**ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ**

**ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ**

**ΕΑΡΙΝΟ ΕΞΑΜΗΝΟ 2021-2022**

**ΜΑΘΗΜΑ «ΔΙΚΤΥΑ ΥΠΟΛΟΓΙΣΤΩΝ»**

**ΔΙΔΑΣΚΩΝ: ΓΕΩΡΓΙΟΣ Δ. ΣΤΑΜΟΥΛΗΣ, ΚΑΘΗΓΗΤΗΣ**

**ΒΟΗΘΟΙ: ΔΙΟΝΥΣΙΟΣ ΔΑΜΑΣΙΩΤΗΣ, ΙΑΚΩΒΟΣ ΠΙΤΤΑΡΑΣ, ΘΕΟΔΟΣΗΣ ΓΙΑΝΝΟΠΟΥΛΟΣ**

***3η Σειρά Ασκήσεων***

Μαθήτριες:

|  |  |
| --- | --- |
| Πελαγία Ροδίτη | p3190346 |
| Ελευθερία Ντούλια | p3180129 |
| Αγγελική Ναούμ | s6180070 |

1. Πρωτόκολλο TCP

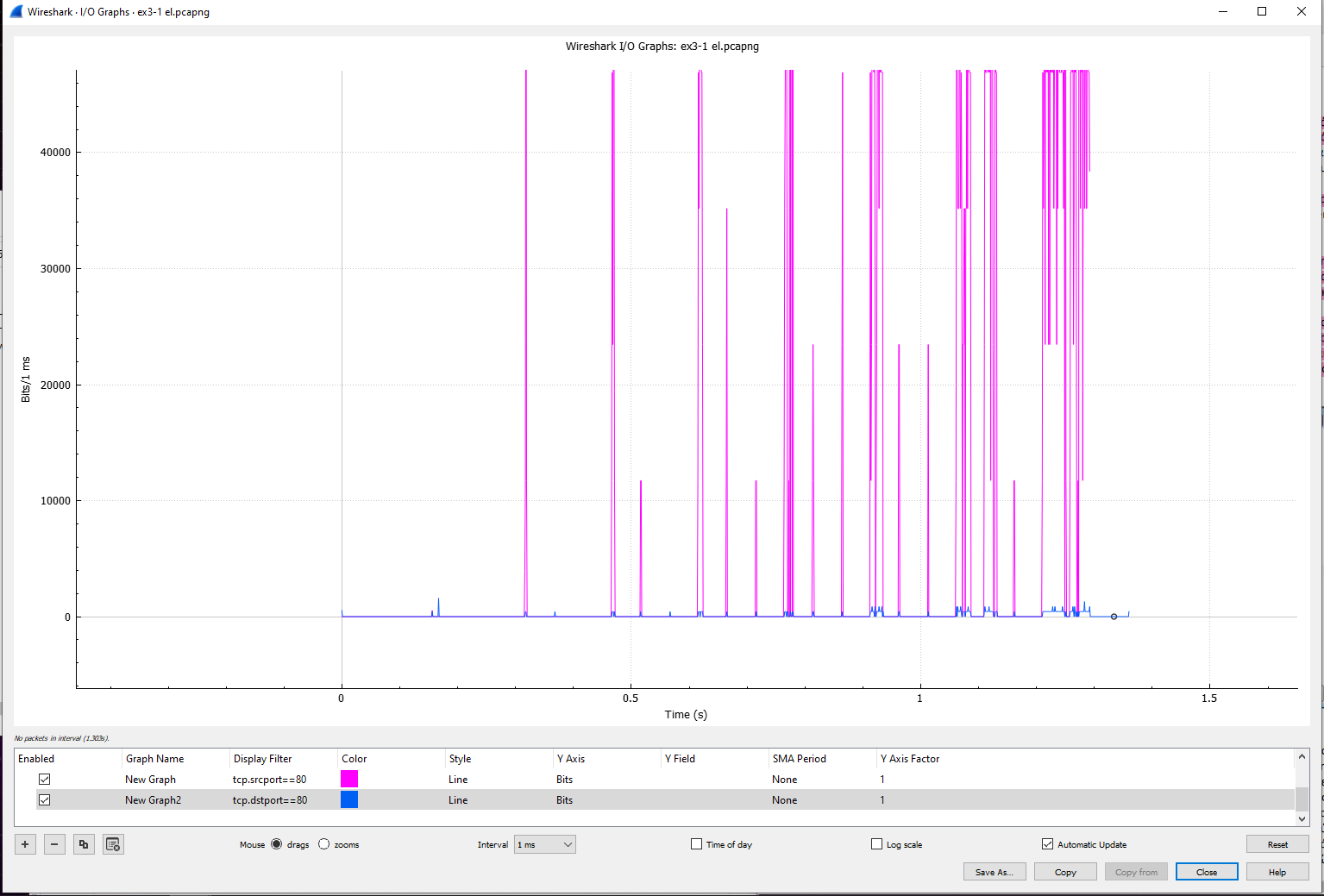
Εικόνα που περιέχει κείμενο

Περιγραφή που δημιουργήθηκε αυτόματα

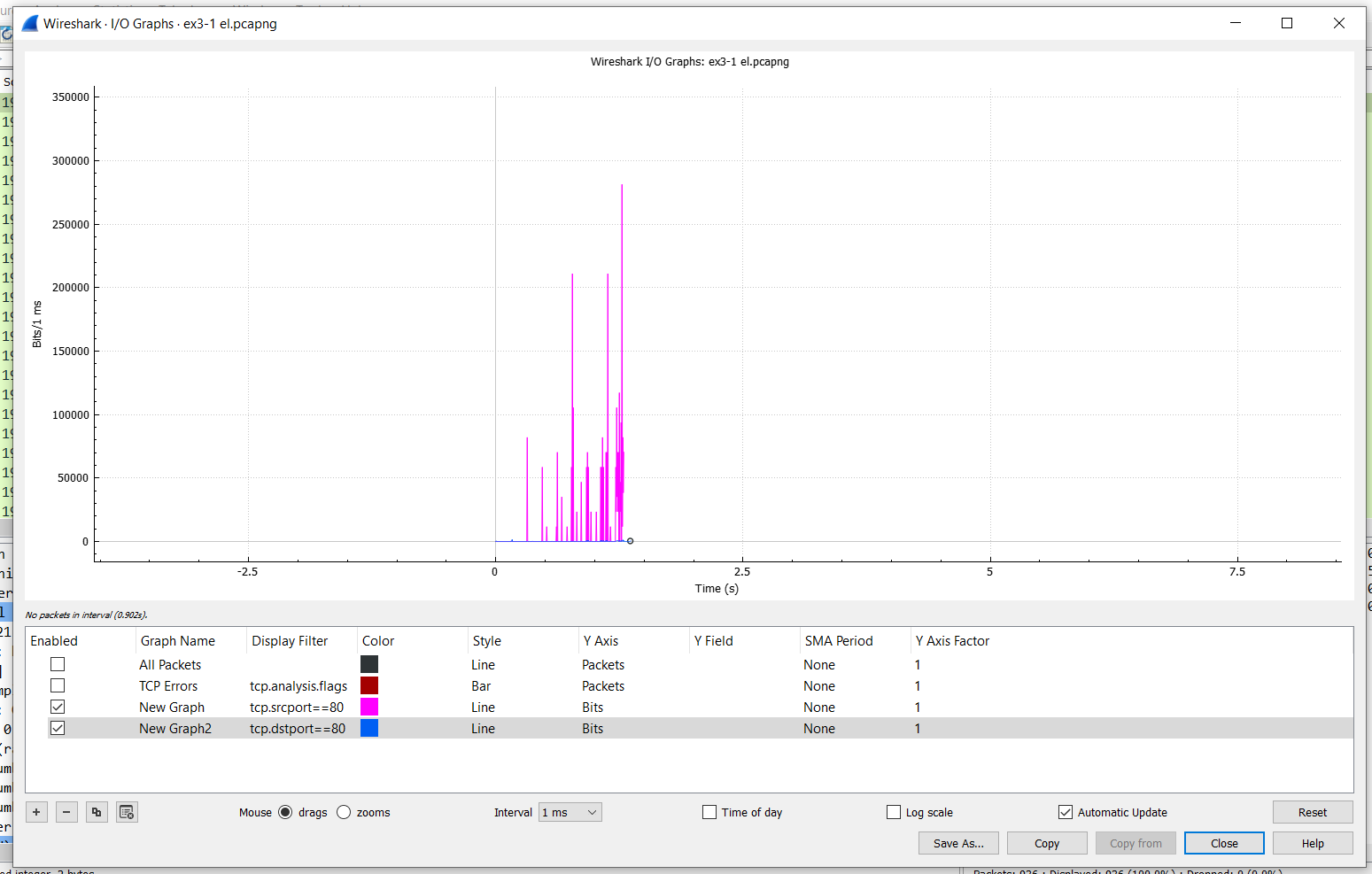
Εικόνα που περιέχει κείμενο

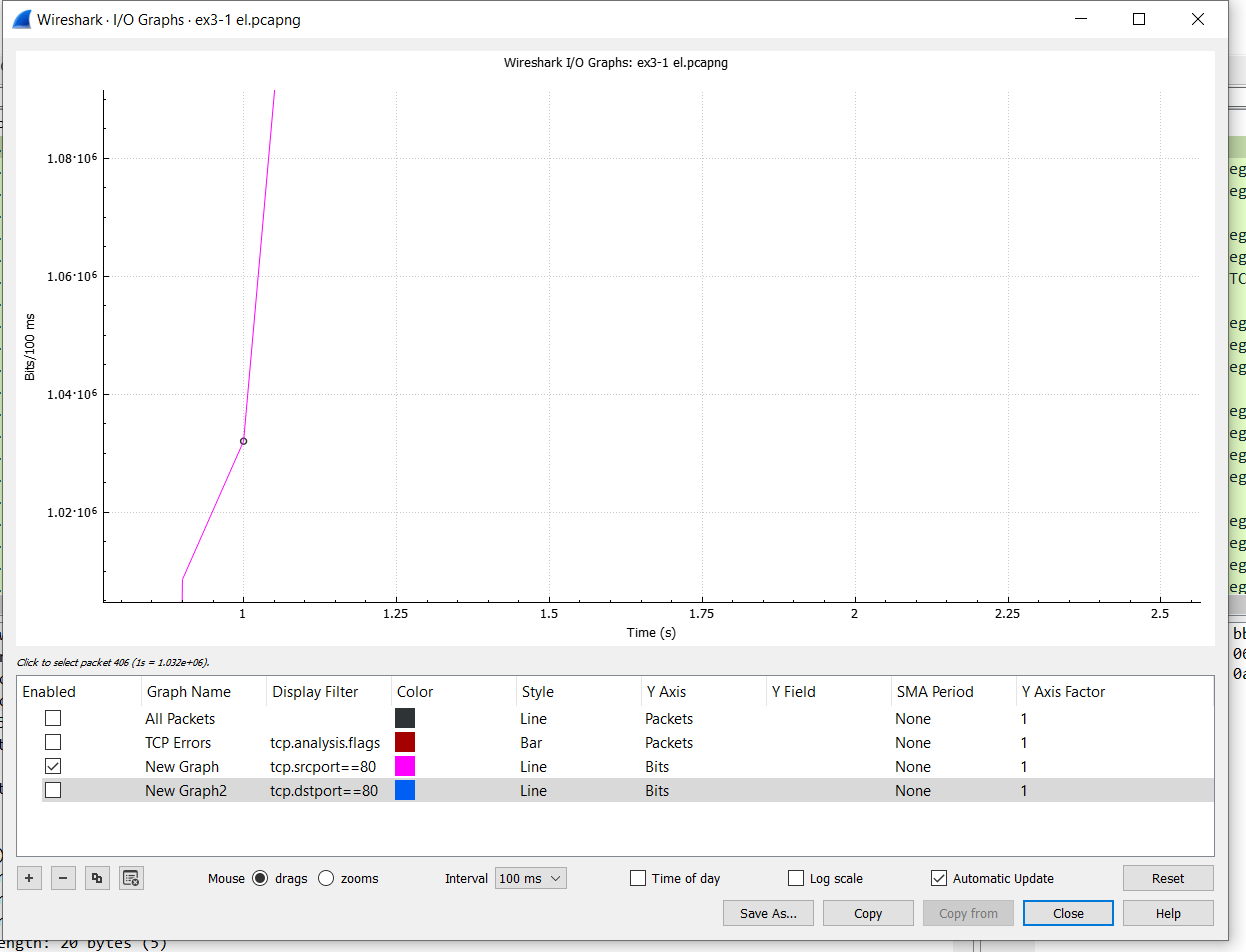
Περιγραφή που δημιουργήθηκε αυτόματα

IO Graph:



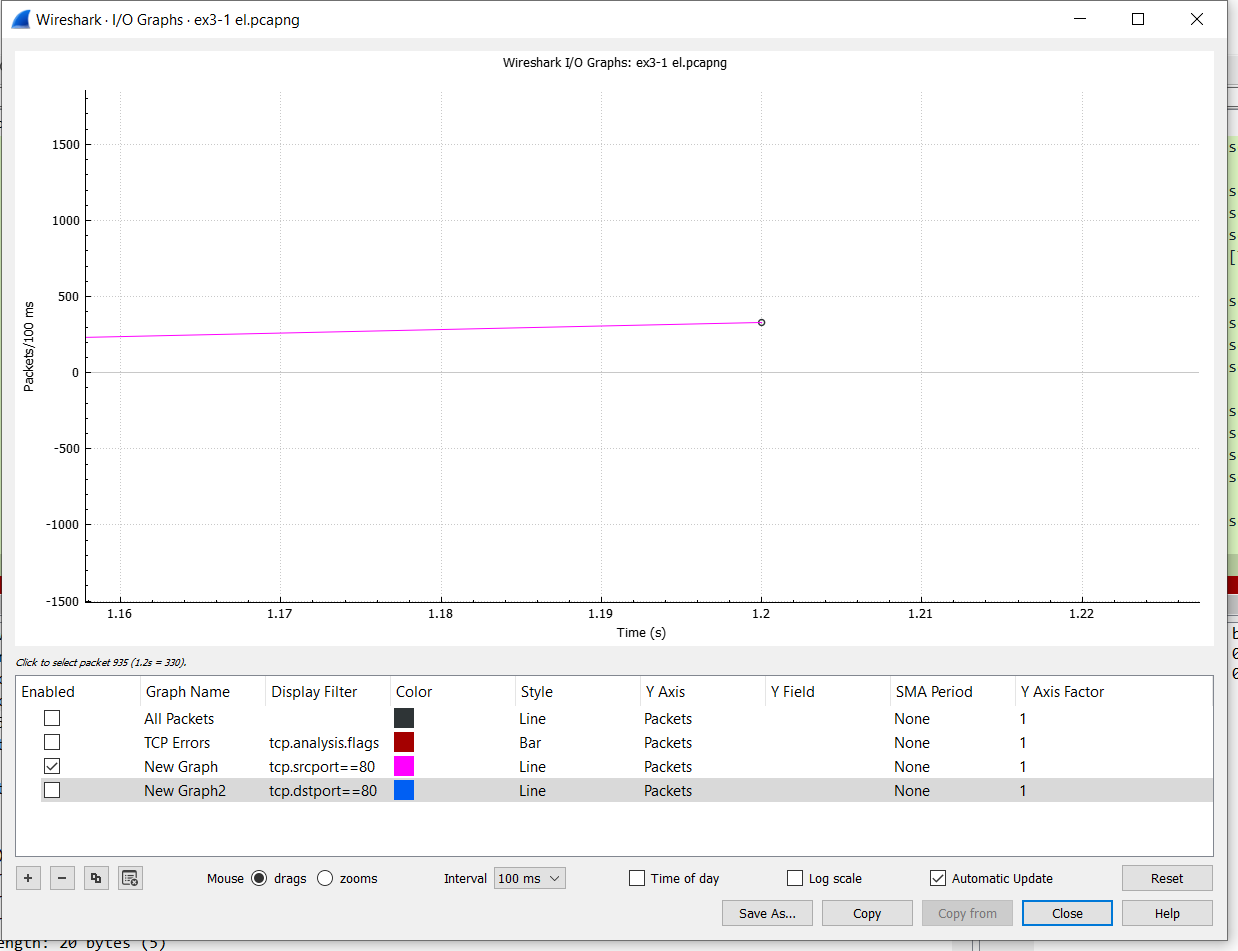
Μέγιστος ρυθμός λήψης περίπου 2,8 Mbps (μωβ γραμμή). Παρατηρούμε ότι ο ρυθμός λήψης εμφανίζει σημαντικές αυξομειώσεις.



**Ερωτήσεις**:

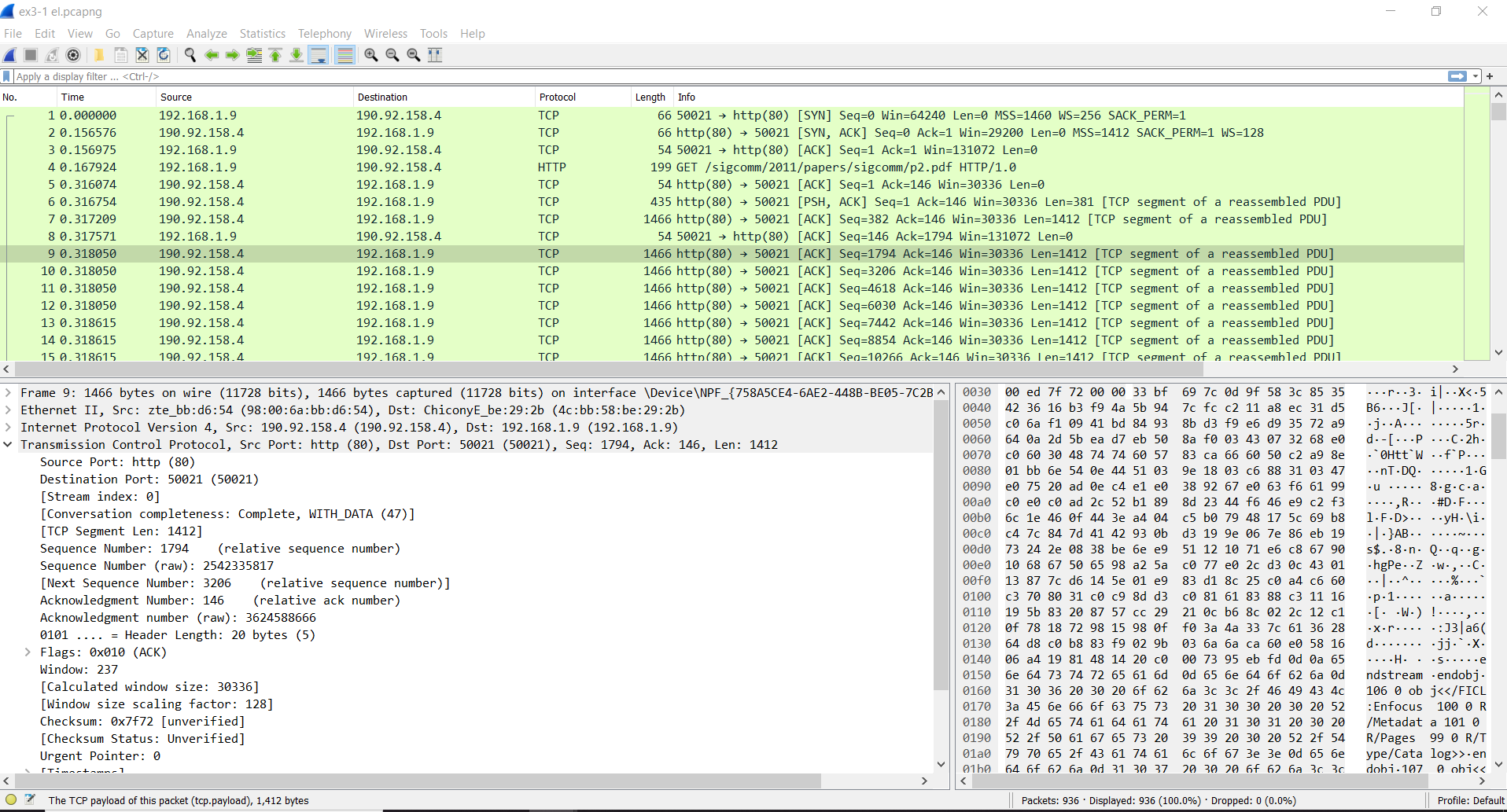
Ο μέγιστος ρυθμός λήψης δεδομένων σε bits/second είναι περίπου 2,81 Mbps, σύμφωνα με το γράφημα.

Ο μέγιστος ρυθμός λήψης δεδομένων σε packets /second είναι 330 πακέτα το δευτερόλεπτο.

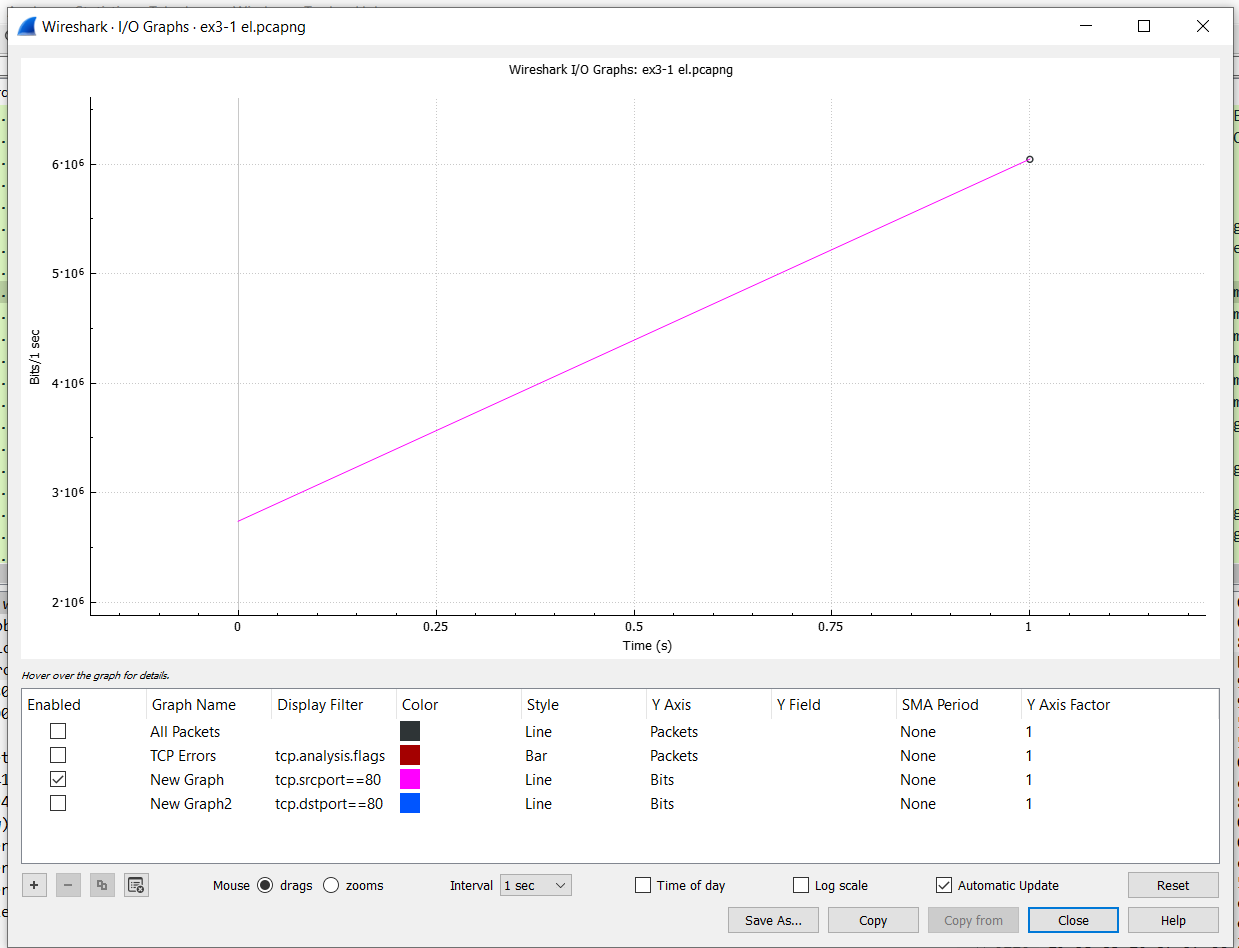


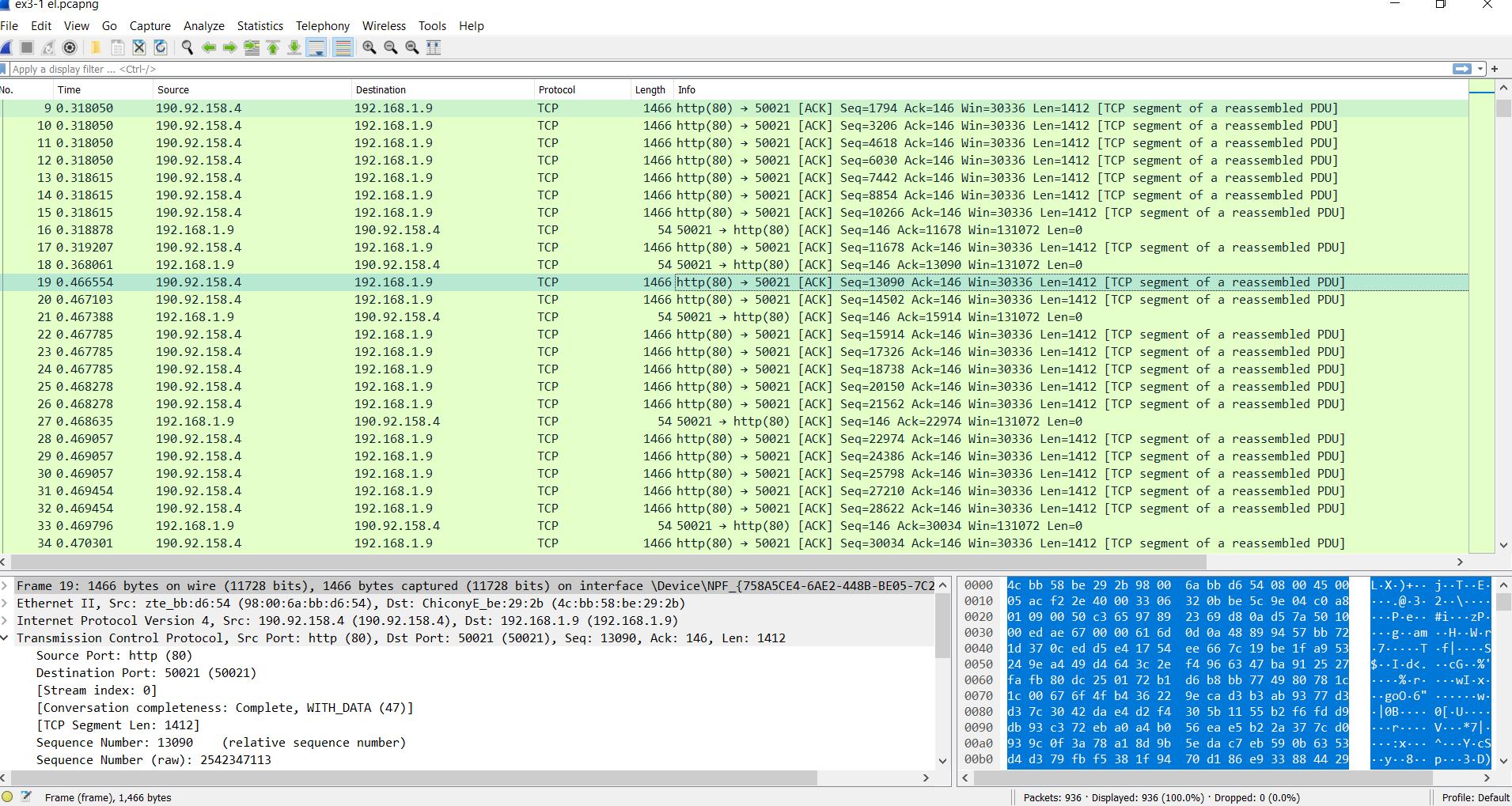
1. Το ποσοστό αυτού του ρυθμού λήψης που αφορά τη λήψη του ωφέλιμου φορτίου του πακέτο μπορεί να υπολογιστεί ως εξής:

Έστω το πακέτο Νο.9, όπως βλέπουμε στη φωτογραφία, το οποίο αποτελεί πακέτο που στέλνεται από τον server που ζητήσαμε να κατεβάσουμε το pdf αρχείο, και περιέχει μέσα μέρος της συνολικής πληροφορίας που χρησιμοποιείται να να συντεθεί στο τέλος το ενιαίο pdf αρχείο που ζητήσαμε.

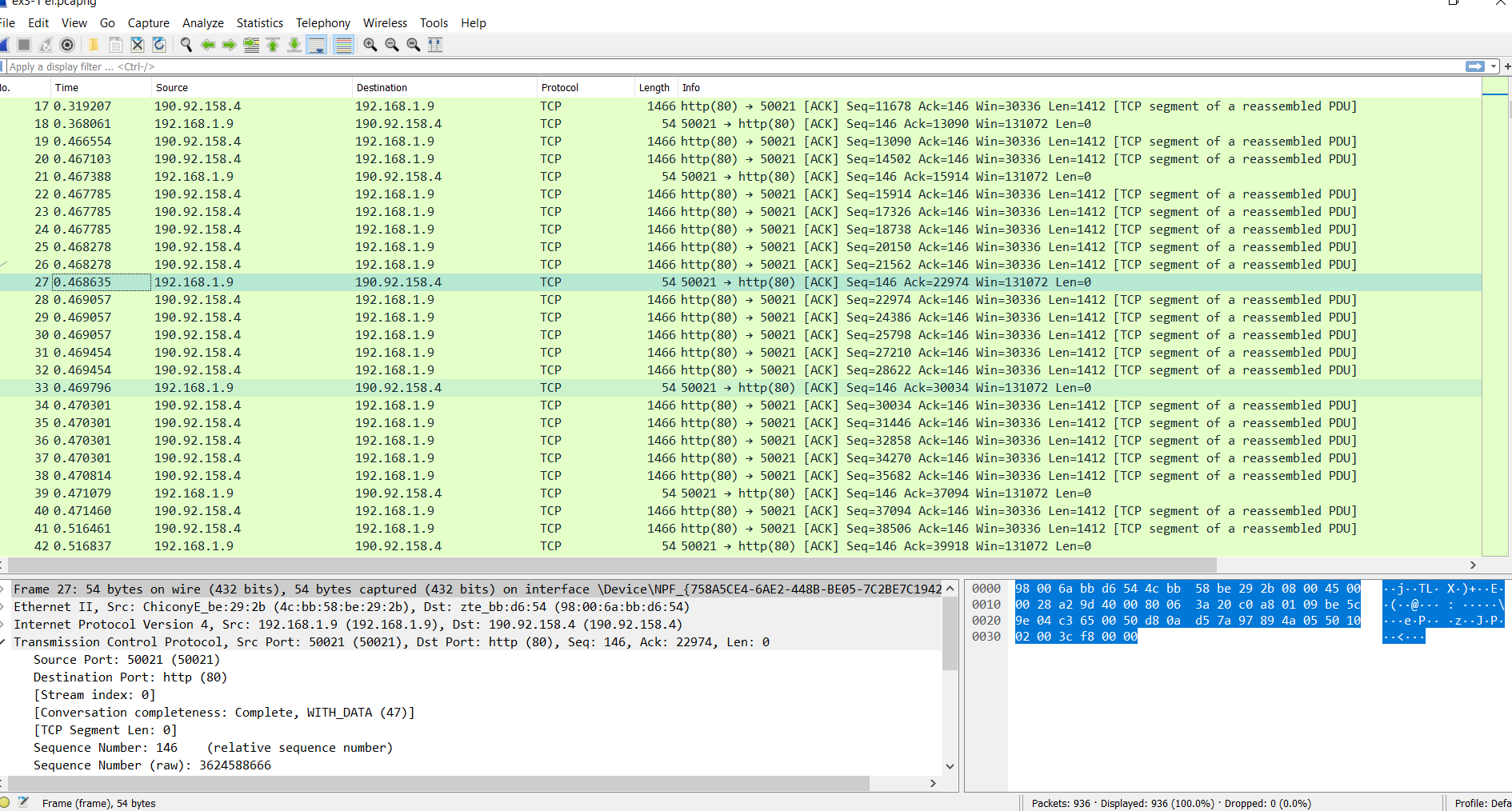


To ποσοστό του ωφέλιμου φορτίου (TCP payload) είναι 96,31%, αφού το συνολικό πλαίσιο αποτελείται από 1.466 bits και το TCP payload καταλαμβάνει 1.412 bits.

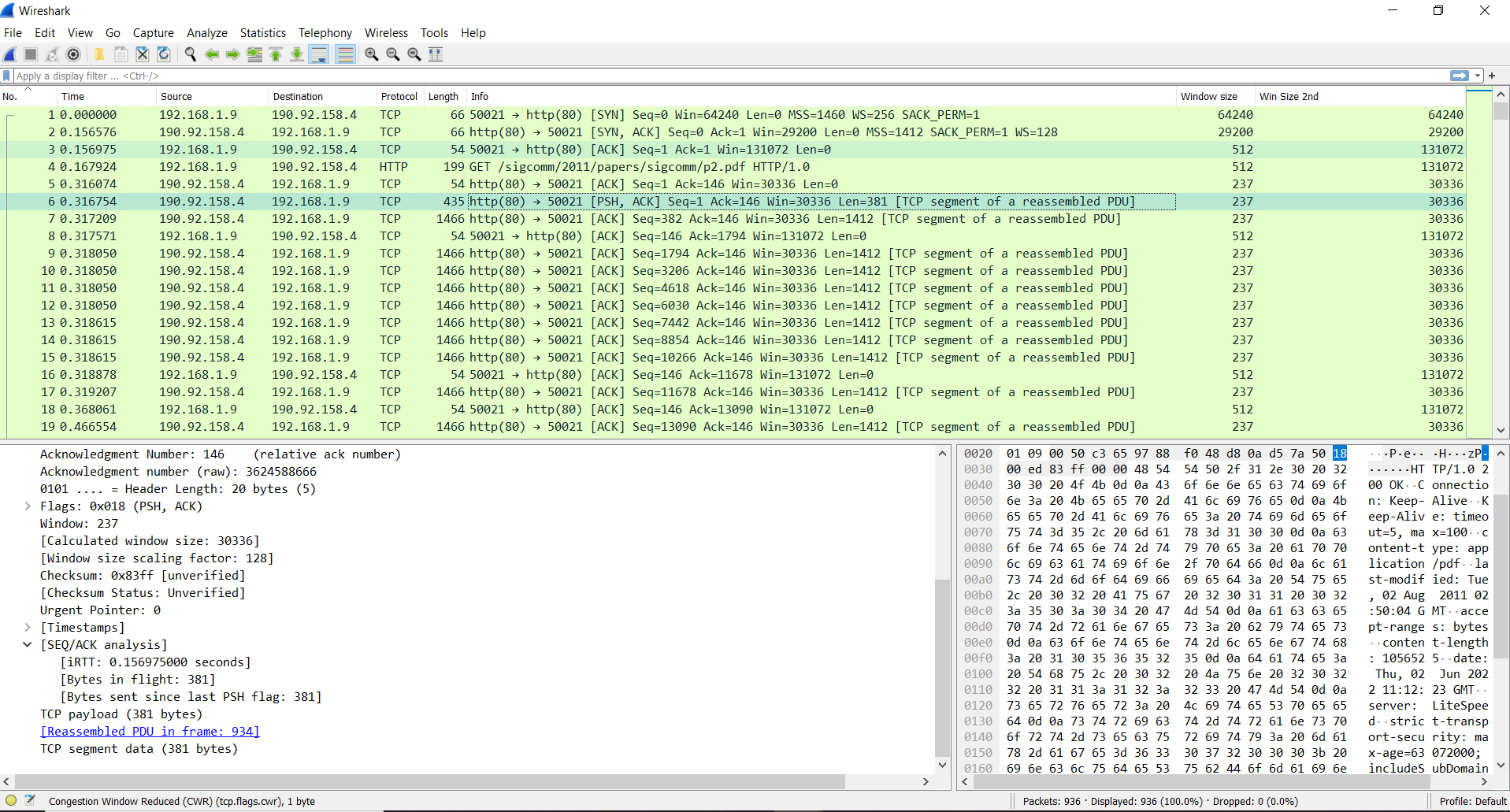
1. Ο ρυθμός αποστολής δεδομένων σε packets/second και bits/second λόγω των πακέτων που περιέχουν τα ACK;
2. Τα πακέτα 19, 20, 21



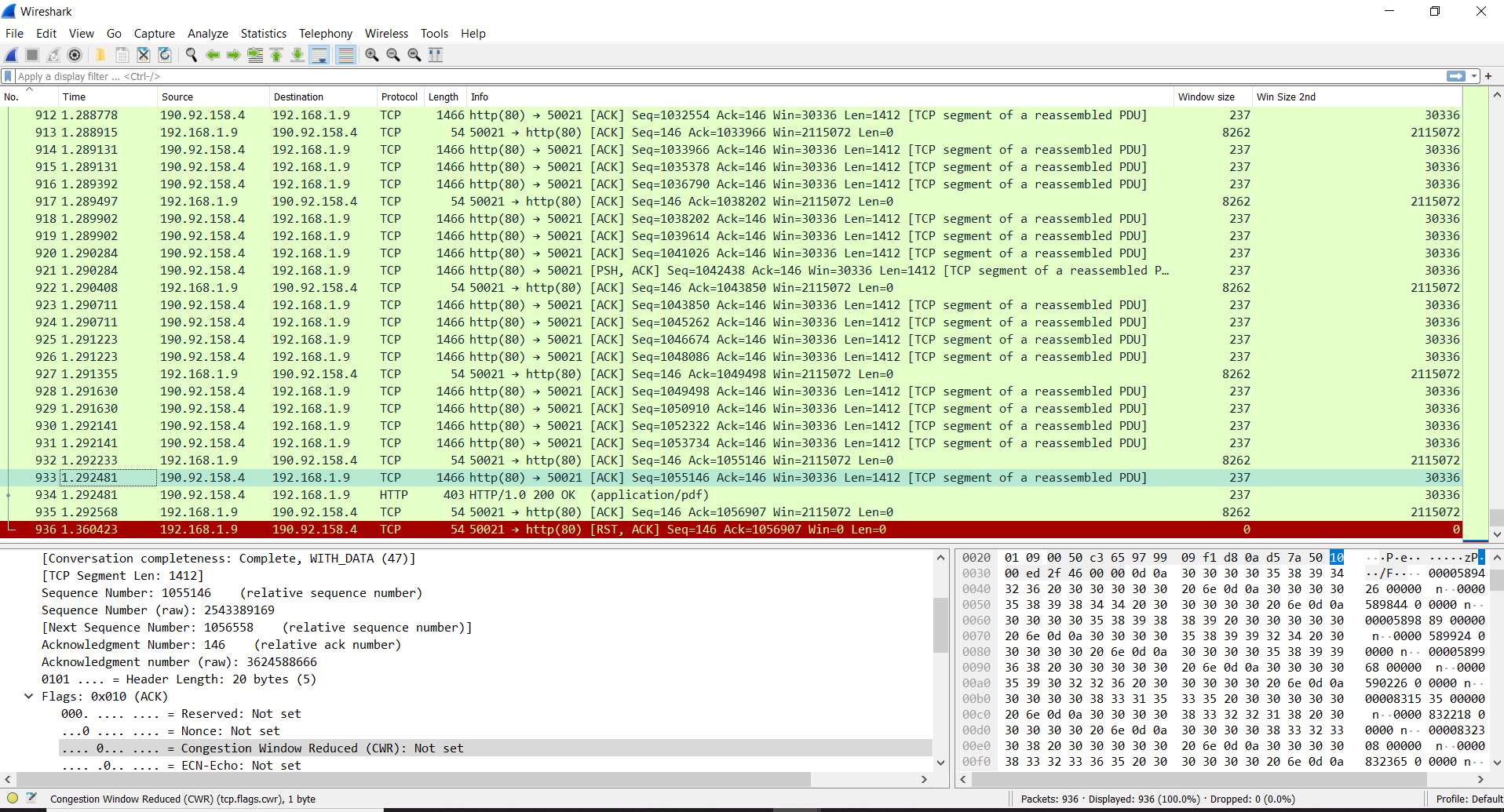
b)) Πακέτα 27 έως 32



d) Το μέγεθος του παραθύρου είναι 30,336 bytes, όπως φαίνεται και στην παρακάτω εικόνα:



1. Εφόσον το πιο πρόσφατα ληφθέν τμήμα TCP από τον server έχει αριθμό ακολουθίας 1055146, το επόμενο TCP τμήμα που μεταδίδεται θα πρέπει να έχει ACK 1055146 + 1412= 1056558

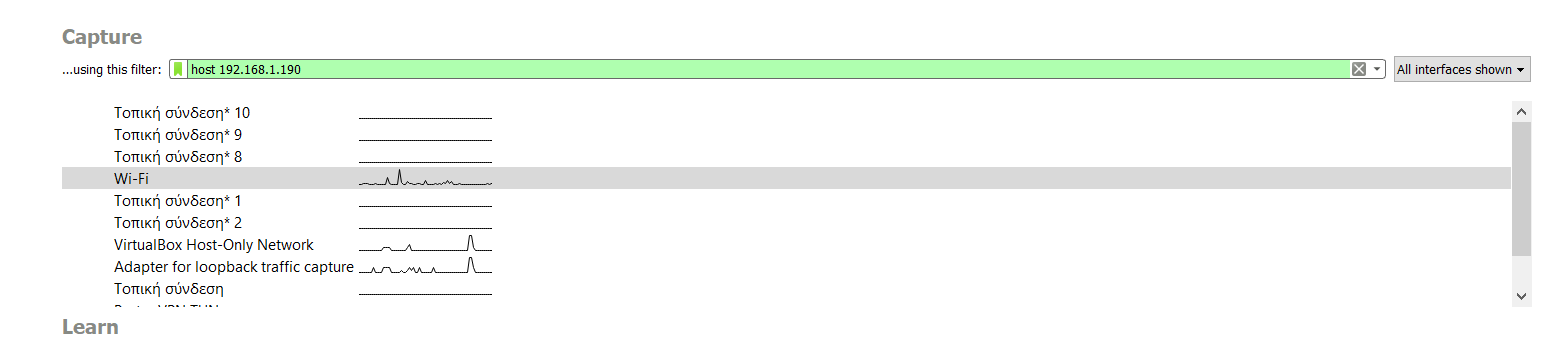


2) Πρωτόκολλο UDP

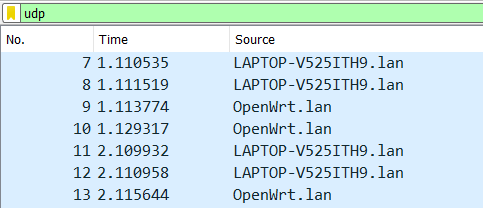
1. Σαν φίλτρο σύλληψης χρησιμοποιήσαμε:

host 192.168.1.190

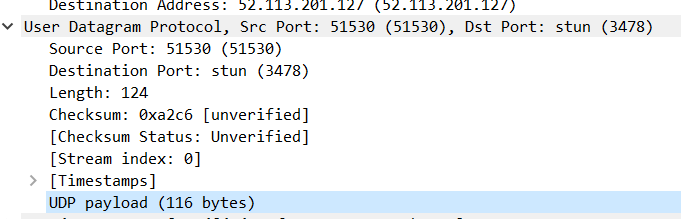
όπου 192.168.1.190 είναι η ip μας.

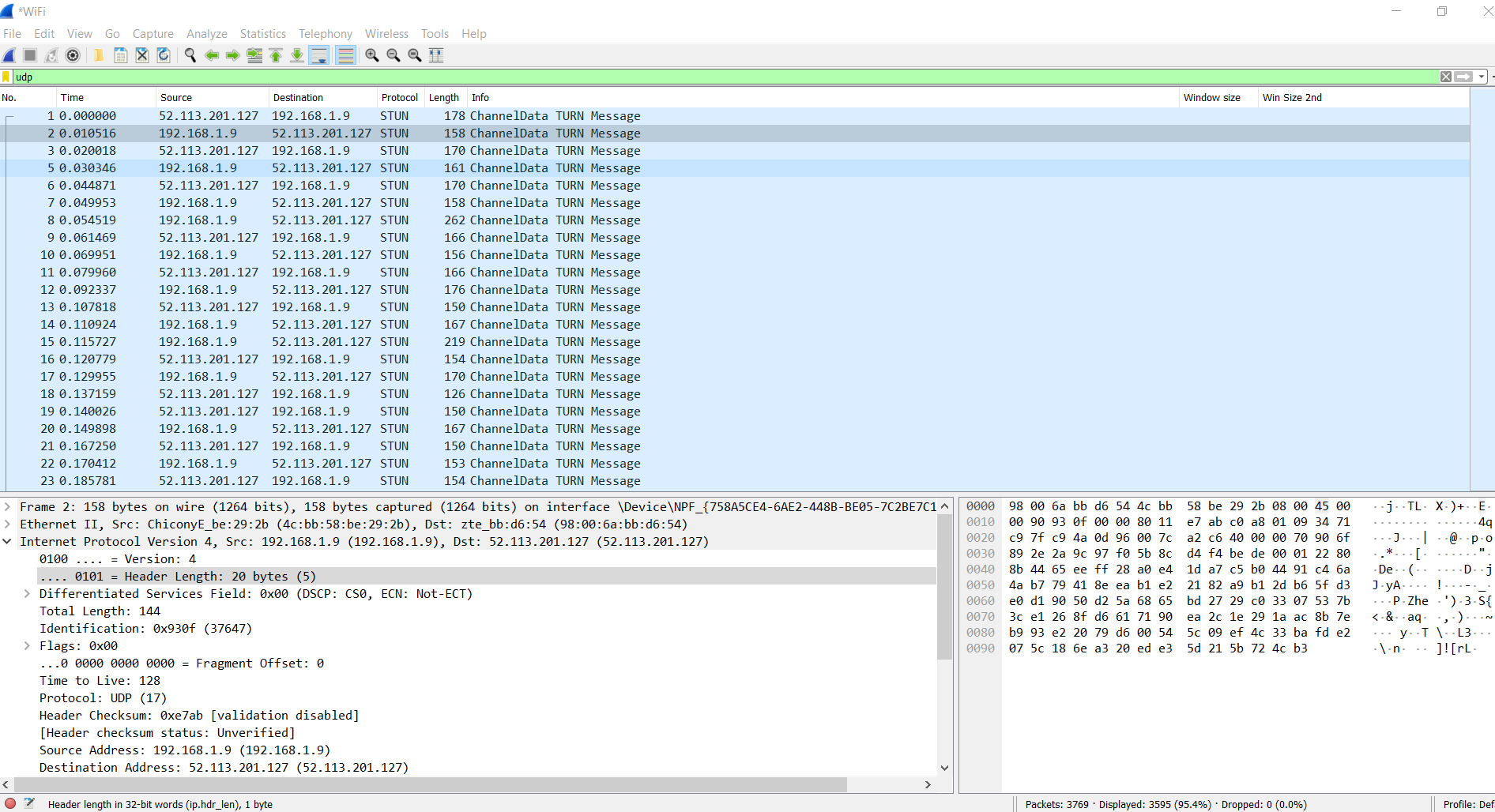
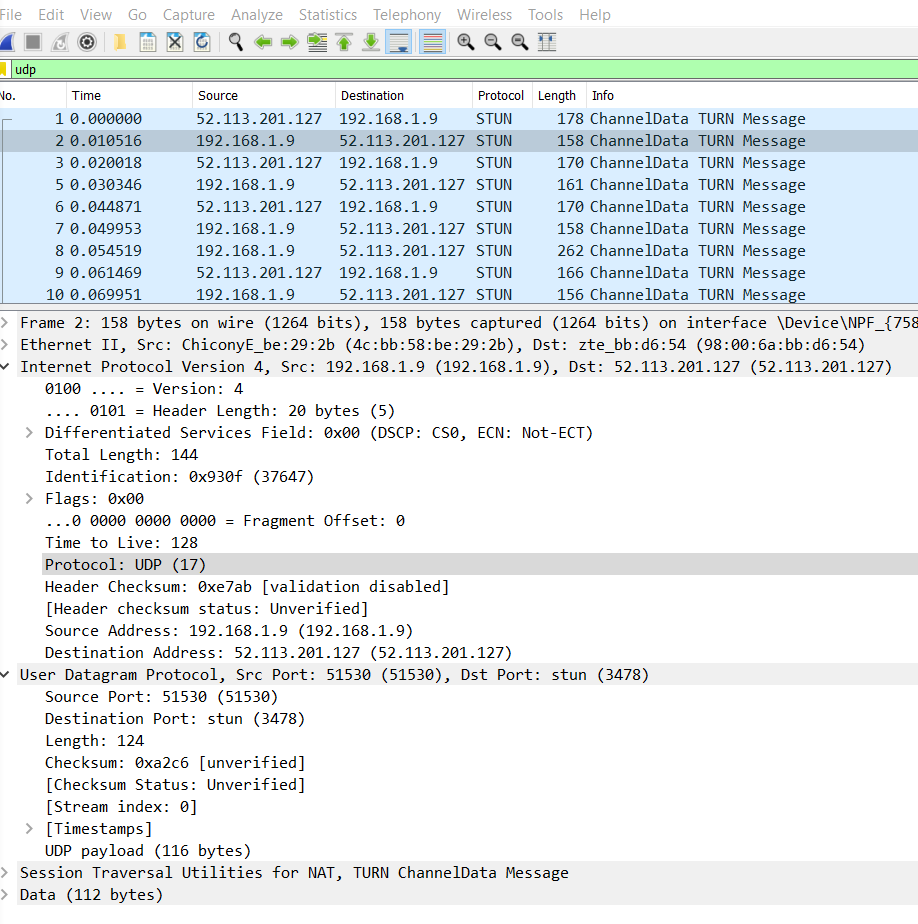


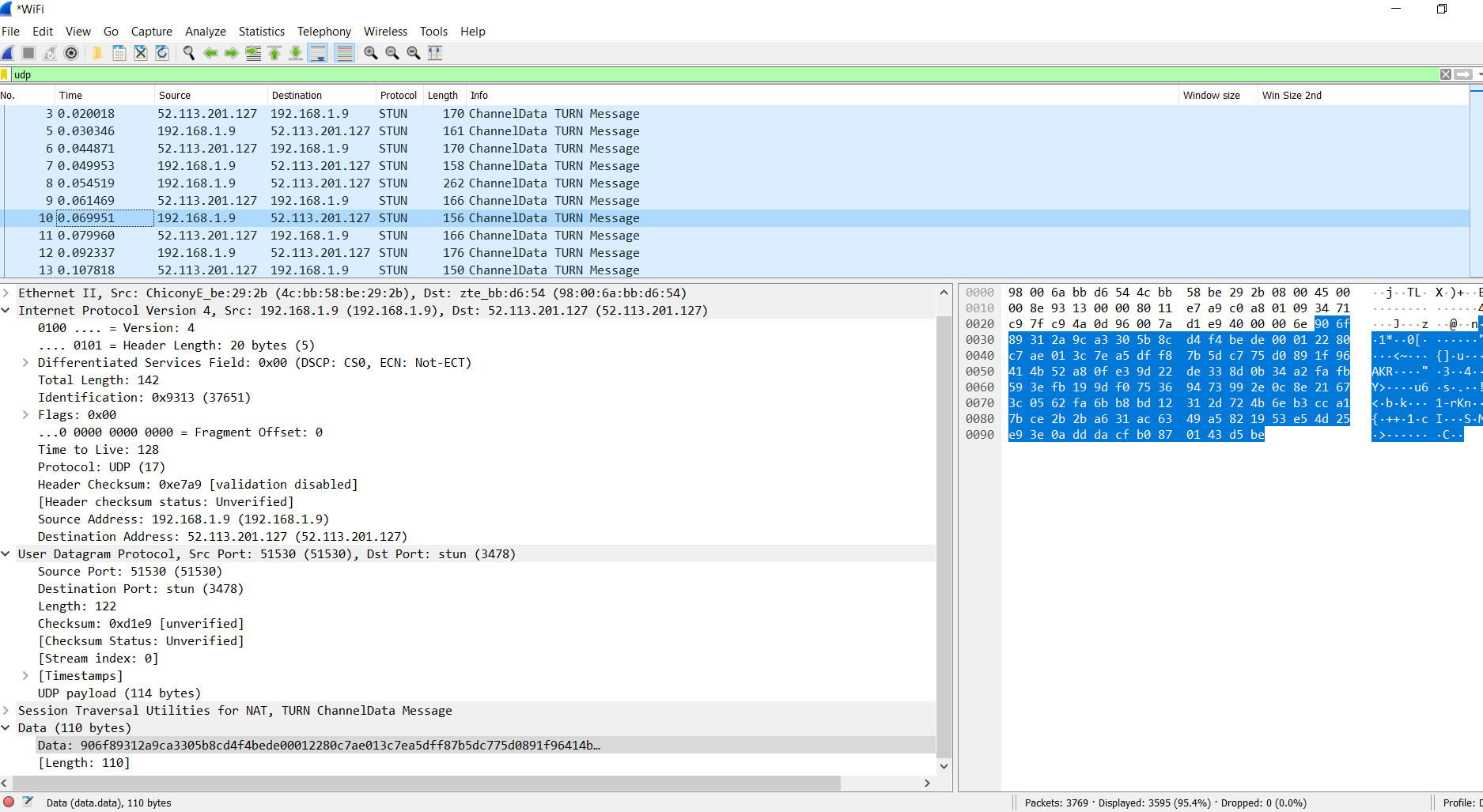
1. Ως φίλτρο απεικόνισης βάλαμε στο display filter ‘udp’.



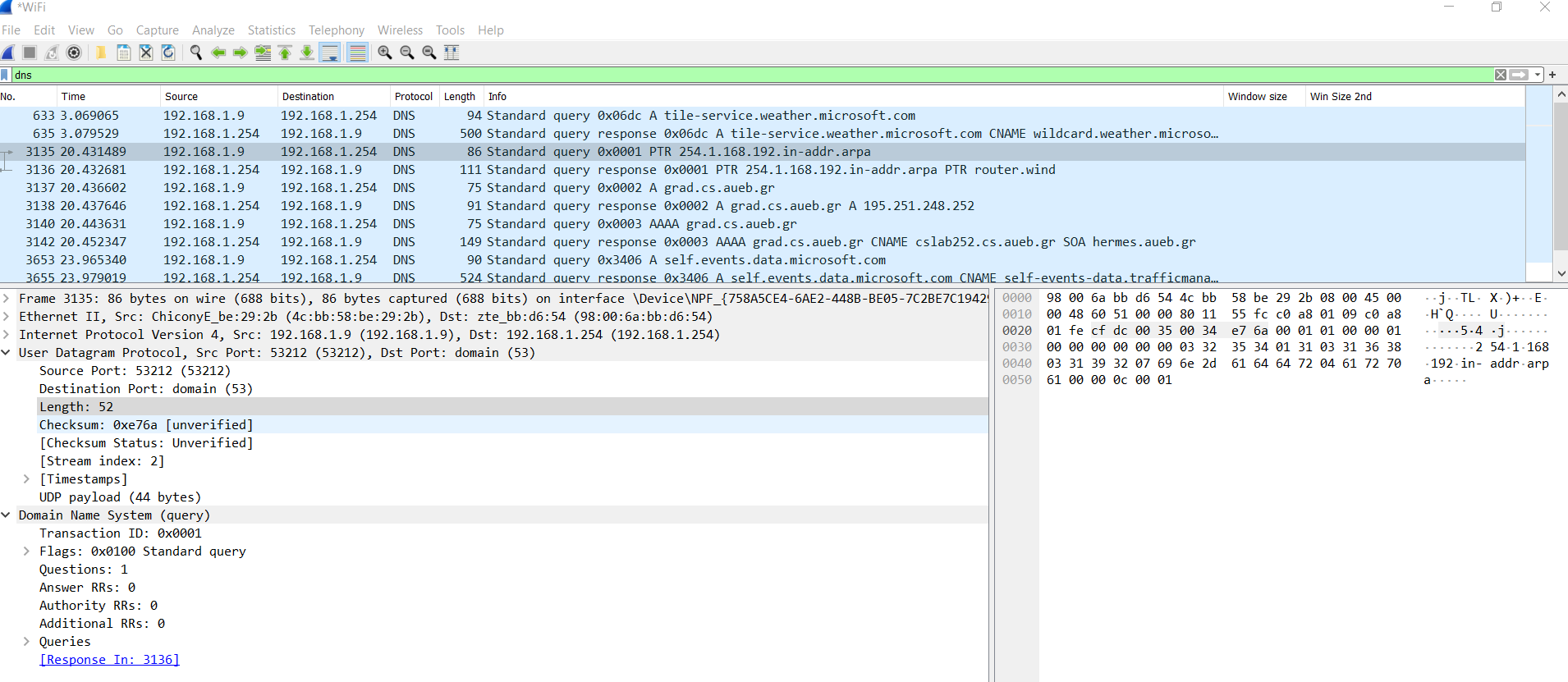
1. source port: 16 bit, destination port: 16 bit, length = 16 bits, checksum = 16 bits



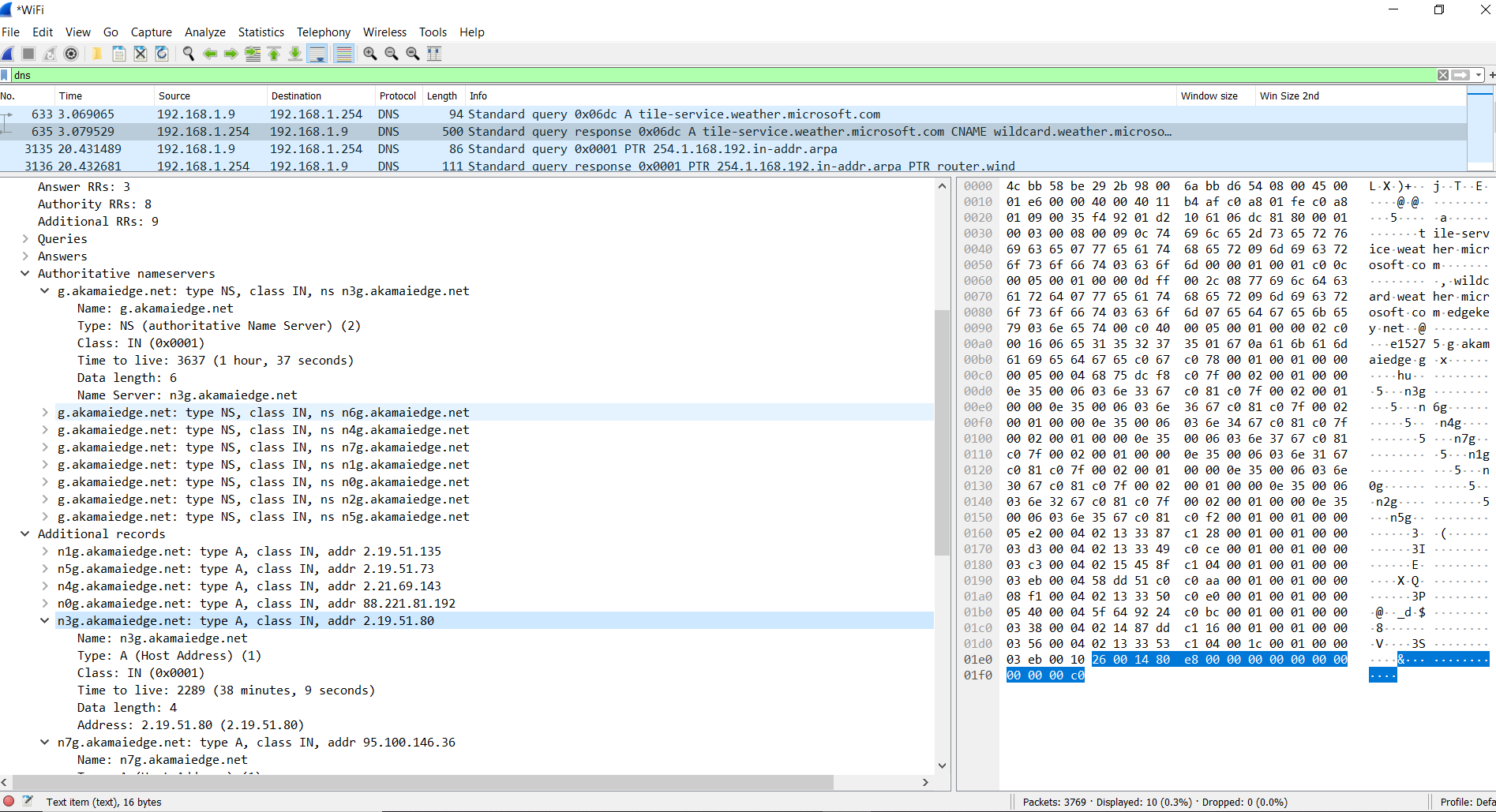
1. Header length: 8 bytes
2.  O αριθμός πρωτοκόλλου για το UDP στην επικεφαλίδα του πακέτου IP εντός του οποίου ενθυλακώνεται είναι το 17.
3. Το μήκος του τμήματος (segment) βάσει του μεγέθους του πακέτου αυτού, είναι 142 - 20 = 122 Bytes. Δηλαδή από το συνολικό μέγεθος του ΙΡ πακέτου (142) αφαιρούμε την επικεφαλίδα του ΙΡ πακέτου μας, και στο τέλος μένουν τα bytes που αντιστοιχούν στο UDP segment.



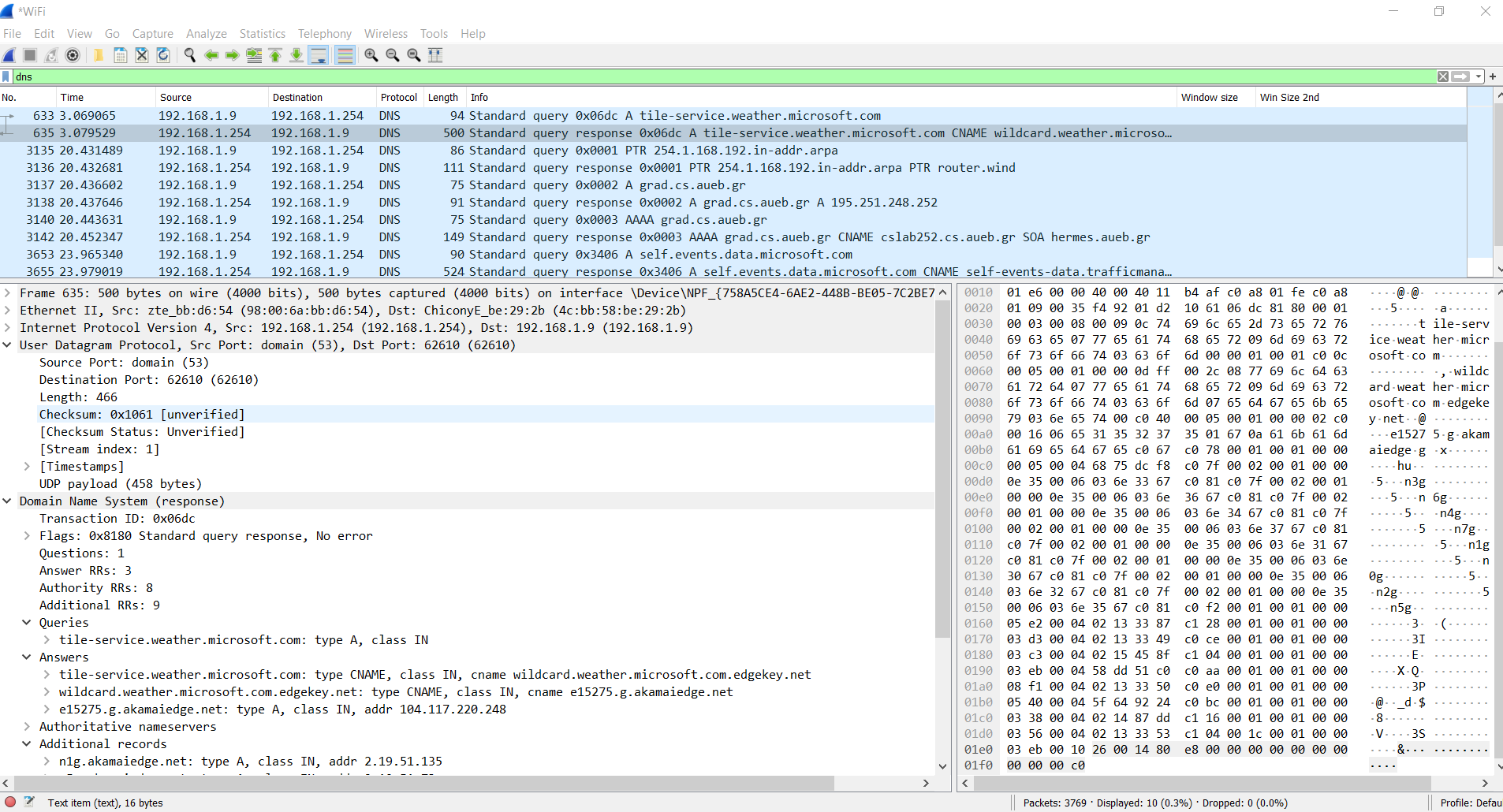
1. Προσδιορίζει το συνολικό μέγεθος του πακέτου UDP σε Bytes, συμπεριλαμβανομένου και της επικεφαλίδας.
2. Το μέγιστο μέγεθος τμήματος για UDP είναι 65.535 bytes (8 byte header + 65,527 bytes of data), διότι το μέγεθος του πεδίου που αντιστοιχεί στο προσδιορισμό του μεγέθους του segment είναι 2 bytes, επομένως ο μεγαλύτερος αριθμός που μπορούμε να πάρουμε από 2 bytes είναι ο 65.535 (1111111111111111).
3. Στην εικόνα φαίνεται ότι το πρωτόκολλο που χρησιμοποιείται είναι το UDP.



1. Ένας από τους nameservers είναι ο n3g.akamaiedge.net με IP address: 2.19.51.80



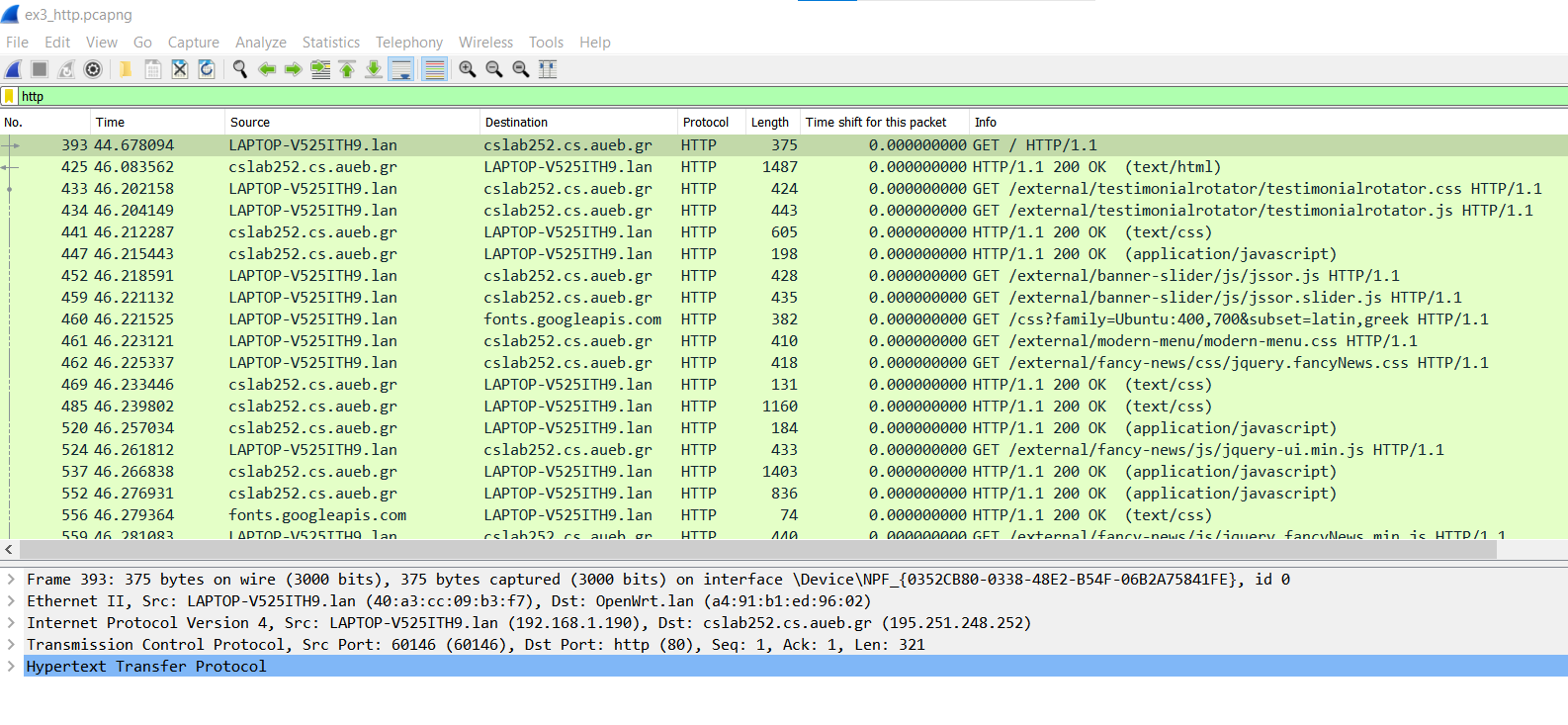
1. Source port: 53, destination port: 62610



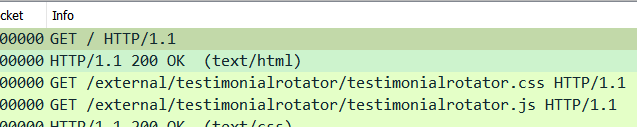
1. Αντιστοιχεί η θύρα 53 (well-known port)

**3) Πρωτόκολλο HTTP**

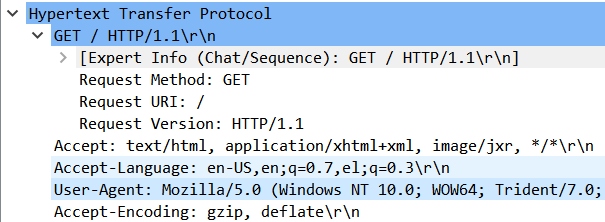
1. Το φίλτρο απεικόνισης που χρησιμοποιήσαμε είναι http.



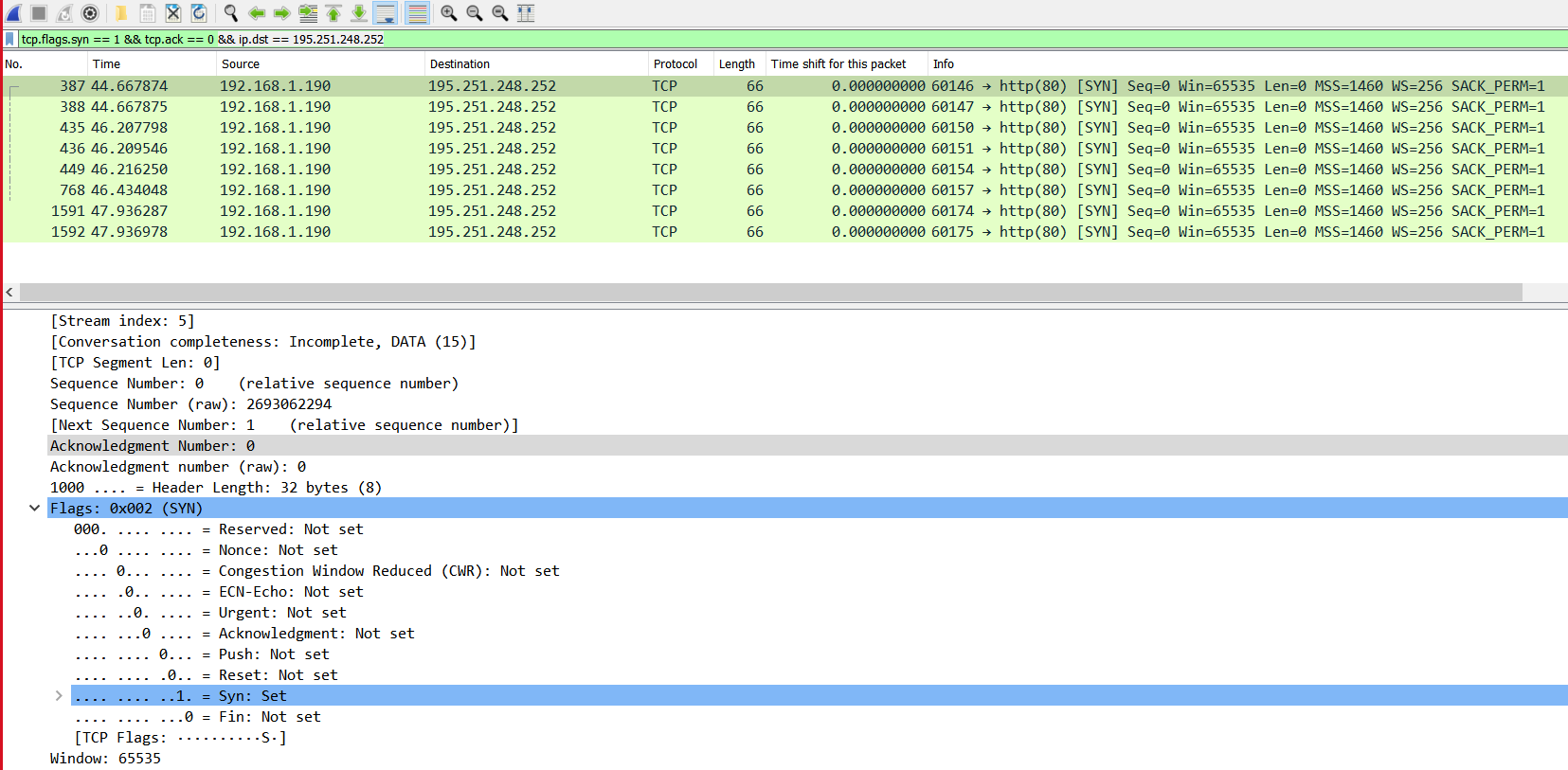
1. H έκδοση HTTP που χρησιμοποίησε ο πλοηγός μας είναι η 1.1.

****

1. Οι γλώσσες που υποστηρίζει είναι αγγλικά και ελληνικά.

****

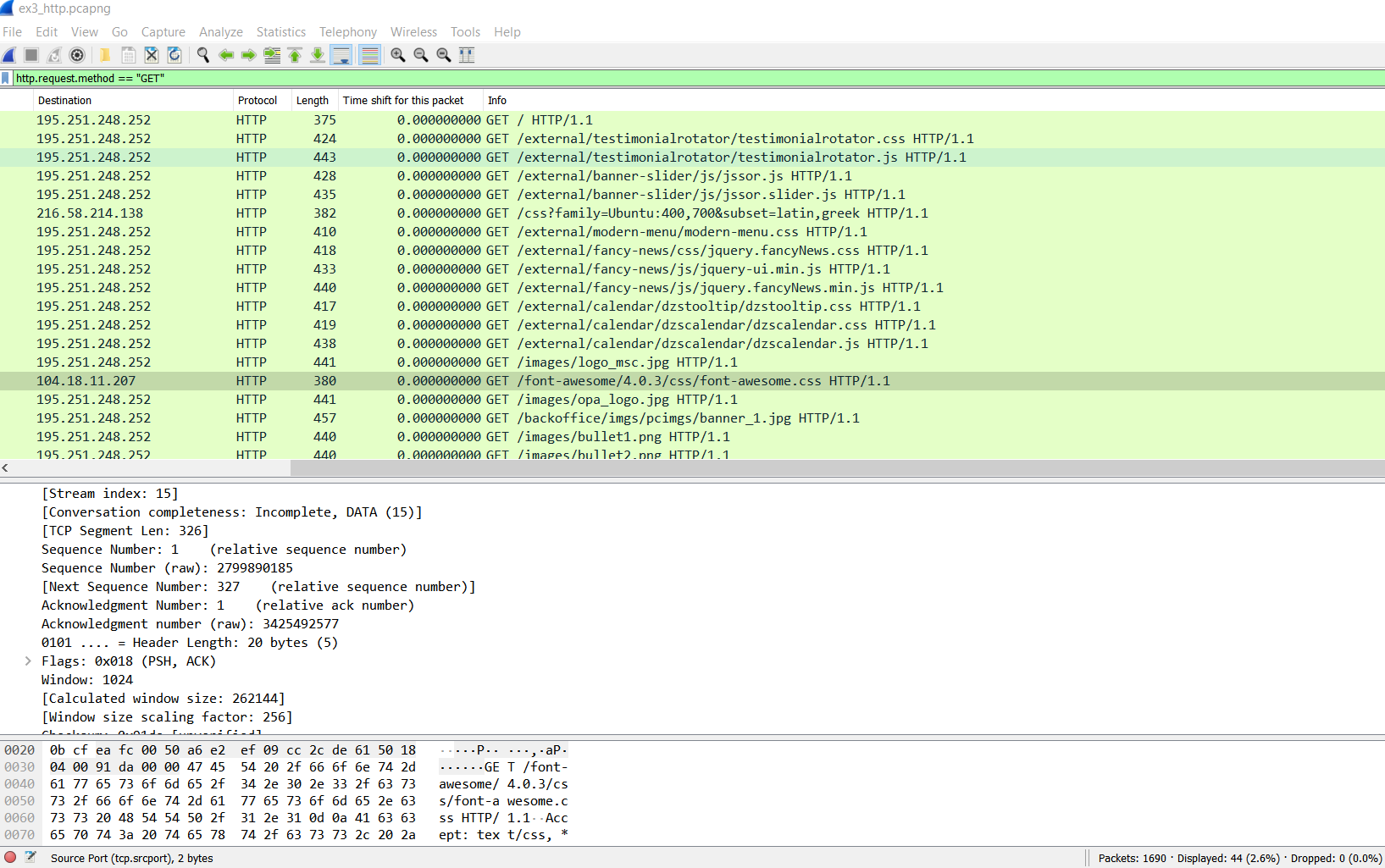
1. To φίλτρο που χρησιμοποιήσαμε είναι tcp.flags.syn == 1 && tcp.ack == 0 && ip.dst == 195.251.248.252.



Χρησιμοποιήσαμε αυτά τα φίλτρα διότι θέλουμε τα πρώτα segments από κάθε tcp 3 way handshake που έχει πραγματοποιηθεί κατά την διάρκεια της αναζήτησης της ζητούμενης ιστοσελίδας. Επομένως, το πρώτο βήμα της τριπλής χειραψίας το SYN bit από τον υπολογιστή μας είναι 1 (set) και το ACK bit είναι 0 (αφού περιμένουμε να το “επιστρέψει” με την τιμή “1” ο server).

Oι συνδέσεις TCP είναι 8 και οι θύρες πηγής είναι 60146, 60147, 60150, 60151, 60154, 60157, 60174, 60175.

1. Το φίλτρο που χρησιμοποιήσαμε για να εμφανιστούν μόνο τα HTTP Requests είναι http.request.method == "GET" και αποστάλθηκαν 44 εντολές προς τον εξυπηρετητή ιστού.

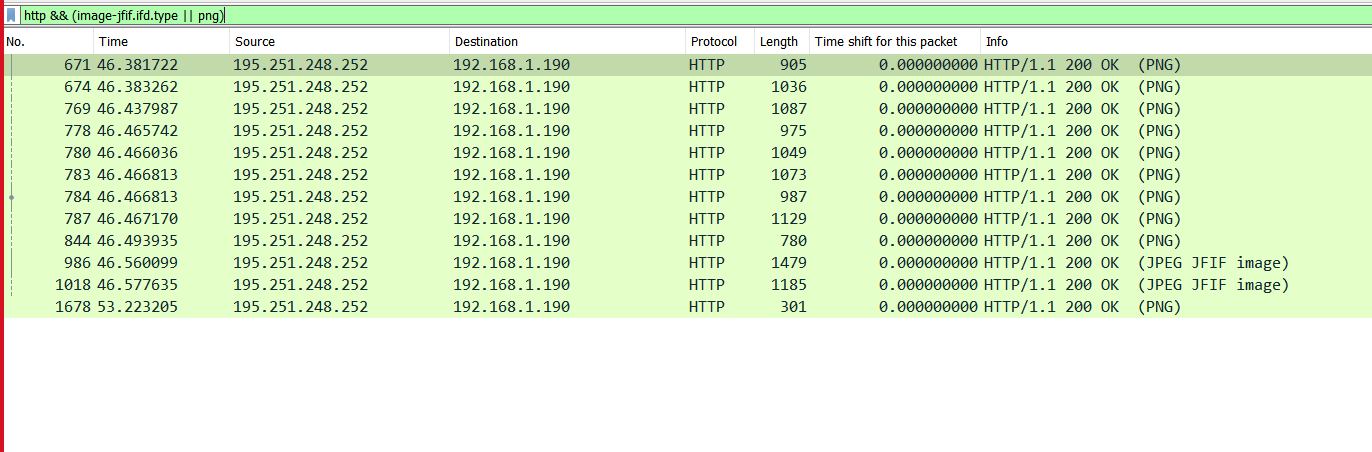


1. Η αρχική γραμμή των αποκρίσεων περιέχει 3 μέρη. Αρχικά έχει το όνομα της μεθόδου με κεφαλαία γράμματα, σε εμάς είναι το GET, την τοπική διαδρομή (local path) του αιτούμενου πόρου και τέλος την έκδοση του πρωτοκόλλου που χρησιμοποιείται, στην περίπτωσή μας HTTP 1.1.
2. Στο 1ο βήμα [SYN], ο client θέλει να εγκαθιδρύσει σύνδεση με τον server, οπότε στέλνει ένα tcp syn μήνυμα με το SYNbit = 1 και sequence number= 0 , έτσι ώστε ο εξυπηρετητής να ξέρει ότι ο πελάτης θέλει να επικοινωνήσει μαζί του, και με τι sequence number θα ξεκινάει τα segments που του στέλνει.

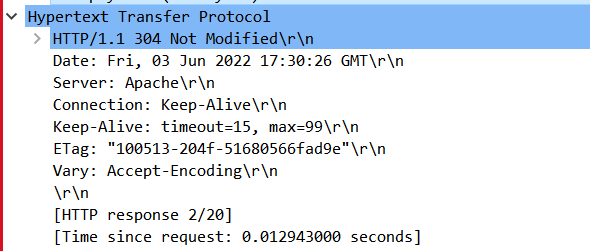
Στο 2ο βήμα [SYN, ACK] ο εξυπηρετητής απαντά στον πελάτη με ένα SYNACK μήνυμα, (θέτει το ACKbit=1). Δηλαδή με το ACK σηματοδοτεί ότι είναι αυτό το μήνυμα απάντηση στο προηγούμενο που είχε στείλει ο πελάτης και παράλληλα στέλνει στον πελάτη ACKnum= 1, το οποίο είναι το sequence number του πελάτη αυξημένο κατά ένα, για να δηλώσει ότι απαντάει για το συγκεκριμένο ερώτημα. Επίσης επιλέγει ένα δικό του αρχικό sequence number = 0 που θα χρησιμοποιείται για αυτή τη σύνδεση.

Στο 3ο βήμα ο πελάτης κάνει acknowledge την απάντηση του εξυπηρετητή και εγκαθιδρύεται μία αξιόπιστη σύνδεση μεταξύ τους, έτσι ώστε να ξεκινήσει η ανταλλαγή δεδομένων (ίσως σε αυτό το σημείο αποστέλλονται και δεδομένα από το πελάτη προς τον εξυπηρετητή) . Συγκεκριμένα, στέλνει πίσω ένα μήνυμα ACK ως απάντηση για το SYNACK μήνυμα του server, αυξάνει το ACKnum του εξυπηρετητή κατά 1 (ACKnum = 1) και στέλνει το δικό του Sequence number όπως ήταν κατά 1 μονάδα αυξημένο από τον εξυπηρετητή, στο προηγούμενο βήμα (Sequence number = 0).

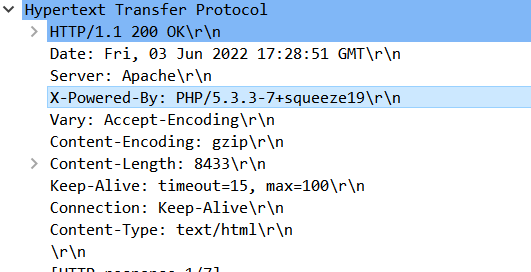
1. O πλοηγός ιστού κατέβασε 12 εικόνες και όπως μπορούμε να δούμε και παρακάτω βλέπουμε ότι επιστρέφονται από την ίδια ip.



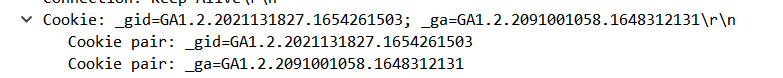
1. Ο κωδικός κατάστασης (status code) που επιστρέφει ο εξυπηρετητής ως απόκριση στο πρώτο μήνυμα HTTP τύπου GET είναι ο 304, τo οποίο υποδηλώνει ότι η σελίδα που ζητήσαμε με αυτό το αίτημα, δεν έχει τροποποιηθεί από την τελευταία φορά που την ζητήσαμε.



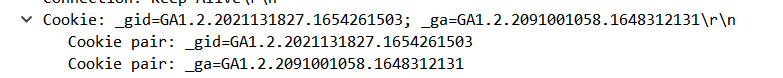
1. Σε μία πιο δυναμικά μεταβαλλόμενη σελίδα, αν την κατεβάσουμε δεύτερη φορά σε χρόνο που έχει αλλάξει το περιεχόμενό της, κατεβαινει εκ νέου και το status code έχει την τιμή 200.
2. Την πρώτη φορά που κατεβάζουμε ο κωδικός κατάστασης (status code) που επιστρέφει ο εξυπηρετητής ως απόκριση στο πρώτο μήνυμα HTTP είναι 200. To περιεχόμενο αυτό τροποποιήθηκε τελευταία φορά Fri, 03 Jun 2022 17:28:51.



1. Στο πρώτο get ερώτημα βλέπουμε cookie pair όπως φαίνεται στο παρακάτω σχήμα.



1. Όπως και στο προηγούμενο ερώτημα, στο πεδίο cookie βλέπουμε και πάλι το pair.



**4) Γενικές ερωτήσεις και ασκήσεις**

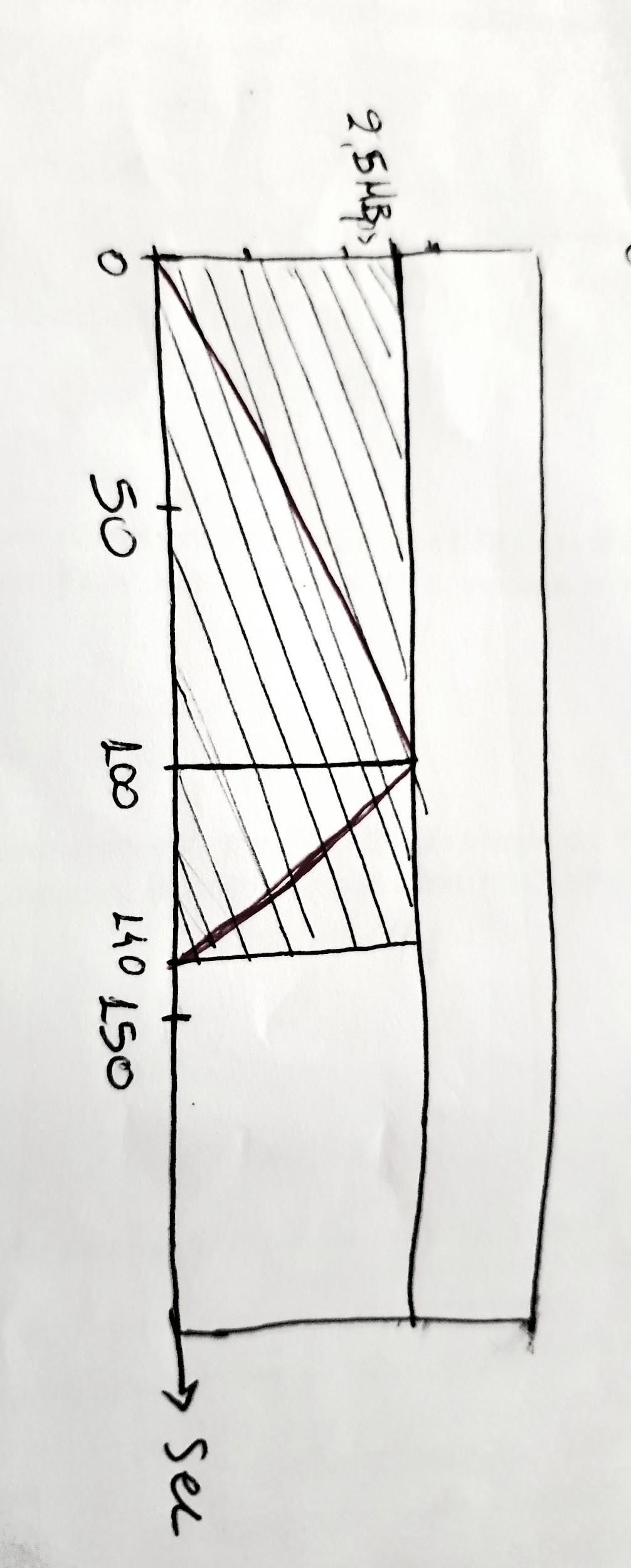
**Άσκηση 1**

α) Ο υπολογιστής χρειάζεται χρόνο για να στείλει τα δεδομένα στον κουβά ίσο με:

t1 = χρειάζονται για να μεταφερθούν όλα τα δεδομένα από τον υπολογιστή στον κουβά.

Ο υπολογιστής στέλνει δεδομένα στο δίκτυο με ρυθμό 2.5MByte/s , επομένως ο κουβάς γεμίζει, αφού και ο ρυθμός μετάδοσης στο σύνδεσμο από τον υπολογιστή προς τον κουβά είναι 3.5MByte/s, με ρυθμό: 3.5 - 2.5 = 1MBps.

Άρα η ελάχιστη χωρητικότητα που πρέπει να έχει ο κουβάς προκειμένου να μην συμβεί απώλεια δεδομένων είναι: 1 MBps \* 100 sec = 100 MBytes.

β) Με το πέρας των 100 δευτερολέπτων, με την ριπή έχουν αποσταλεί όλα τα δεδομένα στον κουβά και αυτός έχει στείλει στο δίκτυο: 2.5MByte/s \* 100s = 250ΜΒ. Οπότε απομένουν να σταλούν ακόμη 350 – 250 = 100ΜΒ στο δίκτυο, τα οποία χρειάζονται χρόνο ίσο με:

t1 = .

Άρα η κίνηση εξόδου θα φαίνεται από το ακόλουθο διάγραμμα:

Για τα πρώτα 100 δευτερόλεπτα ο κουβάς γεμίζει με ρυθμό 1 MByte/s , και μετα ενώ έχει μέσα δεδομένα όγκου 100ΜΒ, για τα επόμενα 40 δευτερόλεπτα αδειάζει με ρυθμό 2.5MByte/s.

(μέγιστη τιµή = 100ΜB την χρονική στιγμή t=100 msec)

γ) Αν η χωρητικότητα του κουβά είναι ίση με 200ΜΒ, η μεγαλύτερη χρονική διάρκεια της ριπής (burst) από τον υπολογιστή, έστω t2, ώστε να μην συμβεί απώλεια δεδομένων πρέπει να είναι:

200MB = 1ΜBps \* t2 => t2 = 200 sec είναι η μεγαλύτερη χρονική διάρκεια της ριπής.

**Άσκηση 2**

Ρυθμός παραγωγής κουπονιών = 10 Mbytes/sec

µέγιστος ρυθμός µμετάδοσης δεδοµένων = 50 Mbytes/sec

χωρητικότητα = 1 Mbyte = 8 Mbits

α) Η μέγιστη διάρκεια της ριπής εξόδου όταν ο κουβάς είναι αρχικά γεμάτος είναι:

S = χωρητικότητα / (ρυθµός µετάδοσης δεδοµένων − Ρυθµός παραγωγής κουπονιών) =>

S = 1 MB / (50MBps - 10MBps) = 1/40 = 0.025 sec = 25msec

β)

Αρχικά ο κουβάς είναι γεμάτος με συνολικό όγκο δεδομένων 8 Mbits.

Για τη ριπή εισόδου διάρκειας 40 msec, μόνο τα πρώτα 25 msec μπορεί να εξέρχεται όγκος δεδομένων με ρυθμό 50 Mbytes/sec, δηλαδή την ταχύτητα του δικτύου, δηλαδή όσο διαρκεί η μέγιστη διάρκεια της ριπής εξόδου. Δηλαδή τα πρώτα 25 msec συνολικά θα εξέλθει όγκος δεδομένων ίσος με: 50 Mbytes/sec \* 25msec =

Τα υπόλοιπα 15 msec (40msec - 25msec) θα εξέρχονται τα δεδομένα με ρυθμό ίσο με τον ρυθμό παραγωγής κουπονιών, δηλαδή 10 Mbytes/sec = 80 Mbits/sec. Δηλαδή συνολικά o όγκος δεδομένων θα είναι ίσος με:

(δεδομένων συνολικά φεύγουν με την ριπή των 40msec)

To προφίλ κίνησης εξόδου και περιεχομένου του κουβά είναι το εξής:

