σωστό.

### PARITY CHECK

Η αναγνώριση του λάθους γίνεται μέσω μιας μεθόδου που ονομάζεται parity check, δηλαδή επαλήθευση ισότητας. Το parity check με λίγα λόγια λειτουργεί επικολλώντας ένα bit στο τέλος κάθε μεταβίβασης. Ανάλογα με τη λειτουργία του parity check είτε ως odd είτε ως even, το bit στην κατάληξη κάθε σήματος είναι τέτοιο ώστε να διαμορφώνει ένα άθροισμα από bits "1", μονό ή ζυγό αντίστοιχα. Το Modem που λαμβάνει το σήμα εξετάζει τον αριθμό των bits "1", μετά το πέρας κάθε μεταβίβασης και εάν διαπιστώσει ότι δε συμφωνεί με το προσυμφωνημένο parity, ζητά την επανάληψη της αποστολής του. Με τον τρόπο αυτό, το μηχάνημα μειώνει σημαντικά τις πιθανότητες κάποιου λάθους στην αποστολή των δεδομένων και προστατεύει το χρήστη από πιθανά προβλήματα στη δουλειά του.

- ❖ Πλεονεκτήματα του PARITY CHECK:
  - Τα σφάλματα σε μια θορυβώδη γραμμή μπορούν να εντοπιστούν γρήγορα και μόνο η λανθασμένη λέξη πρέπει να επαναμεταδοθεί.
- ❖ Μειονεκτήματα του PARITY CHECK:
  - Δεν είναι δυνατός ο εντοπισμός όλων των σφαλμάτων, επομένως πρέπει να χρησιμοποιήσετε έναν μηχανισμό ανίχνευσης σφαλμάτων υψηλότερου επιπέδου, όπως το CRC.

---

### ARQ (Automatic Repeat reQuest)

Μια ακόμα μέθοδος ελέγχου σφαλμάτων για τη μετάδοση δεδομένων είναι αυτή του αυτόματου αιτήματος επανάληψης ( ARQ ), γνωστό και ως αυτόματη επανάληψη ερωτήματος. Αυτή χρησιμοποιεί επιβεβαιώσεις (μηνύματα που αποστέλλονται από τον παραλήπτη που υποδεικνύουν ότι έχει λάβει σωστά ένα πακέτο ) και χρονικά όρια (καθορισμένες χρονικές περιόδους που επιτρέπεται να παρέλθει πριν από τη λήψη μιας επιβεβαίωσης) για να επιτευχθεί αξιόπιστη μετάδοση δεδομένων μέσω ενός αναξιόπιστου καναλιού επικοινωνίας .

Αυτο γίνεται 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, από 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 και έτσι ζητείται επαναποστολή του μηνύματος ή να εφαρμόζεται κάποια τεχνική διόρθωσης του.

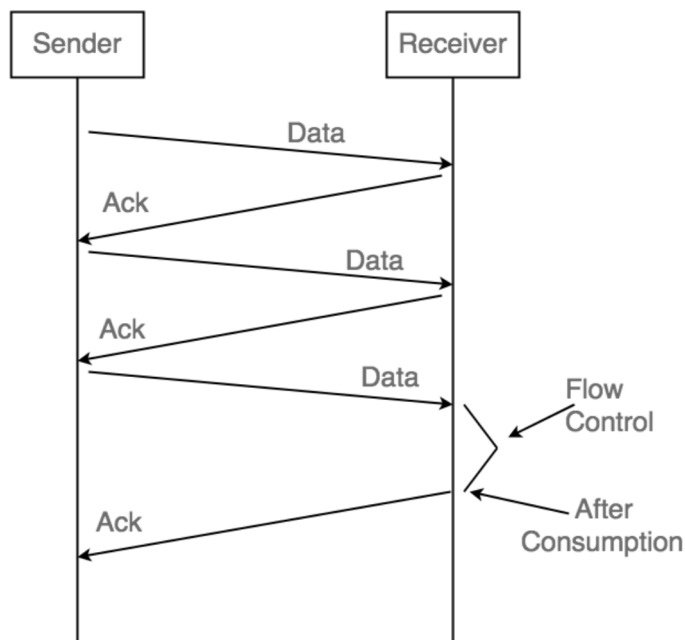
Εάν ο αποστολέας δεν λάβει μια επιβεβαίωση πριν από το χρονικό όριο, εκπέμπει εκ νέου το πακέτο μέχρι να λάβει μια επιβεβαίωση ή να υπερβεί έναν προκαθορισμένο αριθμό αναμεταδόσεων.

Οι παραλλαγές των πρωτοκόλλων ARQ περιλαμβάνουν το

- **Stop-and-wait ARQ**

(ΣΤΗΝ ΕΡΓΑΣΙΑ ΑΥΤΗ ΕΓΙΝΕ Η ΧΡΗΣΗ ΑΥΤΟΥ ΤΟΥ ΠΡΩΤΟΚΟΛΛΟΥ)

Το Stop and Wait ARQ, που ονομάζεται επίσης και εναλλασσόμενο πρωτόκολλο, είναι μια μέθοδος που χρησιμοποιείται σε συστήματα αμφίδρομης επικοινωνίας για την αποστολή πληροφοριών μεταξύ δύο συνδεδεμένων συσκευών (αποστολέα και δέκτη). Αναφέρεται ως stop and wait ARQ επειδή η λειτουργία αυτού του πρωτοκόλλου είναι να στέλνει ένα πακέτο τη φορά. Μετά την αποστολή ενός πλαισίου ή ενός πακέτου, ο αποστολέας δεν στέλνει άλλα πακέτα μέχρι να λάβει μια επιβεβαίωση από τον παραλήπτη. Επιπλέον, ο αποστολέας διατηρεί ένα αντίγραφο του σταλθέντος πακέτου. Αφού λάβει το επιθυμητό πλαίσιο, ο δέκτης στέλνει μια επιβεβαίωση. Εάν η επιβεβαίωση δεν φτάσει στον αποστολέα πριν από την καθορισμένη ώρα, γνωστή ως timeout, ο αποστολέας στέλνει ξανά το ίδιο πακέτο. Το χρονικό όριο μηδενίζεται μετά από κάθε μετάδοση καρέ. Το παραπάνω σενάριο απεικονίζει μια κατάσταση διακοπής και αναμονής, επομένως αυτός ο μηχανισμός ελέγχου ονομάζεται Stop and wait ARQ.



- **Go-Back-N ARQ**

Το Go-Back-N ARQ είναι ένας τύπος του πρωτοκόλλου ARQ, στο οποίο η διαδικασία αποστολής συνεχίζει την αποστολή πολλών πλαισίων ή πακέτων ακόμη και χωρίς λήψη πακέτου επιβεβαίωσης από τον δέκτη. Η διαδικασία του παραλήπτη παρακολουθεί τον αριθμό σειράς του επόμενου πακέτου που αναμένει να λάβει και στέλνει αυτόν τον αριθμό σειράς με κάθε επιβεβαίωση στον αποστολέα. Ο δέκτης θα αφαιρέσει οποιοδήποτε πακέτο δεν έχει τον επιθυμητό αριθμό σειράς που αναμένει και θα στείλει ξανά μια επιβεβαίωση για το τελευταίο σωστό πλαίσιο. Υπάρχουν μόνο δύο πιθανότητες ένα πλαίσιο να μην ταιριάζει με τον αύξοντα αριθμό: είναι είτε διπλό πλαίσιο ενός υπάρχοντος πλαισίου είτε ένα πλαίσιο εκτός σειράς που πρέπει να σταλεί αργότερα, ο παραλήπτης αναγνωρίζει αυτό το σενάριο και στέλνει μια επιβεβαίωση σήμα ανάλογα. Μόλις ο αποστολέας στείλει όλα τα καρέ στο παράθυρό του, θα αναγνωρίσει ότι όλα τα καρέ από το πρώτο χαμένο πλαίσιο και θα επιστρέψει στον αριθμό σειράς του τελευταίου σήματος επιβεβαίωσης που έλαβε από τον δέκτη *pr* και συνεχίσει τη διαδικασία ξανά. Το μόνο μειονέκτημα αυτού του τύπου συστήματος είναι ότι έχει ως

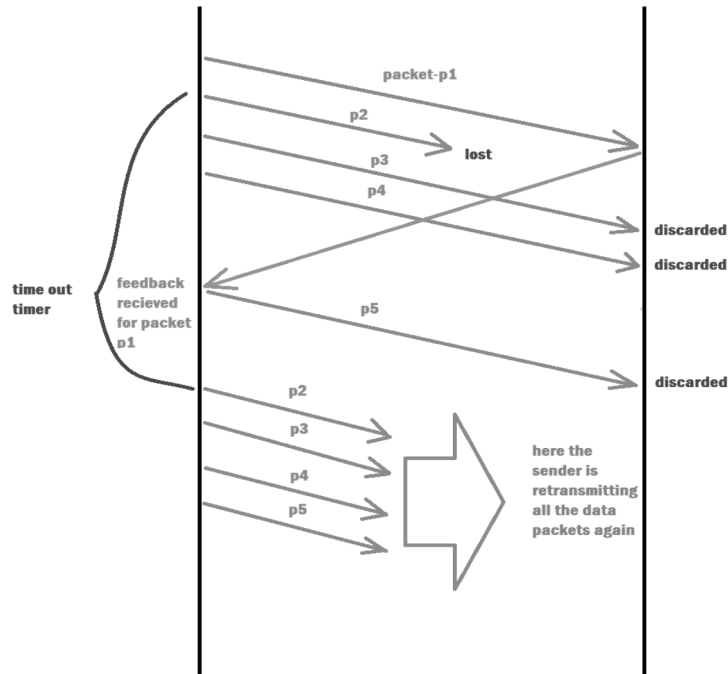
# ΔΙΚΤΥΑ ΥΠΟΛΟΓΙΣΤΩΝ 1

Όνομ/νυμο: Κυπαρίσσης Κυπαρίσσης

A.E.M.:

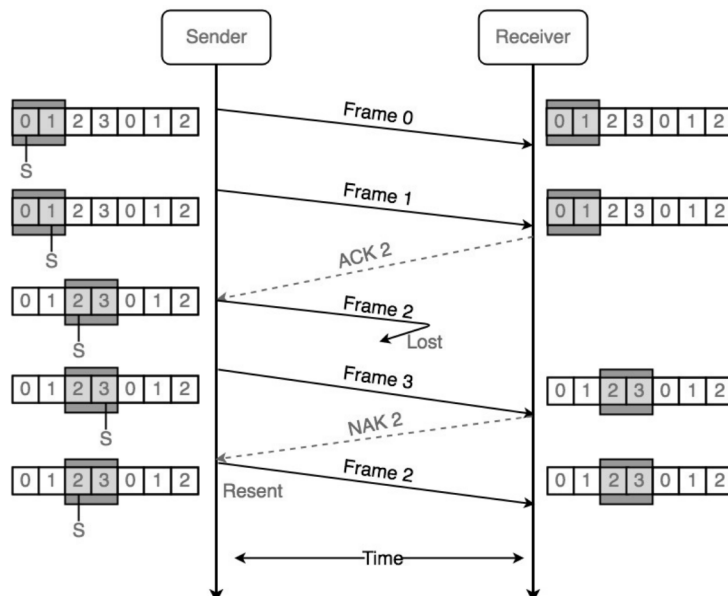
[ 19 ]

αποτέλεσμα την αποστολή πακέτων πολλές φορές: εάν κάποιο πλαίσιο χαθεί ή βρέθηκε ότι είναι κατεστραμμένο, τότε αυτό το πλαίσιο και όλα τα επόμενα πλαίσια στο παράθυρο αποστολής θα επαναμεταδοθούν. Αυτό το πρωτόκολλο είναι πιο αποτελεσματικό από το Stop and wait ARQ καθώς δεν υπάρχει χρόνος αναμονής.



- Selective Repeat ARQ

Ο μηχανισμός πρωτοκόλλου επιλεκτικής επανάληψης ARQ/Επιλεκτικής απόρριψης είναι παρόμοιος με τον μηχανισμό πρωτοκόλλου Go-Back-N. Όμως στο Selective Repeat ARQ η διαδικασία αποστολής συνεχίζεται ακόμα και όταν διαπιστωθεί ότι ένα πλαίσιο είναι κατεστραμμένο ή χαμένο. Αυτό γίνεται έχοντας τον δέκτη να παρακολουθεί τον αριθμό σειράς του παλαιότερου καρέ που δεν έχει λάβει και να στέλνει τον αντίστοιχο αριθμό σειράς με το σήμα επιβεβαίωσης. Εάν ένα πλαίσιο δεν ληφθεί στο τέλος του δέκτη, ο αποστολέας συνεχίζει να στέλνει τα επόμενα καρέ μέχρι να αδειάσει το παράθυρό του. Μόλις ολοκληρωθεί αυτή η διαδικασία διόρθωσης σφαλμάτων, η διαδικασία συνεχίζεται από εκεί που σταμάτησε. Σε αντίθεση με το πρωτόκολλο Go back-N, αυτό δεν στέλνει ένα πακέτο πολλές φορές.



# ΔΙΚΤΥΑ ΥΠΟΛΟΓΙΣΤΩΝ 1

Όνομ/νυμο: Κυπαρίσσης Κυπαρίσσης

A.E.M.:

[ 20 ]

Γενικά, όμως, και τα τρία πρωτόκολλα συνήθως χρησιμοποιούν κάποια μορφή πρωτοκόλλου συρόμενου παραθύρου για να βοηθήσουν τον αποστολέα να προσδιορίσει ποια (εάν υπάρχουν) πακέτα πρέπει να επαναμεταδοθούν. Αυτά τα πρωτόκολλα βρίσκονται στη ζεύξη δεδομένων ή στα επίπεδα μεταφοράς (επίπεδα 2 , 4) του μοντέλου OSI .

- ❖ Πλεονεκτήματα του ARQ:
  - Οι μηχανισμοί ανίχνευσης σφαλμάτων και διόρθωσης είναι αρκετά απλοί σε σύγκριση με τις άλλες τεχνικές.
  - Ένας πολύ απλούστερος εξοπλισμός αποκωδικοποίησης μπορεί να χρησιμοποιηθεί σε σύγκριση με τις άλλες τεχνικές.
- ❖ Μειονεκτήματα του ARQ:
  - Ένα μέσο ή ένα κανάλι με υψηλό ποσοστό σφάλματος μπορεί να προκαλέσει υπερβολική μετάδοση των πλαισίων ή των πακέτων πληροφοριών.
  - Το υψηλό ποσοστό σφάλματος στο κανάλι μπορεί επίσης να οδηγήσει σε απώλεια πληροφοριών, μειώνοντας επομένως την απόδοση ή την παραγωγικότητα του συστήματος.

## Τελευταία σχόλια και συμπεράσματα

---

Τα Modem λοιπόν λειτουργούν ως διαμεσολαβητές των υπολογιστών στη μεταξύ τους επικοινωνία. Είναι τα μέσα που αξιοποιούν το ήδη εξαπλωμένο, φθηνό τηλεφωνικό δίκτυο πραγματοποιώντας ανταλλαγές δεδομένων κάθε φύσεως σε συστήματα σε όλο τον κόσμο.

Κυρίως όμως χαρίζουν νέες δυνατότητες και προοπτικές επικοινωνίας στους κατόχους των υπολογιστών αυτών. Άλλωστε ακόμα και ο αχανής κόσμος του Internet στηρίζει τη λειτουργία και κυρίως την αλματώδη εξάπλωσή του στις μικρές αυτές συσκευές.

# ΔΙΚΤΥΑ ΥΠΟΛΟΓΙΣΤΩΝ 1

Όνομ/νυμo: Κυπαρίσσης Κυπαρίσσης

A.E.M.:

[ 21 ]

## ΑΝΑΦΟΡΕΣ ΑΠΟ ΣΥΝΔΕΣΜΟΥΣ ΚΑΙ ΒΙΒΛΙΟΓΡΑΦΙΑ

---

- <http://web.teiep.gr/noc/Helpdesk/P10/services/PDF/modem.pdf>
- [http://www.idc-online.com/technical\\_references/pdfs/electronic\\_engineering/Command\\_and\\_Data\\_Modes.pdf](http://www.idc-online.com/technical_references/pdfs/electronic_engineering/Command_and_Data_Modes.pdf)
- <https://www.computerhope.com/jargon/s/seriport.htm>
- <https://www.contec.com/support/basic-knowledge/daq-control/serial-communicatin/#:~:text=and%20synchronous%20communication-,What%20is%20serial%20communication%3F,one%20bit%20at%20a%20time.>
- <https://www.thoughtco.com/history-of-the-modem-4077013>
- <https://www.geeksforgeeks.org/what-is-arq-automatic-repeat-request/>
- [http://www.trine2.net.au/Modem-Tutorial\(YES!\)/modem.htm#Full\\_Duplex](http://www.trine2.net.au/Modem-Tutorial(YES!)/modem.htm#Full_Duplex)
- Stallings William, Κατσαβούνης Στέφανος (επιμέλεια) (2018) “Επικοινωνίες υπολογιστών και δεδομένων, 10η Έκδοση”
- <https://en.wikipedia.org/wiki/Histogram>
- <https://en.wikipedia.org/wiki/Skewness#Introduction>