Ονοματεπώνυμο: Ανδρέας Στάμος (03120***)	**Ομάδα:** 1
Όνομα PC/ΛΣ: linux / Ubuntu 22.04.2 LTS	**Ημερομηνία:** 24/10/2023
Διεύθυνση IP: 192.168.1.10	**Διεύθυνση MAC:** DE–3F–DC–B2–E0–D0

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.

Άσκηση 1

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

1.2 0% απώλεια, μέση καθυστέρηση 47.837ms

1.3 Echo 1 RTT: 47.4ms
Echo 2 RTT: 48.4ms
Echo 3 RTT: 47.7ms

1.4 Έχει αλλάξει η διεύθυνση IP που αντιστοίχει στο domain www.mit.edu. Μάλιστα η IP διεύθυνση αλλάξε και όσο εκτελούσα την άσκηση, καθώς αφού πέρασε λίγη ώρα από το αρχικό ping, επιχείρησα να ξανακάνω ping. Το γεγονός αυτό, λογικά, οφείλεται σε DNS Load Balancing. Πιο συγκεκριμένα εκτιμάται πως οι διαχειριστές του www.mit.edu έχουν πολλούς εξυπηρετητές για την ιστοσελίδα και επιθυμούν να κατανείμουν την κίνηση σε αυτούς. Ένας τρόπος για να το κάνουν αυτό, είναι να ρυθμίσουν τον DNS εξυπηρετητή ώστε όταν ερωτηθεί για την IP διεύθυνση του domain τους σε κάποιους χρήστες να επιστρέφει την διεύθυνση ενός εξυπηρετητή και σε κάποιους αλλούς την διεύθυνση ενός άλλου εξυπηρετητή.

1.5 Να καταγραφούν πλαισία που έχουν διεύθυνση MAC προορισμού η οποία δεν είναι ομάδας, δηλαδή που έχουν προορισμό μία και μόνο κάρτα δικτύου.

1.6 `ip`

1.7 `icmp && ip.addr == 23.214.226.42`

1.8 Echo request

1.9 Διεύθυνση IP πηγής: 192.168.1.10
Διεύθυνση IP προορισμού: 23.214.226.42

1.10 Echo reply

1.11 Διεύθυνση IP πηγής: 23.214.226.42
Διεύθυνση IP προορισμού: 192.168.1.10

1.12 Echo 1 RTT: 47.4ms
Echo 2 RTT: 48.4ms
Echo 3 RTT: 47.7ms
Συμφωνούν.

Άσκηση 2

2.1
```
ping -c 5 192.168.1.1
ping -c 5 192.168.1.10
ping -c 5 127.0.0.1
```

2.2 5

2.3 Ο default gateway

2.4 Όχι. Το ICMP οδηγείται στον οδηγό loopback. Αυτό γίνεται σαφές αν δούμε τον τρόπο δρομολόγησης πακέτων του Linux. Το Linux διαθέτει πολλούς πίνακες δρομολόγησης. Όταν λάβει ένα πακέτο για δρομολόγηση, κοιτάζει την Routing Policy Database και εξετάζει σε αύξουσα σειρά του rule τους πίνακες δρομόλογησης. Με την εντολή `ip rule list` βλέπουμε την Routing Policy Database. Βλέπουμε ότι ο πίνακας δρομόλογησης local έχει αριθμό rule 0, που σημαίνει ότι θα εξεταστεί πρώτος.
Με την εντολή `ip route show table local` βλέπουμε τον πίνακα δρομολόγησης local, στον οποίο υπάρχει εγγραφή για την διεύθυνση του ίδιου υπολογιστή προς τον οδηγό loopback.
Επιβεβαιώνουμε ότι το Linux δρομολογεί τα πακέτα προς την διεύθυνση του ίδιου υπολογιστή προς τον οδηγό loopback, με την εντολη: `ip route get {LOCAL IP ADDRESS}`
Έτσι, το Linux προωθεί το IP πακέτο με το ICMP Echo Request προς τον οδηγό loopback. Εκτελεί όμοια, έπειτα, για το ICMP Echo Reply. Όμως έχουμε ζητήσει στο Wireshark να καταγράφει πακέτα που περνάνε από την κανονική μας κάρτα δικτύου, οπότε και δεν καταγράφει το πακέτο αυτό.

Συμπλήρωση: Μπορούμε να παρακάμψουμε την δρομολόγηση του Linux και να στειλούμε το IP πακέτο του ICMP χειροκίνητα προς τον default gateway με χρήση της εντολής `nping -icmp -e {INTERFACE NAME} --dest-mac {DEFAULT GATEWAY MAC ADDRESS} {LOCAL IP ADDRESS}`.

Κατάγραφοντας τα πακέτα με το Wireshark παρατήρουμε ότι το πακέτο φεύγει από τον υπολογιστή μας, πηγαίνει στον δρομολογητή, ο δρομολογητής το δρομολογεί προς στον υπολογιστή μας, λαμβάνουμε πίσω το ICMP Echo Request, και ταυτόχρονα ο δρομολογητής στέλνει και `ICMP Redirect` προς τον υπολογιστή μου (ως αποστολέα), πως δεν έπρεπε να το στείλουμε προς αυτόν, αλλά απευθείας προς τον υπολογιστή μας. Το Linux προσπαθεί να στείλει απάντηση ICMP Echo Reply προς την IP διεύθυνση του υπολογιστή μου. Επειδή είναι στο ίδιο υποδίκτυο με αυτο, στέλνει ARP Request για την MAC διεύθυνση της IP διεύθυνσης του υπολογιστή μου (ως αποστολεά του ICMP Echo Request). Αυτό το ARP Request θα έπρεπε να το απαντήσει η ίδια κάρτα δικτύου, που προφανώς δεν το κάνει, διότι υπό φυσιολογικές συνθήκες αυτό το ARP Request θα έπαιζε τον ρόλο του ARP Announcement. Εφόσον το Linux δεν βρει την MAC διεύθυνση που αντιστοίχει στην IP διεύθυνση που πρέπει να στείλει το Echo Reply, δεν στέλνει Echo Reply.

Πάραυτα μπορούμε με την εντολή `nping -icmp -e {INTERFACE NAME} --dest-mac {DEFAULT GATEWAY MAC ADDRESS} -S {ANOTHER DEVICE IP ADDRESS} {LOCAL IP ADDRESS}` να κάνουμε το IP πακέτο του ICMP να φαίνεται πως προήλθε από μια άλλη IP διεύθυνση (φροντίζουμε να είναι του τοπικού υποδικτύου ωστέ να μην κόψει το πακέτο το NAT του δρομολογητή και, επίσης, αφού θα είναι στο τοπικό υποδίκτυο, να είναι υπαρκτή ώστε όταν ο υπολογιστής στείλει ARP Request να λάβει από κάποιον απάντηση ARP Reply). Με άλλα λόγια, επιλέγω να στείλω το IP πακέτο του ICMP με διεύθυνση προορισμού MAC την διεύθυνση του δρομολογητή, διεύθυνση IP προέλευσης την διεύθυνση IP του κινητού μου τηλεφώνου και διεύθυνση IP προορισμού, όπως και πριν, την διεύθυνση IP του υπολογιστή μου. Παρατηρούμε τα εξής τότε στο Wireshark:

Το IP πακέτο ICMP Echo Request πηγαίνει στον δρομολογητή, ο δρομολογητής το στέλνει στον προορισμό του, δηλαδή στον υπολογιστή μου. Παραδόξως, στέλνει στον υπολογιστή μου, ένα ICMP Redirect, το οποίο όμως απευθύνεται προς την διεύθυνση του κινητού μου. Εικάζουμε πως όταν ο δρομολογήτης έλαβε το ICMP Echo Request κατέγραψε στον ARP πίνακα πως η διεύθυνση IP του κινητού τηλεφώνου μου αντιστοιχεί στην MAC διεύθυνση του υπολογιστή μου, για αυτό και το ICMP Echo Request ήρθε σε εμένα. (αυτό θα μπορούσε να συμβεί σε φυσιολογικές συνθήκες, μάλλον αν ο υπολογιστής μου λειτουργούσε ως Wi-Fi αναμεταδότης). Το Linux λαμβάνει το ICMP Echo Request, βλέπει ως διεύθυνση αποστολέα την διεύθυνση του κινητού μου τηλέφωνου, παρατηρεί πως είναι στο ίδιο υποδίκτυο, στέλνει στο υποδίκτυο ένα ARP Request για να βρει την MAC διεύθυνσή του, το κινητό μου απαντάει, και έπειτα ο υπολογιστής μου στέλνει στο κινητό μου ένα ICMP Echo Reply, παρόλο που το κινητό μου δεν είχε στείλει ποτέ ICMP Echo Request!

2.5 Όχι. Όπως και στο 2.4, στον πίνακα δρομολόγησης local, που εξετάζεται πρώτος, υπάρχει εγγραφή για την διεύθυνση 127.0.0.1 προς τον οδηγό loopback. Συνεπώς το Linux δρομολογεί το ICMP πακέτο προς τον οδηγό loopback. Όμοια έπειτα για το ICMP Echo Reply.

2.6 Σύμφωνα με το διάγραμμα που δίνεται στην εκφώνηση:
Το Λειτουργικό Σύστημα δρομολογεί τα πακέτα με προορισμό την IP διεύθυνση του ίδιου υπολογιστή πρώτα προς τον οδηγό Ethernet, ο οποίος αναλαμβάνει έπειτα να τα δρομολογήσει προς τον οδηγό loopback. Αντίθετα το Λειτοργικό Σύστημα δρομολογεί ταπακέτα με προορισμό την IP διεύθυνση 127.0.0.1 απευθείας προν τον οδηγό loopback.

Ωστόσο, όπως σχολιάστηκε στο 2.4 και στο 2.5, το Linux και από όσο φαίνεται και τα Windows XP, δεν κάνουν αυτό που παρουσιάζεται στο διάγραμμα. Αντίθετα, έχουν στον πίνακα δρομολόγησης μια εγγραφή για την διεύθυνση του ίδιου υπολογιστή προς τον οδηγό loopback. Ενδεχομένως σε άλλα ΛΣ, εκτός των Unix/Linux και Windows, να ισχύει αυτό που εικονίζει το διάγραμμα.

2.7 Παρατηρούμε πως ενώ στον φυλλομετρητή φορτώνονται κανονικά και οι δύο ιστοσελίδες, αδυνατούμε να κάνουμε `ping` στο `www.netflix.com`, ενώ μπορούμε κανονικά να κάνουμε στο `www.amazon.com`. Εκτιμάμε

πως ο διαχειριστής του www.netflix.com έχει μπλοκάρει στον υπολογιστή του εξυπηρετητή (ή στο γενικότερο δίκτυο), μέσω κάποιου τείχους προστασίας, τα πακέτα ICMP.

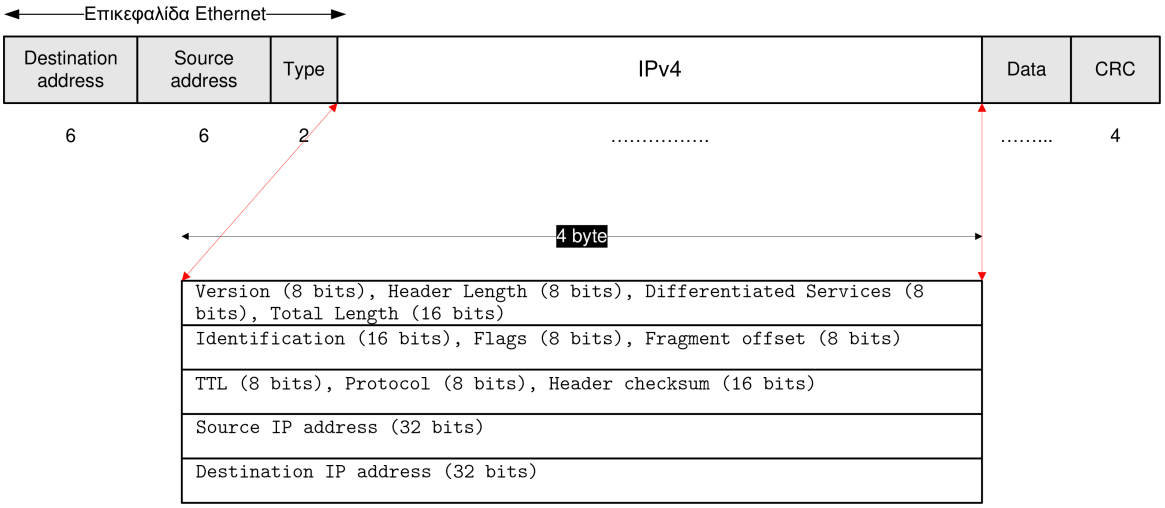
Ενδεχομένως αυτό να έχει αποφασιστεί προκειμένου να αποφευχθεί μια επίθεση **ping flood**, όπου ένας κακόβουλος χρήστης στέλνει, ενδεχομένως από πολλούς υπολογιστές, μαζικά ICMP Echo Request, προκειμένου να επιβάλλει στον εξυπηρετητή να απαντήσει με ICMP Echo Reply, καταναλώνοντας πολλούς πόρους τόσο του δίκτυου όσο και της ΚΜΕ, τόσο πολλούς, ώστε είτε να επιβραδυνθεί η εξυπηρέτηση των φυσιολογικών αιτήματων είτε ακόμα και να γίνει αδύνατη.

Άσκηση 3

3.1 `host 147.102.40.15`

3.2 `ip.src == 192.168.1.10`

3.3



3.4 Total Length, Identification και Header Checksum

3.5 Ναι

3.6 ελάχιστο 52 bytes και μέγιστο 109 bytes

3.7 Έχει τιμή 0x10, που σύμφωνα με την Wikipedia σημαίνει High-throughput data.

3.8 Το πεδίο Identification έχει διαφορετικές τιμές για κάθε IP πάκετο, που είναι λογικό αφού τα πακέτα ξεκινάνε από τον υπολογιστή, δεν έχουν περάσει ακόμα από δίκτυο, οπότε ακόμα δεν είχαν την "ευκαιρία" να υποστούν Fragmentation. Επίσης η σημαία Don't Fragment έχει τεθεί σε 1, που σημαίνει ότι ούτως ή άλλως τα IP πάκετα δεν θα υφίστανται Fragmentation, οπότε και καθένα θα έχει διαφορετικό Identification.

3.9 1

3.10 0

3.11 0x06 που αντιστοιχεί σε TCP

3.12 Το Header Checksum κατακερματίζει την IP επικεφαλίδα. Εφόσον η IP επικεφαλίδα, είναι επόμενο πως θα αλλάζει και το Header Checksum.

Άσκηση 4

4.1 `ping -4 -S {ICMP DATA SIZE} -M do {TARGET IP ADDRESS}`

4.2 Με δοκιμές βρίσκουμε 1472 bytes.

Το ίδιο μέγεθος βρίσκουμε και θεωρητικά βρίσκοντας με `ip link` το MTU (1500 bytes) και αφαιρώντας από αυτό το μέγεθος της IPv4 επικεφαλίδας (20 bytes) και της ICMP επικεφαλίδας (8 bytes). Είναι: $1500-20-8 = 1472$ bytes.

4.3 Αφού το μέγιστο για το οποίο αποστέλλεται χωρίς θρυμματισμό είναι 1472 bytes, το ελάχιστο για να απαιτείται θρυμματισμός είναι 1473 bytes. Πράγματι επιτρέποντας θρυμματισμό (με την εντολή `ping -4 -S {ICMP DATA SIZE} -M do {TARGET IP ADDRESS}`) παρατηρούμε ότι από 1473 bytes και πάνω, το IP πακέτο υφίσταται θρυμματισμό.

4.4 `not broadcast && not multicast`

4.5 `ip.addr == 192.168.1.1`

4.6 Όχι. Σύμφωνα με το `ip link` η κάρτα δικτύου έχει MTU 1500 bytes. Το ίδιο το Λειτουργικό Σύστημα θα αρνηθεί να μεταδώσει παραπάνω, επιστρέφοντας μήνυμα λάθους όταν του ζητηθεί να μεταδόσει πλαίσιο με δεδομένα πάνω από 1500 bytes.

4.7 Εκτελώ την εντολή `ip link` και βρίσκω MTU 1500 bytes.

Εξάλλου, όπως σχολιάσαμε και στο 4.3, αφού το μέγιστο ICMP payload είναι 1472 bytes το μέγιστο πλαίσιο που μπορεί να μεταφερθεί (δηλαδή το MTU) είναι 1472 bytes + 8 bytes (ICMP Header) + 20 bytes (IP Header) = 1500 bytes.

4.8 Μέγιστο μέγεθος ICMP Data = Μέγιστο μέγεθος IPv4 πακέτου − 20 bytes IPv4 επικεφαλίδας − 8 bytes ICMP επικεφαλίδας = 65507 bytes

4.9 Επιτυγχάνει. Είναι λογικό αφού το πακέτο δρομολογείται προς τον οδηγό loopback που σύμφωνα με το `ip link` έχει MTU 65535, δηλαδή το μέγιστο μέγεθος ενός IPv4 πακέτου.

4.10 65535 (είναι το μέγιστο μέγεθος IPv4 πακέτου και το παράγαμε στο ερώτημα 4.9)

4.11 Όχι έχει μεταφερθεί ως πολλά.

4.12 Μετράμε 5 πακέτα. Πράγματι απαιτούνται:

$$\left\lceil \frac{\text{μέγεθος ICMP data}}{\text{μέγεθος ICMP data που χωρά σε 1 πακέτο}} \right\rceil = \left\lceil \frac{6000}{1472} \right\rceil = 5 \text{ πακέτα}$$

4.13

No	Identification	Don't Fragment Bit	More Fragments Bit	Fragment Offset
0	0x20d3	0	1	0
1	0x20d3	0	1	185
2	0x20d3	0	1	370
3	0x20d3	0	1	555
4	0x20d3	0	1	740

4.14 Το More Fragments έχει τιμή 1.

4.15 Το Fragment Offset έχει τιμη 0.

4.16 1480 bytes

4.17 Το Fragment Offset έχει μη μηδενική τιμή.

4.18 Ναι. Αυτό αναγνωρίζεται από την επικεφαλίδα αφού το More Fragments έχει τιμή 1.

4.19 Fragment Offset, More Fragments, Total Length και Header Checksum

4.20 Version, IHL, DSCP, Identification, Flags, TTL, Protocol, Source IP Address, Destination IP Address

4.21 Το Fragment Offset ισούται με το πλήθος των bytes των προηγούμενων fragments διαιρεμένο με το 8.

4.22 Το τελευταίο πακέτο έχει τιμή 740 στο Fragment Offset, που αντιστοιχεί σε $740 \cdot 8 = 5920$ bytes, που σημαίνει ότι το τελευταίο πακέτο ξεκινά από το 5920ο byte του αρχικού μη θρυμματισμένου IPv4 πακέτου, το οποίο είναι σωστό αφού έχουν προηγηθεί: 4 θραύσματα $\cdot$ 1480 $\frac{\text{bytes}}{\text{θραύσμα}} = 5920$ bytes.