

**Όνοματεπώνυμο:** Χαράλαμπος Καμπουγέρης

**Ομάδα:** 3, Τετάρτη 10:45-12:30, Αιθ.Α2

**Όνομα PC/ΛΣ:** CHARALAMPOSS-MacBook-Air/ macOS 14.0

**Ημερομηνία:** 05/12/2023

**Διεύθυνση IP:** 147.102.237.19 και 147.102.202.118

**Διεύθυνση MAC:** 0c:e4:41:e1:6c:74

## **Εργαστηριακή Άσκηση 7**

### **Πρωτόκολλα TCP και UDP**

#### **Άσκηση 1**

**1.1** ip host 147.102.237.19

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

**1.3** Destination Port: 23

**1.4** tcp.dstport == 23

**1.5** SYN flag

**1.6** Κάνει 11 προσπάθειες και στις 2 περιπτώσεις

**1.7** Στις περιπτώσεις Α και Β έχουμε πανομοιότυπα αποτελέσματα και χρονική απόσταση μεταξύ των διαδοχικών προσπαθειών εγκατάστασης σύνδεσης είναι:

1 sec -> 1 sec -> 1 sec -> 1 sec -> 1 sec -> 2 sec -> 4 sec -> 8 sec -> 16 sec -> 32 sec

**1.8** Το μόνο που αλλάζει είναι το Sequence Number (raw)

**1.9** Γίνεται μόνο το πρώτο βήμα της χειραψίας, στο οποίο έχουμε sec = 0, ack = 0

**1.10** Φαίνεται να εγκαταλείπει την προσπάθεια καθώς το flag της απόλυσης (FIN) δεν είναι ενεργοποιημένο

**1.11** ip.addr == 147.102.40.1 and tcp

**1.12** Κάνει μόνο μια προσπάθεια

**1.13** Η διαφορά είναι ότι στην περίπτωση της σύνδεσης Γ λαμβάνουμε μήνυμα απόρριψης σύνδεσης. Συγκεκριμένα έχουμε ACK = 1 και RST = 1.

**1.14** 1 bit flags: ACK, RST, SYN

**1.15** Το flag RST

**1.16** Header Length: 20 bytes  
Data Length: 0 bytes

**1.17** Source Port: 2 bytes  
Destination Port: 2 bytes  
Sequence Number: 4 bytes  
Acknowledgment Number: 4 bytes  
Header Length: 4 bits  
Flags: 12 bits  
Window: 2 bytes  
Checksum: 2 bytes  
Urgent Pointer: 2 bytes

**1.18** Το Header Length έχει τιμή 5 HEX άρα και 5 DEC και δηλώνει ότι η επικεφαλίδα έχει μήκος  $5 \times 4 = 20$  bytes γιατί έχει πάντα μήκος πολλαπλάσιο του 4 (32 bits)

**1.19** Το header length μας δίνει πόσες 32 bit λέξεις υπάρχουν στην επικεφαλίδα TCP. Άρα, στην περίπτωση μας υπάρχουν  $32 \times 5 = 160$  bits = 20 bytes

**1.20** Όχι

**1.21** Το μήκος προκύπτει από το άθροισμα των επικεφαλίδων IPv4 και TCP που είναι 20 byte η κάθε μία. Οπότε συνολικά 40 bytes.

**1.22** Header Length = 44 bytes

**1.23** Ναι, υπάρχει. Αυτό οφείλεται στην ύπαρξη του προαιρετικού πεδίου options το οποίο έχει μήκος 24 bytes

## Άσκηση 2

**2.1** tcp and ip.host == 147.102.40.15

**2.2** Στη θύρα 21

**2.3** Με τη θύρα 20

**2.4** tcp.port == 21

**2.5** Για την εγκατάσταση σύνδεσης ανταλλάσσονται 3 τεμάχια TCP

**2.6** Οι σημαίες SYN και ACK

**2.7**

1<sup>ο</sup> τεμάχιο: 44 bytes

2<sup>ο</sup> τεμάχιο: 40 bytes

3<sup>ο</sup> τεμάχιο: 32 bytes

**2.8** Όλα τα τεμάχια έχουν μήκος 0

**2.9** Διαρκεί συνολικά 0,011762 sec

**2.10** Ναι συμφωνεί

**2.11** Το sequence number που στέλνουμε εμείς έχει τιμή απόλυτη 3793394127 και σχετική 0, ενώ το sequence number που στέλνει ο εξυπηρετητής έχει τιμή απόλυτη 1843615541 και σχετική 0

**2.12** Προκύπτει από το Sequence Number του πελάτη συν1 (εδώ είναι  $0 + 1 = 1$ )

**2.13** Το Acknowledgement Number είναι ίδιο με αυτό της αποδοχής σύνδεσης από τον εξυπηρετητή (δηλαδή 1) και το Sequence Number είναι ίδιο με το Acknowledgement Number διότι ο πελάτης αποστέλλει το τεμάχιο που ζήτησε ο εξυπηρετητής.

**2.14** Είναι μηδέν

**2.15** Το κάθε ένα από αυτά τα 2 πεδία είναι 4 bytes  $\rightarrow$  32bits. Οπότε μπορούν να φτάσουν έως  $2^{32}$

**2.16**  $\text{tcp.port} == 21$  and  $\text{tcp.len} == 0$  and  $(\text{tcp.seq} == 0$  or  $\text{tcp.seq} == 1)$  and  $(\text{tcp.ack} == 0$  or  $\text{tcp.ack} == 1)$

**2.17** Και στις δυο συνδέσεις, το παράθυρο που ανακοινώνει ο υπολογιστής μου είναι 65535 bytes

**2.18** Το παράθυρο εξυπηρετητή είναι πάλι 65535 bytes

**2.19** Στο πεδίο Window

**2.20** Ανακοινώνουν Window Scale = 6 και στις 2 συνδέσεις

**2.21** Στα options και συγκεκριμένα στο Window Scale

**2.22**  $\text{MSS} = 1460$  (υπολογιστή μου)

**2.23** Είναι πακέτο IPv4 οπότε θα είναι ίση με  $\text{MTU} - 40$

**2.24** Στα options: Maximum Segment Size

**2.25**  $\text{MSS} = 536$  (εξυπηρετητή)

**2.26** Παρατηρώ ότι  $\text{MSS} = \text{MTU} - 40 = 576 - 40 = 536$

**2.27** Προφανώς, το μεγαλύτερο τεμάχιο θα έχει μέγεθος ίσο με τη μικρότερη  $\text{MSS} + 20$  bytes (IPv4 Header) + 20 bytes (TCP Header), άρα 576 bytes

**2.28** Το FIN flag

**2.29** Ο εξυπηρετητής

**2.30** 4

**2.31** Header Length = 32 bytes

**2.32** Data Length = 0

**2.33** Το IPv4 πακέτο έχει μήκος 52 bytes, τα οποία προκύπτουν ως άθροισμα της IPv4 επικεφαλίδας (20 bytes) και του τεμαχίου TCP (32 bytes). Επίσης το πλαίσιο Ethernet έχει μήκος 66 bytes, τα οποία προκύπτουν ως άθροισμα της επικεφαλίδας Ethernet (14 bytes) και του πακέτου IPv4 (52 bytes)

**2.34** Το ίδιο ισχύει φυσικά και για το IPv4 πακέτο και το πλαίσιο Ethernet του εξυπηρετητή

**2.35** Απο τον εξυπηρετητή στάλθηκαν 357 bytes, και απο τον υπολογιστή μου στάλθηκαν 107 bytes

**2.36** Εντόπισα τα τελευταία πακέτα που εστειλα και ελαβα και βρήκα το relative sequence number τους

**2.37** tcp.port == 20

**2.38** Ο υπολογιστής μου ανακοινώνει MSS=1460 bytes, ενώ ο εξυπηρετητής ανακοινώνει MSS = 536

**2.39** 536 bytes

**2.40** 0.009543 sec

**2.41** Όχι

**2.42** 121 πακέτα

**2.43** Έστειλε 27 τεμάχια ACK

**2.44** Window: 2055

**2.45** Όχι και ο λόγος σχετίζεται με το γεγονός ότι το window size προσαρμόζεται αναλόγως το bandwidth. Έχουμε χαμηλο bandwidth και για αυτό το window size μειώθηκε σύμφωνα με αλγόριθμο του TCP

**2.46** Ναι αλλάζει και η μικρότερη καταγεγραμμένη τιμή είναι η 1802

**2.47** Εάν ο υπολογιστής ανέφερε μηδενική τιμή για το μέγεθος του παραθύρου (Window: 0) στην κεφαλίδα TCP, αυτό συνήθως υποδεικνύει ότι η προσωρινή μνήμη λήψης TCP του υπολογιστή είναι πλήρης ή ότι ο υπολογιστής δεν μπορεί να λάβει περισσότερα δεδομένα εκείνη τη στιγμή. Ένα μηδενικό μέγεθος παραθύρου ουσιαστικά λέει στον διακομιστή να σταματήσει την αποστολή δεδομένων έως ότου το μέγεθος του παραθύρου γίνει ξανά μη μηδενικό.

**2.48** Μήκος πλαισίου: 590 bytes  
Ethernet Header: 14 bytes  
IP Header: 20 bytes  
TCP Header: 32 bytes (περιέχονται 12 bytes options)

**2.49** Ναι καθώς είναι μικρότερο από τη μέγιστη τιμή ( $524 < 536$ )

**2.50** Θα γινόταν fragmentation

**2.51** 61442 bytes από τον εξυπηρετητή και 0 από τον υπολογιστή μου

**2.52** 61442 bytes σε 0.047159 secs αρα έχουμε 1302.8 Kbytes / sec

**2.53** Συμφωνα με το Wireshark, δεν υπήρξαν αναμεταδόσεις

### Άσκηση 3

**3.1** tcp.port==20

**3.2** 94.65.141.44

**3.3** RTT = 0.014674 sec μεγαλύτερο από αυτό του βρήκαμε στο ερώτημα 2.41

**3.4** Παρατηρώ ότι ο εξυπηρετητής στέλνει τα τεμάχια σε ομάδες και μάλιστα το μέγεθος παραθύρου αυξάνεται εκθετικά με το χρόνο καθώς το αντέχει η σύνδεση με τον υπολογιστή

**3.5** Έστειλε 4 τεμάχια και αυτό είναι σύμφωνο με ό,τι προβλέπει το RFC 5681 καθώς το MSS είναι μικρότερο από 1095

**3.6** Στο δεύτερο RTT στέλνονται 6 τεμάχια, στο τρίτο 10 τεμάχια και στο τέταρτο 16 τεμάχια

**3.7** Στάλθηκαν 2 στο πρώτο, 3 στο δεύτερο και 5 στο τρίτο δηλαδή τα μισά από τα αντίστοιχα του προηγούμενου ερωτήματος

**3.8** Ναι είναι παρόμοια, ωστόσο φτάνουν περισσότερα πακέτα σε κάθε RTT

## Άσκηση 4

**4.1** filter: udp

**4.2** Source Port: 2 bytes  
Destination Port: 2 bytes  
Length: 2 bytes  
Checksum: 2 byte

**4.3** 8 bytes

**4.4** Το IPv4 πακέτο έχει συνολικό μέγεθος 61 bytes και μήκος επικεφαλίδας 20 bytes. Άρα το μέγεθος του δεδομενογράμματος είναι 41 bytes

**4.5** Το συνολικό μέγεθος του δεδομενογράμματος UDP (header+payload)

**4.6** Για μεταφορά μηδενικών δεδομένων το συνολικό μήκος θα είναι όσο της επικεφαλίδας δηλ. 8 bytes

**4.7** Μέγιστο μέγεθος μηνύματος =  $65,535 - 20 - 8 \text{ bytes} = 65,507 \text{ bytes}$   
(μέγιστο μήκος πακέτου IPv4 - μήκος επικεφαλίδας IPv4 - μήκος επικεφαλίδας UDP). Το ελάχιστο μέγεθος μηνύματος προφανώς είναι 0 byte

**4.8** Θα είναι  $576 - 20 - 8 = 548 \text{ bytes}$

**4.9** Ναι, MDNS

**4.10** filter: dns

**4.11** 2001:648:2000:2000::1

**4.12** Source Port: 63615, Destination Port: 53

**4.13** Source Port: 53, Destination Port: 63615

**4.14** Στη 53