

Όνοματεπώνυμο: Χαράλαμπος Καμπουγέρης

Όνομα PC/ΛΣ: DESKTOP-N90CRE0

Ομάδα: 1, Τρίτη 10:45-13:30, Αιθ.Α4

Ημερομηνία: 18/03/2024

## Εργαστηριακή Άσκηση 4

### Εισαγωγή στη δρομολόγηση

#### Άσκηση 1

**1.1** Η IP είναι ο συνολικός αριθμός 32 bit που αποδίδεται σε κάθε διεπαφή για την αναπαράσταση του στο δίκτυο. Αριθμός δικτύου είναι το πρώτο μέρος της IP που χαρακτηρίζει το δίκτυο στο οποίο βρίσκεται η διεπαφή.

**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, μας 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** Εάν ο πίνακας δρομολόγησης περιέχει εγγραφή για τη συσκευή, τότε το πακέτο θα αποσταλεί απευθείας μέσω της υποδεικνυόμενης διεπαφής, διότι έχουμε ταίριασμα μήκους 32 bit.

**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 20810 = 11010000, επομένως ανήκει στην κλάση C.

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

**1.10** 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).

**1.11** Subnet 1 : 10.11.12.0/25 (10.11.12.0 - 10.11.12.127)  $\rightarrow$  100 devices

Subnet 2 : 10.11.12.128/26 (10.11.12.128 - 10.11.12.191)  $\rightarrow$  60 devices

Subnet 3 : 10.11.12.192/27 (10.11.12.192 - 10.11.12.223)  $\rightarrow$  20 devices

Subnet 4 : 10.11.12.224/28 (10.11.12.224 - 10.11.12.240) → 10 devices

**1.12** Ναί υπάρχει, μπορεί να έχει μέχρι 15 υπολογιστές (δεν μπορούμε να χρησιμοποιήσουμε τη broadcast IP)

**1.13** Αναζητούμε στην πραγματικότητα τη θέση του δεξιότερου κοινού bit στο 3ο 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 4η θέση (ξεκινώντας από τα αριστερά), επομένως τα συμπυκνώνουμε στο μπλοκ 171.12.0.0/20.

## Άσκηση 2

**2.1** Εκτελώ την εντολή “ifconfig em0 IP” με την αντίστοιχη IP που μου δίνεται για το κάθε μηχάνημα

**2.2** Χρησιμοποιήθηκε η επιλογή “Generate new MAC addresses for all network adapters”, 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 ή ARP θα δυσλειτουργούσαν.

**2.3** Από το PC1:

```
root@PC:~ # 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=13.467 ms

--- 192.168.1.2 ping statistics ---
1 packets transmitted, 1 packets received, 0.0% packet loss
round-trip min/avg/max/stddev = 13.467/13.467/13.467/0.000 ms
```

```
root@PC:~ # 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=1.872 ms

--- 192.168.1.18 ping statistics ---
1 packets transmitted, 1 packets received, 0.0% packet loss
round-trip min/avg/max/stddev = 1.872/1.872/1.872/0.000 ms
```

```
root@PC:~ # ping -c 1 192.168.1.29
PING 192.168.1.29 (192.168.1.29): 56 data bytes
^C
--- 192.168.1.29 ping statistics ---
1 packets transmitted, 0 packets received, 100.0% packet loss
```

#### 2.4 Από το PC2:

```
root@PC:~ # 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
^C
--- 192.168.1.18 ping statistics ---
1 packets transmitted, 0 packets received, 100.0% packet loss
```

```
root@PC:~ # 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
^C
--- 192.168.1.29 ping statistics ---
1 packets transmitted, 0 packets received, 100.0% packet loss
```

#### 2.5 Από το PC4:

```
root@PC:~ # 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
^C
--- 192.168.1.1 ping statistics ---
1 packets transmitted, 0 packets received, 100.0% packet loss
```

```
root@PC:~ # 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
^C
--- 192.168.1.2 ping statistics ---
1 packets transmitted, 0 packets received, 100.0% packet loss
```

```

root@PC:~ # 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=1.737 ms

--- 192.168.1.18 ping statistics ---
1 packets transmitted, 1 packets received, 0.0% packet loss
round-trip min/avg/max/stddev = 1.737/1.737/1.737/0.000 ms

```

## 2.6 Από το PC3:

```

root@PC:~ # 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=0.809 ms

--- 192.168.1.1 ping statistics ---
1 packets transmitted, 1 packets received, 0.0% packet loss
round-trip min/avg/max/stddev = 0.809/0.809/0.809/0.000 ms

```

```

root@PC:~ # ping -c 1 192.168.1.2
PING 192.168.1.2 (192.168.1.2): 56 data bytes
^C

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

```

## 2.7

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. Επομένως Η διεύθυνση δεν ανήκει στο υποδίκτυο στο οποίο βρισκόμαστε.

## 2.8 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.

**2.9** Εκτελούμε τις εντολές ανάθεσης διευθύνσεων όπως πριν, αλλά με μάσκα το /28 αυτή τη φορά. (ifconfig em0 192.168.1.X/28)

**2.10** 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”

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

### Άσκηση 3

#### 3.1

```
root@PC:~ # ifconfig em0 192.168.1.14/28
root@PC:~ # ifconfig em1 192.168.1.17/28
```

**3.2** Μέσω των ρυθμίσεων του VirtualBox, στο πεδίο Network.

**3.3** 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 65535 bytes
22:09:12.990742 ARP, Request who-has 192.168.1.14 tell 192.168.1.1, length 46
22:09:12.990840 ARP, Request who-has 192.168.1.14 tell 192.168.1.14, length 28
22:09:12.990888 ARP, Reply 192.168.1.14 is-at 08:00:27:29:de:a1 (oui Unknown), length 28
22:09:12.991563 IP 192.168.1.1 > 192.168.1.14: ICMP echo request, id 35075, seq 0, length 64
22:09:12.991591 IP 192.168.1.14 > 192.168.1.1: ICMP echo reply, id 35075, seq 0, length 64
```

**3.4** Καταγράφουμε τα παρακάτω εκτελώντας “tcpdump -i em1” στον R1, όπου και βλέπουμε πακέτα ARP (requests, reply και gratuitous) και ICMP (request και reply):

```
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 65535 bytes
22:11:30.859280 ARP, Request who-has 192.168.1.17 tell 192.168.1.18, length 46
22:11:30.859969 ARP, Request who-has 192.168.1.17 tell 192.168.1.17, length 28
22:11:30.860056 ARP, Reply 192.168.1.17 is-at 08:00:27:31:9e:e0 (oui Unknown), length 28
22:11:30.860973 IP 192.168.1.18 > 192.168.1.17: ICMP echo request, id 29955, seq 0, length 64
22:11:30.861026 IP 192.168.1.17 > 192.168.1.18: ICMP echo reply, id 29955, seq 0, length 64
```

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

**3.6** Δε καταγράφουμε τίποτα και λαμβάνουμε επίσης “No route to host”.

**3.7** Διότι τα δυο PC δεν ανήκουν στο ίδιο υποδίκτυο και ο Router δεν έχει δημιουργηθεί ώστε να προωθεί τα πακέτα

**3.8**

```
root@PC:~ # arp -a
? (192.168.1.14) at 08:00:27:29:de:a1 on em0 expires in 393 seconds [ethernet]
? (192.168.1.1) at 08:00:27:b0:fe:04 on em0 permanent [ethernet]
```

**3.9**

```
root@PC:~ # arp -a
? (192.168.1.2) at 08:00:27:9c:00:87 on em0 permanent [ethernet]
```

**3.10**

```
root@PC:~ # arp -a
? (192.168.1.17) at 08:00:27:31:9e:e0 on em1 permanent [ethernet]
? (192.168.1.18) at 08:00:27:01:1c:01 on em1 expires in 473 seconds [ethernet]
? (192.168.1.14) at 08:00:27:29:de:a1 on em0 permanent [ethernet]
? (192.168.1.1) at 08:00:27:b0:fe:04 on em0 expires in 335 seconds [ethernet]
```

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

**3.12**

```
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 65535 bytes
22:26:35.440331 ARP, Request who-has 192.168.1.1 tell 192.168.1.14, length 28
22:26:35.440831 ARP, Reply 192.168.1.1 is-at 08:00:27:b0:fe:04 (oui Unknown), length 46
22:26:35.440847 IP 192.168.1.14 > 192.168.1.1: ICMP echo request, id 17667, seq 0, length 64
22:26:35.443043 IP 192.168.1.1 > 192.168.1.14: ICMP echo reply, id 17667, seq 0, length 64
22:26:37.758477 ARP, Request who-has 192.168.1.2 tell 192.168.1.14, length 28
22:26:37.759047 ARP, Reply 192.168.1.2 is-at 08:00:27:9c:00:87 (oui Unknown), length 46
22:26:37.759062 IP 192.168.1.14 > 192.168.1.2: ICMP echo request, id 17923, seq 0, length 64
22:26:37.759536 IP 192.168.1.2 > 192.168.1.14: ICMP echo reply, id 17923, seq 0, length 64
```

**3.13**

```

root@PC:~ # arp -a
? (192.168.1.17) at 08:00:27:31:9e:e0 on em1 permanent [ethernet]
? (192.168.1.14) at 08:00:27:29:de:a1 on em0 permanent [ethernet]
? (192.168.1.1) at 08:00:27:b0:fe:04 on em0 expires in 1120 seconds [ethernet]
? (192.168.1.2) at 08:00:27:9c:00:87 on em0 expires in 1123 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

**3.14** περιέχουν και τη MAC address της διεπαφής R1 στο LAN1, διότι προηγουμένως το PC1 και το PC2 έλαβαν πακέτο από αυτήν, επομένως και αποθήκευσαν τη MAC διεύθυνση.

**3.15** Παρατηρούμε πως προστέθηκαν οι MAC διευθύνσεις των διεπαφών των PC2 και PC4:

```

root@PC:~ # arp -a
? (192.168.1.29) at 08:00:27:03:c8:62 on em1 expires in 1195 seconds [ethernet]
? (192.168.1.17) at 08:00:27:31:9e:e0 on em1 permanent [ethernet]
? (192.168.1.18) at 08:00:27:01:1c:01 on em1 expires in 1192 seconds [ethernet]
? (192.168.1.14) at 08:00:27:29:de:a1 on em0 permanent [ethernet]
? (192.168.1.1) at 08:00:27:b0:fe:04 on em0 expires in 611 seconds [ethernet]
? (192.168.1.2) at 08:00:27:9c:00:87 on em0 expires in 614 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 65535 bytes
22:36:15.984397 ARP, Request who-has 192.168.1.18 tell 192.168.1.17, length 28
22:36:15.984926 ARP, Reply 192.168.1.18 is-at 08:00:27:01:1c:01 (oui Unknown), length 46
22:36:15.984939 IP 192.168.1.17 > 192.168.1.18: ICMP echo request, id 22019, seq 0, length 64
22:36:15.985312 IP 192.168.1.18 > 192.168.1.17: ICMP echo reply, id 22019, seq 0, length 64
22:36:19.120674 ARP, Request who-has 192.168.1.29 tell 192.168.1.17, length 28
22:36:19.121204 ARP, Reply 192.168.1.29 is-at 08:00:27:03:c8:62 (oui Unknown), length 46
22:36:19.121213 IP 192.168.1.17 > 192.168.1.29: ICMP echo request, id 22275, seq 0, length 64
22:36:19.121676 IP 192.168.1.29 > 192.168.1.17: ICMP echo reply, id 22275, seq 0, length 64

```

### 3.16

IP Address	MAC Address
192.168.1.29	08:00:27:03:c8:62
192.168.1.17	08:00:27:31:9e:e0
192.168.1.18	08:00:27:01:1c:01
192.168.1.14	08:00:27:29:de:a1
192.168.1.1	08:00:27:b0:fe:04
192.168.1.2	08:00:27:9c:00:87

**3.17** 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.

```
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 65535 bytes
22:46:01.243746 ARP, Request who-has 192.168.1.5 tell 192.168.1.14, length 28
22:46:02.252605 ARP, Request who-has 192.168.1.5 tell 192.168.1.14, length 28
22:46:03.258177 ARP, Request who-has 192.168.1.5 tell 192.168.1.14, length 28
```

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

```
root@PC:~ # arp -a
? (192.168.1.29) at 08:00:27:03:c8:62 on em1 expires in 461 seconds [ethernet]
? (192.168.1.17) at 08:00:27:31:9e:e0 on em1 permanent [ethernet]
? (192.168.1.18) at 08:00:27:01:1c:01 on em1 expires in 458 seconds [ethernet]
? (192.168.1.14) at 08:00:27:29:de:a1 on em0 permanent [ethernet]
? (192.168.1.5) at (incomplete) on em0 expired [ethernet]
```

**3.19** 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

**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** Δε βλέπουμε καμία διαφορά, λαμβάνουμε πάλι το μήνυμα “No route to host”.

**4.4** Παρατηρούμε πως δεν υπάρχει διαδρομή για το LAN2.

```
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



```

root@PC:~ # netstat -rn
Routing tables

Internet:
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** Πάλι δεν ανταποκρίνεται, αλλά αυτή τη φορά δεν υπάρχει το μήνυμα "no route to host".

**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 στο PC3 δεν έχει οριστεί προκαθορισμένη πύλη.

**4.9** Εκτελούμε στο PC3 την εντολή "route add default 192.168.1.17".

**4.10** Πλέον υπάρχει επικοινωνία, γιατί ο PC3 ξέρει πλέον που να στείλει τα ICMP replies (στο router).

**4.11** Βλέπουμε ένα ενδιάμεσο βήμα μεταξύ αποστολέα και παραλήπτη:

```

root@PC:~ # 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)  0.636 ms  0.579 ms  0.568 ms
 2  192.168.1.18 (192.168.1.18)  0.744 ms  0.933 ms  0.990 ms

```

**4.12** arp -d -a

**4.13** Στο R1 σε μία κονσόλα εκτελούμε "tcpdump -vnve -i em0", ενώ σε μία δεύτερη "tcpdump -vnve -i em1".

**4.14** ping -c 1 192.168.1.18

**4.15**

MAC source : 08:00:27:b0:fe:04 (PC1)

MAC destination : 08:00:27:29:de:a1 (R1)

IPv4 source : 192.168.1.1 (PC1)

IPv4 destination : 192.168.1.18 (PC3)

**4.16**

MAC source : 08:00:27:31:9e:e0 (R1)  
MAC destination : 08:00:27:01:1c:01 (PC3)  
IPv4 source : 192.168.1.1 (PC1)  
IPv4 destination : 192.168.1.18 (PC3)

**4.17** Οι διευθύνσεις IP μένουν σταθερές, αλλά μεταβάλλονται οι διευθύνσεις MAC όσο το πακέτο προωθείται από δρομολογητή σε δρομολογητή. (Από τη στιγμή που η διεύθυνση IP δεν ανήκει στο τοπικό δίκτυο, αυτή προωθείται κατευθείαν στο default gateway ο οποίος είναι υπεύθυνος να βρει αυτός τη MAC του προορισμού, διατηρώντας ωστόσο τη διεύθυνση IP, γιατί αλλιώς το πακέτο θα χανόταν)

**4.18** Εκτελούμε την εντολή “ssh lab@192.168.1.18” από το PC1.

**4.19**

```
lab@PC:~ % netstat -an | grep 192.168.1.1
tcp4      0      44 192.168.1.18.22      192.168.1.1.20329    ESTABLISHED
```

Όπως βλέπουμε, χρησιμοποιείται το πρωτόκολλο μεταφοράς TCP, τοπική θύρα σύνδεσης είναι η 22 (στο PC3), ενώ απομακρυσμένη είναι η 20329.

**4.20** netstat -p tcp → Δεν παρατηρούμε κάτι γιατί ο R1 δεν μπορεί να δει δεδομένα που βρίσκονται στο στρώμα μεταφοράς

## Άσκηση 5

**5.1** Για το PC3: ifconfig em0 192.168.1.29/28

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** arp -d -a

**5.3** tcpdump -i em0 arp or icmp → R1

**5.4** tcpdump -i em0 arp or icmp → PC4

**5.5** ping -c 1 192.168.1.X με X = {2, 18, 29}

Τα ping είναι επιτυχή

**5.6** Βλέπουμε τα περιεχόμενα των πινάκων ARP στα PC1, PC2, PC3, PC4, R1:

```
root@PC:~ # arp -a
? (192.168.1.14) at 08:00:27:29:de:a1 on em0 expires in 1186 seconds [ethernet]
? (192.168.1.1) at 08:00:27:b0:fe:04 on em0 permanent [ethernet]
? (192.168.1.2) at 08:00:27:9c:00:87 on em0 expires in 1165 seconds [ethernet]
```

```
root@PC:~ # arp -a
? (192.168.1.1) at 08:00:27:b0:fe:04 on em0 expires in 1155 seconds [ethernet]
? (192.168.1.2) at 08:00:27:9c:00:87 on em0 permanent [ethernet]
```

```
root@PC:~ # arp -a
? (192.168.1.17) at 08:00:27:31:9e:e0 on em0 expires in 1132 seconds [ethernet]
? (192.168.1.18) at 08:00:27:01:1c:01 on em0 permanent [ethernet]
```

```
root@PC:~ # arp -a
? (192.168.1.29) at 08:00:27:03:c8:62 on em0 permanent [ethernet]
? (192.168.1.17) at 08:00:27:31:9e:e0 on em0 expires in 1109 seconds [ethernet]
```

```
root@PC:~ # arp -a
? (192.168.1.29) at 08:00:27:03:c8:62 on em1 expires in 1075 seconds [ethernet]
? (192.168.1.17) at 08:00:27:31:9e:e0 on em1 permanent [ethernet]
? (192.168.1.18) at 08:00:27:01:1c:01 on em1 expires in 1072 seconds [ethernet]
? (192.168.1.14) at 08:00:27:29:de:a1 on em0 permanent [ethernet]
? (192.168.1.1) at 08:00:27:b0:fe:04 on em0 expires in 1072 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 08:00:27:29:de:a1, 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 08:00:27:03:c8:62, 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 -a -d” 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** Ναι, ήταν επιτυχές. Έφτασε με ένα βήμα

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

```
root@PC:~ # arp -a
? (192.168.1.17) at 08:00:27:31:9e:e0 on em0 expires in 1049 seconds [ethernet]
? (192.168.1.18) at 08:00:27:01:1c:01 on em0 permanent [ethernet]
```

```
root@PC:~ # arp -a
? (192.168.1.29) at 08:00:27:03:c8:62 on em0 permanent [ethernet]
? (192.168.1.17) at 08:00:27:31:9e:e0 on em0 expires in 1113 seconds [ethernet]
? (192.168.1.18) at 08:00:27:01:1c:01 on em0 expires in 1113 seconds [ethernet]
```

```
root@PC:~ # arp -a
? (192.168.1.29) at 08:00:27:03:c8:62 on em1 expires in 1134 seconds [ethernet]
? (192.168.1.17) at 08:00:27:31:9e:e0 on em1 permanent [ethernet]
? (192.168.1.18) at 08:00:27:01:1c:01 on em1 expires in 1134 seconds [ethernet]
? (192.168.1.14) at 08:00:27:29:de:a1 on em0 permanent [ethernet]
```

### 5.11

PC3 → R1 : arp request  
R1 → PC3 : arp reply  
PC3 → R1 : icmp request  
R1 → PC4 : arp request  
R1 → PC3 : icmp redirect  
PC4 → R1 : arp reply  
PC3 → PC4 : icmp request  
PC4 → PC3 : arp request  
PC3 → PC4 : arp reply  
PC4 → PC3 : icmp reply

**5.12** Το PC3 αναζητεί τη MAC διεύθυνση του R1 (αφού το PC4 δεν ανήκει στο ίδιο υποδίκτυο), ενώ το PC4 του PC3.

**5.13** Γιατί δεν ανήκει στο ίδιο υποδίκτυο.

(Το PC3 “βλέπει” τις διευθύνσεις 192.168.1.16 - 192.168.1.23 στο LAN2)

**5.14** Το κάνει redirect στο PC4

**5.15** Στάλθηκε απευθείας στο PC3, γιατί ο PC3 ανήκει στο υποδίκτυο του PC4

**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 του 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 δρομολόγησής του.

**5.18** Η default gateway χάνεται

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

```
Internet:
Destination      Gateway          Flags      Netif Expire
127.0.0.1         link#2           UH          lo0
192.168.1.16/28   link#1           U          em0
192.168.1.18      link#1           UHS         lo0
```

#### 5.19

```
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** Πλέον τα requests πηγαίνουν κατευθείαν στο PC4

**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.

```
root@PC:~ # traceroute 192.168.1.29
traceroute to 192.168.1.29 (192.168.1.29), 64 hops max, 40 byte packets
 1  192.168.1.29 (192.168.1.29)  0.749 ms  0.900 ms  0.685 ms
```

## Άσκηση 6

**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** Στο R1: ifconfig em0.5 create/ ifconfig em0.6 create/ ifconfig em1.5 create/ ifconfig em1.6 create

**6.7** Μπορούμε να κάνουμε κανονικά ring στις διεπαφές em0, em0.6, αλλά όχι στην em0.5

**6.8** Μπορούμε να κάνουμε στην em0 και στην em0.5, αλλά όχι στην em0.6.

**6.9** Απέτυχαν διότι προσπάθησαν να κάνουν ring σε VLANs για τα οποία δε διέθεταν διεπαφή.

**6.10** Ναι, μπορούμε

**6.11** Ναι, μπορούμε

**6.12** Όχι, δε μπορούμε να κάνουμε σε καμία.

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

**6.14** Ναι, πλέον επιτυγχάνουν

**6.15**

(PC1) → 08:00:27:b0:fe:04

(PC2) → 08:00:27:9c:00:87

(PC3) → 08:00:27:01:1c:01

**6.16** tcpdump -e

**6.17** 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.

Επιγραμματικά:

PC3 → PC1 : arp request

PC1 → PC3 : arp reply

PC3 → PC1 : icmp request

PC1 → PC2 : arp request  
PC2 → PC1 : arp reply  
PC1 → PC2 : icmp request  
PC2 → PC1 : icmp reply  
PC1 → PC3 : icmp reply

**6.18** Όχι δεν είναι.

**6.19** Απαντά στα ICMP requests, αλλά οι απαντήσεις δεν φτάνουν ποτέ στον PC3 γιατί το PC4 δεν έχει ως default gateway το PC1, αλλά το R1

**6.20** Πλέον επιτυγχάνει κανονικά.

Η διαδρομή που ακολουθείται είναι η εξής:

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