



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

ΟΝΟΜΑΤΕΠΩΝΥΜΟ: ΓΚΟΥΜΕ ΛΑΟΥΡΕΝΤΙΑΝ

ΑΜ: 031 18 014

ΕΞΑΜΗΝΟ: 8^ο

ΟΜΑΔΑ: 3

ΕΡΓΑΣΤΗΡΙΟ ΔΙΚΤΥΩΝ ΥΠΟΛΟΓΙΣΤΩΝ



ΕΡΓΑΣΤΗΡΙΑΚΗ ΑΣΚΗΣΗ 4: ΕΙΣΑΓΩΓΗ ΣΤΗ ΔΡΟΜΟΛΟΓΗΣΗ

Άσκηση 1: Διευθύνσεις IP

1.1) Ο αριθμός δικτύου είναι ένα τμήμα της διεύθυνσης IP, το οποίο δηλώνει τον χώρο διευθύνσεων αυτού του δικτύου.

1.2) Επειδή έχουμε το /22, θα κάνουμε πρακτικά τη λογική πράξη AND μεταξύ του 192.220.147.2 και του 255.255.252.0, επομένως αριθμός δικτύου είναι το 192.220.144.0.

1.3) Μας έχει δοθεί το μπλοκ διευθύνσεων 198.20.0.0/22. Επομένως, μάς είναι διαθέσιμα 10 bits για υποδίκτυα και hosts. Θέλουμε τουλάχιστον 100 συσκευές ανά υποδίκτυο, επομένως αναζητούμε n τέτοιο ώστε $2^n > 100 \Rightarrow n = 7$. Άρα από τα 10 διαθέσιμα bits, τα 7 θα χρησιμοποιηθούν για hosts, επομένως μας μένουν 3 bits για υποδίκτυα, άρα συνολικά $2^3 = 8$ υποδίκτυα.

1.4) Η κλάση C.

1.5) Οι 10.50.10.10 και οι 192.168.56.207.

1.6) Σε περίπτωση που είχαμε ταίριασμα μήκους 32 bit, τα πακέτα θα στέλνονταν μέσω της κατάλληλης επαφής.

1.7) Έχουμε το δίκτυο 10.50.10.0/23 \rightarrow 00001010.00110010.00001010.00000000 με μάσκα 11111111.11111111.11111110.00000000, επομένως διεύθυνση εκπομπής είναι η 00001010.00110010.00001011.11111111 ή αλλιώς 10.50.11.255.

1.8) Το πρώτο byte έχει τιμή $208_{10} = 11010000$, επομένως ανήκει στην κλάση C.

1.9) Έχουν δεσμευτεί 17 bits για το δίκτυο, επομένως απομένουν $32-17 = 15$ bits για συσκευές, δίνοντας μας συνολικά $2^{15} = 32.768$ διαθέσιμες διευθύνσεις. (Στην πραγματικότητα 32.766, καθώς η πρώτη είναι δεσμευμένη για το δίκτυο και η τελευταία για broadcast).

1.10) Το Ε.Μ.Π. έχει διευθύνσεις που έχουν ως πρώτο byte το 147, επομένως είναι κλάσης B.

1.11) Αρχικά, χωρίζουμε το μπλοκ στις υποδιευθύνσεις 10.11.12.0/25 και 10.11.12.128/25. Εξ αυτών, αναθέτουμε στο δεύτερο υποδίκτυο τους 100 υπολογιστές (χωράει $128-2=126$). Χωρίζουμε το υποδίκτυο 10.11.12.0/25 στα υποδίκτυα 10.11.12.0/26 και 10.11.12.64/26, οπότε και αναθέτουμε στο υποδίκτυο 10.11.12.64/26 τους 60 υπολογιστές (χωράει $64-2=62$). Στη συνέχεια, χωρίζουμε το υποδίκτυο 10.11.12.0/26 στα υποδίκτυα 10.11.12.0/27 και 10.11.12.32/27, οπότε και

αναθέτουμε στο 2^ο εξ αυτών τους 20 υπολογιστές (χωράει $32-2=20$). Χωρίζουμε ξανά το πρώτο υποδίκτυο 10.11.12.0/27 σε 10.11.12.0/28 και 10.11.12.16/28. Αναθέτουμε στο 2^ο υποδίκτυο τους 10 υπολογιστές (χωράει $16-2=14$) και μας περισσεύει το υποδίκτυο 10.11.12.0/28.

1.12) Όπως είδαμε υπάρχει χώρος για ένα ακόμη υποδίκτυο, το οποίο μπορεί να χωρέσει 14 υπολογιστές.

1.13) Αναζητούμε στην πραγματικότητα τη θέση του δεξιότερου κοινού bit στο 3^ο byte των διευθύνσεων που δίνονται. Στο byte αυτό έχουμε τις εξής τιμές:

- 4 → 0000 0100
- 5 → 0000 0101
- 6 → 0000 0110
- 7 → 0000 0111
- 8 → 0000 1000

Παρατηρούμε πως εμφανίζεται το bit 0 στη 4^η θέση (ξεκινώντας από τα αριστερά), επομένως τα συμπυκνώνουμε στο μπλοκ 171.12.0.0/20. (Σημείωση: Διαφορετικά, θα μπορούσαμε να συντμήσουμε τις πρώτες 4 διευθύνσεις στο μπλοκ 171.12.4.0/22 και την τελευταία στο 171.12.8.0/21)

Άσκηση 2: Ένα απλό δίκτυο

2.1) Χρησιμοποιήθηκε η επιλογή “Generate new MAC addresses for all network adapters”, ούτως ώστε κάθε εικονική κάρτα δικτύου των μηχανών να έχει δική της MAC διεύθυνση. Αν αντιθέτως είχαμε για παράδειγμα 2 κάρτες δικτύου στο ίδιο δίκτυο με τις ίδιες MAC διευθύνσεις, τότε πρωτόκολλα όπως το DHCP για την απόδοση IP διευθύνσεων θα δυσλειτουργούσαν. Αυτό πιθανόν να συνέβαινε σε περίπτωση που χρησιμοποιούσαμε cloning, χωρίς να έχουμε ορίσει την παραπάνω επιλογή.

2.2) Καταγράφουμε τα αποτελέσματα παρακάτω με τη σειρά:

```
root@PC1:~ # ping -c 1 192.168.1.2
PING 192.168.1.2 (192.168.1.2): 56 data bytes
64 bytes from 192.168.1.2: icmp_seq=0 ttl=64 time=1.076 ms

--- 192.168.1.2 ping statistics ---
1 packets transmitted, 1 packets received, 0.0% packet loss
round-trip min/avg/max/stddev = 1.076/1.076/1.076/0.000 ms
```

```
root@PC1:~ # ping -c 1 192.168.1.18
PING 192.168.1.18 (192.168.1.18): 56 data bytes
64 bytes from 192.168.1.18: icmp_seq=0 ttl=64 time=2.196 ms

--- 192.168.1.18 ping statistics ---
1 packets transmitted, 1 packets received, 0.0% packet loss
round-trip min/avg/max/stddev = 2.196/2.196/2.196/0.000 ms
```

```
root@PC1:~ # ping -c 1 192.168.1.29
PING 192.168.1.29 (192.168.1.29): 56 data bytes

--- 192.168.1.29 ping statistics ---
1 packets transmitted, 0 packets received, 100.0% packet loss
```

2.3) Αντίστοιχα από το PC2:

```
root@PC2:~ # ping -c 1 192.168.1.18
PING 192.168.1.18 (192.168.1.18): 56 data bytes
ping: sendto: No route to host

--- 192.168.1.18 ping statistics ---
1 packets transmitted, 0 packets received, 100.0% packet loss
```

```
root@PC2:~ # ping -c 1 192.168.1.29
PING 192.168.1.29 (192.168.1.29): 56 data bytes
ping: sendto: No route to host

--- 192.168.1.29 ping statistics ---
1 packets transmitted, 0 packets received, 100.0% packet loss
```

2.4) Αντίστοιχα από το PC4:

```
root@PC4:~ # ping -c 1 192.168.1.1
PING 192.168.1.1 (192.168.1.1): 56 data bytes
ping: sendto: No route to host

--- 192.168.1.1 ping statistics ---
1 packets transmitted, 0 packets received, 100.0% packet loss
```

```
root@PC4:~ # ping -c 1 192.168.1.2
PING 192.168.1.2 (192.168.1.2): 56 data bytes
ping: sendto: No route to host

--- 192.168.1.2 ping statistics ---
1 packets transmitted, 0 packets received, 100.0% packet loss
```

```
root@PC4:~ # ping -c 1 192.168.1.18
PING 192.168.1.18 (192.168.1.18): 56 data bytes
64 bytes from 192.168.1.18: icmp_seq=0 ttl=64 time=2.117 ms

--- 192.168.1.18 ping statistics ---
1 packets transmitted, 1 packets received, 0.0% packet loss
round-trip min/avg/max/stddev = 2.117/2.117/2.117/0.000 ms
```

2.5) Αντίστοιχα, από το PC3:

```
root@PC3:~ # ping -c 1 192.168.1.1
PING 192.168.1.1 (192.168.1.1): 56 data bytes
64 bytes from 192.168.1.1: icmp_seq=0 ttl=64 time=1.949 ms

--- 192.168.1.1 ping statistics ---
1 packets transmitted, 1 packets received, 0.0% packet loss
round-trip min/avg/max/stddev = 1.949/1.949/1.949/0.000 ms
```

```
root@PC3:~ # ping -c 1 192.168.1.2
PING 192.168.1.2 (192.168.1.2): 56 data bytes

--- 192.168.1.2 ping statistics ---
1 packets transmitted, 0 packets received, 100.0% packet loss
```

2.6) Το μήνυμα “No route to host” εμφανίστηκε στα εξής ping:

- 1) PC2 προς το PC3
- 2) PC2 προς το PC4
- 3) PC4 προς το PC1
- 4) PC4 προς το PC2

Ο λόγος που αυτά απέτυχαν, είναι πως εάν βάλουμε τη μάσκα της διεύθυνσης του αποστολέα στη διεύθυνση του παραλήπτη, θα δούμε πως τα υποδίκτυα είναι διαφορετικά μεταξύ τους. Αναλυτικότερα:

- 1) Αρχικά, βρίσκουμε το υποδίκτυο του PC2, εφαρμόζοντας τη μάσκα του στη διεύθυνση του και παίρνουμε το 192.168.1.0. Στη συνέχεια, βρίσκουμε του PC3 και παίρνουμε 192.168.1.16, άρα $192.168.1.0 \neq 192.168.1.16$.
- 2) Το υποδίκτυο του PC4 με τη μάσκα του PC2 γίνεται $192.168.1.16 \neq 192.168.1.0$.
- 3) Το υποδίκτυο του PC4 είναι το 192.168.1.16. Εφαρμόζοντας τη μάσκα του στη διεύθυνση του PC1 παίρνουμε το 192.168.1.0, άρα είναι επομένως σε διαφορετικά υποδίκτυα.
- 4) Εφαρμόζοντας τη μάσκα του υποδικτύου του PC4 στο PC2 παίρνουμε 192.168.1.0, το οποίο είναι διαφορετικό του 192.168.1.16.

2.7) Δε λαμβάνουμε απάντηση στο ping στις εξής περιπτώσεις:

- 1) PC1 προς PC4
- 2) PC3 προς PC2

Όπως είδαμε από το 2.6, το PC4 αδυνατεί να στείλει στο PC1 και αντίστοιχα το PC2 αδυνατεί να στείλει στο PC3.

2.8) Εκτελούμε τις εντολές ανάθεσης διευθύνσεων όπως πριν, αλλά με μάσκα το /28 αυτή τη φορά.

2.9) Αποτυγχάνουν πλέον τα ping:

1. Από το PC1 στο PC3 λαμβάνουμε πλέον “No route to host”
2. Από το PC3 στο PC1 λαμβάνουμε πλέον “No route to host”

2.10) Σχετικά με τα ping της 2.7, αυτά πλέον μας απαντάνε με “No route to host”.

Άσκηση 3: Ένα απλό δίκτυο

3.1) Μέσω των ρυθμίσεων του VirtualBox, στο πεδίο Network.

3.2) Καταγράφουμε τα παρακάτω εκτελώντας “tcpdump -i em0” στον R1, όπου και βλέπουμε πακέτα ARP (requests, reply και gratuitous) και ICMP (request και reply):

```
root@PC:~ # tcpdump -i em0
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on em0, link-type EN10MB (Ethernet), capture size 262144 bytes
21:37:45.641176 ARP, Request who-has 192.168.1.1 tell 192.168.1.1, length 46
21:37:45.641185 ARP, Request who-has 192.168.1.14 tell 192.168.1.1, length 46
21:37:45.641351 ARP, Request who-has 192.168.1.14 tell 192.168.1.14, length 28
21:37:45.641401 ARP, Reply 192.168.1.14 is-at 08:00:27:13:a7:15 (oui Unknown), length 28
21:37:45.641831 ARP, Request who-has 192.168.1.2 tell 192.168.1.2, length 46
21:37:45.642525 IP 192.168.1.1 > 192.168.1.14: ICMP echo request, id 53250, seq 0, length 64
21:37:45.642617 IP 192.168.1.14 > 192.168.1.1: ICMP echo reply, id 53250, seq 0, length 64
```

3.3) Καταγράφουμε επίσης αμφότερα τα είδη μηνυμάτων:

```
root@PC:~ # tcpdump -i em1
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on em1, link-type EN10MB (Ethernet), capture size 262144 bytes
21:40:20.299617 ARP, Request who-has 192.168.1.18 tell 192.168.1.18, length 46
21:40:20.299626 ARP, Request who-has 192.168.1.17 tell 192.168.1.18, length 46
21:40:20.299719 ARP, Request who-has 192.168.1.17 tell 192.168.1.17, length 28
21:40:20.299772 ARP, Reply 192.168.1.17 is-at 08:00:27:5a:86:2d (oui Unknown), length 28
21:40:20.299776 ARP, Request who-has 192.168.1.29 tell 192.168.1.29, length 46
21:40:20.300212 IP 192.168.1.18 > 192.168.1.17: ICMP echo request, id 53250, seq 0, length 64
21:40:20.300279 IP 192.168.1.17 > 192.168.1.18: ICMP echo reply, id 53250, seq 0, length 64
```

3.4) Παρατηρούμε πως λαμβάνουμε μήνυμα “No route to host” και δε καταγράφονται πακέτα ARP/ICMP ούτε στο LAN1 αλλά ούτε και στο LAN2.

3.5) Όμοια, δε καταγράφουμε τίποτα και λαμβάνουμε επίσης “No route to host”.

3.6) Στο πρώτο ring, από το PC1 στο PC3, εφαρμόζοντας τη μάσκα υποδικτύου στο PC1 βλέπουμε πως αυτό ανήκει στο δίκτυο 192.168.1.0, επομένως όταν πάει να στείλει στο PC3, εφαρμόζει τη δική του μάσκα στο PC3 και βλέπει πως αυτό ανήκει στο υποδίκτυο 192.168.1.16, το οποίο είναι διαφορετικό από το δικό του. Το ίδιο πράγμα συμβαίνει και στο αντίστροφο ring με ανεστραμμένους τους ρόλους των PC1 και PC3.

3.7) Είναι τα παρακάτω:

```
root@PC1:~ # arp -a
? (192.168.1.14) at 08:00:27:13:a7:15 on em0 expires in 445 seconds [ethernet]
? (192.168.1.1) at 08:00:27:a7:4d:d1 on em0 permanent [ethernet]
```

3.8) Είναι τα παρακάτω:

```
root@PC2:~ # arp -a
? (192.168.1.2) at 08:00:27:d5:2f:28 on em0 permanent [ethernet]
```

3.9) Είναι τα παρακάτω:

```
root@PC:~ # arp -a
? (192.168.1.17) at 08:00:27:5a:86:2d on em1 permanent [ethernet]
? (192.168.1.18) at 08:00:27:20:c6:95 on em1 expires in 504 seconds [ethernet]
? (192.168.1.14) at 08:00:27:13:a7:15 on em0 permanent [ethernet]
? (192.168.1.1) at 08:00:27:a7:4d:d1 on em0 expires in 349 seconds [ethernet]
```

3.10) Παραμένουν μόνο οι MAC διευθύνσεις των καρτών δικτύου του στα LAN1 (em0) και LAN2 (em1).

3.11) Καταγράφουμε τα παρακάτω:

```
root@PC:~ # tcpdump -i em0 'arp or icmp'
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on em0, link-type EN10MB (Ethernet), capture size 262144 bytes
21:56:43.211728 ARP, Request who-has 192.168.1.1 tell 192.168.1.14, length 28
21:56:43.212452 ARP, Reply 192.168.1.1 is-at 08:00:27:a7:4d:d1 (oui Unknown), length 46
21:56:43.212525 IP 192.168.1.14 > 192.168.1.1: ICMP echo request, id 39684, seq 0, length 64
21:56:43.212895 IP 192.168.1.1 > 192.168.1.14: ICMP echo reply, id 39684, seq 0, length 64
21:56:52.407230 ARP, Request who-has 192.168.1.2 tell 192.168.1.14, length 28
21:56:52.408012 ARP, Reply 192.168.1.2 is-at 08:00:27:d5:2f:28 (oui Unknown), length 46
21:56:52.408075 IP 192.168.1.14 > 192.168.1.2: ICMP echo request, id 40708, seq 0, length 64
21:56:52.408805 IP 192.168.1.2 > 192.168.1.14: ICMP echo reply, id 40708, seq 0, length 64
```

3.12) Πλέον εντοπίζονται τα παρακάτω περιεχόμενα στον ARP πίνακα του R1:

```

root@PC:~ # arp -a
? (192.168.1.17) at 08:00:27:5a:86:2d on em1 permanent [ethernet]
? (192.168.1.14) at 08:00:27:13:a7:15 on em0 permanent [ethernet]
? (192.168.1.1) at 08:00:27:a7:4d:d1 on em0 expires in 1126 seconds [ethernet]
? (192.168.1.2) at 08:00:27:d5:2f:28 on em0 expires in 1135 seconds [ethernet]

```

Παρατηρούμε πως προστέθηκαν οι εγγραφές για τις MAC διευθύνσεις των PC1 και PC2, τις οποίες το R1 έμαθε μέσω των ARP request που προκλήθηκαν στο προηγούμενο ερώτημα λόγω του ping στα PC1/PC2.

3.13) Προστέθηκε όπως βλέπουμε η MAC διεύθυνση της διεπαφής του R1 στο LAN1, διότι προηγουμένως το PC1 έλαβε πακέτο από αυτήν, επομένως και αποθήκευσε τη MAC διεύθυνση της διεπαφής.

```

root@PC1:~ # arp -a
? (192.168.1.14) at 08:00:27:13:a7:15 on em0 expires in 1016 seconds [ethernet]
? (192.168.1.1) at 08:00:27:a7:4d:d1 on em0 permanent [ethernet]

```

3.14) Παρατηρούμε πως προστέθηκαν οι MAC διευθύνσεις των διεπαφών των PC2 και PC4, ενώ η καταγραφή φαίνεται επίσης παρακάτω:

```

root@PC:~ # arp -a
? (192.168.1.29) at 08:00:27:e2:93:3a on em1 expires in 1196 seconds [ethernet]
? (192.168.1.17) at 08:00:27:5a:86:2d on em1 permanent [ethernet]
? (192.168.1.18) at 08:00:27:20:c6:95 on em1 expires in 1192 seconds [ethernet]
? (192.168.1.14) at 08:00:27:13:a7:15 on em0 permanent [ethernet]
? (192.168.1.1) at 08:00:27:a7:4d:d1 on em0 expires in 829 seconds [ethernet]
? (192.168.1.2) at 08:00:27:d5:2f:28 on em0 expires in 838 seconds [ethernet]

```

```

root@PC:~ # tcpdump -i em1 'arp or icmp'
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on em1, link-type EN10MB (Ethernet), capture size 262144 bytes
22:02:46.602600 ARP, Request who-has 192.168.1.18 tell 192.168.1.17, length 28
22:02:46.602980 ARP, Reply 192.168.1.18 is-at 08:00:27:20:c6:95 (oui Unknown), length 46
22:02:46.603044 IP 192.168.1.17 > 192.168.1.18: ICMP echo request, id 46340, seq 0, length 64
22:02:46.604235 IP 192.168.1.18 > 192.168.1.17: ICMP echo reply, id 46340, seq 0, length 64
22:02:50.997246 ARP, Request who-has 192.168.1.29 tell 192.168.1.17, length 28
22:02:50.998024 ARP, Reply 192.168.1.29 is-at 08:00:27:e2:93:3a (oui Unknown), length 46
22:02:50.998090 IP 192.168.1.17 > 192.168.1.29: ICMP echo request, id 47364, seq 0, length 64
22:02:50.998898 IP 192.168.1.29 > 192.168.1.17: ICMP echo reply, id 47364, seq 0, length 64

```

3.15) Δημιουργούμε τον παρακάτω πίνακα με τις αντιστοιχίσεις:

<u>IP Address</u>	<u>MAC Address</u>
192.168.1.29	08:00:27:e2:93:3a
192.168.1.17	08:00:27:5a:86:2d
192.168.1.18	08:00:27:20:c6:95
192.168.1.14	08:00:27:13:a7:15
192.168.1.1	08:00:27:a7:4d:d1
192.168.1.2	08:00:27:d5:2f:28

3.16) Σε αυτή την περίπτωση παράγεται ένα ARP request ανά ping, καθώς σε κάθε ping το PC1 προσπαθεί να βρει τη διεύθυνση MAC του παραλήπτη του ping, χωρίς, ωστόσο επιτυχία.

```
root@PC1:~ # tcpdump
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on em0, link-type EN10MB (Ethernet), capture size 262144 bytes
22:11:10.521775 ARP, Request who-has 192.168.1.5 tell 192.168.1.1, length 28
22:11:11.530806 ARP, Request who-has 192.168.1.5 tell 192.168.1.1, length 28
22:11:12.543842 ARP, Request who-has 192.168.1.5 tell 192.168.1.1, length 28
```

3.17) Παρουσιάζεται το παρακάτω, όπου μας ενημερώνει πως είναι ανολοκλήρωτη η MAC διεύθυνση της συγκεκριμένης IP:

```
root@PC1:~ # arp -a
? (192.168.1.14) at 08:00:27:13:a7:15 on em0 expires in 170 seconds [ethernet]
? (192.168.1.1) at 08:00:27:a7:4d:d1 on em0 permanent [ethernet]
? (192.168.1.5) at (incomplete) on em0 expired [ethernet]
```

3.18) Παρατηρούμε πως αποστέλλεται ένα ARP request ανά ping. Όταν, επιπλέον, φτάσουμε τα 6 αντί να μη λάβουμε καμία απάντηση όπως στα προηγούμενα λαμβάνουμε το μήνυμα “ping: sendto: Host is down”.

Άσκηση 4: Προεπιλεγμένος δρομολογητής

4.1) Εκτελούμε στον R1 την εντολή “sysctl net.inet.ip.forwarding=1”.

4.2) Να προσθέσουμε τη γραμμή “ gateway_enable=“YES” ” στο αρχείο /etc/rc.conf του R1.

4.3) Προς το παρόν δε βλέπουμε καμία διαφορά, λαμβάνουμε πάλι το μήνυμα “No route to host”.

4.4) Παρατηρούμε πως δεν υπάρχει διαδρομή για το LAN2.

```
root@PC1:~ # netstat -rn
Routing tables

Internet:
Destination        Gateway             Flags               Netif  Expire
127.0.0.1           link#2              UH                  lo0
192.168.1.0/28      link#1              U                   em0
192.168.1.1         link#1              UHS                 lo0
```

4.5) Εκτελούμε στο PC1 την εντολή “route add default 192.168.1.14”.

4.6) Προστέθηκε η εγγραφή default με διεύθυνση την 192.168.1.14.

Destination	Gateway	Flags	Netif	Expire
default	192.168.1.14	UGS	em0	
127.0.0.1	link#2	UH	lo0	
192.168.1.0/28	link#1	U	em0	
192.168.1.1	link#1	UHS	lo0	

4.7) Αυτή τη φορά, εκτελώντας ping από το PC1 στο PC3 και καταγράφοντας το LAN1 από το R1 βλέπουμε πως το ping στάλθηκε επιτυχώς (δε λάβαμε μήνυμα “No route to host”), ωστόσο δε λήφθηκε καμία απάντηση.

4.8) Εκτελούμε σε διαφορετικές κονσόλες του R1 τις εντολές “tcpdump -i em0 icmp” και “tcpdump -i em1 icmp” για να καταγράψουμε ICMP πακέτα στα LAN1 και LAN2 αντίστοιχα. Αυτό που παρατηρούμε είναι πως καταγράφονται κανονικά ICMP Request προς τον προορισμό, χωρίς ωστόσο να καταγράφεται κάποιο Reply, αφού στον R3 δεν έχει οριστεί προκαθορισμένη πύλη.

4.9) Εκτελούμε στο PC3 την εντολή “route add default 192.168.1.17”.

4.10) Πλέον, όχι μόνο έχουμε επιτυχής αποστολή του ICMP Request, αλλά λαμβάνουμε κανονικά και το ICMP Reply. Τα βήματα που συμβαίνουν είναι τα εξής:

- 1) Το PC1 πάει να στείλει πακέτο στο PC3, ωστόσο βλέπει πως είναι σε διαφορετικά δίκτυα, επομένως το στέλνει στο default gateway, εν προκειμένω στο em0 του R1.
- 2) Το R1, έχει καταγραφή για τη διεύθυνση IP του παραλήπτη (του PC3) στην em1 του, επομένως στέλνει το πακέτο από αυτή την κάρτα δικτύου απευθείας στο PC3.
- 3) Το PC3 λαμβάνει το μήνυμα και επειδή προφανώς το PC1 είναι επίσης σε διαφορετικό δίκτυο από αυτό, προωθεί την απάντησή του στο default gateway του, δηλαδή στο em1 του R1.
- 4) Με τη σειρά του, το R1 ακολουθεί πλέον αντίστροφη πορεία και στέλνει το πακέτο από το em0 απευθείας στο PC1.

4.11) Βλέπουμε ένα ενδιάμεσο βήμα μεταξύ αποστολέα και παραλήπτη:

```
root@PC1:~ # traceroute 192.168.1.18
traceroute to 192.168.1.18 (192.168.1.18), 64 hops max, 40 byte packets
 1  192.168.1.14 (192.168.1.14)  0.987 ms  1.043 ms  0.968 ms
 2  192.168.1.18 (192.168.1.18)  1.989 ms  2.012 ms  2.444 ms
```

4.12) Εκτελούμε “arp -ad” σε καθέναν εξ αυτών.

4.13) Στο R1 σε μία κονσόλα εκτελούμε “tcpdump -vnne -i em0”, ενώ σε μία δεύτερη “tcpdump -vnne -i em1”.

4.14) Εκτελούμε “ping -c 1 192.168.1.18” από το PC1.

4.15) Στο LAN1, πηγή αποτελεί η διεύθυνση MAC/IP 08:00:27:a7:4d:d1/192.168.1.1, ενώ προορισμό η διεύθυνση MAC/IP 08:00:27:13:a7:15/192.168.1.18.

4.16) Στο LAN2, πηγή αποτελεί η διεύθυνση MAC/IP 08:00:27:5a:86:2d/192.168.1.1, ενώ προορισμό η διεύθυνση MAC/IP 08:00:27:20:c6:95/192.168.1.18.

4.17) Παρατηρούμε πως καθ' όλη τη δρομολόγηση οι διευθύνσεις IP πηγής/προορισμού παρέμειναν αναλλοίωτες. Αυτό που μεταβλήθηκε είναι οι διευθύνσεις MAC, ως εξής:

- Στο LAN1 εντοπίζεται ως πηγή του ICMP Echo Request η MAC Address που αντιστοιχεί στο PC1 με παραλήπτη την MAC της διεπαφής em0 του R1.
- Στο LAN2 εντοπίζεται ως πηγή του ICMP Echo Request η MAC Address που αντιστοιχεί στην διεπαφή em1 του R1 με παραλήπτη την MAC του PC3.

4.18) Για το ερώτημα αυτό έπρεπε να ενεργοποιήσουμε τον SSH Daemon, μετακινώντας τον χαρακτήρα '#' ένα κενό πιο μακριά από το "YES" (ώστε να μην είναι κολλητά) στο αρχείο /etc/rc.conf των PC1 και PC3. Ελέγχουμε εάν είναι enabled το SSH με την εντολή sysrc sshd_enable. Στη συνέχεια, εκτελούμε την εντολή /etc/rc.d/sshd start προκειμένου να εκκινήσουμε τον SSH daemon. Αφού εκτελέσουμε τα παραπάνω βήματα και στο PC1 αλλά και στο PC3, εκτελούμε την εντολή "ssh lab@192.168.1.18" από το PC1.

4.19) Εκτελώντας "netstat -an" στο PC1, βλέπουμε μεταξύ άλλων την παρακάτω γραμμή:

```
lab@PC3:~ % netstat -an
Active Internet connections (including servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         (state)
tcp4      0      44 192.168.1.18.22         192.168.1.1.50847      ESTABLISHED
```

Όπως βλέπουμε, χρησιμοποιείται το πρωτόκολλο μεταφοράς TCP, τοπική θύρα σύνδεσης είναι η 22 (στο PC3 ουσιαστικά), ενώ απομακρυσμένη είναι η 50.847.

4.20) Παρατηρούμε πως δεν εμφανίζεται κανένα αποτέλεσμα, μιας και επιβάλλαμε να δούμε πληροφορίες για το πρωτόκολλο TCP, ενώ ο δρομολογητής ελέγχει δεδομένα μέχρι και το πρωτόκολλο δικτύου.

Άσκηση 5: Προθέματα δικτύου και δρομολόγηση

5.1) Εκτελούμε στα PC1/PC2 την εντολή "route add default 192.168.1.14", ενώ στα PC3/PC4 την εντολή "route add default 192.168.1.17".

5.2) Εκτελούμε παντού “arp -a”.

5.3) Στο R1 εκτελούμε “tcpdump -i em0 ‘icmp or arp’”.

5.4) Στο PC4 εκτελούμε την εντολή “tcpdump icmp or arp”.

5.5) Ναι, όλα.

5.6) Βλέπουμε τα περιεχόμενα των πινάκων ARP στα PC1, PC2, PC3, PC4, R1 με τη σειρά:

```
root@PC1:~ # arp -a
? (192.168.1.14) at 08:00:27:13:a7:15 on em0 expires in 1130 seconds [ethernet]
? (192.168.1.1) at 08:00:27:a7:4d:d1 on em0 permanent [ethernet]
? (192.168.1.2) at 08:00:27:d5:2f:28 on em0 expires in 1119 seconds [ethernet]

root@PC2:~ # arp -a
PC1 (192.168.1.1) at 08:00:27:a7:4d:d1 on em0 expires in 1074 seconds [ethernet]
? (192.168.1.2) at 08:00:27:d5:2f:28 on em0 permanent [ethernet]

root@PC3:~ # arp -a
? (192.168.1.17) at 08:00:27:5a:86:2d on em0 expires in 1035 seconds [ethernet]
? (192.168.1.18) at 08:00:27:20:c6:95 on em0 permanent [ethernet]

root@PC4:~ # arp -a
? (192.168.1.29) at 08:00:27:e2:93:3a on em0 permanent [ethernet]
? (192.168.1.17) at 08:00:27:5a:86:2d on em0 expires in 997 seconds [ethernet]

root@PC:~ # arp -a
? (192.168.1.29) at 08:00:27:e2:93:3a on em1 expires in 967 seconds [ethernet]
? (192.168.1.17) at 08:00:27:5a:86:2d on em1 permanent [ethernet]
? (192.168.1.18) at 08:00:27:20:c6:95 on em1 expires in 964 seconds [ethernet]
? (192.168.1.14) at 08:00:27:13:a7:15 on em0 permanent [ethernet]
? (192.168.1.1) at 08:00:27:a7:4d:d1 on em0 expires in 964 seconds [ethernet]
```

5.7) Έχουμε την παρακάτω ακολουθία μηνυμάτων μεταξύ των PC1, R1, PC4 :

- i. ARP, Request who-has 192.168.1.14 tell 192.168.1.1, όπου το PC1 ρωτάει για τη MAC της IP 192.168.1.14
- ii. ARP, Reply 192.168.1.14 is-at 08:00:27:13:a7:15, όπου το R1 υπό τη διεύθυνση 192.168.1.14 απαντάει με τη MAC του στο PC1
- iii. ICMP echo request, IP 192.168.1.1 > 192.168.1.29, όπου το PC1 στέλνει το ICMP echo request του με τελικό προορισμό τη διεύθυνση 192.168.1.29, αλλά απευθείας στη MAC address του em0 του R1
- iv. ARP, Request who-has 192.168.1.29 tell 192.168.1.17, όπου το R1 ρωτάει για τη MAC του PC4
- v. ARP, Reply 192.168.1.29 is-at 08:00:27:e2:93:3a, όπου το PC4 απαντάει στο R1 πληροφορώντας το για τη MAC του
- vi. ICMP echo request, IP 192.168.1.1 > 192.168.1.29, όπου το R1 προωθεί το πακέτο του PC1 στο PC4
- vii. ICMP echo reply, IP 192.168.1.29 > 192.168.1.1, όπου το PC4 απαντάει στη διεύθυνση του PC1

- viii. ICMP echo reply, IP 192.168.1.29 > 192.168.1.1, όπου το R1 προωθεί την απάντηση του PC4 στο PC1

5.8) Εκτελούμε “arp -ad” σε κάθε μηχανήμα και στη συνέχεια “tcpdump -e icmp or arp” στα PC3, PC4, ενώ “tcpdump -e -i em1 ‘icmp or arp’” στο R1.

5.9) Το ping ήταν επιτυχές, ωστόσο λάβαμε τις έξτρα πληροφορίες (ένα IP header) που φαίνονται παρακάτω και το μήνυμα Redirect Host (New addr: 192.168.1.29).

```
root@PC3:~ # ping -c 1 192.168.1.29
PING 192.168.1.29 (192.168.1.29): 56 data bytes
36 bytes from 192.168.1.17: Redirect Host(New addr: 192.168.1.29)
 0r HL TOS Len  ID Flg  off TTL Pro  cks      Src      Dst
  4  5  00 0054 7951   0 0000  40  01 7dd8 192.168.1.18 192.168.1.29

64 bytes from 192.168.1.29: icmp_seq=0 ttl=64 time=3.473 ms

--- 192.168.1.29 ping statistics ---
1 packets transmitted, 1 packets received, 0.0% packet loss
round-trip min/avg/max/stddev = 3.473/3.473/3.473/0.000 ms
```

5.10) Βλέπουμε τα περιεχόμενα των ARP πινάκων των R1, PC3, PC4:

```
root@PC:~ # arp -a
? (192.168.1.29) at 08:00:27:e2:93:3a on em1 expires in 1034 seconds [ethernet]
? (192.168.1.17) at 08:00:27:5a:86:2d on em1 permanent [ethernet]
? (192.168.1.18) at 08:00:27:20:c6:95 on em1 expires in 1034 seconds [ethernet]
? (192.168.1.14) at 08:00:27:13:a7:15 on em0 permanent [ethernet]

root@PC3:~ # arp -a
? (192.168.1.17) at 08:00:27:5a:86:2d on em0 expires in 976 seconds [ethernet]
? (192.168.1.18) at 08:00:27:20:c6:95 on em0 permanent [ethernet]

root@PC4:~ # arp -a
? (192.168.1.29) at 08:00:27:e2:93:3a on em0 permanent [ethernet]
? (192.168.1.17) at 08:00:27:5a:86:2d on em0 expires in 955 seconds [ethernet]
? (192.168.1.18) at 08:00:27:20:c6:95 on em0 expires in 955 seconds [ethernet]
```

5.11) Έχουμε την παρακάτω αλληλουχία μηνυμάτων:

- Το PC3 κάνει broadcast ένα ARP request ρωτώντας τη MAC διεύθυνση της IP 192.168.1.17
- Το R1 λαμβάνει το broadcast και απαντάει στο PC3 με τη MAC του
- Το PC1 στέλνει στην παραπάνω MAC του R1 το ICMP echo request με προορισμό τη διεύθυνση 192.168.1.29
- Το R1 κάνει broadcast ένα ARP Request, όπου ρωτάει για τη MAC της διεύθυνσης 192.168.1.29.
- Προτού λάβει απάντηση από το broadcast, απαντάει στο ICMP Echo request του PC3 με ICMP redirect 192.168.1.29 to host 192.168.1.29.
- Το R1 μαθαίνει τη MAC του PC4
- Το R1 προωθεί στο PC4 το ICMP Echo request του PC3

- viii. Το PC4 κάνει broadcast ένα ARP Request με σκοπό να μάθει τη MAC της διεύθυνσης 192.168.1.18
- ix. Το PC3 απαντάει (χωρίς διαμεσολάβηση) στο ARP request του PC4 ενημερώνοντάς το με τη MAC του
- x. Τέλος, ο PC4 απαντάει στο ICMP echo request του PC3

5.12) Το PC3 αναζητεί τη MAC διεύθυνση του R1, ενώ το PC4 του PC3.

5.13) Το PC3 είναι στο δίκτυο 192.168.1.16, ενώ όταν πάει να στείλει στη διεύθυνση 192.168.1.29, εάν εφαρμόσει τη μάσκα του βλέπει πως το PC4 είναι στο δίκτυο 192.168.1.24, άρα προωθεί το πακέτο προς δρομολόγηση στην προκαθορισμένη πύλη.

5.14) Ο R1 το προωθεί στο PC4.

5.15) Στάλθηκε απευθείας στο PC3.

5.16) Εκτελούμε στα PC3, PC4 “tcpdump -e icmp”, ενώ στο R1 “tcpdump -i em1 icmp”.

5.17) Αυτό που βλέπουμε να συμβαίνει, είναι πως σε κάθε ICMP echo request μήνυμα, το PC3 στέλνει το ping στην IP_{PC4}, αυτό μεταφέρεται στο R1, το R1 στέλνει στο PC3 Icmp redirect, προωθεί το request στο PC4 και τέλος εκείνο απαντάει κατευθείαν στο PC3. Ο PC3 ουσιαστικά αγνοεί τα redirects, καθώς η προτεινόμενη διεύθυνση παράκαμψης δεν ανήκει στο υποδίκτυο της διεπαφής του στο LAN2, διαφορετικά θα ενημέρωνε κατάλληλα τον πίνακα δρομολόγησης του.

5.18) Διαγράφηκε η προκαθορισμένη διαδρομή.

5.19) Αφού εκτελέσουμε την εντολή, καταγράφουμε τον εξής πίνακα δρομολόγησης για κάποια υποδίκτυα (αλλά και για το ίδιο το PC3) του 192.168.0.0/16:

192.168.1.16/28	link#1	U	em0
192.168.1.18	link#1	UHS	lo0
192.168.1.24/29	192.168.1.17	UGS	em0

5.20) Παρατηρούμε πως στο πρώτο Ping, το ταίριασμα μεγαλύτερου προθέματος στέλνει το πακέτο στο R1, καθώς η μάσκα /29 στη διεύθυνση του PC4 το κάνει να ανήκει στο υποδίκτυο 192.168.1.24/29, επομένως προωθείται στην 192.168.1.17 για να γίνει rerouting. Στη συνέχεια, μετά το πρώτο ping, το PC3 λαμβάνει το icmp redirect, αλλά αυτή τη φορά δε το αγνοεί, αφού αν εφαρμόσει το subnet mask του στη διεύθυνση του PC4 βλέπει πως είναι στο ίδιο LAN οπότε και το προωθεί πλέον κατευθείαν εκεί.

5.21) Έχει προστεθεί πλέον η παρακάτω εγγραφή, η οποία διαφέρει στο ότι η δρομολόγηση σε αυτήν γίνεται μέσω του στρώματος ζεύξης δεδομένων, σε αντίθεση με τα άλλα υποδίκτυα του 192.168.0.0/16 που γίνεται μέσω του R1.

192.168.1.16/28	link#1	U	em0
-----------------	--------	---	-----

5.22) Δε μπορεί να επικοινωνήσει με μηχανήματα του LAN1, καθώς δεν υπάρχει προεπιλεγμένη πύλη προκειμένου να κάνει το κατάλληλο routing.

5.23) Στέλνει κατευθείαν τα πακέτα στο PC4, αφού υπάρχει ήδη εγγραφή στον πίνακα δρομολόγησής του που αφορά αποκλειστικά το PC4 και του λέει να τα στείλει στο τοπικό δίκτυο.

Άσκηση 6: Router on a stick

6.1) Εκτελούμε στο R1 τις εντολές “ifconfig bridge1 create” και “ifconfig bridge1 addm em0 addm em1 up”.

6.2) Εκτελούμε στο PC1 τις εντολές “ifconfig em0.5 create inet 192.168.5.1/24” και “ifconfig em0.6 create inet 192.168.6.1/24”.

6.3) Εκτελούμε στο PC2 “ifconfig em0.5 create inet 192.168.5.2/24”.

6.4) Στο PC3 “ifconfig em0.6 create inet 192.168.6.18/24”.

6.5) Στο PC4 “ifconfig em0.5 create inet 192.168.5.29/24”.

6.6) Μπορούμε να κάνουμε κανονικά ping στις διεπαφές em0, em0.6, αλλά όχι στην em0.5, καθώς λαμβάνουμε το παρακάτω σφάλμα:

```
root@PC3:~ # ping -c 1 192.168.5.1
PING 192.168.5.1 (192.168.5.1): 56 data bytes
36 bytes from 192.168.1.17: Destination Host Unreachable
  Ur  HL  TOS  Len   ID  Flg  off  TTL  Pro  cks      Src      Dst
   4   5   00  0054  7afd   0  0000   40   01  7848 192.168.1.18 192.168.5.1

--- 192.168.5.1 ping statistics ---
1 packets transmitted, 0 packets received, 100.0% packet loss
```

6.7) Μπορούμε να κάνουμε στην em0 και στην em0.5, αλλά όχι στην em0.6.

6.8) Απέτυχαν διότι προσπάθησαν να κάνουν ping σε VLANs για τα οποία δε διέθεταν διεπαφή, επομένως η αναζήτηση παραλήπτη των πακέτων αυτών έγινε στα LAN1 και LAN2. Αν π.χ. κάναμε καταγραφή στο R1, θα βλέπαμε πως όταν το PC3 στέλνει στην διεπαφή em0.6 του PC1, κανένα πακέτο δε περνάει από το R1.

6.9) Ναι πλέον μπορούμε κανονικά.

6.10) Ναι, μπορούμε κανονικά.

6.11) Όχι, δε μπορούμε να κάνουμε σε καμία.

6.12) Στο PC1 εκτελούμε την εντολή “sysctl net.inet.ip.forwarding=1” και στο PC2 την “route add default 192.168.1.1”.

6.13) Ναι, πλέον επιτυγχάνουν.

6.14) Καταγράφουμε τις MAC:

- PC1: 08:00:27:a7:4d:d1
- PC2: 08:00:27:d5:2f:28
- PC3: 08:00:27:20:c6:95

6.15) Εκτελούμε “tcpdump -e” στα μηχανήματα.

6.16) Το PC3 δεν έχει διεπαφή στο VLAN 5, επομένως έκανε broadcast ARP request για να βρει τη MAC του default gateway του, δηλαδή της 192.168.6.1. Στο ερώτημα αυτό, απάντησε το PC1 δίνοντας του τη MAC του. Σε αυτή τη MAC, και με προορισμό τη διεύθυνση 192.168.1.2 έστειλε το VLAN ICMP echo request το PC3. Αφού το PC1 λάβει το μήνυμα, η διεύθυνση 192.168.1.1 αποστέλλει ARP request με σκοπό να βρει τη MAC address του 192.168.1.2. Το PC2 απαντάει στο broadcast αυτό, ενημερώνοντας με την MAC του. Το PC1 προωθεί το ICMP echo request (ethertype IPv4 αυτή τη φορά 802.1Q που ήταν το αρχικό) στο PC2. Το PC2 απαντάει με ICMP echo reply (ethertype IPv4) στο PC1, το οποίο αφού λάβει την απάντηση, την στέλνει στο PC3 ξανά ως 802.1Q αντί για IPv4.

6.17) Όχι δεν είναι.

6.18) Ενώ το PC4 απαντάει κανονικά στα ping request που λαμβάνει, τα στέλνει στο R1, δηλαδή στην προκαθορισμένη του πύλη, το οποίο δε ξέρει που να τα προωθήσει ώστε να πάνε στη διεπαφή 192.168.6.18, επομένως απλά τα απορρίπτει.

6.19) Πλέον επιτυγχάνει κανονικά. Η διαδρομή που ακολουθείται είναι η εξής:

- a) Το PC3 στέλνει το request, το οποίο πάει στην MAC του 192.168.6.1, που είναι και η προκαθορισμένη του πύλη
- b) Αφού τα λάβει το PC1, τα προωθεί στο PC4 βάσει της MAC που έχει για τη διεύθυνση 192.168.5.29
- c) Από εκεί, το PC4 απαντάει και στέλνει την απάντηση στο default gateway του, το οποίο είναι αντιστοιχισμένο στη MAC του PC1
- d) Το PC1 έχει αντιστοίχιση για την IP 192.168.5.29, επομένως προωθεί κατευθείαν την απάντηση στο PC3.