ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

ΟΝΟΜΑΤΕΠΩΝΥΜΟ: ΓΚΟΥΜΕ ΛΑΟΥΡΕΝΤΙΑΝ
ΑΜ: 031 18 014
ΕΞΑΜΗΝΟ: 7^{ο}
ΟΜΑΔΑ: 4
MAC ADDRESS: B4-69-21-1B-6C-FF
IPv4: Άσκ1: 10.3.20.27, Άσκ2: 10.3.20.27, Άσκ3 και 4: 10.3.20.21
ΌΝΟΜΑ ΥΠΟΛΟΓΙΣΤΗ: LAPTOP-B2DVAJKK
ΛΕΙΤΟΥΡΓΙΚΟ ΣΥΣΤΗΜΑ: WINDOWS 10

ΔΙΚΤΥΑ ΥΠΟΛΟΓΙΣΤΩΝ

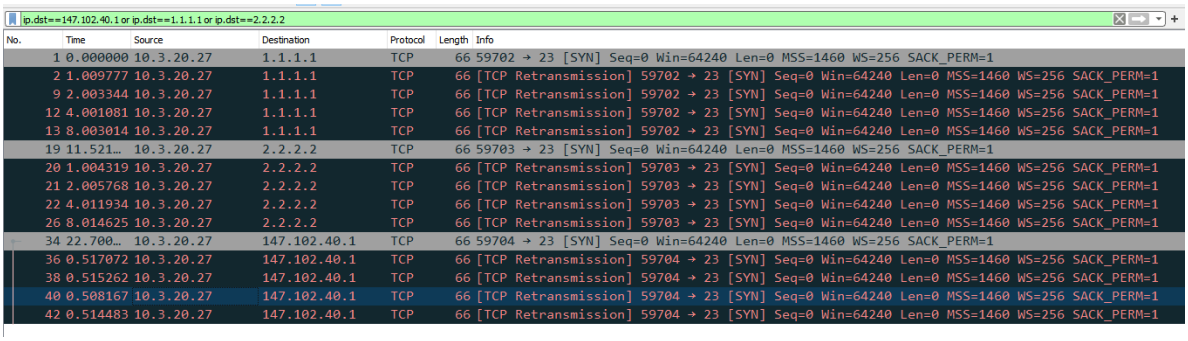
ΕΡΓΑΣΤΗΡΙΑΚΗ ΑΣΚΗΣΗ 7: ΠΡΩΤΟΚΟΛΛΑ TCP ΚΑΙ UDP

Άσκηση 1: Μετάδοση δεδομένων με TCP

1.1) Φίλτρο σύλληψης: **host 10.3.20.27**. Τα αποτελέσματα του command line φαίνονται παρακάτω:

```
C:\Users\Άλεξ>telnet 1.1.1.1
Σύνδεση με 1.1.1.1...Δεν ήταν δυνατό το άνοιγμα σύνδεσης με τον κεντρικό υπολογιστή, στη θύρα 23: Η σύνδεση απέτυχε

C:\Users\Άλεξ>telnet 2.2.2.2
Σύνδεση με 2.2.2.2...Δεν ήταν δυνατό το άνοιγμα σύνδεσης με τον κεντρικό υπολογιστή, στη θύρα 23: Η σύνδεση απέτυχε

C:\Users\Άλεξ>telnet 147.102.40.1
Σύνδεση με 147.102.40.1...Δεν ήταν δυνατό το άνοιγμα σύνδεσης με τον κεντρικό υπολογιστή, στη θύρα 23: Η σύνδεση απέτυχε
```

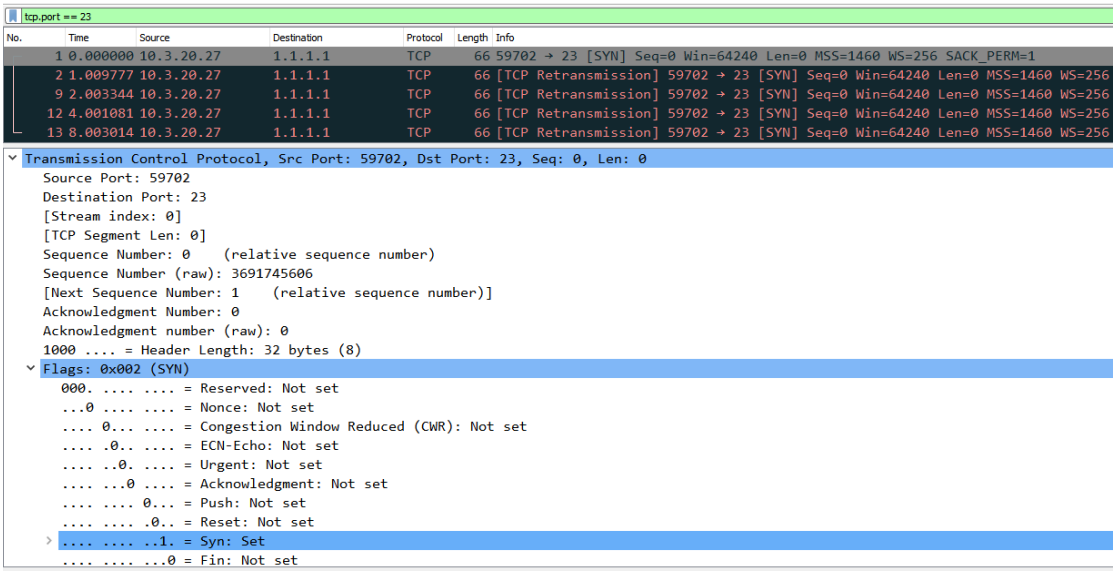
1.2) Φίλτρο απεικόνισης: **ip.dst==1.1.1.1 or ip.dst==2.2.2.2 or ip.dst=147.102.40.1** και βλέπουμε τα εξής:



1.3) Επιλέγοντας οποιοδήποτε από τα παραπάνω πακέτα, βλέπουμε στο TCP Layer το πεδίο **Destination Port: 23**, το οποίο αντιστοιχεί στο πρωτόκολλο **Telnet**.

1.4) Φίλτρο απεικόνισης: **tcp.port==23**.

1.5) Επιλέγοντας το πρώτο πακέτο TCP, παρατηρούμε πως το flag που είναι set για την έναρξη της επικοινωνίας είναι το **SYN**.



1.6) Όπως φαίνεται και στο 1.2, γίνονται **5 προσπάθειες σύνδεσης για κάθε μία από τις προσπάθειες Α και Β (10 συνολικά)**, εκ των οποίων η πρώτη αποτελεί την προσπάθεια εκκίνησης εγκατάστασης και οι υπόλοιπες 4 επαναμετάδοση.

1.7) Στο στιγμιότυπο του 1.2 μπορούμε να δούμε πως οι προσπάθειες γίνονται σε **χρονικές στιγμές που είναι δυνάμεις του 2**. Συγκεκριμένα, αν η πρώτη γίνεται τη στιγμή 0 (εκκίνηση εγκατάστασης), τότε έχουμε επαναμετάδοση τις στιγμές $1=2^0$, $2=2^1$, $4=2^2$, $8=2^3$.

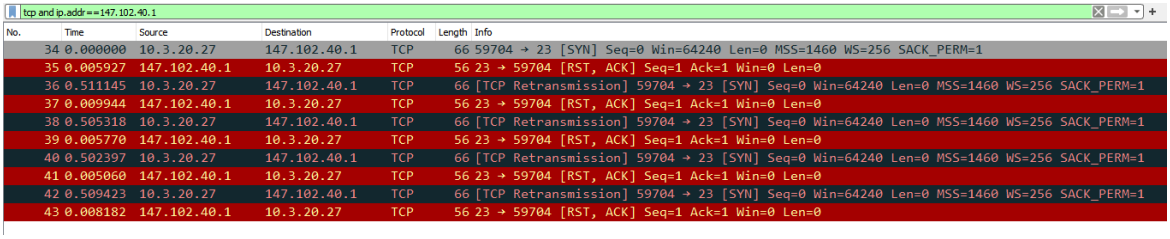
1.8) Δε παρατηρείται κάποια **ουσιαστική διαφορά**.

1.9) Στις προσπάθειες Α και Β παρατηρείται μόνο η προσπάθεια εκκίνησης εγκατάστασης (αποστολή SYN από τον client), ενώ στην περίπτωση Γ λαμβάνουμε επιπλέον από τον σέρβερ τα flags ACK και RST, με τα οποία γνωστοποιεί πως έλαβε το SYN και απορρίπτει τη σύνδεση. Οπότε παρατηρήσαμε το **πρώτο βήμα κυρίως**.

1.10) Σε κανένα τεμάχιο δε παρατηρήθηκε η σημαία FIN ως set, οπότε απλά **εγκαταλείπει την προσπάθεια**.

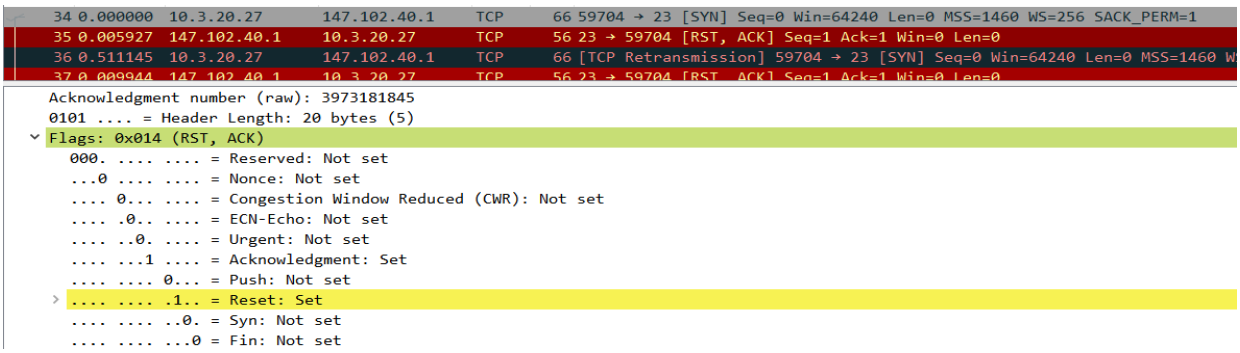
1.11) Φίλτρο απεικόνισης: **tcp and ip.dst==147.102.40.1**.

1.12) Γίνονται και εδώ συνολικά **5 προσπάθειες**.



1.13) Η διαφορά με τις περιπτώσεις Α και Β είναι πως ε**δώ λαμβάνουμε απόκριση από τον server, η οποία έχει τα flags ACK και RST ενεργοποιημένα**. Αυτό σημαίνει πως ο συγκεκριμένος σέρβερ υπάρχει, αλλά απορρίπτει τέτοιου είδους συνδέσεις όπως και περιμέναμε.

1.14) Επιλέγοντας το πακέτο 35 (πρώτη απάντηση TCP από τον 147.102.40.1), παρατηρούμε τα εξής flags μήκους 1 bit: **Reserved, Nonce, CWR, ECN-Echo, Urgent, Acknowledgment, Push, Reset, Syn και Fin:**



1.15) Η σημαία **Reset: Set** δηλώνει άρνηση εγκατάστασης.

1.16) Το ανωτέρω τεμάχιο αποτελείται από **20 bytes επικεφαλίδας και 0 bytes δεδομένων**.

1.17) Για την επικεφαλίδα του TCP τεμαχίου αυτού έχουμε:

Source Port (16 bits)		**Destination Port (16 bits)**
Sequence Number (32 bits)		
Acknowledgment Number (32 bits)		
Header Length (4 bits)	**Flags (12 bits)**	**Window (16 bits)**
Checksum (16 bits)		**Urgent Pointer (16 bits)**

1.18) Σύμφωνα με την ιστοσελίδα, το μέγεθος της επικεφαλίδας TCP προσδιορίζεται από το πεδίο **Data Offset** το οποίο όπως λέει μας δίνει την τιμή σε λέξεις των 32 bits. Αντιθέτως, στο Wireshark το πεδίο αυτό ονομάζεται **Header Length** και για το συγκεκριμένο πακέτο έχει τιμή $0101 = 5_{10}$, το οποίο μας δίνει 20 bytes, όσα δηλαδή βρήκαμε προηγουμένως.

1.19) Όπως αναφέραμε, το πεδίο Header Length έχει στα περιεχόμενα του τεμαχίου την τιμή $5_{16} = 0101 = 5_{10}$ οπότε αφού **μετράει σε λέξεις των 32 bits** (4 bytes), το γινόμενο 4 * 5 μας δίνει 20 bytes.

1.20) Δεν υπάρχει πεδίο το οποίο να μας πληροφορεί για το συνολικό μήκος του τεμαχίου.

1.21) Το μήκος του τεμαχίου TCP μπορεί να βρεθεί εάν α**πό το συνολικό μήκος του IPv4 πακέτου (πεδίο Total Length) αφαιρέσουμε το μήκος της IPv4 επικεφαλίδας (πεδίο Header Length)**. Εάν θέλουμε να βρούμε τα δεδομένα του TCP segment, θα πρέπει από την τιμή που προέκυψε να αφαιρέσουμε την τιμή του TCP Header Length. (*Σημείωση: Τα παραπάνω έχουν τιμές για λέξεις των 32 bits).*

1.22) Το μήκος επικεφαλίδας του πρώτου τεμαχίου TCP που στέλνει ο υπολογιστής μας στο 147.102.40.1 για την εγκατάσταση της TCP σύνδεσης είναι **32 bytes**.

1.23) Παρατηρούμε πως υπάρχει διαφορά στο μήκος των παραπάνω 2 τεμαχίων κατά 12 bytes, η οποία και οφείλεται στο πεδίο **Options το οποίο δεν υπήρχε στην απάντηση από τον 147.102.40.1, ενώ καταλαμβάνει 12 bytes στο πρώτο TCP τεμάχιο που εμείς αποστέλουμε**. Να σημειωθεί πως στη συγκεκριμένη περίπτωση προέκυψε μέγεθος επικεφαλίδας 32 bytes, το οποίο είναι ακέραιο πολλαπλάσιο μιας λέξης (32 bits). Εάν προέκυπτε διαφορετικά, θα υπήρχε TCP Padding προκειμένου να γίνει το μήκος ακέραιο πολλαπλάσιο των 4 bytes.

Άσκηση 2:

2.1) Φίλτρο σύλληψης: **(tcp) and (ip host edu-dy.cn.ntua.gr)**.

2.2) Για την έναρξη της επικοινωνίας προσπαθεί να συνδεθεί στη **θύρα 21**, η οποία και αντιστοιχεί στο πρωτόκολλο FTP ελέγχου.

2.3) Αντίστοιχα για τη σύνδεση μεταφοράς δεδομένων, συνδέεται στη **θύρα 20**, η οποία και αντιστοιχεί στο πρωτόκολλο FTP μεταφοράς δεδομένων.

2.4) Φίλτρο απεικόνισης: **tcp.port==21**.

2.5) Όπως ήταν αναμενόμενο και φαίνεται παρακάτω, **ανταλλάσσονται 3 πακέτα**.

tcp.port==21

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	10.3.20.27	147.102.40.15	TCP	66	58244 → 21 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=1 SACK_PERM=1
2	0.004804	147.102.40.15	10.3.20.27	TCP	66	21 → 58244 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=536 WS=64 SACK_PERM=1
3	0.004913	10.3.20.27	147.102.40.15	TCP	54	58244 → 21 [ACK] Seq=1 Ack=1 Win=8192 Len=0

2.6) Αρχικά η σημαία SYN στο πρώτο τεμάχιο από εμάς προς τον σέρβερ, στη συνέχεια οι σημαίες SYN και ACK κατά την απόκριση του σέρβερ και τέλος η σημαία ACK από εμάς προς τον σέρβερ, οπότε γενικά χρησιμοποιούνται οι **σημαίες SYN και ACK**.

2.7) Το μέγεθος των επικεφαλίδων TCP των παραπάνω τεμαχίων είναι **32, 32 και 20 bytes αντίστοιχα**.

2.8) Το **μέγεθος δεδομένων των τεμαχίων αυτών είναι μηδενικό**, καθώς όπως γνωρίζουμε κατά την τριπλή χειραψία δεν ανταλλάσσονται δεδομένα.

2.9) Από το στιγμιότυπο στο 2.5, παρατηρούμε πως η τριπλή χειραψία διαρκεί **0.004913 seconds**.

2.10) Επιλέγοντας 1 εκ των 2 ACK τεμάχια -αφού αυτά διαθέτουν το [SEQ/ACK analysis] πεδίο- βλέπουμε πως το iRTT συμφωνεί με τον παραπάνω χρόνο.

```
▾ [SEQ/ACK analysis]
    [This is an ACK to the segment in frame: 2]
    [The RTT to ACK the segment was: 0.000109000 seconds]
    [iRTT: 0.004913000 seconds]
```

2.11) Η δική μας πλευρά ανακοινώνει τον σχετικό/απόλυτο **αριθμό σειράς 0/2242961906**, ενώ η πλευρά του σέρβερ τον **σχετικό/απόλυτο αριθμό σειράς 0/260366001**.

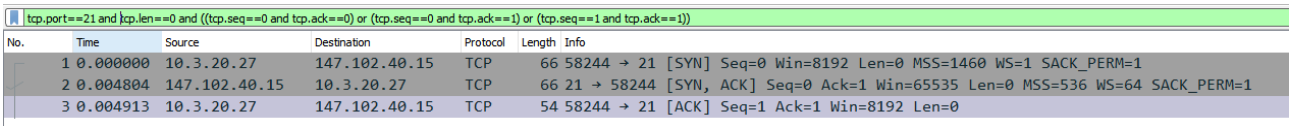
2.12) Παρατηρώντας το τεμάχιο TCP με το οποίο ο ftp server δηλώνει πως αποδέχεται τη σύνδεση, βλέπουμε το **relative ACK number να είναι 1**. Προκύπτει μια τέτοια τιμή, καθώς **ταυτίζεται με το Next Sequence Number** και αφού πριν είχαμε ως Sequence Number το 0, επόμενη τιμή είναι το 1. Σε απόλυτες τιμές, προκύπτει ως η τιμή του Sequence Number που έλαβε αυξημένη κατά 1.

2.13) Αναφορικά με το 3^{ο} τεμάχιο της τριπλής χειραψίας, το raw Sequence Number του είναι το **raw Acknowledgment Number του προηγούμενου τεμαχίου**, ενώ το raw Acknowledgment Number του είναι το r**aw Sequence Number του προηγούμενου τεμαχίου αυξημένο κατά 1**.

2.14) Το μήκος δεδομένων των τριών τεμαχίων της τριπλής χειραψίας είναι **μηδενικό**, αφού όπως γνωρίζουμε δεν ανταλλάσσονται δεδομένα κατά τη διαδικασία αυτή.

2.15) Παρατηρώντας στο παράθυρο λεπτομερειών βλέπουμε πως τα πεδία Sequence Number και Acknowledgment Number καταλαμβάνουν 4 bytes, επομένως η μέγιστη τιμή που μπορούν να λάβουν είναι **2^{32} -1 = 4.294.967.295**.

2.16) Προκειμένου να δούμε τα τεμάχια τριπλής χειραψίας σχετικά με τη θύρα ελέγχου FTP εφαρμόζουμε το φίλτρο:
tcp.len==0 and ((tcp.seq==0 and tcp.ack==0) or (tcp.seq==0 and tcp.ack==1) or (tcp.seq==1 and tcp.ack==1))
και βλέπουμε τα παρακάτω τεμάχια:



Εάν θέλαμε να δούμε την τριπλή χειραψία για την εγκατάσταση μεταφοράς δεδομένων τότε θα βάζαμε tcp.port==20 στην αρχή, ενώ αν θέλαμε να δούμε και τις 2 τριπλέτες δε θα βάζαμε καθόλου το **tcp.port==** στην αρχή του φίλτρου.

2.17) Από το παραπάνω στιγμιότυπο, βλέπουμε στο πρώτο τεμάχιο την τιμή **Win=8.192** και στο δεύτερο **Win=65.535**, τα οποία αντιστοιχούν στο μέγεθος παραθύρου λήψης που ανακοινώνει ο υπολογιστής μας και ο σερβερ αντίστοιχα.

2.18) Η σχετική πληροφορία υπάρχει στο πεδίο **Window** της TCP επικεφαλίδας.

2.19) Αναφερόμενοι στα 3 παραπάνω πακέτα **η μικρότερη τιμή** παραθύρου είναι **8.192, ενώ η μεγαλύτερη 65.535**. Στη γενική περίπτωση, βλέπουμε πως το πεδίο Window καταλαμβάνει 2 bytes, επομένως μπορεί να λάβει μέγιστη τιμή 2^{16}-1 = **65.535**, ενώ η ελάχιστη τιμή που μπορεί να λάβει είναι **0** όταν δεν υπάρχει διαθέσιμος χώρος στον buffer τη στιγμή που στέλνεται το τεμάχιο.

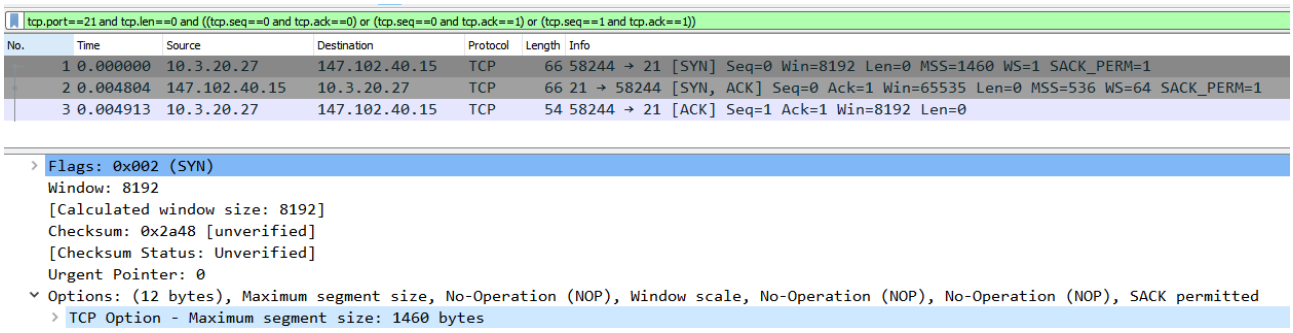
2.20) Από το στιγμιότυπο του ερωτήματος 2.16 βλέπουμε πως ο υπολογιστής μας ανακοινώνει **MSS = 1460 bytes**.

2.21) Ανοίγοντας το τερματικό μας κάνουμε non-fragmented ping στο www.google.com αρχικά για icmp payload 1473 bytes, το οποίο θα μας έδινε MTU = 1473 + 20 (IP header) + 8 (ICMP header). Λαμβάνουμε στην περίπτωση αυτή μήνυμα σφάλματος. Δοκιμάζοντας, ωστόσο για icmp payload 1472 bytes η αποστολή γίνεται κανονικά (ισοδυναμεί με MTU 1500 bytes), όπως βλέπουμε παρακάτω:

```
C:\Users\Άλεξ>ping -n 1 -f -l 1473 www.google.com

Pinging www.google.com [216.58.205.68] with 1473 bytes of data:
Packet needs to be fragmented but DF set.

Ping statistics for 216.58.205.68:
    Packets: Sent = 1, Received = 0, Lost = 1 (100% loss),

C:\Users\Άλεξ>ping -n 1 -f -l 1472 www.google.com

Pinging www.google.com [216.58.205.68] with 1472 bytes of data:
Reply from 216.58.205.68: bytes=68 (sent 1472) time=29ms TTL=119

Ping statistics for 216.58.205.68:
    Packets: Sent = 1, Received = 1, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 29ms, Maximum = 29ms, Average = 29ms
```

Άρα η διεπαφή του υπολογιστή μας έχει MTU 1500 bytes. Επανερχόμενοι στην αρχική καταγραφή μας, το **MSS = 1460 bytes προκύπτει από το MTU 1500 bytes εάν αφαιρέσουμε το IP header (20 bytes) και το ελάχιστο μέγεθος ενός TCP header (20 bytes)**.

2.22) Η τιμή MSS βρίσκεται στο υποπεδίο **TCP-Option – Maximum segment size του πεδίου Options της TCP επικεφαλίδας**.



2.23) Στο παραπάνω στιγμιότυπο, βλέπουμε πως ο edu-dy.cn.ntua.gr ανακοινώνει **MSS = 536 bytes**.

2.24) Η τιμή MSS αυτή, προκύπτει **αφαιρώντας 40 bytes από την τιμή της MTU = 576** για τους ίδιους λόγους που εξηγήσαμε προηγουμένως.

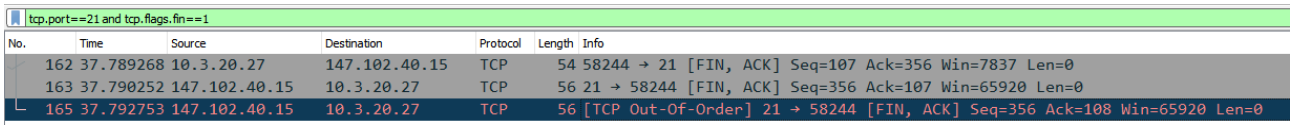
2.25) Λόγω του παραπάνω περιορισμού, το μεγαλύτερο τεμάχιο TCP που μπορεί να στείλει ο υπολογιστής μας (επικεφαλίδα + δεδομένα) είναι **556 bytes**.

2.26) Η σημαία **FIN**.

2.27) Φίλτρο απεικόνισης: **tcp.port==21 and tcp.flags.fin==1**.

2.28) Ο υπολογιστής μας εκκινεί τη διαδικασία απόλυσης σύνδεσης.

2.29) Ανταλλάσσονται **3 TCP τεμάχια**. (Λόγω Wi-Fi το τελευταίο πακέτο λήφθηκε ως Out-Of-Order)



2.30) Και οι 3 TCP επικεφαλίδες έχουν **μέγεθος 20 bytes**.

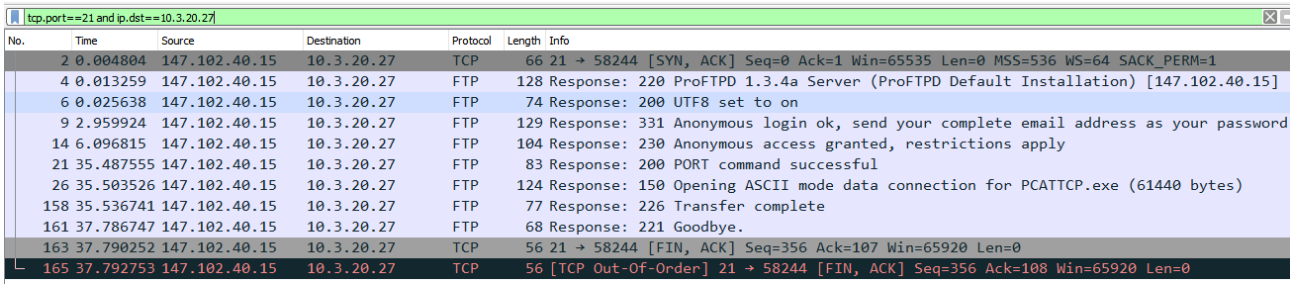
2.31) Έχουν **μηδενικό μέγεθος δεδομένων**.

2.32) Επιλέγοντας το πακέτο 162 με το οποίο ο υπολογιστής μας εκκινεί την απόλυση της σύνδεσης βλέπουμε στην IP επικεφαλίδα το πεδίο Total Length να έχει τιμή 40. Η τιμή αυτή προκύπτει **ως το άθροισμα της IP επικεφαλίδας -το οποίο από το πεδίο Header Length του IP βλέπουμε πως είναι ίσο με 20- με το συνολικό μέγεθος του TCP τεμαχίου, το οποίο είδαμε πως δεν έχει δεδομένα παρά μόνο 20 bytes επικεφαλίδας**.

2.33) Ακριβώς **όμοια με το 2.29** προκύπτει το μήκος του πακέτου IPv4, το οποίο μεταφέρει το αντίστοιχο τεμάχιο από το edu-dy.cn.ntua.gr.

2.34) Η πλευρά του σέρβερ μετέδωσε **<u>συνολικά</u> 965 bytes**, ενώ ο υπολογιστής μας **874 bytes.**

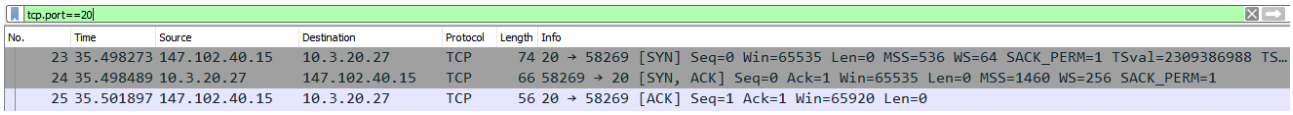
2.35) Για να βρούμε π.χ. όσα μας έστειλε ο σέρβερ εφαρμόσαμε το φίλτρο **tcp.port==21 and ip.dst==10.3.20.27** και αθροίσαμε τα bytes της στήλης Length.



Για να βρούμε το αντίστροφο, αλλάξαμε το φίλτρο σε **tcp.port==21 and ip.dst==147.102.40.15** και ακολουθήσαμε την ίδια διαδικασία.

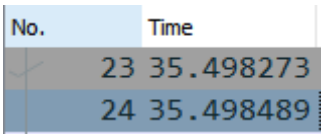
2.36) Φίλτρο απεικόνισης **tcp.port==20**.

2.37) Όπως βλέπουμε και στο παρακάτω στιγμιότυπο, ανακοινώνονται τιμές **MSS 536 και 1460** από την πλευρά του edu-dy.cn.ntua.gr και τη δικιά μας αντίστοιχα.



2.38) Το μέγεθος του μεγαλύτερου τεμαχίου TCP που μπορεί να στείλει ο σέρβερ στον υπολογιστή μας ανέρχεται σε **556 bytes**, 536 του payload (MSU) και επιπλέον 20 της επικεφαλίδας TCP.

2.39) Η διαφορά στον χρόνο που καταγράφονται τα πρώτα 2 τεμάχια, η οποία και αντιστοιχεί στο RTT είναι **0.000216 seconds**.



2.40) Ο υπολογιστής μας δε στέλνει ACK για κάθε πακέτο που λαμβάνει. Συγκεκριμένα, κατά χρονική σειρά στέλνει ανά **4, 6, 6, 8, 8, 10, 7, 3, 12, 14, 14, 16 και 6 πακέτα**.

2.41) Κατά τη μεταφορά του αρχείου, οι τιμές παραθύρου που ανακοινώνει ο υπολογιστής μας παραμένουν **αμετάβλητες και ίσες με 512 bytes**. Το ότι δεν αλλάζουν οι τιμές, σημαίνει πως δεν υπάρχει υπερφόρτωση του buffer του υπολογιστή μας.

2.42) Συνολικά είναι **590 bytes**. Όσον αφορά τις επικεφαλίδες, έχουν μέγεθος 14/20/20 για τα Ethernet/IP/TCP πρωτόκολλα αντίστοιχα.

2.43) Στο πεδίο Total Length της IP επικεφαλίδας βλέπουμε την τιμή 576 bytes. Δεδομένου ότι οι επικεφαλίδες IP και TCP είναι αθροιστικά 40 bytes, το μέγεθος των δεδομένων του TCP τεμαχίου ανέρχεται σε 536 bytes, το **οποίο συμφωνεί με το 2.38**.

2.44) Διαβάζοντας το documentation βλέπουμε πως **υπό κανονικές συνθήκες δε θα στέλνονταν ποτέ δεδομένα μεγαλύτερα από την παραπάνω τιμή**. Υπάρχει περίπτωση μόνο να σταλούν δεδομένα μεγαλύτερα από αυτά που μπορεί να διαχειριστεί το ενδιάμεσο δίκτυο που μεσολαβεί των 2 κόμβων, οπότε και εκεί να

πρέπει να γίνει fragmentation από τον αποστολέα των πακέτων και **να αποφευχθεί να γίνει από τους δρομολογητές**.

2.45) Εφαρμόζουμε ξανά το φίλτρο απεικόνισης tcp.port==20. Επιλέγουμε το πρώτο πακέτο TCP που έστειλε ο υπολογιστής μας και βλέπουμε το Sequence Number (raw) του, το οποίο είναι: 1842084077. Στη συνέχεια επιλέγουμε το τελευταίο πακέτο TCP που έστειλε ο υπολογιστής μας, το οποίο έχει τιμή Sequence Number (raw): 1842084078, επομένως κατά τη σύνδεση δεδομένων ο υπολογιστής μας έστειλε **1 μόνο byte** δεδομένων. Εφαρμόζοντας την ίδια μέθοδο αντίστροφα, βρίσκουμε πως ο server μας έστειλε **61.616 bytes**. Οι τιμές αυτές είναι ουσιαστικά η τιμή του πεδίου **Sequence Number (relative)** για τον κάθε host.

2.46) Εφαρμόζουμε ξανά το φίλτρο **ftp-data**. Από εκεί βρίσκουμε τη διαφορά χρόνου καταγραφής μεταξύ τελευταίου και πρώτου πακέτου, η οποία είναι 0.27675 sec. Εφόσον προηγουμένως βρήκαμε πόσα data μας έστειλε, ο ρυθμός μετάδοσης που μας τα έστειλε είναι (61.616 bytes / 0.27675 sec) = 222,641 Kbytes/sec.

2.47) Δεν εντοπίστηκε κάποιο Retransmission TCP πακέτο, επομένως **δεν υπήρξαν αναμεταδόσεις τεμαχίων**.

Άσκηση 3: Αποφυγή συμφόρησης στο TCP

3.1) Φίλτρο απεικόνισης: **tcp.port==20**.

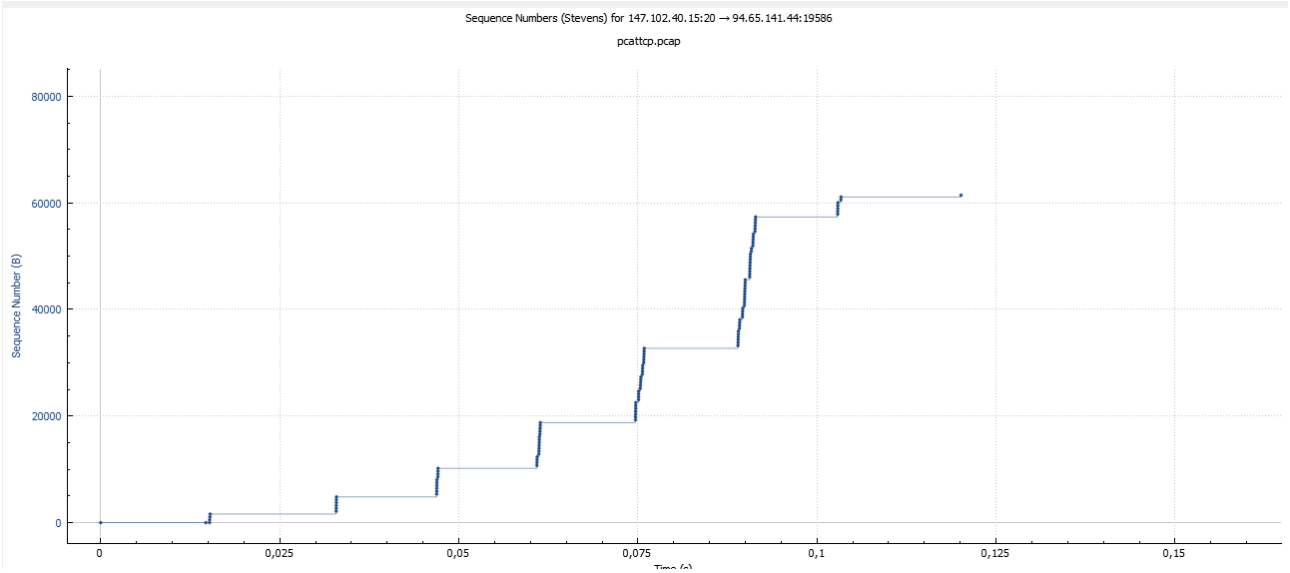
3.2) Παρακάτω βλέπουμε τα πακέτα της τριμερούς χειραψίας, επομένως η IP του υπολογιστή που κατέβασε το PCATTCP.exe είναι **94.65.141.44**.

No.	Time	Source	Destination	Protocol	Length	Info
20	29.690533	147.102.40.15	94.65.141.44	TCP	74	20 → 19586 [SYN] Seq=0 Win=65535 Len=0 MSS=536 WS=64 SACK_PERM=1 TSval=32103835 TSec…
21	29.705159	94.65.141.44	147.102.40.15	TCP	66	19586 → 20 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1452 WS=256 SACK_PERM=1
22	29.705207	147.102.40.15	94.65.141.44	TCP	54	20 → 19586 [ACK] Seq=1 Ack=1 Win=65920 Len=0

3.3) Εάν επιλέξουμε το 2^{ο} πακέτο από τα παραπάνω, βλέπουμε στο TCP header και στο ψευδο-υποπεδίο [Timestamps] ότι το RTT είναι **0.014626s** χρόνος τάξεις μεγέθους μεγαλύτερος σε σχέση με αυτόν του 2.39.

```
▾ [Timestamps]
    [Time since first frame in this TCP stream: 0.014626000 seconds]
    [Time since previous frame in this TCP stream: 0.014626000 seconds]
```

3.4) Παρατηρούμε την εξής κλιμακωτή μορφή:



Με εξαίρεση τα 2 πρώτα TCP πακέτα, τα οποία είναι μέρος της τριπλής χειραψίας, παρατηρούμε πως τα FTP data στέλνονται σε “ριπές” πακέτων. **Δηλαδή στέλνονται πολλά μαζί το ένα πίσω από το άλλο ανά τακτά χρονικά διαστήματα, κάθε φορά περισσότερα σε πλήθος από την προηγούμενη στη γενική περίπτωση** (προφανώς στο τέλος το “σκαλοπάτι” είναι μικρότερο, καθώς στάλθηκαν όλα τα δεδομένα εκεί).

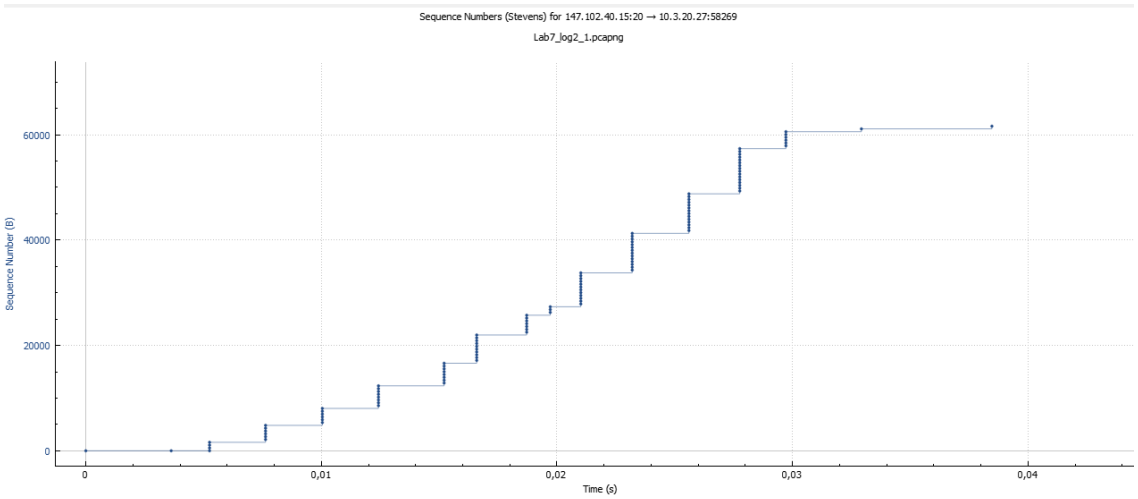
3.5) Στην παράγραφο 3.1 του RFC 5681 διαβάζουμε το εξής:

```
IW, the initial value of cwnd, MUST be set using the following
guidelines as an upper bound.

If SMSS > 2190 bytes:
    IW = 2 * SMSS bytes and MUST NOT be more than 2 segments
If (SMSS > 1095 bytes) and (SMSS <= 2190 bytes):
    IW = 3 * SMSS bytes and MUST NOT be more than 3 segments
if SMSS <= 1095 bytes:
    IW = 4 * SMSS bytes and MUST NOT be more than 4 segments
```

Όπου SMSS = Sender MSS και IW η αρχική τιμή του cwnd (congestion window, ένα όριο από τη μεριά του αποστολέα για τα δεδομένα που μπορεί να στείλει σε ένα δίκτυο πριν λάβει ACK). Παρατηρούμε από το παράθυρο Sequence Numbers, 4 **πακέτα κατά το πρώτο RTT**, το οποίο συμφωνεί πλήρως με τον τρίτο περιορισμό, ότι δηλαδή μπορούν να σταλούν μέχρι και 4 Segments (δεδομένου ότι ο edu-dy.cn.ntua.gr βλέπουμε πως έχει MSS 536 bytes).

3.6) Στο **δεύτερο RTT έστειλε 6 τεμάχια, ενώ στο τρίτο 10 τεμάχια**. Αυτό που στην πραγματικότητα συμβαίνει είναι πως σε κάθε RTT αποστολής δεδομένων στέλνονται ολοένα και περισσότερα ACK σχεδόν ταυτόχρονα δίνοντάς μας την ψευδαίσθηση ότι στέλνονται σε μία ριπή περισσότερα πακέτα, ενώ στην πραγματικότητα η ριπή αυτή αποτελείται από αυξανόμενο αριθμό μικρότερων. Αυτό επιβεβαιώνει και τον Slow Start αλγόριθμο, καθώς ξεκινάει με μικρό ρυθμό μετάδοσης για να δοκιμάσει το δίκτυο και αυξάνει σταδιακά.

3.7) Ενεργοποιούμε το φίλτρο απεικόνισης tcp.port==20 στην καταγραφή μας. Βλέπουμε το εξής διάγραμμα αριθμών σειράς συναρτήσει του χρόνου από τον edu-dy.cn.ntua.gr προς τον υπολογιστή μας.



Συγκρίνοντας τα παραπάνω διαγράμματα, βλέπουμε ότι **ποιοτικά είναι ίδια**. Στο πρώτο RTT βλέπουμε μετάδοση 4 FTP πακέτων, ενώ στο 2ο και 3ο από 6 πακέτα. Σε αντίθεση με πριν, όμως, **ο υπολογιστής μας δε κάνει πολλαπλά ACK σχεδόν ταυτόχρονα όπως ο 94.65.141.44**, αλλά 1 κάθε φορά, οπότε και δε παρατηρείται απαραίτητα αύξηση του ρυθμού αποστολής των πακέτων.

Άσκηση 4: Μετάδοση δεδομένων με UDP

Με την προεργασία έχουμε τα εξής στο τερματικό μας:

```
D:\>ipconfig/flushdns

Windows IP Configuration

Successfully flushed the DNS Resolver Cache.

D:\>nslookup edu-dy.cn.ntua.gr
Server:  pdns0.grnet.gr
Address:  62.217.126.164

Non-authoritative answer:
Name:    edu-dy.cn.ece.ntua.gr
Address:  147.102.40.15
Aliases:  edu-dy.cn.ntua.gr
```

4.1) Φίλτρο σύλληψης: **udp**.

4.2) Σχετικά με το πρώτο UDP datagram που έστειλε ο υπολογιστής μας έχουμε τα εξής πεδία:

- **Source Port (2 bytes)**
- **Destination Port (2 bytes)**
- **Length (2 bytes)**

- **Checksum (2 bytes)**

4.3) Το **συνολικό μήκος της επικεφαλίδας UDP είναι 8 bytes**.

4.4) Το δεδομενόγραμμα αυτό είναι ενθυλακωμένο σε ένα πακέτο IPv4 και έχει συνολικό μέγεθος **53 bytes**.

4.5) Το πεδίο Length εκφράζει το **συνολικό μέγεθος του UDP datagram**.

4.6) Προφανώς το **ελάχιστο μέγεθος** δεδομενογραμμάτων UDP που μπορεί να μεταφερθεί από ένα πακέτο IPv4 είναι **8 bytes** και αυτό συμβαίνει όταν το datagram δε μεταφέρει δεδομένα παρά μόνο την επικεφαλίδα. Σχετικά με τη μέγιστη τιμή, είδαμε πως το πεδίο Length παραπάνω αφορά το συνολικό μήκος και ότι είναι 2 bytes = 16bits, επομένως, θα περιμέναμε η μέγιστη τιμή που μπορεί να λάβει είναι $2^{16} - 1 = 65.535$ bytes. Ωστόσο, στο σημείο αυτό πρέπει να θυμηθούμε πως το πεδίο Total Length της IPv4 επικεφαλίδας είναι επίσης 2 bytes, με μέγιστη τιμή 65.535 bytes, επομένως η **μέγιστη τιμή που μπορεί να λάβει ένα UDP Datagram είναι 65.535 – 20 = 65.515 bytes**, όπου 20 bytes το ελάχιστο μέγεθος μιας IP επικεφαλίδας.

4.7) Το πεδίο Header Length ενός IPv4 πακέτου αποτελείται από 4 bits, επομένως παίρνει μέγιστη τιμή 15_{10} και δεδομένου ότι μετράει το μέγεθος σε λέξεις των 4 bytes, το μέγιστο IPv4 header είναι 60 bytes. Επομένως, προκειμένου ένα πακέτο UDP να σταλεί/παραληφθεί με βεβαιότητα πρέπει να έχει συνολικό μήκος μέχρι και (576-60) = **516 bytes**.

4.8) Δε παρατηρήθηκαν πρωτόκολλα πέραν του DNS.

4.9) Φίλτρο απεικόνισης: **dns**.

4.10) Η IPv4 διεύθυνση του DNS server που χρησιμοποιήθηκε είναι **62.217.126.164**.

4.11) Θύρα προέλευσης/προορισμού για την ερώτηση στον DNS Server: **50.972/53**.

4.12) Θύρα προέλευσης/προορισμού για την απόκριση του DNS Server: **53/50.972**. Εδώ να σημειωθεί πως τα query/response για την IP του edu-dy.cn.ntua.gr είναι τα πακέτα 3 και 4 αντίστοιχα παρακάτω και όχι τα 1, 2.

dns

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	10.3.20.21	62.217.126.164	DNS	87	Standard query 0x0001 PTR 164.126.217.62.in-addr.arpa
2	0.035512	62.217.126.164	10.3.20.21	DNS	115	Standard query response 0x0001 PTR 164.126.217.62.in-addr.arpa PTR pdns0.grnet.gr
3	0.039700	10.3.20.21	62.217.126.164	DNS	77	Standard query 0x0002 A edu-dy.cn.ntua.gr
4	0.807657	62.217.126.164	10.3.20.21	DNS	121	Standard query response 0x0002 A edu-dy.cn.ntua.gr CNAME edu-dy.cn.ece.ntua.gr A 147.102.…
5	0.811519	10.3.20.21	62.217.126.164	DNS	77	Standard query 0x0003 AAAA edu-dy.cn.ntua.gr
6	0.816068	62.217.126.164	10.3.20.21	DNS	159	Standard query response 0x0003 AAAA edu-dy.cn.ntua.gr CNAME edu-dy.cn.ece.ntua.gr SOA psy…

4.13) Η **θύρα 53** αντιστοιχεί στο πρωτόκολλο **DNS**.