

Ονοματεπώνυμο: ΙΟΡΔΑΝΙΔΗΣ ΚΡΙΤΩΝ
Ομάδα: 1
Όνομα PC/OS: Kriton's Air / MacOS 12.6
Ημερομηνία: 26/10/2022
Διεύθυνση IP: 147.102.236.182
Διεύθυνση MAC: –

ΕΡΓΑΣΤΗΡΙΑΚΗ ΑΣΚΗΣΗ 4: ΠΡΩΤΟΚΟΛΛΟ IPv4 ΚΑΙ ΘΡΥΜΜΑΤΙΣΜΟΣ

Άσκηση 1

- 1.1 ping -c 3 www.mit.edu
- 1.2 Με αυτό το φίλτρο δεν βλέπουμε κίνηση δικτύου broadcast και multicast, δηλαδή βλέπουμε μόνο τα μηνύματα απο και προς τον υπολογιστή μας συγκεκριμένα (unicast)
- 1.3 0.0% packet loss και μέση καθυστέρηση 34,639 ms
- 1.4 round-trip min/avg/max/stddev = 30.013/34.639/37.991/3.379 ms
- 1.5 Το wireshark δίνει τιμές 0.035728, 0,370803 και 0,029903. Υπάρχει μια απειροελάχιστη (αμελητέα) απόκλιση απο τις τιμές του terminal
- 1.6 Το φίλτρο ip
- 1.7 ip.addr == 147.102.236.182 and icmp
- 1.8 echo requests
- 1.9 Για τα echo requests η IPv4 πηγής είναι 147.102.236.182 και η IPv4 προορισμού είναι η 92.123.12.49
- 1.10 echo replies
- 1.11 Για τα echo requests η IPv4 πηγής είναι 92.123.12.49 και η IPv4 προορισμού είναι η 147.102.236.182
- 1.12 Έχει αλλάξει η διεύθυνση IPv4 του www.mit.edu

Άσκηση 2

- 2.1 ping -c 5 <destination address>
- 2.2 Μόνο 5 μηνύματα έχουν καταγραφεί
- 2.3 Το default gateway δηλαδή η διεύθυνση 147.102.236.200
- 2.4 Όχι, διότι αυτά πάνε στον οδηγό loopback και δεν βγαίνουν στο τοπικό δίκτυο οπότε δεν εντοπίζονται από το wireshark
- 2.5 Και πάλι όχι διότι ό,τι δεν βγαίνει στο τοπικό δίκτυο δεν εντοπίζεται από το wireshark
- 2.6 Η διαφορά είναι ότι όταν στέλνω request στον εαυτό μου, αυτό περνά απο τον Ethernet driver ο οποίος το ανακατευθύνει στον οδηγό loopback, ενώ όταν στέλνω μήνυμα στη διεύθυνση του loopback αυτό πάει κατευθείαν στον οδηγό loopback ο οποίος το στέλνει στην είσοδο πακέτων
- 2.7 Παρατηρώ ότι το Netflix δεν στέλνει πίσω echo replies σε αντίθεση με το Amazon το οποίο στέλνει. Αυτό μπορεί να οφείλεται στο ότι το Netflix έχει ενεργοποιημένο κάποιο firewall που κόβει τα ICMP πακέτα

Άσκηση 3

- 3.1 host 147.102.40.15
- 3.2 ip.src==147.102.236.182
- 3.3
Version: 1 byte
Header Length: 1 byte
Differentiated Services Field: 1 byte

Total Length: 2 bytes
Identification: 2 bytes
Flags: 1 byte
Fragment Offset: 2 bytes
Time to Live: 1 byte
Protocol: 1 byte
Header Checksum: 2 bytes
Source Address: 4 bytes
Destination Address: 4 bytes

3.4 Αλλάζουν τα Differentiated Services Field, Total Length και Header Checksum

3.5 Ναι (20 bytes)

3.6 Το μικρότερο έχει μήκος 52 bytes και το μεγαλύτερο 128 bytes

3.7 Το Differentiated Services Field παίρνει τιμες 0,8 και 10 (hex) οι οποίες αντιστοιχούν σε Standard, Low-priority data και High-throughput data ποιότητες υπηρεσιών αντίστοιχα

3.8 Το identification έχει παντού τιμη 0

3.9 Έχει παντού τη τιμή 1

3.10 Έχει παντού τη τιμή 0

3.11 Έχει παντού τη τιμή 6 και αντιστοιχεί σε πρωτόκολλο TCP

3.12 Το Header Checksum προκύπτει απο τα byte του καθε πακέτου IPv4. Από τη στιγμή που σε καθε τέτοιο πακέτο αλλάζουν καποια byte, είναι αναμενόμενο να αλλάζει και το Header Checksum

Άσκηση 4

4.1 ping -o -D -s <packet_size> <destination address>

(Σε windows: ping -n 1 -f -l <size> <destination address>)

4.2 Με δοκιμές βρίσκω ότι η μέγιστη τιμή είναι 1472 bytes

4.3 Προκύπτει, λοιπόν, ότι η ελάχιστη τιμή για την οποία απαιτείται θρυμματισμός είναι τα 1473 bytes

4.4 not broadcast and not multicast

4.5 ip.addr == 147.102.38.11

4.6 Όχι, διότι δεν στέλνεται κανένα μήνυμα

4.7 Το MTU της διεπαφής του υπολογιστή μου είναι 1514 bytes σύμφωνα με το wireshark

4.8 Πρέπει το πεδίο data της επικεφαλίδας ICMP να έχει μέγεθος 1472 bytes

4.9 Ναι επιτυγχάνει και μάλιστα προς την δική μου IP μπορώ να στείλω πακέτα με έως και 1492 bytes

4.10 1514 bytes

4.11 Όχι, έχει σπάσει σε πολλά μικρότερα πακέτα

4.12 Χρειάστηκαν 5 πακέτα διότι κάθε πακέτο μπορεί να μεταφέρει το πολύ 1480 bytes και $6000/1480=4,05$. Άρα χρειάζονται 4 πακέτα μέγιστου μήκους και ένα ακόμη το οποίο είναι μικρότερο σε μήκος

4.13

ΘΡΑΥΣΜΑΤΑ	Identification	Don't Fragment Bit	More Fragments Bit	Fragment Offset
1o	7DB9	0	1	0
2o	7DB9	0	1	1480
3o	7DB9	0	1	2960
4o	7DB9	0	1	4440
5o	7DB9	0	0	5920

- 4.14 To More Fragments που έχει τιμή 1
- 4.15 To Fragment Offset που έχει τιμή 0
- 4.16 1514 bytes (συνολικό μήκος πακέτου)
- 4.17 To Fragment Offset που δεν έχει τιμή 0
- 4.18 Ναι
- 4.19 To More Fragments έχει τιμή 1
- 4.20 To Fragment Offset μόνο
- 4.21 Παρατηρώ ότι κάθε τιμή προκύπτει από την αντίστοιχη τιμή του προηγούμενου fragment προσθέτοντας 1480, ξεκινώντας από το πρώτο που έχει τιμή 0. Οπότε είναι αναμενόμενο το 4ο fragment να έχει τιμή $4440 = 1480 \times 3$ (3 fragments πριν από αυτό) και το 5ο να έχει $5920 = 1480 \times 4$ (4 fragments πριν από αυτό)
- 4.22 To Fragment Offset, το More Fragments Bit (συγκεκριμένα μόνο στο 5ο fragment έχει τιμή 0 αφού είναι το τελευταίο fragment) και το Total Length