

Εργαστηριακή Άσκηση 7 Πρωτόκολλα TCP και UDP

Θοδωρής Φρίζος Παπαρρηγόπουλος

el18040

Ομάδα 4η.

Ipn6: 2a02:587:4f0e:2b65:34b0:9e4a:d912:2f8a

MAC: 40-1C-83-38-F7-20

Λειτουργικό: Windows

DESKTOP-1403ER3

Άσκηση 1

1.1) host 192.168.1.9

1.2) ip.addr == 1.1.1.1 or ip.addr == 2.2.2.2 or ip.addr == 147.102.40.1

1.3) Port 23

1.4) tcp.port == 23

1.5) Η σημαία SYN

1.6) 5 προσπάθειες

1.7)

tcp.port == 23						
No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	192.168.1.9	1.1.1.1	TCP	66	56116 → 23 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
8	1.008887	192.168.1.9	1.1.1.1	TCP	66	[TCP Retransmission] [TCP Port numbers reused] 56116 → 23 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
15	2.006125	192.168.1.9	1.1.1.1	TCP	66	[TCP Retransmission] [TCP Port numbers reused] 56116 → 23 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
45	4.002383	192.168.1.9	1.1.1.1	TCP	66	[TCP Retransmission] [TCP Port numbers reused] 56116 → 23 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
87	8.003828	192.168.1.9	1.1.1.1	TCP	66	[TCP Retransmission] [TCP Port numbers reused] 56116 → 23 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
184	13.076136	192.168.1.9	2.2.2.2	TCP	66	56151 → 23 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
203	1.007431	192.168.1.9	2.2.2.2	TCP	66	[TCP Retransmission] [TCP Port numbers reused] 56151 → 23 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
204	2.009954	192.168.1.9	2.2.2.2	TCP	66	[TCP Retransmission] [TCP Port numbers reused] 56151 → 23 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
232	4.004606	192.168.1.9	2.2.2.2	TCP	66	[TCP Retransmission] [TCP Port numbers reused] 56151 → 23 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
262	8.006750	192.168.1.9	2.2.2.2	TCP	66	[TCP Retransmission] [TCP Port numbers reused] 56151 → 23 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
322	12.882102	192.168.1.9	147.102.40.1	TCP	66	56196 → 23 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
323	0.014032	147.102.40.1	192.168.1.9	TCP	54	23 → 56196 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
348	0.507728	192.168.1.9	147.102.40.1	TCP	66	[TCP Retransmission] [TCP Port numbers reused] 56196 → 23 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
349	0.012909	147.102.40.1	192.168.1.9	TCP	54	23 → 56196 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
353	1.989525	192.168.1.9	147.102.40.1	TCP	66	[TCP Retransmission] [TCP Port numbers reused] 56196 → 23 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
354	0.013507	147.102.40.1	192.168.1.9	TCP	54	23 → 56196 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
360	3.995733	192.168.1.9	147.102.40.1	TCP	66	[TCP Retransmission] [TCP Port numbers reused] 56196 → 23 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
361	0.012433	147.102.40.1	192.168.1.9	TCP	54	23 → 56196 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
433	7.997949	192.168.1.9	147.102.40.1	TCP	66	[TCP Retransmission] [TCP Port numbers reused] 56196 → 23 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
435	0.011863	147.102.40.1	192.168.1.9	TCP	54	23 → 56196 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0

1.8) Είναι ίδια

1.9) Μόνο το 1ο βήμα καθώς στέλνει ο client SYN segment

1.10) Εγκαταλείπει απλώς, καθώς δεν παρατηρώ κάποιο τεμάχιο FIN

1.11) ip.addr == 147.102.40.1

1.12) 5 προσπάθειες

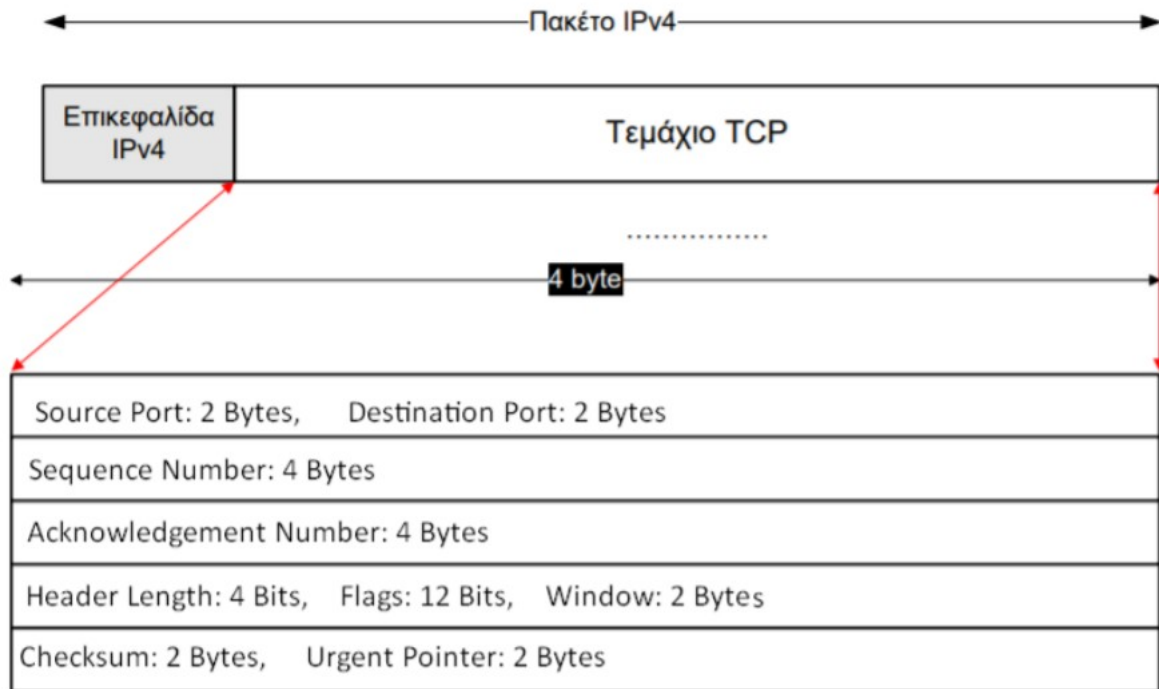
1.13) Λαμβάνουμε για κάθε προσπάθεια απάντηση τύπου RST (reject connection). (Bit Reset στα flags set)

1.14) Ack και Rst

1.15) Η σημαία Reset

1.16) Επικεφαλίδα TCP 20 Bytes και δεδομένα 0 Bytes (γιατί το τεμάχιο στέλνεται για την εγκατάσταση σύνδεσης)

1.17)



1.18) Το πεδίο Data Offset (Header Length στο Wireshark)

1.19) Το μήκος της επικεφαλίδας είναι πολλαπλάσιο των 4 Bytes. Άρα η τιμή 5 που παρατηρούμε στο πακέτο σημαίνει $4 \times 5 = 20$ Bytes

1.20) Δεν υπάρχει

1.21) $\text{Payload} = \text{Total Length (IPv4 Header)} - \text{IPv4 Header Length} - \text{TCP Header Length}$

1.22) Είναι 32 Bytes (τιμή 8 στο Header Length)

1.23) Ναι, αυτό συμβαίνει καθώς ο υπολογιστής μου, προσπαθώντας να εγκαταστήσει σύνδεση προσθέτει options στην επικεφαλίδα για να κάνει συγχρονισμό, ενώ ο server απλά απορρίπτει την σύνδεση και δεν χρειάζεται options.

Ασκηση 2

- 2.1) host edu-dy.cn.ntua.gr and tcp
- 2.2) Θύρα 21
- 2.3) Θύρα 20
- 2.4) tcp.port == 21
- 2.5) Ανταλλάσσονται 3 τεμάχια
- 2.6) Σημαίες: SYN και ACK (SYN, SYN+ACK, ACK)
- 2.7) Τα πρώτα 2 έχουν μέγεθος 32 Bytes (εξαιτίας των Options), ενώ το τελευταίο έχει μέγεθος 20 Bytes
- 2.8) 32 bytes, 32 bytes, 20 bytes
- 2.9) Δεν περιλαμβάνουν δεδομένα

1	0.000000	192.168.1.9	147.102.40.15	TCP	66 61880 → 21 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=1 SACK_PERM=1
2	0.032961	147.102.40.15	192.168.1.9	TCP	66 21 → 61880 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=536 WS=64 SACK_PERM=1
3	0.033163	192.168.1.9	147.102.40.15	TCP	54 61880 → 21 [ACK] Seq=1 Ack=1 Win=8192 Len=0

2.10) Ναι συμφωνεί

▼ [SEQ/ACK analysis]

[This is an ACK to the segment in frame: 2]

[The RTT to ACK the segment was: 0.000202000 seconds]

[iRTT: 0.033163000 seconds]

2.11)

```
Sequence Number: 1 (relative sequence number)
Sequence Number (raw): 4141778900
[Next Sequence Number: 1 (relative sequence number)]
Acknowledgment Number: 1 (relative ack number)
Acknowledgment number (raw): 3558942654
```

- 2.12) Είναι Client Sequence Number + 1
- 2.13) Αρχικό Client Sequence Number + 1 ως νέο Sequence Number και αρχικό Server Sequence Number + 1 ως νέο Acknowledgement Number.
- 2.14) Μηδέν
- 2.15) Αφού τα αντίστοιχα πεδία έχουν μέγεθος 32 bits, τότε μέγιστη τιμή είναι η $2^{32} - 1$
- 2.16) tcp.flags.syn == 1 or tcp.flags.ack == 1) and (tcp.ack_raw == 0 or tcp.ack_raw == 2094594523 or tcp.ack_raw == 1938938557) and tcp.flags.push == 0
- 2.17) Υπολογιστής μου: 8192 Server: 65535
- 2.18) Στο πεδίο Window
- 2.19) Min: 0 (σταματάω τον αποστολέα), Max: 65535
- 2.20) MSS 1460
- 2.21) Είναι MTU – IPv4 Header – TCP Header = 1500 – 20 – 20 (ελάχιστες τιμές επικεφαλίδων).
- 2.22) Πρώτα 4 Bytes του πεδίου Options
- 2.23) 536
- 2.24) Είναι MTU – IPv4 Header – TCP Header = 576 – 20 – 20
- 2.25) 536 + Header TCP = 556 Bytes
- 2.26) Η σημαία FIN (τελευταίο bit στα flags)
- 2.27) tcp.flags.fin == 1
- 2.28) Ο Server
- 2.29) Συνολικά 4 τεμάχια
- 2.30) 20 bytes

2.31) 0

2.32) Είναι $\text{Total Length} = \text{Header IPv4} + \text{Header TCP} + \text{Payload} = 20 + 20 + 0 = 40$

2.33) Το ίδιο

2.34) Μεταδόθηκαν συνολικά 121 από τον υπολογιστή μου (Sequence Number τελευταίου – 1 για το FIN, ACK) και 375 (Sequence Number Τελευταίου – 1 για το FIN – 1 για το ACK) από τον Server.

2.35) Sequence Number του τελευταίου τεμαχίου που στέλνει ο υπολογιστής μου – Sequence Number του πρώτου και αντίστοιχα και για τον Server

2.36) $\text{tcp.port} == 20$

2.37) Server: 535 κι εγώ 1460

2.38) 556 Bytes (536 + TCP Header)

2.39) 0.000139sec

2.40) Κατά μέσο όρο κάθε 3 τεμάχια

2.41) Κατά την εγκατάσταση της σύνδεσης ανακοινώνω 65535 και μετά αλλάζει την 1 η φορά, γίνεται 131072 και παραμένει σταθερή από εκεί και έπειτα

2.42) Frame: 590 Bytes, Ethernet Header: 14 Bytes, IPv4 Header: 20 Bytes, TCP Header: 20 Bytes

2.43) Ναι είναι 536 Bytes, που συμφωνεί με το MSS που έχει δηλώσει ο αποστολέας (που συνδέεται άμεσα με την MTU του)

2.44) Θα έκανε Fragment τα δεδομένα, με την προϋπόθεση ότι η σημαία Don't Fragment δεν είναι 1

2.45) Στέλνει 61442 Bytes συνολικά

2.46) 600 κB/s

2.47) Δεν παρατηρώ αναμετάδοση. (Θα έβλεπα κάποιο τεμάχιο με ίδιο Sequence number με κάποιο προηγούμενο)

Άσκηση 3

3.1) `tcp.port == 20`

3.2) 94.65.141.44

3.3) Περίπου 0.015 sec. Είναι μεγαλύτερη

3.4) Παρατηρώ πως στέλνονται κάποια τεμάχια μαζί και μετά μεσολαβεί ένα αρκετά μεγάλο χρονικό διάστημα, στο οποίο ο Server περιμένει ACK

3.5) Έστειλα 4 τεμάχια. Συμφωνεί πλήρως με το link, αφού το MSS είναι 536 Bytes, άρα το $IW = 4 * MSS \text{ Bytes}$ ($= < 4 \text{ fragments}$)

3.6) Στο 2ο 6 τεμάχια και στο 3ο 10 τεμάχια. Παρατηρούμε πως αυξάνονται κατά πολλαπλάσια του 2 (θέμα σύμβασης)

3.7) Ακολουθεί παρόμοια λογική. Πρώτα στέλνει 10 και μετά αυξάνει το window συμφόρησης κατά 10 MSS. Αυτό συμβαίνει, καθώς έχει αλλάξει η σύμβαση και πλέον το window size αυξάνεται κατά κάποια πολλαπλάσια του 1460

Άσκηση 4

4.1) udp

4.2)

Source Port: 2 bytes

Destination Port: 2 bytes

Length: 2 bytes

Checksum: 2 bytes

4.3) 8 bytes

4.4) 94 bytes

4.5) Payload + UDP Header

4.6) Το ελάχιστο είναι 8 Bytes που είναι μόνο το header. Το μέγιστο είναι $65535 - 20 = 65515$, καθώς 65535 είναι το μέγιστο πακέτο IPv4 και 20 η ελάχιστη επικεφαλίδα IPv4

4.7) 566 bytes

4.8) Ναι, STUN, DB-LSP-DISC/JSON, QUIC, SSDP

4.9) dns

4.10) fe80::71:9ec9:ad21:dbed

4.11)

Source Port: 59571

Destination Port: 53

4.12)

Source Port: 53

Destination Port: 59571

4.13) Θύρα 53