



Εργαστήριο Δικτύων Υπολογιστών

ΕΡΓΑΣΤΗΡΙΑΚΗ ΑΣΚΗΣΗ 5
ΣΤΑΤΙΚΗ ΔΡΟΜΟΛΟΓΗΣΗ

Κουστένης Χρίστος | el20227 | 19/03/2024

Άσκηση 1: Δρομολόγηση σε ένα βήμα

1.1

```
ifconfig em0 192.168.2.1/24 --> PC1
```

```
ifconfig em0 192.168.2.2/24 --> PC2
```

1.2

```
sysrc ifconfig_em0="inet 192.168.1.1 netmask 255.255.255.0"
```

```
sysrc ifconfig_em1="inet 192.168.2.1 netmask 255.255.255.0"
```

1.3

```
gateway_enable="YES"
```

1.4

```
service netif restart && service routing restart
```

1.5

```
route add -net 192.168.2.0/24 192.168.1.1 --> PC1
```

1.6

```
netstat -r
```

UGS

U : Η διαδρομή είναι ενεργή

G : Ο προορισμός είναι πύλη, που θα αποφασίσει για το πώς θα προωθήσει τα πακέτα περαιτέρω.

S : Η διαδρομή έχει οριστεί στατικά.

1.7

Ενώ στάλθηκε το ping, δε λαμβάνουμε απάντηση.

1.8

Το LAN1 παρατηρούμε πως το PC1 στέλνει τα ICMP Echo requests του στη διεπαφή em0 του R1, ενώ στο LAN2 το R1 στέλνει μέσω της em1 τα αιτήματα στο PC2. Επομένως, ενώ ο PC2 λαμβάνει κανονικά τα requests του PC1, αδυνατεί να απαντήσει, καθώς δε ξέρει προς τα που πρέπει να προωθήσει τα replies.

1.9

```
route add -net 192.168.1.0/24 192.168.2.1 --> PC2
```

1.10

Ναι.

1.11

Ο πίνακας δρομολόγησης του R1, όπως φαίνεται έχει ήδη την απαραίτητη πληροφορία για δρομολόγηση στα LAN1 (192.168.1.0/24) και LAN2 (192.168.2.0/24), οπότε και δεν απαιτείται κάποια επιπλέον ρύθμιση.

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

Internet:
Destination        Gateway             Flags      Netif Expire
localhost           link#3              UHS        lo0
192.168.1.0/24      link#1              U          em0
192.168.1.1         link#1              UHS        lo0
192.168.2.0/24      link#2              U          em1
192.168.2.1         link#2              UHS        lo0

Internet6:
Destination        Gateway             Flags      Netif Expire
::/96              localhost           URS        lo0
localhost           link#3              UHS        lo0
::ffff:0.0.0.0/96  localhost           URS        lo0
fe80::/10          localhost           URS        lo0
fe80::%lo0/64      link#3              U          lo0
fe80::1%lo0        link#3              UHS        lo0
ff02::/16          localhost           URS        lo0
root@PC:~ #
```

Άσκηση 2: Proxy ARP

2.1

route del 192.168.2.0/24 --> PC1

2.2

ifconfig em0 inet 192.168.1.2/20 --> PC1

2.3

Το PC1 βρίσκεται στο υποδίκτυο 192.168.0.0. Εάν εφαρμόσουμε τη μάσκα του υποδικτύου του στις διευθύνσεις των PC2, PC3 βλέπουμε πως το PC1 τα αντιλαμβάνεται σα να ανήκουν στο ίδιο υποδίκτυο.

2.4

Όχι, δεν είναι. Λαμβάνουμε σφάλμα « ping: sendto: Host is down ».

2.5

Το ping είναι επιτυχές, καθώς ο δρομολογητής λειτουργεί ως proxy, επομένως απαντάει με τη δική του MAC στα ARP requests του PC1, δεδομένου ότι το PC2 βρίσκεται σε υποδίκτυο στο οποίο ο R1 ξέρει πώς να δρομολογήσει πακέτα για εκεί.

2.6

Γιατι στο PC3 δεν έχουμε προσθέσει στο πίνακα προώθησης την εγγραφή για το 192.168.1.0/24 .

2.7

`route add -net 192.168.1.0/24 192.168.2.1 --> PC3`

2.8

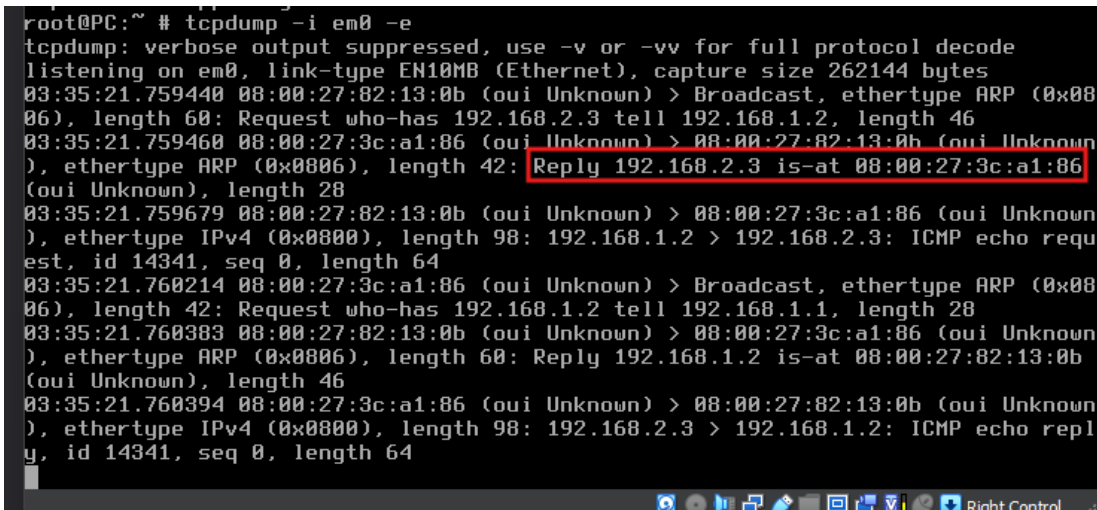
`arp -ad`

2.9

Στο R1 εκτελούμε σε μία κονσόλα «`tcpdump -ei em0`» και σε μία δεύτερη «`tcpdump -ei em1`».

2.10

Βάζει στο ARP reply τη δικιά του MAC διεύθυνση.



```
root@PC:~ # tcpdump -i em0 -e
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on em0, link-type EN10MB (Ethernet), capture size 262144 bytes
03:35:21.759440 08:00:27:82:13:0b (oui Unknown) > Broadcast, ethertype ARP (0x08006), length 60: Request who-has 192.168.2.3 tell 192.168.1.2, length 46
03:35:21.759460 08:00:27:3c:a1:86 (oui Unknown) > 08:00:27:82:13:0b (oui Unknown), ethertype ARP (0x08006), length 42: Reply 192.168.2.3 is-at 08:00:27:3c:a1:86 (oui Unknown), length 28
03:35:21.759679 08:00:27:82:13:0b (oui Unknown) > 08:00:27:3c:a1:86 (oui Unknown), ethertype IPv4 (0x0800), length 98: 192.168.1.2 > 192.168.2.3: ICMP echo request, id 14341, seq 0, length 64
03:35:21.760214 08:00:27:3c:a1:86 (oui Unknown) > Broadcast, ethertype ARP (0x08006), length 42: Request who-has 192.168.1.2 tell 192.168.1.1, length 28
03:35:21.760383 08:00:27:82:13:0b (oui Unknown) > 08:00:27:3c:a1:86 (oui Unknown), ethertype ARP (0x08006), length 60: Reply 192.168.1.2 is-at 08:00:27:82:13:0b (oui Unknown), length 46
03:35:21.760394 08:00:27:3c:a1:86 (oui Unknown) > 08:00:27:82:13:0b (oui Unknown), ethertype IPv4 (0x0800), length 98: 192.168.2.3 > 192.168.1.2: ICMP echo reply, id 14341, seq 0, length 64
```

Επίσης, βλέπουμε πως το το R1 απαντάει με τη MAC του em0 του, παρόλο που το ring έχει ως προορισμό το PC3.

2.11

Προς τη MAC 08:00:27:3C:A1:86 (em0 του R1).

2.12

Από τη MAC 08:00:27:44:58:03 (em1 του R1).

2.13

Παρατηρήσαμε τα παρακάτω πακέτα:

- PC1 κάνει broadcast ARP request για να μάθει την MAC address της διεύθυνσης 192.168.2.3
- Το R1 απαντάει με ARP reply δίνοντάς του τη MAC της διεπαφής em0 ως MAC της 192.168.2.3, αφού το έχουμε δηλώσει ως proxy.
- Το PC1 στέλνει στο em0 του R1 το ICMP echo request.

- Το PC1, μέσω της em1 κάνει broadcast ένα ARP request με σκοπό να μάθει την MAC της 192.168.2.3
- Το PC3 απαντάει με ARP Reply στο παραπάνω broadcast με τη MAC διεύθυνσή του.
- Το R1 προωθεί στο PC3 το ICMP echo request μέσω της em1
- Το PC3 απαντάει στην em1 του R1 με ICMP echo reply, με τελικό αποδέκτη τη διεύθυνση 192.168.1.2.
- Το R1 κάνει broadcast ένα ARP Request μέσω της em0, ώστε να μάθει την MAC της 192.168.1.2, μιας και δε φαίνεται να την αποθήκευσε από το προηγούμενο broadcast του PC1
- Το PC1 απαντάει με τη MAC διεύθυνσή του στο R1
- Το R1 προωθεί το ICMP echo reply στο PC1 με $MAC_{destination} = MAC_{PC1}$ και $MAC_{source} = MAC_{em0_R1}$

2.14

Το ping θα επιτυγχάνει όσο το PC1 νομίζει πως το PC3 είναι στο ίδιο υποδίκτυο με αυτό. Επομένως, το μέγιστο μήκος προθέματος είναι 22, καθώς αν βάλουμε 23, το PC1 αντιλαμβάνεται πως το PC3 ανήκει στο 192.168.2.0/23, ενώ το ίδιο το PC1 ανήκει στο 192.168.0.0/23, άρα από τα 23 bits και μετά απαιτείται δρομολόγηση, για την οποία δεν έχουμε ορίσει κάποια πύλη στο PC1, οπότε και το Ping θα αποτυγχάνει. (Σημείωση: Μέχρι τα 22 bits, το PC1 αντιλαμβάνεται αμφότερα μηχανήματα στο υποδίκτυο 192.168.0.0/22)

2.15

```
ifconfig em0 192.168.1.2/23 --> PC1
```

2.16

```
route add -net 192.168.2.0/24 -interface em0 --> PC1
```

2.17

Με « **netstat -r** » βλέπουμε πως ως πύλη για το δίκτυο 192.168.2.0/24 εμφανίζεται η διεπαφή em0.

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

Internet:
Destination        Gateway             Flags               Netif  Expire
localhost           link#2              UH                  lo0
192.168.0.0/23      link#1              U                   em0
192.168.1.2         link#1              UHS                 lo0
192.168.2.0/24      link#1              US                  em0

Internet6:
Destination        Gateway             Flags               Netif  Expire
::/96               localhost           URS                 lo0
localhost           link#2              UHS                 lo0
::ffff:0.0.0.0/96   localhost           URS                 lo0
fe80::/10           localhost           URS                 lo0
fe80::%lo0/64       link#2              U                   lo0
fe80::1%lo0         link#2              UHS                 lo0
ff02::/16           localhost           URS                 lo0
root@PC:~ #
```

2.18

Πλέον το Ping επιτυγχάνει, καθώς το ταίριασμα μεγαλύτερου προθέματος γίνεται με το υποδίκτυο 192.168.2.0/24, οπότε και το em0 κάνει τα κατάλληλα ARP requests, ώστε να στείλει τα ICMP πακέτα και λαμβάνει απαντήσεις από το proxy ARP, δηλαδή το R1, το οποίο απαντάει σα να ήταν το PC3.

2.19

```
sysctl net.link.ether.inet.proxyall=0 --> R1
```

2.20

```
route change -net 192.168.2.0/24 192.168.1.1 --> PC1
```

2.21

```
ifconfig em0 192.168.1.2/24 --> PC1
```

2.22

Εξαφανίστηκε.

2.23

```
route add -net 192.168.2.0/24 192.168.1.1 --> PC1
```

2.24

```
ifconfig em0 delete 192.168.2.3
```

Uncheck cable connected on adapter

Άσκηση 3 : Δρομολόγηση σε περισσότερα βήματα

3.1

R1

```
sysrc ifconfig_em1="inet 172.17.17.1 netmask 255.255.255.252"
```

```
service netif restart
```

3.2

R2

```
sysrc ifconfig_em0="inet 172.17.17.2 netmask 255.255.255.252"
```

```
sysrc ifconfig_em1="inet 192.168.2.1 netmask 255.255.255.0"
```

```
service netif restart
```

3.3

gateway_enable="YES" (υπήρχε ήδη η γραμμή αυτή γιατί το R2 φτιάχτηκε ως linked clone του R1)

service routing restart

3.4

Destination Host Unreachable

```

root@PC:~ # ping -c 1 192.168.2.2
PING 192.168.2.2 (192.168.2.2): 56 data bytes
92 bytes from 192.168.1.1: Destination Host Unreachable
Vr HL TOS Len ID Flg off TTL Pro cks Src Dst
 4 5 00 0054 4f32 0 0000 3f 01 a822 192.168.1.2 192.168.2.2

--- 192.168.2.2 ping statistics ---
1 packets transmitted, 0 packets received, 100.0% packet loss
root@PC:~ #

```

3.5

Στο LAN1 παράγονται μηνύματα ICMP echo request και ICMP host unreachable.

Στο WAN1 δεν παράγεται καθόλου κίνηση γιατί ο R1 δεν έχει κάποια εγγραφή ώστε να προωθήσει τα echo requests.

3.6

tracert 192.168.2.2 --> PC2

```
root@PC:~ # traceroute 192.168.2.2
traceroute to 192.168.2.2 (192.168.2.2), 64 hops max, 40 byte packets
 1  192.168.1.1 (192.168.1.1)  0.360 ms  0.225 ms  0.186 ms
 2  192.168.1.1 (192.168.1.1)  0.180 ms !H  0.188 ms !H  0.218 ms !H
root@PC:~ #
```

Εκτελώντας « **man traceroute** » βλέπουμε πως το « **!H** » αναφέρεται στο Host Unreachable.

3.7

```
route add -net 192.168.2.0/24 172.17.17.2--> R1
```

3.8

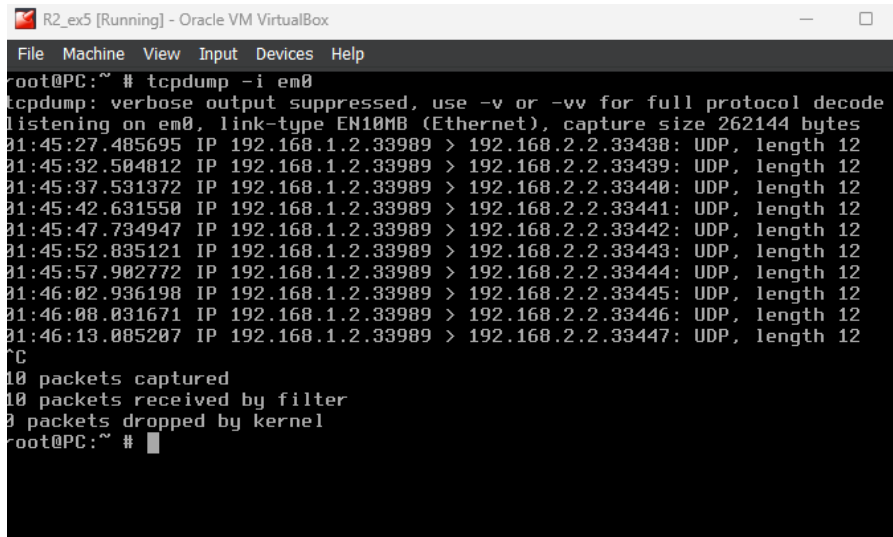
Πλέον, το ping του PC1 στο PC2 επιτυγχάνει ως Request, ωστόσο δε λαμβάνουμε πίσω το Reply, καθώς όταν στέλνει το Reply το PC2, αυτό αδυνατεί να προωθηθεί από το R2, επομένως στέλνεται ένα « ICMP host 192.168.1.2 unreachable » από το R2 στο PC2.

```
01:00:46.521844 IP 192.168.2.1 > 192.168.2.2: ICMP host 192.168.1.2 unreachable  
length 92
```

3.9

- IP 192.168.1.2 > 192.168.2.2: **ICMP echo request**, το οποίο είναι το πακέτο του PC1 που το R2 προωθεί στο PC2
- IP 192.168.2.2 > 192.168.1.2: **ICMP echo reply**, το οποίο είναι το πακέτο που το PC2 στέλνει στο R2 με τελικό προορισμό το PC1.
- IP 192.168.2.1 > 192.168.2.2: **ICMP host 192.168.1.2 unreachable**, το οποίο είναι η απάντηση που το R2 στέλνει στο PC2, ενημερώνοντας το πως δε μπορεί να προωθήσει το προηγούμενο reply.

3.10



```
R2_ex5 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
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
01:45:27.485695 IP 192.168.1.2.33989 > 192.168.2.2.33438: UDP, length 12
01:45:32.504812 IP 192.168.1.2.33989 > 192.168.2.2.33439: UDP, length 12
01:45:37.531372 IP 192.168.1.2.33989 > 192.168.2.2.33440: UDP, length 12
01:45:42.631550 IP 192.168.1.2.33989 > 192.168.2.2.33441: UDP, length 12
01:45:47.734947 IP 192.168.1.2.33989 > 192.168.2.2.33442: UDP, length 12
01:45:52.835121 IP 192.168.1.2.33989 > 192.168.2.2.33443: UDP, length 12
01:45:57.902772 IP 192.168.1.2.33989 > 192.168.2.2.33444: UDP, length 12
01:46:02.936198 IP 192.168.1.2.33989 > 192.168.2.2.33445: UDP, length 12
01:46:08.031671 IP 192.168.1.2.33989 > 192.168.2.2.33446: UDP, length 12
01:46:13.085207 IP 192.168.1.2.33989 > 192.168.2.2.33447: UDP, length 12
^C
10 packets captured
10 packets received by filter
0 packets dropped by kernel
root@PC:~ #
```

Στο WAN1 δε παρατηρούμε πακέτα ICMP, παρά μόνο UDP, με αποστολέα το 192.168.1.2.33989 και παραλήπτη τη διεύθυνση 192.168.2.2.XXXXX με διαφορετική κάθε φορά θύρα προορισμού προκειμένου ο κόμβος-παραλήπτης να μην επεξεργαστεί τα UDP packets. Καταγράφουμε τα εν λόγω πακέτα, καθώς αυτά αποστέλλονται μέσω του traceroute με ένα μικρό TTL μέχρι να ληφθεί απάντηση ICMP time exceeded.

3.11

Στο LAN2 βλέπουμε να προωθούνται τα ανωτέρω UDP πακέτα από το R2 στο PC2, ενώ επιπλέον βλέπουμε ως απάντηση από το PC2 (192.168.2.2) στο R2 (192.168.1.2) μηνύματα ICMP 192.168.2.2 udr port XXXXX unreachable, όπου XXXXX η εκάστοτε θύρα προορισμού.

3.12

Το PC2 αποκρίνεται στο R2 λέγοντας πως ήταν unreachable το port του μηνύματος που έλαβε. Δε παράγονται ICMP destination unreachable μηνύματα ως απόκριση στα ICMP που παράγει το PC2, καθώς σε αυτή την περίπτωση θα προκαλούνταν loop στο σύστημα.

3.13

```
route add -net 192.168.1.0/24 172.17.17.1 --> R2
```

3.14


```

root@PC:~ # tcpdump -i en0
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on en0, link-type EN10MB (Ethernet), capture size 262144 bytes
02:40:37.584137 IP 192.168.1.2.34069 > 192.168.2.2.33438: UDP, length 12
02:40:37.584163 IP 172.17.17.2 > 192.168.1.2: ICMP time exceeded in-transit, length 48
02:40:37.584918 IP 192.168.1.2.34069 > 192.168.2.2.33439: UDP, length 12
02:40:37.584938 IP 172.17.17.2 > 192.168.1.2: ICMP time exceeded in-transit, length 48
02:40:37.585276 IP 192.168.1.2.34069 > 192.168.2.2.33440: UDP, length 12
02:40:37.585285 IP 172.17.17.2 > 192.168.1.2: ICMP time exceeded in-transit, length 48
02:40:37.585727 IP 192.168.1.2.34069 > 192.168.2.2.33441: UDP, length 12
02:40:37.586012 IP 192.168.2.2 > 192.168.1.2: ICMP 192.168.2.2 udp port 33441 unreachable, length 48
02:40:37.586688 IP 192.168.1.2.34069 > 192.168.2.2.33442: UDP, length 12
02:40:37.586902 IP 192.168.2.2 > 192.168.1.2: ICMP 192.168.2.2 udp port 33442 unreachable, length 48
02:40:37.587276 IP 192.168.1.2.34069 > 192.168.2.2.33443: UDP, length 12
02:40:37.587492 IP 192.168.2.2 > 192.168.1.2: ICMP 192.168.2.2 udp port 33443 unreachable, length 48

```

Πλέον μπορούμε να κάνουμε κανονικά traceroute. Στο WAN1 παράγονται μηνύματα τύπου **ICMP time exceeded in-transit (172.17.17.2 > 192.168.1.2)**, ενώ επιπλέον καταγράφονται μηνύματα **ICMP 192.168.2.2 udp port 3344X (X={1,2,3}) unreachable (192.168.2.2 > 192.168.1.2)**. Τα ICMP time exceeded μηνύματα που παρήχθησαν οφείλονται στο γεγονός ότι στη διεπαφή 172.17.17.2 του R2 μηδενίστηκε το TTL (TTL = 2) της δεύτερης τριάδας πακέτων που απεστάλησαν από το traceroute του PC1.

3.15

```

root@PC:~ # ping 172.17.17.1
PING 172.17.17.1 (172.17.17.1): 56 data bytes
ping: sendto: No route to host
ping: sendto: No route to host
ping: sendto: No route to host
^C
--- 172.17.17.1 ping statistics ---
3 packets transmitted, 0 packets received, 100.0% packet loss
root@PC:~ #

```

Λαμβάνουμε ως απάντηση « **No route to host** », πράγμα που οφείλεται στο γεγονός ότι ο πίνακας δρομολόγησης του PC2 δε περιλαμβάνει εγγραφές ούτε για το υποδίκτυο της διεύθυνσης 172.17.17.1, αλλά ούτε και έχει default gateway, ώστε το πακέτο να δρομολογηθεί από εκεί.

3.16

route del 192.168.1.0/24 --> PC2

3.17

route add default 192.168.2.1

3.18

Το ping εκτελείται με επιτυχία.

3.19

Λόγω το default gateway στο πίνακα δρομολόγησης του PC2 τη δεύτερη φορά θα στείλει σωστά το ping στο R2 ο οποίος με τη σειρά του θα το προωθήσει στο R1.

Άσκηση 4: Ένα πιο πολύπλοκο δίκτυο με εναλλακτικές διαδρομές

4.1

```
ifconfig em0 192.168.2.3/24 --> PC3
```

```
route add -net 192.168.1.0/24 192.168.2.1 --> PC3
```

4.2

```
sysrc ifconfig_em2="inet 172.17.17.5 netmask 255.255.255.252" --> R1
```

```
service netif restart --> R1
```

4.3

```
sysrc ifconfig_em2="inet 172.17.17.9 netmask 255.255.255.252" --> R2
```

```
service netif restart --> R2
```

4.4

```
sysrc ifconfig_em0="inet 172.17.17.6 netmask 255.255.255.252" --> R3
```

```
sysrc ifconfig_em1="inet 172.17.17.10 netmask 255.255.255.252" --> R3
```

```
service netif restart --> R3
```

4.5

Υπάρχει ήδη η ζητούμενη γραμμή γιατί το R3 είναι linked clone.

```
service routing restart
```

4.6

```
route add -net 192.168.2.0/24 172.17.17.2 --> R1
```

4.7

```
route add -net 192.168.1.0/24 172.17.17.1 --> R2
```

4.8

```
route add -net 192.168.1.0/24 172.17.17.5 --> R3
```

```
route add -net 192.168.2.0/24 172.17.17.9 --> R3
```

4.9

```
route add -host 192.168.2.3 172.17.17.6
```

```
netstat -r
```

192.168.2.3	172.17.17.6	UGHS	em2
-------------	-------------	------	-----

Η σημαία H.

4.10

```
traceroute 192.168.2.2
```

```
root@PC:~ # traceroute 192.168.2.2
traceroute to 192.168.2.2 (192.168.2.2), 64 hops max, 40 byte packets
 1  192.168.1.1 (192.168.1.1)  0.298 ms  0.239 ms  0.204 ms
 2  172.17.17.2 (172.17.17.2)  15.076 ms  0.423 ms  0.370 ms
 3  192.168.2.2 (192.168.2.2)  1.052 ms  0.628 ms  0.579 ms
root@PC:~ #
```

Βλέπουμε συνολικά 3 βήματα, το 3ο εκ των οποίων γίνεται στο destination (PC2).

4.11

```
ping 192.168.2.2
```

```
root@PC:~ # ping 192.168.2.2
PING 192.168.2.2 (192.168.2.2): 56 data bytes
64 bytes from 192.168.2.2: icmp_seq=0 ttl=62 time=1.002 ms
64 bytes from 192.168.2.2: icmp_seq=1 ttl=62 time=0.866 ms
64 bytes from 192.168.2.2: icmp_seq=2 ttl=62 time=0.974 ms
64 bytes from 192.168.2.2: icmp_seq=3 ttl=62 time=0.846 ms
64 bytes from 192.168.2.2: icmp_seq=4 ttl=62 time=0.832 ms
^C
--- 192.168.2.2 ping statistics ---
5 packets transmitted, 5 packets received, 0.0% packet loss
round-trip min/avg/max/stddev = 0.832/0.904/1.002/0.070 ms
root@PC:~ #
```

ttl = 62. Άρα 2 βήματα.

4.12

4 βήματα.

```
root@PC:~ # traceroute 192.168.2.3
traceroute to 192.168.2.3 (192.168.2.3), 64 hops max, 40 byte packets
 1  192.168.1.1 (192.168.1.1)  0.364 ms  0.238 ms  0.209 ms
 2  172.17.17.6 (172.17.17.6)  1.281 ms  0.391 ms  0.417 ms
 3  172.17.17.2 (172.17.17.2)  1.191 ms  0.434 ms  0.401 ms
 4  192.168.2.3 (192.168.2.3)  0.893 ms  0.650 ms  0.658 ms
root@PC:~ #
```

4.13

```
root@PC:~ # ping 192.168.2.3
PING 192.168.2.3 (192.168.2.3): 56 data bytes
64 bytes from 192.168.2.3: icmp_seq=0 ttl=62 time=1.033 ms
64 bytes from 192.168.2.3: icmp_seq=1 ttl=62 time=1.007 ms
64 bytes from 192.168.2.3: icmp_seq=2 ttl=62 time=0.946 ms
^C
--- 192.168.2.3 ping statistics ---
3 packets transmitted, 3 packets received, 0.0% packet loss
round-trip min/avg/max/stddev = 0.946/0.996/1.033/0.036 ms
root@PC:~ #
```

ttl = 62. Άρα 2 βήματα.

4.14

To ICMP echo request ακολουθεί τη διαδρομή PC1 → R1 → R3 → R2 → PC3.

4.15

Αντιθέτως, το ICMP echo reply ακολουθεί τη διαδρομή PC3 → R2 → R1 → PC1, δεδομένου πως είχαμε ορίσει στατική εγγραφή στον R2 ώστε να προωθούνται πακέτα προς το LAN1 μέσω του R1, ενώ είχαμε ορίσει επίσης στατική εγγραφή στο R1, έτσι ώστε πακέτα προς το PC3 να διέρχονται από το R3.

4.16

`tcpdump -i em1 --> R2`

4.17

Όχι, δεν παρατηρούμε κανένα είδος πακέτου.

4.18

Ναι, παράγονται τα παρακάτω πακέτα.

```
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
16:53:34.635512 ARP, Request who-has 192.168.2.3 tell 192.168.2.1, length 28
16:53:34.635772 ARP, Reply 192.168.2.3 is-at 08:00:27:56:13:3a (oui Unknown), length 46
16:53:34.635782 IP 192.168.1.2.34800 > 192.168.2.3.33444: UDP, length 12
16:53:34.636023 IP 192.168.2.3 > 192.168.1.2: ICMP 192.168.2.3 udp port 33444 unreachable, length 48
16:53:39.786493 IP 192.168.1.2.34800 > 192.168.2.3.33445: UDP, length 12
16:53:39.786772 IP 192.168.2.3 > 192.168.1.2: ICMP 192.168.2.3 udp port 33445 unreachable, length 48
16:53:44.809573 IP 192.168.1.2.34800 > 192.168.2.3.33446: UDP, length 12
16:53:44.809843 IP 192.168.2.3 > 192.168.1.2: ICMP 192.168.2.3 udp port 33446 unreachable, length 48
16:53:49.875869 IP 192.168.1.2.34800 > 192.168.2.3.33447: UDP, length 12
16:53:49.876187 IP 192.168.2.3 > 192.168.1.2: ICMP 192.168.2.3 udp port 33447 unreachable, length 48
16:53:54.898967 IP 192.168.1.2.34800 > 192.168.2.3.33448: UDP, length 12
16:53:54.899258 IP 192.168.2.3 > 192.168.1.2: ICMP 192.168.2.3 udp port 33448 unreachable, length 48
```

Παρατηρούμε την παρουσία πακέτων *ICMP 192.168.2.3 udp port XXXXX unreachable*.

Το traceroute ωστόσο αποτυγχάνει καθώς ο R2 δεν μπορεί να προωθήσει replies πίσω στο PC1 αφού η διαδρομή που του ορίσαμε στατικά να ακολουθήσει μέσω WAN1 έχει βλάβη.

4.19

Ναι.

4.20

route change -net 192.168.2.0/24 172.17.17.6 --> R1

route change -net 192.168.1.0/24 172.17.17.10 -->R2

4.21

R1

route show 192.168.2.2

route show 192.168.2.3

```
root@PC:~ # route show 192.168.2.2
route to: 192.168.2.2
destination: 192.168.2.0
mask: 255.255.255.0
gateway: 172.17.17.6
fib: 0
interface: em2
flags: <UP,GATEWAY,DONE,STATIC>
rcvpipe sendpipe ssthresh rtt,msec mtu weight expire
0 0 0 0 1500 1 0
root@PC:~ # route show 192.168.2.3
route to: 192.168.2.3
destination: 192.168.2.3
gateway: 172.17.17.6
fib: 0
interface: em2
flags: <UP,GATEWAY,HOST,DONE,STATIC>
rcvpipe sendpipe ssthresh rtt,msec mtu weight expire
0 0 0 0 1500 1 0
root@PC:~ #
```

Η διαφορά που παρατηρούμε είναι πως, για το PC2 το destination είναι το subnet 192.168.2.0/24, ενώ για το PC3 το destination είναι η ίδια η IP του (192.168.2.3), μιας και είχαμε ορίσει προηγουμένως στατική εγγραφή προς αυτό από το R1 μέσω του R3. Για τον ίδιο λόγο έχουμε ενεργοποιημένη και τη σημαία HOST στην δεύτερη δρομολόγηση.

4.22

Μεταξύ των 2 τελευταίων εγγραφών, επιλέγεται η τελευταία, καθώς έχουμε ταίριασμα μήκους 32 bits.

```

root@PC:~ # netstat -r -4
Routing tables

Internet:
Destination      Gateway          Flags           Netif Expire
localhost         link#4          UH              lo0
172.17.17.0/30    link#2          U               em1
172.17.17.1       link#2          UHS             lo0
172.17.17.4/30    link#3          U               em2
172.17.17.5       link#3          UHS             lo0
192.168.1.0/24    link#1          U               em0
192.168.1.1       link#1          UHS             lo0
192.168.2.0/24    172.17.17.6     UGS             em2
192.168.2.3       172.17.17.6     UGHS            em2
root@PC:~ #

```

4.23

```
route change -net 192.168.2.0/24 172.17.17.5 --> R3
```

4.24

Όχι, δεν είναι επιτυχές.

```

root@PC:~ # ping 192.168.2.2
PING 192.168.2.2 (192.168.2.2): 56 data bytes
92 bytes from 172.17.17.6: Time to live exceeded
  vr HL TOS Len  ID Flg  off TTL Pro  cks      Src      Dst
   4  5  00 0054 948d   0 0000  01  01 a0c7 192.168.1.2 192.168.2.2

92 bytes from 172.17.17.6: Time to live exceeded
  vr HL TOS Len  ID Flg  off TTL Pro  cks      Src      Dst
   4  5  00 0054 26f4   0 0000  01  01 0e61 192.168.1.2 192.168.2.2

92 bytes from 172.17.17.6: Time to live exceeded
  vr HL TOS Len  ID Flg  off TTL Pro  cks      Src      Dst
   4  5  00 0054 948e   0 0000  01  01 a0c6 192.168.1.2 192.168.2.2

92 bytes from 172.17.17.6: Time to live exceeded
  vr HL TOS Len  ID Flg  off TTL Pro  cks      Src      Dst
   4  5  00 0054 948f   0 0000  01  01 a0c5 192.168.1.2 192.168.2.2

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

```

4.25

Γίνονται συνεχόμενα redirects μεταξύ 192.168.1.1 και 172.17.17.6 γιατί το PC1 ξεκινάει στέλνοντας το ICMP echo request στο R1 το οποίο το προωθεί στο R3 το οποίο όμως λόγω του ερωτήματος 5.1 το ξαναπροωθεί στο R1 κ.λπ κ.λπ. Μέχρι που κάποια στιγμή μηδενίζεται το TTL. Το μήνυμα Time to live exceeded έρχεται από 172.17.17.6. Αν όμως, βάζαμε ως TTL μια περιττή τιμή τότε θα παίρναμε απάντηση από τη διεπαφή 192.168.1.1

4.26

```
tcpdump -i em0 -ve 'icmp[icmptype] == icmp-echo' --> R3
```

4.27

Στο WAN2 καταγράφηκαν 63 (packets captured) +1 Time Exceeded

4.28

32 ICMP echo requests με πηγή τον R1 και 31 με πηγή τον R2. Αυτό συμβαίνει γιατί by default τα πακέτα στέλνονται με ttl = 64 από το PC1 και έτσι εκτελώντας αλληπαλά τον βρόχο το πακέτο που στείλαμε καταλήγει να πεθαίνει (ttl = 0) στο R3.

4.29

`tcpdump -i em2 -e 'icmp[icmptype] == icmp-echo' --> R1`

`tcpdump -i em0 -e 'icmp[icmptype] == icmp-timxceed' --> R3`

4.30

Εμφανίζονται 64 βήματα, ενώ η διαδρομή που καταγράφεται είναι: PC1 → R1

(192.168.1.1) → R3 (172.17.17.6) → R1 (192.168.1.1) → R3 (172.17.17.6) → R1

(192.168.1.1) → R3 (172.17.17.6) ... → R1 (192.168.1.1) → R3 (172.17.17.6)

4.31

Λαμβάνουμε ως αποτέλεσμα μετά τον τερματισμό της καταγραφής ότι έγιναν capture 2016 πακέτα . Στο πρώτο ICMP echo request του PC1, αναμένουμε πως δε καταγράφηκε τίποτα στο WAN2, καθώς το TTL ήταν 1, οπότε

και απάντησε το R1 αμέσως. Για τα υπόλοιπα ICMP echo requests του PC1:

- 2^ο ICMP echo request: Καταγράφεται 1 στο WAN2
- 3^ο ICMP echo request: Καταγράφονται 2 στο WAN2
- 4^ο ICMP echo request: Καταγράφονται 3 στο WAN2
- ν-οστό ICMP echo request: Καταγράφονται ν-1 στο WAN2

Συνεπώς, εφόσον έχουμε συνολικά 64 requests, ψάχνουμε το άθροισμα: $0 + 1 + 2 + 3 + \dots + 63 = 63 \cdot 64 / 2 = 2016$, όσα και τα πακέτα που καταγράψαμε.

4.32

32 πακέτα → Επειδή η καταγραφή γίνεται στο R3 καταγράφονται μόνο τα πακέτα που είχαν ttl = 0 όταν έφτασαν στον R3 τα οποία θα είναι τα μισά από αυτά που παράχθηκαν και πιο συγκεκριμένα αυτά που είχαν ζυγό ttl όταν παράχθηκαν από το PC1.

Άσκηση 5: Χωρισμός σε υποδίκτυα

WAN1

IP Address: 172.17.17.129
Network Address: 172.17.17.128
Usable Host IP Range: 172.17.17.129 - 172.17.17.130
Broadcast Address: 172.17.17.131
Total Number of Hosts: 4
Number of Usable Hosts: 2

WAN2

IP Address: 172.17.17.133
Network Address: 172.17.17.132
Usable Host IP Range: 172.17.17.133 - 172.17.17.134
Broadcast Address: 172.17.17.135
Total Number of Hosts: 4
Number of Usable Hosts: 2

WAN3

IP Address: 172.17.17.137
Network Address: 172.17.17.136
Usable Host IP Range: 172.17.17.137 - 172.17.17.138
Broadcast Address: 172.17.17.139
Total Number of Hosts: 4
Number of Usable Hosts: 2

5.1

Από το μπλοκ 172.17.17.0/24 μπορούμε να φτιάξουμε 2 υποδίκτυα που να χωράνε 126 hosts, τα 172.17.17.0/25 και 172.17.17.128/25. Ωστόσο, παρατηρούμε πως το υποδίκτυο 172.17.17.128/30 είναι δεσμευμένο από τα WANs. Επομένως, αναθέτουμε στο LAN1 το 172.17.17.0/25 προκειμένου να αποφύγουμε ενδεχόμενες μελλοντικές συγκρούσεις στα ταιριάσματα μήκους. Μας απομένει το 172.17.17.128/25.

LAN1 : 172.17.17.0/25 | $32-25 = 7 \text{ bits} \Rightarrow 2^7 = 128 \text{ hosts}$

IP Address: 172.17.17.0
Network Address: 172.17.17.0
Usable Host IP Range: 172.17.17.1 - 172.17.17.126
Broadcast Address: 172.17.17.127
Total Number of Hosts: 128
Number of Usable Hosts: 126
Subnet Mask: 255.255.255.128

5.2

Από το προηγούμενο μπλοκ μας απομένει το 172.17.17.128/25, το οποίο διαθέτει 2 υποδίκτυα χωρητικότητας 62 κόμβων έκαστο, το 172.17.17.192/26 και το 172.17.17.128/26. Αναθέτουμε στο LAN2 το 172.17.17.192/26.

LAN2 : 172.17.17.192/26 | $32-26 = 6 \text{ bits} \Rightarrow 2^6 = 64 \text{ hosts}$

Αναλυτικά:

IP Address: 172.17.17.192
Network Address: 172.17.17.192
Usable Host IP Range: 172.17.17.193 - 172.17.17.254
Broadcast Address: 172.17.17.255
Total Number of Hosts: 64
Number of Usable Hosts: 62
Subnet Mask: 255.255.255.192

5.3

Αντίστοιχα, το 172.17.17.128/26 που μας έμεινε μπορεί να σπάσει σε 172.17.17.128/27 και 172.17.17.160/27 με 30 υπολογιστές έκαστο. Αναθέτουμε το 172.17.17.160/27 στο LAN3.

LAN3 : 172.17.17.160/27 | $32-27 = 5 \text{ bits} \Rightarrow 2^5 = 32 \text{ hosts}$

IP Address: 172.17.17.160
Network Address: 172.17.17.160
Usable Host IP Range: 172.17.17.161 - 172.17.17.190
Broadcast Address: 172.17.17.191
Total Number of Hosts: 32
Number of Usable Hosts: 30

5.4

ifconfig em1 172.17.17.129/30 --> R1

ifconfig em2 172.17.17.133/30 --> R1

5.5

ifconfig em0 172.17.17.126/25 --> R1

ifconfig em0 172.17.17.1/25 --> PC1

5.6

ifconfig em1 172.17.17.130/30 --> R2

ifconfig em2 172.17.17.138/30 --> R2

5.7

ifconfig em1 172.17.17.193/26 --> R2

ifconfig em0 172.17.17.253/26 --> PC2

ifconfig em0 172.17.17.254/26 --> PC3

5.8

ifconfig em1 172.17.17.134/30 --> R3

ifconfig em2 172.17.17.137/30 --> R3

5.9

ifconfig em0 172.17.17.161/27 --> PC4

ifconfig em0 172.17.17.190/27 --> R3.

5.10

PC1 → route add default 172.17.17.126

PC2 → route add default 172.17.17.193

PC3 → route add default 172.17.17.193

PC4 → route add default 172.17.17.190

5.11

R1

route add -net 172.17.17.192/26 172.17.17.130

route add -net 172.17.17.160/27 172.17.17.130

5.12

R2

```
route add -net 172.17.17.0/25 172.17.17.137
```

```
route add -net 172.17.17.160/27 172.17.17.137
```

5.13

R3

```
route add -net 172.17.17.0/25 172.17.17.133
```

```
route add -net 172.17.17.192/26 172.17.17.133
```

5.14

Εκτελούμε τα ζητούμενα ping και όλα επιτυγχάνουν.

Άσκηση 6: Ταυτόσημες διευθύνσεις IP

6.1

MAC_{PC2} = 08:00:27:8A:BF:6C

MAC_{PC3} = 08:00:27:56:13:3A

6.2

```
ifconfig em0 172.17.17.254/26 --> PC2
```

6.3

Ναι.

```
root@PC:~ # ifconfig em0 172.17.17.254/26
root@PC:~ # Mar 13 12:32:23 PC kernel: arp: 08:00:27:56:13:3a is using my IP add
ress 172.17.17.254 on em0!
```

6.4

Ναι, αντίστοιχο μήνυμα περί χρήσης της IP του από το PC2

```
root@PC:~ # Mar 13 12:29:52 PC kernel: arp: 08:00:27:8a:bf:6c is using my IP add
ress 172.17.17.254 on em0!
```

6.5

Ναι, έχει ορισθεί. Το νόημα είναι για την ειδοποίηση και την αντιμετώπιση του προβλήματος. Δεν απαγορεύεται να έχουν δύο συσκευές την ίδια IP, απλά αυτό είναι πιθανό να προκαλέσει προβλήματα

6.6

Όχι, έχει διαγραφεί λόγω της αλλαγής της IP.

6.7

`route add default 172.17.17.193 --> PC2`

6.8

`arp -ad --> PC2, PC3, R2`

6.9

`tcpdump -i em0 -n arp --> R2`

6.10

`tcpdump -n tcp --> PC2, PC3`

6.11

`ssh lab@172.17.17.254 --> PC1`

```
root@PC:~ # ssh lab@172.17.17.254
Fssh_kex_exchange_identification: read: Connection reset by peer
Connection reset by 172.17.17.254 port 22
```

6.12

Ναι.

6.13

Δεν υπάρχει διαθέσιμη εγγραφή για το PC2, καταγράφεται μόνο το PC3 στη διεύθυνση 172.17.17.254 .

```
root@PC:~ # arp -a
? (172.17.17.137) at 08:00:27:28:a0:82 on em2 expires in 943 seconds [ethernet]
? (172.17.17.138) at 08:00:27:c2:9e:f2 on em2 permanent [ethernet]
? (172.17.17.129) at 08:00:27:44:58:03 on em1 expires in 943 seconds [ethernet]
? (172.17.17.130) at 08:00:27:fe:e8:f3 on em1 permanent [ethernet]
? (172.17.17.193) at 08:00:27:10:c9:86 on em0 permanent [ethernet]
? (172.17.17.254) at 08:00:27:56:13:3a on em0 expires in 943 seconds [ethernet]
root@PC:~ #
```

6.14

Παρατηρούμε πως απάντησε πρώτα το PC2.

```
root@PC:~ # tcpdump -i em0 -n arp
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on em0, link-type EN10MB (Ethernet), capture size 262144 bytes
00:18:10.778410 ARP, Request who-has 172.17.17.254 tell 172.17.17.193, length 28
00:18:10.778917 ARP, Reply 172.17.17.254 is-at 08:00:27:8a:bf:6c, length 46
00:18:10.778971 ARP, Reply 172.17.17.254 is-at 08:00:27:56:13:3a, length 46
^C
3 packets captured
15 packets received by filter
0 packets dropped by kernel
```

6.15

Στο PC3.

6.16

Στο PC3 (κάνουμε ifconfig από την κονσόλα του ssh και βλέπουμε την MAC).

6.17

who --> PC3

```
root@PC:~ # who
root          ttyv0          Mar 12 21:25
lab           pts/0          Mar 13 13:06 (172.17.17.1)
root@PC:~ #
```

netstat -a --> PC3

```
root@PC:~ # netstat -a
Active Internet connections (including servers)
Proto Recv-Q Send-Q Local Address           Foreign Address          (state)
tcp4    0      0 172.17.17.254.ssh       172.17.17.1.63991       ESTABLISHED
tcp4    0      0 *.ssh                   *.*                       LISTEN
tcp6    0      0 *.ssh                   *.*                       LISTEN
tcp4    0      0 localhost.smtp          *.*                       LISTEN
udp4    0      0 *.syslog                *.*                       *
udp6    0      0 *.syslog                *.*                       *
```

Active UNIX domain sockets									
Address	Type	Recv-Q	Send-Q	Inode	Conn	Refs	Nextref	Addr	
16631c00	stream	0	0	0	16631b40	0	0		
16631b40	stream	0	0	0	16631c00	0	0		
165199c0	stream	0	0	0	0	0	0		

6.18

Το PC2 απάντησε πρώτο στο ARP request και έτσι πήρε το πρώτο πακέτο της τριπλής χειραψίας (SYN), μετά το PC2 έστειλε στο PC1 το δεύτερο πακέτο της χειραψίας (SYN, ACK) και το PC1 έστειλε στην διεύθυνση 172.17.17.254 το τρίτο πακέτο της χειραψίας (ACK) όμως μόλις αυτό το τρίτο πακέτο έφτασε στον R2, η εγγραφή του πίνακα ARP είχε αλλάξει και πλέον το τρίτο πακέτο της χειραψίας έφτασε στο PC3 το οποίο δεν γνώριζε τίποτα για τη σύνδεση αυτή. Έτσι στέλνονται RST πακέτα και η σύνδεση απολύεται. Στη δεύτερη προσπάθεια σύνδεσης ο R1 έχει καταγράψει τη διεύθυνση του PC2 στον πίνακα arp του οπότε όλα τα τεμάχια της τριπλής χειραψίας φτάνουν σε αυτόν και η σύνδεση είναι επιτυχής.

```

PC2_ex5 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
root@PC2:~ # tcpdump -i em0 -n tcp
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on em0, link-type EN10MB (Ethernet), capture size 262144 bytes
2:54:35.271872 IP 172.17.17.1.47285 > 172.17.17.254.22: Flags [S], seq 29114000
9, win 65535, options [mss 1460,nop,wscale 6,sack0K,TS val 3856989132 ecr 0], l
ngth 0
2:54:35.271923 IP 172.17.17.254.22 > 172.17.17.1.47285: Flags [S.], seq 3387455
95, ack 2911400020, win 65535, options [mss 1460,nop,wscale 6,sack0K,TS val 399
315050 ecr 3856989132], length 0
2:54:36.301759 IP 172.17.17.254.22 > 172.17.17.1.47285: Flags [S.], seq 3387455
95, ack 2911400020, win 65535, options [mss 1460,nop,wscale 6,sack0K,TS val 399
316000 ecr 3856989132], length 0
2:54:38.516767 IP 172.17.17.254.22 > 172.17.17.1.47285: Flags [S.], seq 3387455
95, ack 2911400020, win 65535, options [mss 1460,nop,wscale 6,sack0K,TS val 399
318295 ecr 3856989132], length 0
2:54:42.729131 IP 172.17.17.254.22 > 172.17.17.1.47285: Flags [S.], seq 3387455
95, ack 2911400020, win 65535, options [mss 1460,nop,wscale 6,sack0K,TS val 399
322508 ecr 3856989132], length 0
C
  packets captured
  packets received by filter
  packets dropped by kernel
root@PC2:~ # who
root
ttyv0      Mar 12 21:03
root@PC2:~ #

PC3_ex5 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Mar 13 12:32:59 PC kernel: arp: 08:00:27:8a:bf:6c is using my IP address 172.17.
17.254 on em0!
root@PC3:~ # arp -ad
172.17.17.193 (172.17.17.193) deleted
root@PC3:~ # tcpdump -i em0 -n tcp
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on em0, link-type EN10MB (Ethernet), capture size 262144 bytes
12:55:11.708798 IP 172.17.17.1.47285 > 172.17.17.254.22: Flags [S.], seq 33874552
96, win 1027, options [nop,nop,TS val 3856989132 ecr 3997315050], length 0
12:55:11.708954 IP 172.17.17.254.22 > 172.17.17.1.47285: Flags [R], seq 33874552
96, win 0, length 0
12:55:11.709906 IP 172.17.17.1.47285 > 172.17.17.254.22: Flags [P.], seq 0:38, a
ck 1, win 1027, options [nop,nop,TS val 3856989132 ecr 3997315050], length 38
12:55:11.709954 IP 172.17.17.254.22 > 172.17.17.1.47285: Flags [R], seq 33874552
96, win 0, length 0
12:55:12.737831 IP 172.17.17.1.47285 > 172.17.17.254.22: Flags [R], seq 29114000
20, win 0, length 0
12:55:14.952562 IP 172.17.17.1.47285 > 172.17.17.254.22: Flags [R], seq 29114000
20, win 0, length 0
12:55:19.165186 IP 172.17.17.1.47285 > 172.17.17.254.22: Flags [R], seq 29114000
20, win 0, length 0
^C
  7 packets captured
  9 packets received by filter
  
```

6.19

Το πρώτο τεμάχιο της τριπλής χειραψίας φτάνει στον PC2 καθώς αυτός απαντάει πρώτος στο arp request και απαντάει με tcp ACK. Το τρίτο τεμάχιο της τριπλής χειραψίας όμως φτάνει στον PC3 ο οποίος όμως αφού δεν έχει λάβει τα προηγούμενα απαντάει με RST και το PC1 απαντάει και αυτός με RST και η προσπάθεια σύνδεσης αποτυγχάνει. Συγχρόνως το αποκομμένο από τον arp table PC2 κάνει 3 επιπλέον απόπειρες να στείλει [SYN,ACK] και να εδραιώσει την tcp σύνδεση.