Όνοματεπώνυμο: Μάρκος Δεληγιάννης		Ομάδα: 3
Όνομα PC/ΛΣ: MarkiniHP / Linux Mint		Ημερομηνία: 26 / 10 / 2022
Διεύθυνση IP: 147.102.38.90	Διεύθυνση MAC: A4-17-31-50-E0-BD	

Εργαστηριακή Άσκηση 4
Πρωτόκολλο IPv4 και θρυμματισμός

Απαντήστε στα ερωτήματα στον χώρο που σας δίνεται παρακάτω και στην πίσω σελίδα εάν δεν επαρκεί. Το φυλλάδιο αυτό θα παραδοθεί στον επιβλέποντα.

1

1.1 ping -4 -c 3 www.mit.edu

1.2 Με την εφαρμογή του φίλτρου συλλαμβάνονται μόνο τα πακέτα unicast (δεν είναι broadcast ούτε multicast). Έτσι βλέπουμε μόνο τα σχετικά με τον υπολογιστή μας πακέτα, χωρίς θόρυβο.

1.3 Ποσοστό απωλειών: 0% (0% packet loss)
Μέση καθυστέρηση: 53.649 ms (rtt avg 53.649 ms)

1.4 1ο ping: 53.8 ms
2ο ping: 53.6 ms
3ο ping: 53.5 ms

1.5 1ο ping: 53.775093 ms
2ο ping: 53.588142 ms (Συμφωνούν με τις τιμές του 1.4)
3ο ping: 53.488006 ms

1.6 “ip”

1.7 “icmp”

1.8 Στάλθηκαν μηνύματα “Echo (ping) request” με πεδίο Type = 8

1.9 IPv4 address πηγής: 147.102.38.90
IPv4 address προορισμού: 104.125.28.249

1.10 Ελήφθησαν μηνύματα “Echo (ping) reply” με πεδίο Type = 0

1.11 IPv4 address πηγής: 104.125.28.249
IPv4 address προορισμού: 147.102.38.90

1.12 Σε σχέση με το παρελθόν έχει αλλάξει η IPv4 διεύθυνση του www.mit.edu και το round trip time (έχει μειωθεί).

.....

2

2.1 ping -4 -c 5 127.0.0.1

.....

2.2 Το Wireshark έχει καταγράψει 5 μηνύματα ICMP Echo request, που αντιστοιχούν σε 1 ping.

2.3 Ο προορισμός τους ήταν ο default gateway 147.102.38.200.

2.4 Δεν παρατηρήσαμε τέτοια μηνύματα, όπως ήταν αναμενόμενο, καθώς αυτά παραμένουν στον υπολογιστή και δεν αποστέλλονται. Έτσι, η βιβλιοθήκη του wireshark δεν τα εντοπίζει.

2.5 Δεν παρατηρήσαμε τέτοια μηνύματα. Η εξήγηση είναι η ίδια με το 2.4.

2.6 Όταν κάνουμε ping στην loopback address τότε το σύστημα γνωρίζει ότι πρόκειται για πακέτο με προορισμό τον εαυτό του και δεν εμπλέκονται drivers καρτών δικτύου. Όταν όμως κάνουμε ping την δική μας IP, τότε εμπλέκονται οι drivers των καρτών δικτύου οι οποίοι επιστρέφουν το πακέτο στον υπολογιστή.

2.7 Παρατηρούμε ότι το www.netflix.com δεν απαντά στα pings (100% packet loss) παρόλο που είναι διαθέσιμο, ενώ το www.amazon.com απαντά κανονικά (0% packet loss). Μία πιθανή εξήγηση είναι ότι το netflix μπλοκάρει τα ping ώστε να προστατευθεί από πιθανές επιθέσεις DoS (Denial of Service), κατά τις οποίες γίνονται πολλαπλά pings σε έναν server ώστε να τον αποτρέψουν από το να απαντήσει σε άλλα αιτήματα.

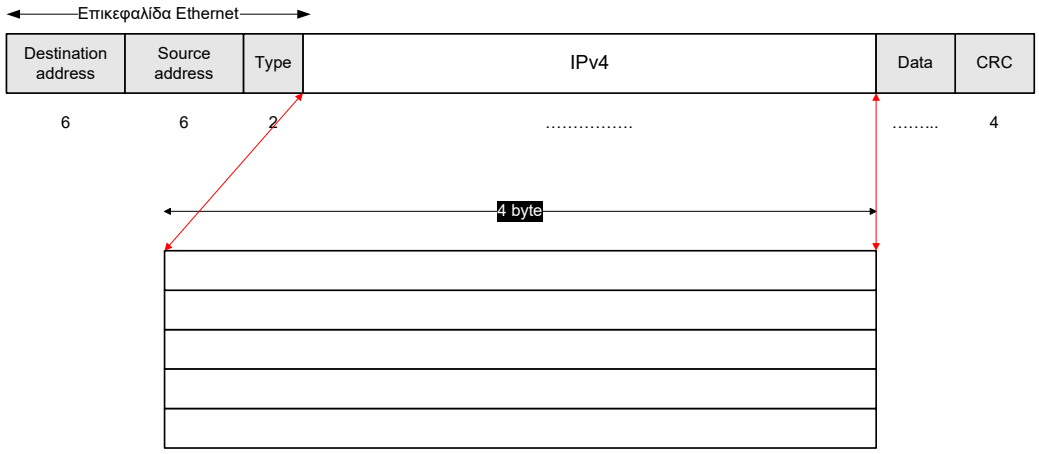
3

3.1 “host 147.102.40.15”

3.2 “ip.src == 147.102.38.90”

3.3 Τα πεδία της επικεφαλίδας με τη σειρά που εμφανίζονται είναι:

Version: 4 bits	Header length: 4 bits	
Differentiated Services Field: 1 byte	Total length: 2 bytes	
Identification: 2 bytes	Flags: 3 bit	Fragment offset: 13 bit
Time to live: 1 byte	Protocol: 1 byte	Header checksum: 2 bytes
Source address: 4 bytes	Destination address: 4 bytes	



3.4 Διαφορετική τιμή έχουν τα πεδία Total Length, Identification και Header Checksum.

.....

3.5 Ναι, 20 bytes.

3.6 Μικρότερο μήκος πακέτου: 32 byte Μεγαλύτερο μήκος πακέτου: 132 byte

3.7 Η τιμή που λαμβάνει είναι 0x00 (standard).

3.8 Παρατηρούμε ότι είναι μοναδικές για κάθε πακέτο.

3.9 Η σημαία don’t fragment έχει τιμή 1.

3.10 Το πεδίο fragment offset έχει τιμή 0.

3.11 Το πεδίο Protocol έχει τιμή 0x06 και αντιστοιχεί στο πρωτόκολλο TCP.

3.12 Το header checksum υπολογίζεται με βάση όλα τα bits του header, συνεπώς οποιαδήποτε αλλαγή στο header αλλάζει με μεγάλη πιθανότητα το header checksum. Επειδή τα headers δεν είναι ακριβώς ίδια μεταξύ τους έχουμε διαφορετική τιμή του πεδίου header checksum.

4 (Σε σύστημα Windows)

4.1 ping -n 1 -f -l [size in bytes] [ipv4 address]

4.2 1472 bytes

4.3 1472+1=1473 bytes

4.4 “No Broadcast and no Multicast”

4.5 “ip.addr==147.102.38.89”

4.6 Δεν παράγονται, διότι το πακέτο πρέπει να θρυμματιστεί και δεν το έχουμε επιτρέψει.

4.7 Από το wireshark είναι Frame size – Eth. header length = 1514 – 14 = 1500. Αυτό μπορεί να προκύψει και από την τιμή του 4.2 (μέγεθος ICMP payload) αν προσθέσουμε τις επικεφαλίδες ICMP και IP (8 και 20 byte).

4.8 Βάζοντας πολύ μεγάλες τιμές μήκους στο ping και διαβάζοντας το error message είναι 65500 bytes.

4.9 Δεν επιτυγχάνει. Με δοκιμή προκύπτει μέγιστη τιμή επιτυχίας 1492 bytes.

4.10 Από 4.8 με προσθήκη των επικεφαλίδων ICMP και IPv4 είναι 65500 + 8 + 20 = 65528 bytes.

4.11 Όχι, το πακέτο θρυμματίζεται.

4.12 Έχουμε μέγεθος πακέτου ICMP 6000+8=6008 και μέγιστο μέγεθος IPv4 data = MTU – 20 bytes = 1480 θα χρειαστούμε $\lceil 6008/1480 \rceil = 5$ πακέτα IPv4.

4.13 1ο: Identification: 0x94c7 / Don’t Fragment Bit: 0 / More Fragments Bit: 1 / Fragment Offset: 0

2ο: Identification: 0x94c7 / Don’t Fragment Bit: 0 / More Fragments Bit: 1 / Fragment Offset: 1480

3ο: Identification: 0x94c7 / Don’t Fragment Bit: 0 / More Fragments Bit: 1 / Fragment Offset: 2960

4ο: Identification: 0x94c7 / Don’t Fragment Bit: 0 / More Fragments Bit: 1 / Fragment Offset: 4440

5ο: Identification: 0x94c7 / Don’t Fragment Bit: 0 / More Fragments Bit: 0 / Fragment Offset: 5920

4.14 Το πεδίο More Fragments Bit = 1

4.15 Το πεδίο Fragment Offset = 0

4.16 Data length + IPv4 header = 1480 + 20 = 1500 bytes.

4.17 Το πεδίο Fragment Offset = 1480 ≠ 0 bytes

4.18 Ναι

4.19 Από το πεδίο More Fragments Bit = 1

4.20 Αλλάζει το Fragment Offset και το Header Checksum.

.....

4.21 Κάθε θραύσμα (εκτός από το τελευταίο) έχει 1480 byte payload (το μέγιστο δυνατό).

Συνεπώς πριν από το 4ο-5ο θραύσμα αντίστοιχα έχουν προηγηθεί 3*1480=4440 και 4*1480=5920 byte αντ.

4.22 Το Fragment Offset, το More Fragments Bit (στο τελευταίο είναι 0), το total length (στο τελευταίο είναι ίσο με 108) και το Header Checksum.

.....