

Όνοματεπώνυμο: Ανδρέας Στάμος (03120***)	Όνομα PC: linux / Ubuntu 22.04.2 LTS
Ομάδα: 1	Ημερομηνία: 05/03/2024

## Εργαστηριακή Άσκηση 4

### Εισαγωγή στη δρομολόγηση

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 Η διεύθυνση IP αφορά σε μια συγκεκριμένη διεπαφή ενός συγκεκριμένου host και έχει μήκος 32bits στο IPv4 και 128bits στο IPv6.

Αντίθετα ο αριθμός δικτύου είναι ένα πρόθεμα της διεύθυνσης IP, αφορά σε ένα δίκτυο και έχει μήκος μικρότερο από 32bits στο IPv4 και από 128bits στο IPv6. Οι hosts που βρίσκονται στο ίδιο δίκτυο, έχουν IP διευθύνσεις με ίδιο αριθμό δικτύου.

- 1.2 192.220.144, διότι  $192.122.147.2 \& ((1 \ll 22) - 1) = 192.220.144.0$

- 1.3 Το τμήμα host μαζί με το τμήμα υποδικτύου είναι  $32 - 22 = 10$  bits. Κάθε υποδίκτυο θέλουμε να έχει 100 IP διευθύνσεις + 1η διεύθυνση που είναι το subnet identifier + τελευταία διεύθυνση που είναι η broadcast διεύθυνση, συνολικά 102 διευθύνσεις. Απαιτούνται  $\lceil \log_2 102 \rceil = 7$  bits για το τμήμα host. Άρα απομένουν  $10 - 7 = 3$  bits για το τμήμα υποδικτύου. οπότε μπορούν να υπάρξουν  $2^3 = 8$  υποδίκτυα.

- 1.4 C

- 1.5 b, d, e

- 1.6 Κάθε διεπαφή είναι ρυθμισμένη σε μια ορισμένη IP διεύθυνση και ένα ορισμένο τμήμα υποδικτύου (είτε χειροκίνητα είτε μέσω DHCP). Έτσι, ο δρομολογητής αρκεί να ελέγξει αν η διεύθυνση παραλήπτη έχει ίδιο τμήμα υποδικτύου με την IP διεύθυνση της υπό εξέταση διεπαφής. Πιο συγκεκριμένα, πρέπει να ελέγξει αν (δική του IP διεύθυνση & μάσκα υποδικτύου) == (IP διεύθυνση παραλήπτη & μάσκα υποδικτύου).

- 1.7 10.50.10.64

- 1.8  $208 = 0b11010000$ , οπότε ανήκει στην κλάση C.

- 1.9 B

- 1.10  $2^{32-17} - 2 = 2^{15} - 2 = 32768 - 2 = 32766$

- 1.11 Τα 4 υποδίκτυα, πρέπει να έχουν αντίστοιχα τμήμα host μήκους  $\lceil \log_2(100 + 2) \rceil = 7$  bits,  $\lceil \log_2(60 + 2) \rceil = 6$  bits,  $\lceil \log_2(20 + 2) \rceil = 5$  bits,  $\lceil \log_2(10 + 2) \rceil = 4$  bits. Σπάμε ιεραρχικά υπόδίκτυα σε υποδίκτυα. Οπότε αναθέτω το 10.11.12.0/25 στο 1ο υποδίκτυο, το 10.11.12.128/26 στο 2ο υποδίκτυο, το 10.11.12.192/27 στο 3ο υποδίκτυο και τέλος το 10.11.12.224/28 στο 4ο υποδίκτυο.

- 1.12 Ναι υπάρχει χώρος για ακόμα 1 υποδίκτυο, το 10.11.12.208/28 με επιτρεπόμενο πλήθος hosts  $2^4 - 2 = 14$ .

- 1.13 4 = 0b0100

5 = 0b0101

6 = 0b0110

7 = 0b0111

8 = 0b1000

Συνεπώς, τα 5 block διευθύνσεων μπορούν να συντημηθούν σε: 171.12.4.0/22 και 171.12.8.0/24.

### Άσκηση 2

- 2.1 Στα PC1, PC2, PC3, PC4:

hostname PCX

ifconfig em0 192.168.1.X/Y up όπου  $(X/Y) = (1/24), (2/28), (18/24), (29/28)$  αντίστοιχα.

- 2.2** Ναι. Τα μηχανήματα βρίσκονται στο ίδιο LAN οπότε προκειμένου να μπορούν να ανταλλάξουν πλαίσια, πρέπει να έχουν διαφορετικές MAC διεύθυνσεις.
- 2.3** Λειτουργούν επιτυχώς προς το PC2, PC3, αποτυγχάνει προς το PC3 (χωρίς όμως μήνυμα No route to host).
- 2.4** Αποτυγχάνουν και τα δύο, με μήνυμα No route to host.
- 2.5** Λειτουργούν επιτυχώς προς το PC3, αποτυγχάνει προς τα PC1, PC2 με μήνυμα No route to host).
- 2.6** Λειτουργεί επιτυχώς προς το PC1, αποτυγχάνει προς το PC2 (χωρίς όμως μήνυμα No route to host).
- 2.7** Το PC1 και το PC3 θεωρούν ότι βρίσκονται στο ίδιο υποδίκτυο με όλα τα μηχανήματα, οπότε προωθούν πλαίσια προς τις διεπαφές τους όταν κάνουν ping προς αυτά. Έτσι τα Echo Request φθάνουν σε όλα τα μηχανήματα, προερχόμενα από τα PC1, PC3.
- Ωστόσο, το PC2 θεωρεί ότι βρίσκεται στο ίδιο υποδίκτυο 192.168.1.0/28 μόνο με το PC1 (ίδιος αριθμός δικτύου), έτσι όταν έρθει ώρα να στείλει Echo Reply προς το PC3 δεν ξέρει προς που να το στείλει (θα έπρεπε να υπάρχει εγγραφή στον πίνακα δρομολόγησης), και επίσης, όταν είναι να στείλει Echo Request, ομοίως, οπότε δείχνει No route to host.
- Όμοια, το PC4 θεωρεί ότι βρίσκεται στο ίδιο υποδίκτυο 192.168.1.16/28 μόνο με το PC4, οπότε συμβαίνει ακριβώς το ίδιο φαινόμενο.
- 2.8** Το πακέτο Echo Reply παραδίδεται στο στρώμα δικτύου. Έτσι όταν τα στρώματα δικτύου των PC2 και PC4 παραλάβουν πακέτα προς τα PC3 και PC1 αντίστοιχα, δεν γνωρίζουν που να στείλουν το πακέτο (καθώς θεωρούν ότι δεν μπορούν να το στείλουν απευθείας), οπότε το απορρίπτουν.
- 2.9** Στα PC1, PC2, PC3, PC4:
- ```
ifconfig em0 192.168.1.X/28 up όπου X = 1,2,18,29 αντίστοιχα (αφού 255.255.255.240 = (1 << 28) - 1)
```
- 2.10** Επιτυγχάνουν μόνο τα ping μεταξύ των PC1 ↔ PC2 και PC3 ↔ PC4, και αποτυγχάνουν όλα τα άλλα με μήνυμα No route to host.
- 2.11** Παρατηρείται μήνυμα No route to host.

## Άσκηση 3

- 3.1** Σταν R1:
- ```
hostname R1
ifconfig em0 192.168.1.14/28 up
ifconfig em1 192.168.1.17/28 up
```
- 3.2** Από τις ρυθμίσεις των μηχανημάτων PC3, PC4 αντίστοιχα στο VirtualBox.
- 3.3** Παρατηρούμε ARP Request, ARP Reply, ICMP Echo Request, ICMP Echo Reply. Το ping επιτυγχάνει.
- 3.4** Όμοια, παρατηρούμε ARP Request, ARP Reply, ICMP Echo Request, ICMP Echo Reply. Το ping επιτυγχάνει.
- 3.5** Με καταγραφή στο ίδιο το PC1 (σε άλλο τερματικό από εκεί που γίνεται το ping), παρατηρούμε ότι δεν παράγεται κανένα πλαίσιο. Το στρώμα δικτύου στο PC1, μόλις παραλάβει το πακέτο προς το PC3, δεν έχει σχετική εγγραφή στους πίνακες δρομολόγησης τους, οπότε αδυνατεί να στείλει οτιδήποτε, και έτσι βλέπουμε και το σχετικό μήνυμα No route to host στο ping.
- 3.6** Όμοια, με καταγραφή στο ίδιο το PC3 (σε άλλο τερματικό από εκεί που γίνεται το ping), παρατηρούμε ότι δεν παράγεται κανένα πλαίσιο. Το στρώμα δικτύου στο PC3, μόλις παραλάβει το πακέτο προς το PC1, δεν έχει σχετική εγγραφή στους πίνακες δρομολόγησης τους, οπότε αδυνατεί να στείλει οτιδήποτε, και έτσι βλέπουμε και το σχετικό μήνυμα No route to host στο ping.
- 3.7** Βλ. ερωτήματα 3.5 και 3.6 όπου απαντάται.
- 3.8** Στο PC1 υπάρχουν ARP εγγραφές (αντιστοιχήσεις IPv4 διεύθυνσης - MAC διεύθυνσης) για το PC1 και την διεπαφή του R1 στο LAN1.
- 3.9** Μόνο για το ίδιο το PC2.
- 3.10** Για το PC1 και τις δύο διεπαφές του R1.

**3.11** Οι ARP εγγραφές για τις δύο διεπαφές του R1 παραμένουν.

**3.12** (εκτελούνται οι εντολές)

**3.13** Υπάρχουν εγγραφές για το PC1, για το PC2 και τις δύο διεπαφές του R1.

Οι εγγραφές για τις δύο διεπαφές του R1 εισήχθησαν αυτόματα, η εγγραφή για το PC1 εισήχθη λόγω του ARP Reply που απέστειλε το PC1 προς τον R1 όταν έγινε το αντίστοιχο ping και, όμοια, η εγγραφή για το PC3 εισήχθη λόγω του ARP Reply που απέστειλε το PC2 προς τον R1 όταν έγινε το αντίστοιχο ping.

**3.14** Στο PC1 υπάρχει ARP εγγραφή για το ίδιο και για την διεπαφή στο LAN1 του R1.

Στο PC2 υπάρχει ARP εγγραφή για το ίδιο και για την διεπαφή στο LAN1 του R1.

**3.15** Προστίθεται εγγραφές για τα PC3, PC4.

**3.16** Από τον πίνακα ARP του R1:

Host - Υποδίκτυο	IPv4 διεύθυνση	MAC διεύθυνση
PC1-LAN1	192.168.1.1	08:00:27:e6:40:e4
PC2-LAN1	192.168.1.2	08:00:27:3f:6b:2d
PC3-LAN2	192.168.1.18	08:00:27:1e:50:74
PC4-LAN2	192.168.1.29	08:00:27:9a:34:15
R1-LAN1	192.168.1.14	08:00:27:00:f2:a5
R1-LAN2	192.168.1.17	08:00:27:20:f9:0e

**3.17** Παράγονται ARP Requests, στα οποία όμως κανείς δεν απαντά αφού κανείς host δεν έχει αυτή την διεύθυνση IP. Συνεπώς ο R1, μην γνωρίζοντας πού να στείλει το ICMP πακέτο, δεν το αποστέλλει.

Προσέχουμε ότι δεν βλέπουμε μήνυμα `Host is down` διότι για να χαρακτηρίσει το ARP έναν Host ως down, περιμένει να μην λάβει απάντηση σε ARP Requests πλήθος φορών ίσο με `net.link.ether.inet.maxtries`, που έχει default τιμή 5 η οποία ισχύει και έδω όπως βλέπει κανείς με `sysctl net.link.ether.inet.maxtries`, την στιγμή που εμείς στείλαμε μόνο 3 requests.

**3.18** Περιέχει εγγραφή για την διεύθυνση 192.168.1.5 χωρίς MAC διεύθυνση, η οποία είναι, δηλαδή, `incomplete`.

Η εγγραφή αυτή παραμένει στην ARP μνήμη για χρόνο πλήθους δευτερολέπτων ίσο με `net.link.ether.inet.wait`. Η μεταβλητή αυτή έχει default τιμή 20, όπως μπορεί να δει κανείς και με `sysctl net.link.ether.inet.wait`. Πράγματι, αν περιμένουμε λίγο και ξαναδοούμε την ARP μνήμη, η εγγραφή για την διεύθυνση 192.168.1.5 έχει διαγραφεί.

**3.19** Παρατηρούμε ότι μόλις φτάσουμε τα 6 αιτήματα, βλέπουμε μήνυμα `Host is down`. Όπως εξηγήθηκε στο ερώτημα 3.17, μετά τα 5 ARP Requests, δηλαδή στο 6ο αίτημα, θεωρεί το ARP ότι ο host είναι down, για αυτό και μόνο στο 6ο ping βλέπουμε το μήνυμα αυτό.

## Άσκηση 4

**4.1** Στον R1:

```
sysctl net.inet.ip.forwarding=1
```

**4.2** Στον R1:

```
echo 'gateway_enable="YES"' >>/etc/rc.conf
```

**4.3** Όχι, εξακολουθεί να αποτυγχάνει, διότι το στρώμα δικτύου του PC1 δεν γνωρίζει ότι πρέπει να δρομολογήσει τα πακέτα εκτός του τοπικού υποδικτύου του προς τον R1.

**4.4** Όχι, υπάρχει μόνο προς το τοπικό υποδίκτυο 192.168.1.0/28, δηλαδή το LAN1. (το LAN2 είναι το 192.168.16.0/28)

**4.5** Στο PC1:

```
route add default 192.168.1.14
```

**4.6** Προστέθηκε η εξής εγγραφή:

**Destination** default, που σημαίνει ότι είναι η προκαθορισμένη πύλη.

**Gateway** 192.168.1.14, που είναι η IP διεύθυνση του R1.

**Flags** UGS, όπου το U σημαίνει πως η εγγραφή είναι ενεργή, το G σημαίνει πως μεσολαβούν δρομολογητές και το S σημαίνει πως είναι εγγραφή που προστέθηκε στατικά στον πίνακα δρομολόγησης.

- 4.7** Το ping δεν είναι επιτυχημένο, ωστόσο πια δεν βλέπουμε μήνυμα `No route to host`, αλλά απλά δεν έρχονται ICMP Echo Replies.

Στην πραγματικότητα, αυτό που συμβαίνει είναι πως τα ICMP Echo Requests φθάνουν στο PC3, αλλά το στρώμα δικτύου του PC3, όταν παραλάβει το ICMP Echo Reply με παραλήπτη το PC1, δεν γνωρίζει πώς να στείλει πακέτα προς το PC1 (δεν υπάρχει σχετική εγγραφή στον πίνακα δρομολόγησης του).

- 4.8** Στον R1 με `tcpdump -i em0 -e -n` βλέπουμε ότι εκπέμπονται στο LAN1 πακέτα ICMP με IP και MAC αποστολέα το PC1, IP παραλήπτη το PC3, και MAC παραλήπτη τον R1.

Στον R1 με `tcpdump -i em1 -e -n` βλέπουμε ότι εκπέμπονται στο LAN2 πακέτα ICMP με IP αποστολέα το PC1, IP παραλήπτη το PC3, MAC αποστολέα το R1 και MAC παραλήπτη το PC3.

Και στις δύο περιπτώσεις πρόκειται για το ίδιο πακέτο ICMP Echo Request.

Η διαφορά βρίσκεται στο ότι πλέον, εκπέμπονται ICMP Echo Requests και αυτά φθάνουν στο LAN2.

- 4.9** Στο PC3:

```
route add default 192.168.1.17
```

- 4.10** Το ping επιτυγχάνει!

Σε συνέχεια της εξήγησης στο ερώτημα 3.7, πλέον το στρώμα δικτύου του PC3 γνωρίζει πώς να στείλει πακέτα προς το PC1, οπότε τα ICMP Echo Replies αποστέλλονται προς τον R1, που με την σειρά του τα στέλνει στο PC1.

- 4.11** Στο PC1:

```
tracert -I 192.168.1.18
```

Παρατηρούμε ότι υπάρχει δύο βήματα, ένα μέχρι την διεπαφή του R1 στο LAN1, και ένα μέχρι το PC3.

- 4.12** Στα PC1, PC3, R1:

```
arp -a -d
```

- 4.13** Σε ένα εικονικό τερματικό στον R1 τρέχουμε:

```
tcpdump -i em0 -e -n -vv
```

και σε ένα δεύτερο εικονικό τερματικό, επίσης στον R1, τρέχουμε:

```
tcpdump -i em1 -e -n -vv
```

- 4.14** Στο PC1:

```
ping -c 1 192.168.1.18
```

- 4.15** MAC προέλευσης: 08:00:27:e6:40:e4 (PC1)

MAC προορισμού: 08:00:27:00:f2:a5 (R1-LAN1)

IPv4 προέλευσης: 192.168.1.1 (PC1)

IPv4 προορισμού: 192.168.1.18 (PC3)

- 4.16** MAC προέλευσης: 08:00:27:20:f9:0e (R1-LAN2)

MAC προορισμού: 08:00:27:1e:50:74 (PC3)

IPv4 προέλευσης: 192.168.1.1 (PC1)

IPv4 προορισμού: 192.168.1.18 (PC3)

- 4.17** Οι διευθύνσεις IP μένουν ίδιες και αντιστοιχούν στον αποστολέα που αρχικά έστειλε το πακέτο, και στον παραλήπτη που τελικά είναι να παραλάβει το πακέτο.

Αντίθετα, οι MAC διευθύνσεις, αλλάζουν κάθε φορά που γίνεται hop, δηλαδή κάθε φορά που το πακέτο περνά σε άλλο υπόδίκτυο. (η αλλαγή αυτή γίνεται από τον δρομολογητή)

**4.18** Στο ttyv1@PC1:

```
ssh lab@192.168.1.18
```

Υπενθυμίζεται, ότι από default δεν επιτρέπεται ssh στον χρήστη root.

**4.19** Στο ttyv0@PC1:

```
netstat -an | grep 192.168.1.18
```

Χρησιμοποιείται το πρωτοκόλλο μεταφοράς TCP, με τοπική θύρα (θύρα πελάτη) την 32654, και απομακρυσμένη θύρα (θύρα εξυπηρετητή) την 22.

**4.20** Στον R1:

```
netstat -p tcp
```

Παρατηρούμε ότι δεν υπάρχει καμία σύνδεση TCP στον R1. Αυτό είναι λογικό, καθώς η δρομολόγηση γίνεται στο στρώμα δικτύου, και σε επίπεδο IP πακέτων, οπότε ο δρομολογητής δεν έχει κάποιον ρόλο σχετικά με την TCP σύνδεση που δημιουργείται end-to-end μεταξύ PC1 - PC3, απλά μεταδίδει τα IP πακέτα που δημιουργούνται.

## Άσκηση 5

**5.1** Στο PC3:

```
ifconfig em0 192.168.1.18/29
```

Στα PC1, PC2:

```
route add default 192.168.1.14
```

Στα PC3, PC4:

```
route add default 192.168.1.17
```

**5.2** Στα PC1, PC2, PC3, PC4, R1:

```
arp -a -d
```

**5.3** Στον R1:

```
tcpdump -i em0 -n 'icmp or arp'
```

**5.4** Στον PC4:

```
tcpdump -i em0 -n 'icmp or arp'
```

**5.5** Και τα 3 pings είναι επιτυχημένα.**5.6** Οι πίνακες ARP περιέχουν τις ακόλουθες εγγραφές ανά μηχανήμα:

**PC1** PC1, PC2, R1-LAN1

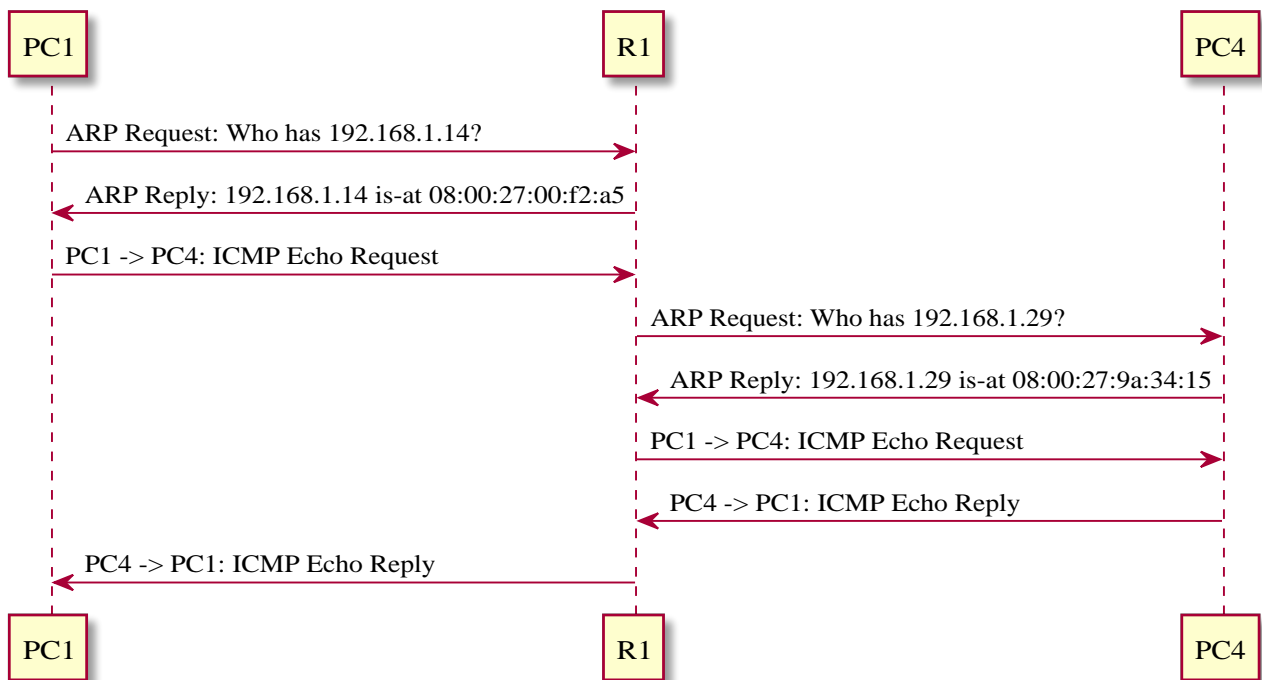
**PC2** PC2, PC1

**PC3** PC3, R1-LAN2

**PC4** PC4, R1-LAN2

**R1** R1-LAN1, R1-LAN2, PC1, PC3, PC4

**5.7**



Σε όλα τα μηχανήματα: `arp -a -d`

Στα PC3, PC4: `tcpdump -i em0 -e -n 'arp or icmp'`

Στο R1: `tcpdump -i em1 -e -n 'arp or icmp'`

- 5.9** Ήταν επιτυχές. Το ping μας δείχνει TTL=64. Είναι σκόπιμο να αναφέρουμε πως το TTL αυτό αφορά το Echo Reply που στέλνει το PC4 προς το PC3. Η πορεία του Echo Request είναι PC3 → R1 → PC4, καθώς ο πίνακας δρομολόγησης του PC3 αναφέρει πως απευθείας στο LAN2 στέλνει προς διευθύνσεις 192.168.1.16/29 και η 192.168.1.29 δεν είναι τέτοια διεύθυνση. Αντίθετα η πορεία του Echo Reply είναι PC4 → PC3 (απευθείας μέσω του LAN2) καθώς ο πίνακας δρομολόγησης του PC4 αντίθετα αναφέρει πως απευθείας στο LAN2 στέλνει προς διευθύνσεις 192.168.1.16/28 και η 192.168.1.18 είναι τέτοια διεύθυνση. Πρακτικά, το PC3 θεωρεί λανθασμένα πως το PC4 δεν είναι στο ίδιο υποδίκτυο μαζί του, οπότε προωθεί τα πακέτα προς τον R1, ενώ το PC4 γνωρίζει την πραγματική κατάσταση, οπότε δεν προωθεί προς τον R1, αλλά στέλνει απευθείας προς το PC3.

**Συμπλήρωση:** Θεωρητικά στην κατάσταση αυτή ο R1 πρέπει να εκδώσει ένα ICMP Redirect προς το PC3. Ωστόσο τα FreeBSD 13.2 και FreeBSD 14.0 για απροσδιόριστο λόγο δεν το εκδίδουν. Δοκιμάστηκε Debian Linux 12 Bookworm στον R1, και πράγματι το ICMP Redirect εξεδόθη.

**Συμπλήρωση No.2 - bug:** Στην περίπτωση που εκδίδει το ICMP Redirect το FreeBSD 10.3 δεν βλέπουμε σχετικό μήνυμα στο ping, ενώ αν το εκδώσει το Debian βλέπουμε σχετικό μήνυμα. Διερευνώντας για ποιον λόγο το ping αποτυγχάνει, **βρέθηκε bug στον κώδικα του ping** και συγκεκριμένα στο αρχείο `sbin/ping/ping.c` του πηγαίου κώδικα του FreeBSD. Πιο συγκεκριμένα γίνονται κάποιες λάθος συγκρίσεις. Το FreeBSD όταν στέλνει μηνύματα ICMP τοποθετεί σε αυτά τα πρώτα 28 bytes του IP πακέτου (νόμιμη συμπεριφορά βάση του RFC 792), ενώ το Linux τοποθετεί το πλήρες IP πακέτο. Επειδή οι συγκρίσεις για το μέγεθος του ICMP συμβαίνουν λάθος, αντί το ping, να συγκρίνει αν το ICMP που το ερχεται έχει μέγεθος τουλάχιστον 28 bytes, το συγκρίνει με ένα μεγαλύτερο αριθμό, αυτό έχει ως αποτέλεσμα να θεωρεί το πακέτο αλλοιωμένο, ενώ δεν είναι.

Κάνοντας την εξής αλλαγή και κάνοντας recompile το ping, το μήνυμα ICMP εμφανίζεται κανονικά!

```

1 diff --git a/sbin/ping/ping.c b/sbin/ping/ping.c
2 index 2fc876e50..666c1b4e6 100644
3 --- a/sbin/ping/ping.c
4 +++ b/sbin/ping/ping.c
5 @@ -1328,7 +1328,7 @@ pr_pack(char *buf, ssize_t cc, struct sockaddr_in *from, struct
6     ↪ timespec *tv)
7     * ICMP header then stop.
8     */
9     if (icmp_data_raw_len <
  
```

```

9  - (ssize_t)(sizeof(struct ip) + sizeof(struct icmp))) {
10 + (ssize_t)(sizeof(struct ip) + ICMP_MINLEN)) {
11      if (options & F_VERBOSE)
12          warnx("quoted data too short (%zd bytes) from %s",
13                icmp_data_raw_len, inet_ntoa(from->sin_addr));
14  @@ -1351,7 +1351,7 @@ pr_pack(char *buf, ssize_t cc, struct sockaddr_in *from, struct
    ↪ timespec *tv)
15      * quoted packets carrying IP options.
16      */
17      if (icmp_data_raw_len <
18          (ssize_t)(oip_header_len + sizeof(struct icmp))) {
19 + (ssize_t)(oip_header_len + ICMP_MINLEN)) {
20          if (options & F_VERBOSE)
21              warnx("inner packet too short (%zd bytes) from %s",
22                    icmp_data_raw_len, inet_ntoa(from->sin_addr));
23  @@ -1360,7 +1360,7 @@ pr_pack(char *buf, ssize_t cc, struct sockaddr_in *from, struct
    ↪ timespec *tv)
24
25      memcpy(&oip, icmp_data_raw, sizeof(struct ip));
26      oicmp_raw = icmp_data_raw + oip_header_len;
27  - memcpy(&oicmp, oicmp_raw, sizeof(struct icmp));
28 + memcpy(&oicmp, oicmp_raw, MIN(cc, (ssize_t)sizeof(struct icmp)));
29
30      if (((options & F_VERBOSE) && uid == 0) ||
31          (!(options & F_QUIET2) &&

```

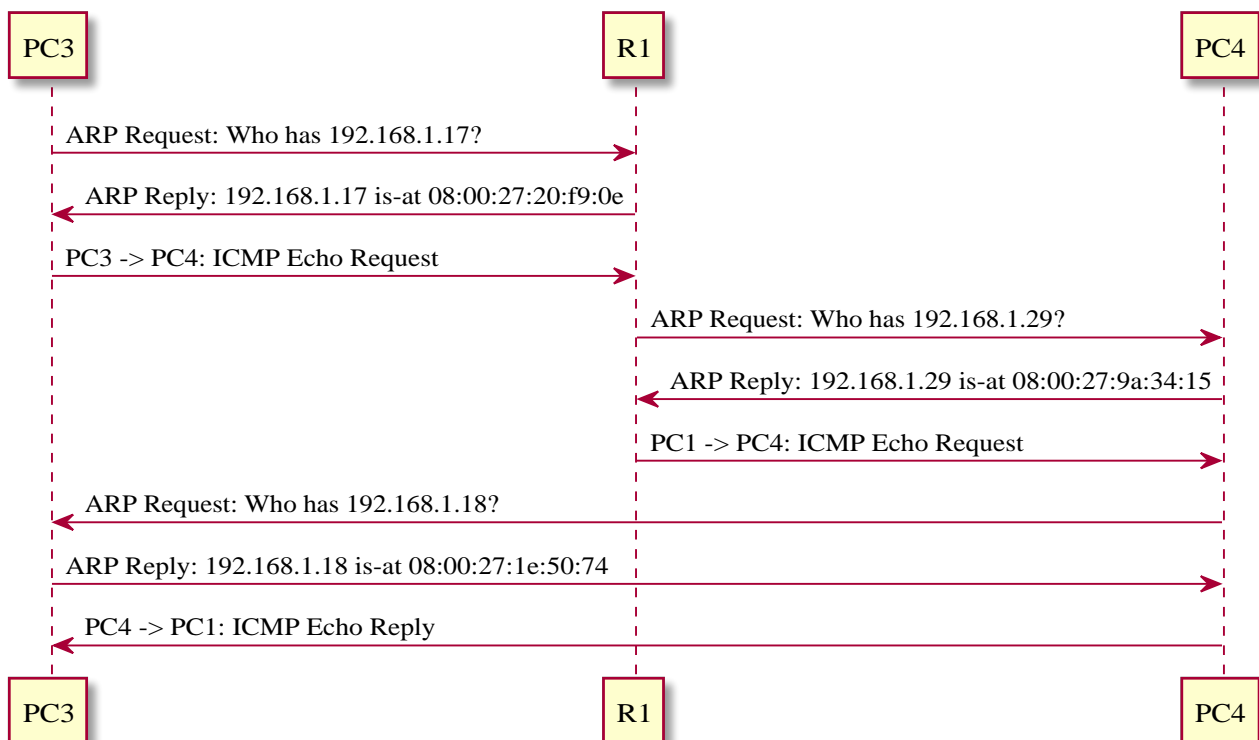
5.10 Οι πίνακες ARP περιέχουν τις ακόλουθες εγγραφές ανά μηχανήμα:

**PC3** PC3, R1-LAN2

**PC4** PC4, PC3, R1-LAN2

**R1-LAN2** R1-LAN2, PC3, PC4

5.11



5.12 Το PC3 αναζητά την MAC διεύθυνση του R1 στην διεπαφή του στο LAN2, ενώ το PC4 αναζητά την MAC διεύθυνση του PC3. Αυτό συμβαίνει για τον λόγο που περιγράφεται στο ερώτημα

[5.13] Εξήγηθηκε και στο ερώτημα .

Το PC3 θεωρεί ότι βρίσκεται στο υποδίκτυο 192.168.1.16/29, οπότε θεωρεί λανθασμένα ότι το PC4 με διεύθυνση 192.168.1.29 δεν βρίσκεται στο υποδίκτυό του. Αυτό αποτυπώνεται στον πίνακα δρομολόγησης που έχει εγγραφή για να προωθεί απευθείας στην em0 τα πακέτα με IP προορισμό 192.168.1.16/29. Έτσι, επικρατεί ο κανόνας της προκαθορισμένης πύλης, και το πακέτο προωθείται προς τον R1, οπότε εκδίδεται ένα ARP Request για το R1.

**5.14** Το προωθεί σε όλες τις περιπτώσεις. Όπως εξηγήθηκε στο ερώτημα 5.9, κάποια ΛΣ εκδίδουν και ICMP Redirect, για να ενημερώσουν το PC3 πως θα μπορούσε να είχε στείλει στο PC4 απευθείας.

**5.15** Απευθείας.

**5.16** Στα PC3, PC4, R1:

```
tcpdump -i emX -e -n -vv 'icmp' όπου X = 0 για PC3, PC4 και X = 1 για R1.
```

**5.17** Βλ. ερώτημα 5.9 για περιγραφή του τι συμβαίνει.

**5.18** Στο PC3:

```
ifconfig em0 192.168.1.18/28 up
```

Η προκαθορισμένη πύλη διαγράφεται.

**5.19** 192.168.1.18 → lo0

192.168.1.26/29 → em0 — προς το 192.168.1.17

192.168.1.16/28 → em0 — απευθείας

**5.20** Στα PC3, PC4, R1:

```
tcpdump -i emX -e -n -vv 'icmp' όπου X = 0 για PC3, PC4 και X = 1 για R1.
```

Μόλις φθάσει το Echo Request στον R1, ο R1 εκδίδει ICMP Redirect και προωθεί το πακέτο προς το PC4.

Το PC3 λαμβάνει το ICMP Redirect και προσθέσει στον πίνακα δρομολόγησης μια εγγραφή για το 192.168.1.29 όπου σημειώνει πως στέλνει στο em0 και συγκεκριμένα στο 192.168.1.29.

**5.21** Έχει προστεθεί η εγγραφή με προορισμό το 192.168.1.29, η οποία δηλώνει πως το πακέτο προς αυτό πρέπει να αποστέλλονται απευθείας προς τον 192.168.1.29 στο em0.

**5.22** Η επικοινωνία αυτή λειτουργεί, αλλά μόνο μονόδρομα, δηλαδή μηχανήματα του LAN1 μπορούν να στείλουν στο PC3 και αυτό λαμβάνει, όμως το PC3 δεν μπορεί να στείλει προς μηχανήματα του LAN1. Αυτό συμβαίνει, διότι το PC3 δεν έχει προκαθορισμένη πύλη, ούτε κάποιον κανόνα δρομολόγησης για τις διευθύνσεις του δικτύου LAN1, οπότε όταν επιχειρεί να στείλει προς διεύθυνση του LAN1, το στρώμα δίκτυου βγάζει το σφάλμα No route to host.

**5.23** Στο PC3:

```
route add default 192.168.1.14
```

Λόγω του ICMP Redirect που παρελήφθη πριν, υπάρχει κανόνας συγκεκριμένα για το 192.168.1.29. Οπότε επιλέγεται αυτός και έτσι το πακέτο στέλνεται απευθείας προς το 192.168.1.29 μέσω της em0.

Αν όμως, αυτός ο κανόνας δεν υπήρχε (αν π.χ. τον διαγράψουμε με `route delete 192.168.1.29`) τότε επιλέγεται ο κανόνας που έχει το μέγιστο prefix, δηλαδή ο 192.168.1.16/29 που δημιουργήσαμε πριν και όχι ο κανόνας της προκαθορισμένης πύλης, και έτσι το πακέτο αποστέλλεται και πάλι προς τον R1. (βέβαια, στην συνέχεια ο R1, θα αποστείλει και πάλι ICMP Redirect, οπότε η σχετική εγγραφή στον πίνακα δρομολόγησης για το 192.168.1.29 δημιουργείται εκ νέου)

## Άσκηση 6

**6.1** Στον R1:

```
ifconfig bridge0 create addm em0 addm em1 up
```

**6.2** Στο PC1:

```
ifconfig em0.5 create 192.168.5.1/24 up
```

```
ifconfig em0.6 create 192.168.6.1/24 up
```



**6.3** Στο PC2:

```
ifconfig em0.5 create 192.168.5.2/24 up
```

**6.4** Στο PC3:

```
ifconfig em0.6 create 192.168.6.18/24 up
```

**6.5** Στο PC4:

```
ifconfig em0.5 create 192.168.5.29/24 up
```

**6.6** Στον R1:

```
ifconfig em0.5 create up  
ifconfig em0.6 create up  
ifconfig em1.5 create up  
ifconfig em1.6 create up
```

**6.7** Όχι. Μπορούμε στην φυσική διεπαφή του (192.168.1.1) και σε εκείνη του VLAN6 (192.168.6.1), αλλά όχι σε εκείνη του VLAN5 (192.168.5.1)**6.8** Όχι. Μπορούμε στην φυσική διεπαφή του (192.168.1.1) και σε εκείνη του VLAN5 (192.168.5.1), αλλά όχι σε εκείνη του VLAN6 (192.168.6.1)**6.9** Το ping προς το 192.168.1.1 πέτυχε διότι το πακέτο δρομολογήθηκε μέσω του δρομολογητή R1 στο επίπεδο δικτύου.

Τα ping των PC3, PC4 προς τα 192.168.6.1, 192.168.5.1 αντίστοιχα, πέτυχαν διότι τα πλαίσια των μηνυμάτων μεταχθήκαν από τον R1 ως γέφυρα, στο επίπεδο ζεύξης δεδομένων (από το bridge0).

Όταν τα PC3, PC4 κάνουν ping προς τα 192.168.5.1, 192.168.6.1 αντίστοιχα, τότε με βάση τους πίνακες δρομολόγησης, τα αντίστοιχα πακέτα πρέπει να δρομολογηθούν προς την προκαθορισμένη πύλη (αφού το PC3 δεν έχει διεπαφή στο VLAN6 και το PC4 δεν έχει διεπαφή στο VLAN5). Ο R1, ως δρομολογητής, δεν γνωρίζει σχετικά με τα δίκτυα 192.168.5.0/24 και 192.168.6.0/24, οπότε δεν γνωρίζει τι να κάνει αυτά τα πακέτα και έτσι το απορρίπτει. Μάλιστα ο R1 στέλνει και Destination Host Unreachable προς το PC3.

**Συμπλήρωση:** Το ping δεν δείχνει το Destination Host Unreachable, λόγω του **bug** που περιγράφηκε στην απάντηση στο ερώτημα 5.9. Κάνοντας την προτεινόμενη αλλαγή, όπως περιγράφεται στην απάντηση στο ερώτημα 5.9, το σχετικό μήνυμα ICMP εμφανίζεται φυσιολογικά.

**6.10** Ναι.

Προς την διεπαφή VLAN6 του PC1 (192.168.6.1) επιτυγχάνει διότι τα PC1, PC3 είναι πράγματι στο ίδιο υποδίκτυο VLAN6.

Προς την κανονική διεπαφή του PC1 στο LAN1 (192.168.1.1) επιτυγχάνει, διότι:

- Το PC3 δεν έχει εγγραφή στον πίνακα δρομολόγησης για το PC1, οπότε στέλνει το Echo Request στην προκαθορισμένη πύλη, που είναι το PC1. Έτσι εκπέμπεται στο VLAN6 ένα IP πακέτο με αποστολέα 192.168.6.18 και παραλήπτη 192.168.1.1.
- Το PC1 λαμβάνει το μήνυμα και επειδή η διεύθυνση είναι δική του, δεν χρειάζεται να το δρομολογήσει, αλλά το στέλνει προς επεξεργασία στο ΛΣ απευθείας.
- Το PC1 πρέπει να απαντήσει Echo Reply τώρα. Το Echo Reply έχει διεύθυνση παραλήπτη 192.168.6.18. Σύμφωνα με τους κανόνες δρομολόγησης στο PC1, αυτό το πακέτο αποστέλλεται απευθείας στο VLAN6. Έτσι φθάνει στο PC3.

**6.11** Ναι. Η επικοινωνία με την διεπαφή του PC2 στο VLAN5 (192.168.5.2) γίνεται απευθείας μέσω του VLAN5, ενώ η επικοινωνία με την κανονική διεπαφή του PC2 στο LAN1 (192.168.1.2) γίνεται μέσω του δρομολογητή R1.**6.12** Δεν μπορούμε με καμία. Για κάνα από τα δίκτυα 192.168.1.0/28 και 192.168.5.0/24 δεν υπάρχει κανόνας δρομολόγησης στο PC3, οπότε επιλέγεται η προκαθορισμένη πύλη, που είναι η διεπαφή του PC1 στο VLAN5. Όμως στο PC1 δεν επιτρέπονται οι προωθήσεις πακέτων (η μεταβλητή `net.inet.ip.forwarding=0`), και έτσι η επικοινωνία αποτυγχάνει.**6.13** Στο PC1:

```
sysctl net.inet.ip.forwarding=1
```

Στο PC2:

```
route change default 192.168.1.1
```

**6.14** Ναι.

Τα Echo Request, Echo Reply προς 192.168.5.2 κάνουν την εξής διαδρομή:

PC3-VLAN6 ↔ PC1-VLAN6 ↔ PC1-VLAN5 ↔ PC2-VLAN5.

(σημειώνεται πως λόγω της διεπαφής em1 στο PC1 υπάρχει κανόνας δρομολόγησης για τις διευθύνσεις 192.168.5.0/24 απευθείας προς το VLAN5.)

Τα Echo Request, Echo Reply προς 192.168.1.2 κάνουν την εξής διαδρομή:

PC3-VLAN6 ↔ PC1-VLAN6 ↔ PC1-LAN1 ↔ PC2-LAN1.

(σημειώνεται πως το Echo Reply από το PC2 προωθείται απευθείας στο PC1 λόγω του κανόνα δρομολόγησης της προεπιλεγμένης πύλης.)

**6.15** Οι MAC διευθύνσεις είναι οι ακόλουθες:

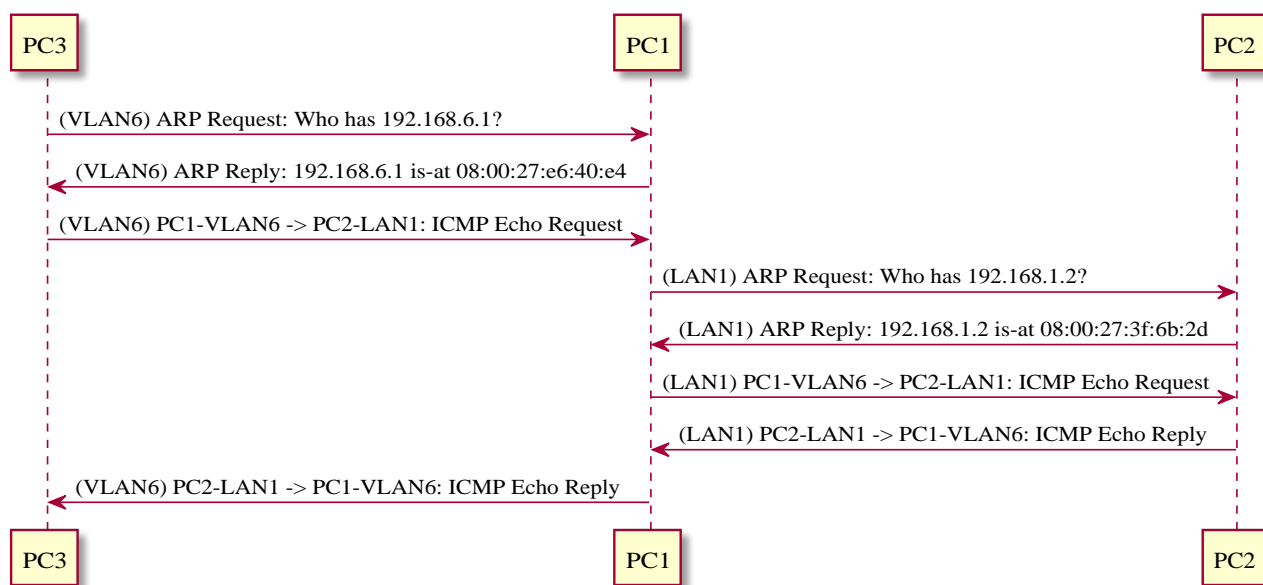
Host - Υποδίκτυο	MAC διεύθυνση
PC1-LAN1	08:00:27:e6:40:e4
PC2-LAN1	08:00:27:3f:6b:2d
PC3-LAN2	08:00:27:1e:50:74

Στα PC1, PC2, PC3:

```
arp -a -d
```

**6.16** Στα PC1, PC2, PC3:

```
tcpdump -i em0 -e -n
```

**6.17****6.18** Όχι.**6.19** Στο PC4:

```
tcpdump -i em0 -e -n
```

Το Echo Request ακολουθεί την πορεία PC3-VLAN6 → PC1-VLAN6 → PC4-VLAN5, οπότε δηλαδή φθάνει στο PC4.

Το PC4 εκδίδει Echo Reply με διεύθυνση αποστολέα 192.168.5.29 και διεύθυνση παραλήπτη 192.168.6.18. Το Echo Reply ακολουθεί την πορεία PC4-LAN1 → R1-LAN2. Ο R1 όμως δεν έχει κανόνα δρομολόγησης για το δίκτυο 192.168.6.0/24, οπότε δεν μπορεί να δρομολογήσει το πακέτο και έτσι το απορρίπτει (αυτό

συμβαίνει καθώς οι διεπαφές `em0.5` και `em0.6` δεν έχουν ρυθμιστεί σε IP διεύθυνση). Κανονικά το R1 θα εξέδιδε Destination Host Unreachable, όμως αυτό το μήνυμα θα έπρεπε να αποσταλλεί στο αποστολέα του IP πακέτου, δηλαδή στην διεύθυνση 192.168.5.29, για την οποία επίσης δεν υπάρχει κανόνας δρομολόγησης στον R1, οπότε δεν αποστέλλεται το ICMP μήνυμα.

#### 6.20 Στο PC4:

```
route change default 192.168.5.1
```

Πλέον το ping επιτυγχάνει. Το Echo Request ακολουθεί την ίδια διαδρομή που περιγράφηκε στο ερώτημα [6.19](#). Όμως, το Echo Reply ακολουθεί πια την διαδρομή PC4-VLAN5 → PC1-VLAN5 → PC3-VLAN6, οπότε φθάνει στο PC3 και έτσι βλέπουμε το ping να επιτυγχάνει.