

Εργαστηριακή Άσκηση 11

Το πρωτόκολλο IPv6

Άσκηση 1

1.1 Εκτελούμε την παρακάτω εντολή στα PC

```
root@R0:~ # sysrc ifconfig_em0_ipv6="inet6 accept_rtadv"  
ifconfig_em0_ipv6: -> inet6 accept_rtadv
```

1.2 Εκτελούμε "service netif restart".

1.3 Έχει αποδοθεί η παρακάτω:

```
inet6 fe80::a00:27ff:fe71:2acf%em0
```

1.4 Έχει αποδοθεί η παρακάτω:

```
inet6 fe80::a00:27ff:fe3c:8ac5%em0
```

1.5 Οι παραπάνω διευθύνσεις είναι link-local και έχουν παραχθεί έχοντας ως πρώτα 64 bit το fe80::/64 και ως τελευταία 64 την EUI-64 MAC address, ο καθένας τη δική του. Για παράδειγμα, όσον αφορά το PC1 παράχθηκε ως εξής:

- Πρώτα 64: fe80::/64
- Τελευταία 64: Από την MAC address 08:00:27:71:2a:cf, αντιστρέφουμε το 7ο bit του πρώτου byte ($08_{16} = 00001000_2$), οπότε και γίνεται 0a:00:27:71:2a:cf και στη συνέχεια παρεμβάλλουμε τα ff:fe στη μέση της MAC, οπότε γίνεται 0a:00:27:ff:fe:71:2a:cf. Συνενώνοντας τα παραπάνω παίρνουμε fe80:0000:0000:0a00:27ff:fe71:2acf, τα οποία απλοποιούνται στο fe80::a00:27ff:fe71:2acf.

Βλέπουμε παρακάτω πληροφορίες για το PC1:

```
root@PC1:~ # ifconfig em0  
em0: flags=8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> metric 0 mtu 1500  
options=9b<RXCSUM,TXCSUM,VLAN_MTU,VLAN_HWTAGGING,VLAN_HWCSUM>  
ether 08:00:27:71:2a:cf  
hwaddr 08:00:27:71:2a:cf  
inet6 fe80::a00:27ff:fe71:2acf%em0 prefixlen 64 scopeid 0x1  
nd6 options=23<PERFORMNUD,ACCEPT_RTADV,AUTO_LINKLOCAL>  
media: Ethernet autoselect (1000baseT <full-duplex>)  
status: active
```

1.6 Εκτελούμε στο PC1 "netstat -rn" και βλέπουμε τις παρακάτω 9 εγγραφές.

Internet6:			
Destination	Gateway	Flags	Netif
Expire			
::/96	:::1	UGRS	lo0
:::1	link#3	UH	lo0
::ffff:0.0.0.0/96	:::1	UGRS	lo0
fe80::/10	:::1	UGRS	lo0
fe80::%em0/64	link#1	U	em0
fe80::a00:27ff:fe71:2acf%em0	link#1	UHS	lo0
fe80::%lo0/64	link#3	U	lo0
fe80::1%lo0	link#3	UHS	lo0
ff02::/16	:::1	UGRS	lo0

1.7 Μία μόνο

1.8 Τις “fe80::%em0/64” και “fe80::%lo0/64” με διεπαφές εξόδου em0 και lo0 αντίστοιχα

1.9 Το ίδιο το PC1

1.10 Πρέπει να προστεθεί το %em0:

```
root@PC1:~ # ping6 fe80::a00:27ff:fe71:2acf
ping6: UDP connect: Network is unreachable
root@PC1:~ # ping6 fe80::a00:27ff:fe71:2acf%em0
PING6(56=40+8+8 bytes) fe80::a00:27ff:fe71:2acf%em0 --> fe80::a00:27ff:fe71:2acf%em0
16 bytes from fe80::a00:27ff:fe71:2acf%em0, icmp_seq=0 hlim=64 time=0.061 ms
16 bytes from fe80::a00:27ff:fe71:2acf%em0, icmp_seq=1 hlim=64 time=0.072 ms
16 bytes from fe80::a00:27ff:fe71:2acf%em0, icmp_seq=2 hlim=64 time=0.052 ms
16 bytes from fe80::a00:27ff:fe71:2acf%em0, icmp_seq=3 hlim=64 time=0.048 ms
^C
--- fe80::a00:27ff:fe71:2acf%em0 ping6 statistics ---
4 packets transmitted, 4 packets received, 0.0% packet loss
round-trip min/avg/max/std-dev = 0.048/0.058/0.072/0.009 ms
```

1.11 Απαιτείται το %em0 στο τέλος.

1.12 Απαντάει το PC1 κάνοντας το Ping από το PC1.

1.13 Παρατηρούμε ότι απαντάνε και τα 2 PC, αφού η διεύθυνση ff02::1 αφορά όλους τους κόμβους στην τοπική ζεύξη:

```
root@PC1:~ # ping6 -I em0 ff02::1
PING6(56=40+8+8 bytes) fe80::a00:27ff:fe71:2acf%em0 --> ff02::1
16 bytes from fe80::a00:27ff:fe71:2acf%em0, icmp_seq=0 hlim=64 time=0.447 ms
16 bytes from fe80::a00:27ff:fe3c:8ac5%em0, icmp_seq=0 hlim=64 time=1.038 ms(DUP !)
```

1.14

```
root@PC1:~ # ifconfig em0 inet6 fd00:1::2/64
```

1.15

```
root@PC2:~ # ifconfig em0 inet6 fd00:1::3/64
```

1.16 Είναι διευθύνσεις μη δρομολογούμενες στο δημόσιο διαδίκτυο, αντίστοιχες με τις ιδιωτικές διευθύνσεις στο IPv4, 10.0.0.0/8, 172.16.0.0/12 και 192.168.0.0/16.

1.17 Υπάρχουν πλέον από 2 σε κάθε PC (ifconfig em0):

```
inet6 fe80::a00:27ff:fe71:2acf%em0 prefixlen 64 scopeid 0x1
inet6 fd00:1::2 prefixlen 64
```

```
inet6 fe80::a00:27ff:fe3c:8ac5%em0 prefixlen 64 scopeid 0x1
inet6 fd00:1::3 prefixlen 64
```

1.18 Βλέπουμε πως υπάρχουν πλέον 11 εγγραφές:

```
root@PC1:~ # netstat -rn6
Routing tables

Internet6:
Destination          Gateway              Flags      Netif
Expire
:::/96                ::1                  UGRS       lo0
::1                   link#3              UH         lo0
::ffff:0.0.0.0/96     ::1                  UGRS       lo0
fd00:1::/64           link#1              U          em0
fd00:1::2             link#1              UHS        lo0
fe80::/10             ::1                  UGRS       lo0
fe80::%em0/64         link#1              U          em0
fe80::a00:27ff:fe71:2acf%em0 link#1              UHS        lo0
fe80::%lo0/64         link#3              U          lo0
fe80::1%lo0           link#3              UHS        lo0
ff02::/16             ::1                  UGRS       lo0
```

1.19 Θα τροποποιήσουμε το αρχείο /etc/hosts σε κάθε PC προσθέτοντας τις παρακάτω γραμμές στα PC1 και PC2 αντίστοιχα:

```
fe80::a00:27ff:fe3c:8ac5%em0      PC2      PC2
```

```
fe80::a00:27ff:fe71:2acf%em0      PC1      PC1
```

1.20 Ναι κανονικά:

```
root@PC1:~ # ping6 PC2
PING6(56=40+8+8 bytes) fe80::a00:27ff:fe71:2acf%em0 --> fe80::a00:27ff:fe3c:8ac5%em0
16 bytes from fe80::a00:27ff:fe3c:8ac5%em0, icmp_seq=0 hlim=64 time=0.990 ms
16 bytes from fe80::a00:27ff:fe3c:8ac5%em0, icmp_seq=1 hlim=64 time=1.482 ms
16 bytes from fe80::a00:27ff:fe3c:8ac5%em0, icmp_seq=2 hlim=64 time=1.219 ms
^C
--- PC2 ping6 statistics ---
3 packets transmitted, 3 packets received, 0.0% packet loss
round-trip min/avg/max/std-dev = 0.990/1.230/1.482/0.201 ms
```

1.21 Δε βλέπουμε καμία εγγραφή.

1.22 man ndp

1.23 Εκτελούμε “ndp -a”:

```
root@PC:~ # ndp -a
Neighbor              Linklayer Address  Netif Expire      1s 5s
fd00:1::2             08:00:27:a9:25:0c  em0 permanent R
fe80::a00:27ff:fe71:2acf%em0 08:00:27:e1:7e:0b  em0 23h57m55s S
fe80::a00:27ff:fe3c:8ac5%em0 08:00:27:a9:25:0c  em0 permanent R
```

1.24 Βλέπουμε 3 εγγραφές και βρίσκονται σε καταστάσεις R(Reachable) και S (Stale)

1.25 `ndp -p`. Δεν υπάρχουν εγγραφές

1.26 `ndp -c`

1.27 Εκτελούμε `tcpdump -vvvni em0`.

1.28 Βλέπουμε 6 IPv6 πακέτα, ένα Neighbor Solicitation με το οποίο ο PC1 ρωτάει για τη MAC του PC2, ένα Neighbor Solicitation με το οποίο ο PC2 ρωτάει για τη MAC του PC1, δύο Neighbor Advertisement που περιέχει την απάντηση στα παραπάνω και στη συνέχεια το ICMPv6 Request και Reply:

1.29 Μεταφέρονται μηνύματα του πρωτοκόλλου ICMPv6, τα οποία προσδιορίζονται από την τιμή 58 του πεδίου next-header.

1.30

- 1) PC1 → PC2 (Neighbor Solicitation)
- 2) PC2 → PC1 (Neighbor Advertisement)
- 3) PC1 → PC2 (ICMPv6 Echo Request)
- 4) PC2 → PC1 (ICMPv6 Echo Reply)
- 5) PC2 → PC1 (Neighbor Solicitation)
- 6) PC1 → PC2 (Neighbor Advertisement)

1.31 Το πρώτο πακέτο που καταγράψαμε έχει διεύθυνση προορισμού την ff02::1:ff3c:8ac5. Η διεύθυνση αυτή προκύπτει προσθέτοντας στο πρόθεμα ff02:0:0:0:0:1:ff00:0/104 τα τελευταία 24 bit της unicast διεύθυνσης του προορισμού. Εν προκειμένω, προορισμός ήταν η διεύθυνση του PC2 (fe80::a00:27ff:fe3c:8ac5), εκ της οποίας τελευταία 24 bits είναι τα 3c:8ac5, οπότε και προκύπτει η παραπάνω διεύθυνση ως **multicast Solicited Node**.

1.32 Αντίστοιχα για το 5ο πακέτο.

1.33 Η εγγραφή που αφορά το PC1 είναι σε κατάσταση Stale ενώ η διάρκεια ζωής της είναι 24 ώρες (στην αρχή)

1.34 Παρατηρήσαμε τις καταστάσεις R (Reachable) και S (Stale)

1.35 Η κατάσταση R διαρκεί 23sec, μετά την παρέλευση των οποίων μεταπίπτει στην κατάσταση S για λίγο, προτού ξαναμπει στην κατάσταση R.

1.36 Η διάρκεια ζωής της κατάστασης S είναι 24ώρες.

1.37 Παρατηρούμε τις ίδιες καταστάσεις, με τη διαφορά πως τώρα όταν λήξουν τα 23s της κατάστασης R, μεταβαίνουμε στην κατάσταση S όπου και ο χρόνος βαίνει μειούμενος από τις 24 ώρες χωρίς να ξαναπάμε στην κατάσταση R.

1.38 Παρατηρούμε επιπλέον πακέτα NS και NA, τα οποία επαναλαμβάνονται ανά 23s, όποτε δηλαδή λήγει η κατάσταση R.

Άσκηση 2

2.1

```
root@R1:~ # sysrc ipv6_gateway_enable="YES"
ipv6_gateway_enable: NO -> YES
root@R1:~ # service routing restart
```

```
root@R2:~ # sysrc ipv6_gateway_enable="YES"
ipv6_gateway_enable: NO -> YES
root@R2:~ # service routing restart
```

2.2

```
root@PC2:~ # ifconfig em0 inet6 fd00:1::3/64 delete
root@PC2:~ # ifconfig em0 inet6 fd00:2::2/64
```

2.3

```
root@R1:~ # vtysh

Hello, this is FRRouting (version 7.5.1).
Copyright 1996-2005 Kunihiro Ishiguro, et al.

R1# configure terminal
R1(config)# interface em0
R1(config-if)# ipv6 address fd00:1::1/64
```

2.4

```
R1(config)# interface em1
R1(config-if)# ipv6 address fd00:3::1/126
```

2.5

```
R2(config)# interface em1
R2(config-if)# ipv6 address fd00:2::1/64
```

2.6

```
R2(config)# interface em0
R2(config-if)# ipv6 address fd00:3::2/126
```

2.7

```
root@PC1:~ # route -6 add default fd00:1::1
add net default: gateway fd00:1::1
```

2.8

```
root@PC2:~ # route -6 add default fd00:2::1  
add net default: gateway fd00:2::1
```

2.9 tcpdump -i em0

2.10 Το ping αποτυγχάνει, αφού το PC1 κάνει στην αρχή NS, ώστε να μάθει που είναι το PC2, χωρίς ωστόσο να παίρνει απάντηση αφού ούτε ο R1 γνωρίζει:

```
root@R1:~ # 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  
23:30:56.035649 IP6 fe80::a00:27ff:fe71:2acf > ff02::1:ff3c:8ac5: ICMP6, neighbo  
r solicitation, who has fe80::a00:27ff:fe3c:8ac5, length 32  
23:30:57.030852 IP6 fe80::a00:27ff:fe71:2acf > ff02::1:ff3c:8ac5: ICMP6, neighbo  
r solicitation, who has fe80::a00:27ff:fe3c:8ac5, length 32  
23:30:58.032026 IP6 fe80::a00:27ff:fe71:2acf > ff02::1:ff3c:8ac5: ICMP6, neighbo  
r solicitation, who has fe80::a00:27ff:fe3c:8ac5, length 32
```

2.11 Όπως βλέπουμε, παράχθηκαν 3 NS με προορισμό την ff02::1:ff3c:8ac5, δηλαδή την Solicited Node του PC2.

2.12 Στον R1: `ipv6 route fd00:2::/64 fd00:3::2`

2.13 Ακόμα δε μπορούμε να κάνουμε ping, καθώς δε μπορεί να δρομολογήσει ο R2 την απάντηση προς το PC1.

2.14 Στον R2: `ipv6 route fd00:2::/64 fd00:3::2`

2.15 Πλέον το ping επιτυγχάνει, όσον αφορά τη στατική διεύθυνση fd00:2::2 που ορίσαμε στον PC2 (ping fd00:2::2).

2.16 Στο R1 σε GCM “interface em0” → “no ipv6 nd suppress-ra”

2.17

```
R1(config-if)# ipv6 nd prefix fd00:1::/64
```

2.18

```
R2(config)# interface em1  
R2(config-if)# no ipv6 nd suppress-ra
```

2.19

```
R2(config-if)# ipv6 nd prefix fd00:2::/64
```

2.20

```
root@PC1:~ # route -6 delete default fd00:1::1  
delete net default: gateway fd00:1::1
```

2.21 "tcpdump -eni em0"

2.22 "service netif restart"

2.23 Καταγράφουμε τα παρακάτω πακέτα:

```
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  
17:41:45.107375 IP6 fe80::a00:27ff:fea9:250c > ff02::2: ICMP6, router solicitation, length 16  
17:41:45.108107 IP6 fe80::a00:27ff:febb:b187 > ff02::1: ICMP6, router advertisement, length 56  
17:41:45.679863 IP6 :: > ff02::1:ffa9:250c: ICMP6, neighbor solicitation, who has fd00:1::a00:27ff:fea9:250c, length 32
```

Έχουμε τα παρακάτω πακέτα:

- 1) Το PC1 στέλνει RS, ώστε να ζητήσει να λαμβάνει μηνύματα RA.
- 2) Ο R1 απαντάει με RA.
- 3) Το PC1 στέλνει NS.

2.24 Το PC1 παράγει NS καθώς ψάχνει να βρεί αν κάποιο άλλο μηχάνημα χρησιμοποιεί την ίδια διεύθυνση IPv6 link-local

2.25 Χρησιμοποιεί την :: γιατί ακόμα δεν έχει ορισθεί η διεύθυνσή του.

2.26 Τη διεύθυνση link-local που αναζητεί στα μηνύματα NS

2.27 RS -> ff02::1:, RA -> ff02::1:, NS -> ff02::1:ffa9:250c (οι κατάλληλες διευθύνσεις multicast)

2.28 33:33:ff:a9:25:0c-> NS
33:33:00:00:00:01 -> RA
33:33:00:00:00:02 -> RS

2.29 Παρατηρούμε πως πλέον η πρώτη εγγραφή έχει επιπρόσθετα το flag A.

2.30

```
inet6 fe80::a00:27ff:fea9:250c%em0 prefixlen 64 scopeid 0x1  
inet6 fd00:1::a00:27ff:fea9:250c prefixlen 64 autoconf
```

2.31 Βλέπουμε πως υπάρχει η προκαθορισμένη διαδρομή και προέκυψε ως η link-state διεύθυνση του δρομολογητή με τον οποίο επικοινωνήσε πριν το PC1 (δηλαδή η em0 του R1):

```

root@PC1:~ # netstat -r -6
Routing tables

Internet6:
Destination      Gateway          Flags      Netif Expire
::/96             localhost       URS        lo0
localhost        link#2          UHS        lo0
::ffff:0.0.0.0/96 localhost       URS        lo0
fd00:1::/64      link#1          U          em0
fd00:1::a00:27ff:f link#1          UHS        lo0
fe80::/10        localhost       URS        lo0
fe80::%em0/64    link#1          U          em0
fe80::a00:27ff:fea link#1          UHS        lo0
fe80::%lo0/64    link#2          U          lo0
fe80::1%lo0      link#2          UHS        lo0
ff02::/16        localhost       URS        lo0

```

2.32 Από το PC2 μπορούμε να κάνουμε ping μόνο στην 2η διεύθυνση του PC1, ενώ από το R1 και στις 2, καθώς είναι στο ίδιο LAN, οπότε μπορεί να επικοινωνήσει και με τη link-local address.

Άσκηση 3

3.1

```

R1(config)# no ipv6 route fd00:2::/64 fd00:3::2
R2(config)# no ipv6 route fd00:1::/64 fd00:3::1

```

3.2

```

R1(config)# router ripng
R1(config-router)# network em0
R1(config-router)# network em1
R2(config)# router ripng
R2(config-router)# network em0
R2(config-router)# network em1

```

3.3 Βλέπουμε μία εγγραφή:

```

R1(config-router)# do show ipv6 route ripng
Codes: K - kernel route, C - connected, S - static, R - RIPng,
       O - OSPFv3, I - IS-IS, B - BGP, N - MHRP, T - Table,
       v - UNC, U - UNC-Direct, A - Babel, D - SHARP, F - PBR,
       f - OpenFabric,
       > - selected route, * - FIB route, q - queued, r - rejected, b - backup
R>* fd00:2::/64 [120/2] via fe80::a00:27ff:fe38:4bd, em1, weight 1, 00:00:29

```


3.4 Επόμενος κόμβος για το fd00:2::/64 είναι η fe80::a00:27ff:fe38:4bd, δηλαδή η link-local address του em0 του R2.

3.5 Μπορούμε χρησιμοποιώντας την private address “ping6 fd00:2::2”.

3.6

```
root@R1:~# tcpdump -vvvni em1 ip6
tcpdump: listening on em1, link-type EN10MB (Ethernet), capture size 262144 bytes
13:11:49.923244 IP6 (class 0xc0, hlim 255, next-header UDP (17) payload length: 52) fe80::a00:27ff:fe4d:fa0a.521 > ff02::9.521: [udp sum ok] ripng-resp 2:
    fd00:1::/64 (1)
    fd00:3::/64 (1)
13:12:08.932326 IP6 (class 0xc0, hlim 255, next-header UDP (17) payload length: 52) fe80::a00:27ff:fe4d:fa0a.521 > ff02::9.521: [udp sum ok] ripng-resp 2:
    fd00:1::/64 (1)
    fd00:3::/64 (1)
13:12:14.300003 IP6 (class 0xc0, hlim 255, next-header UDP (17) payload length: 52) fe80::a00:27ff:fe38:4bd.521 > ff02::9.521: [udp sum ok] ripng-resp 2:
    fd00:2::/64 (1)
    fd00:3::/64 (1)
13:12:40.937586 IP6 (class 0xc0, hlim 255, next-header UDP (17) payload length: 52) fe80::a00:27ff:fe4d:fa0a.521 > ff02::9.521: [udp sum ok] ripng-resp 2:
    fd00:1::/64 (1)
    fd00:3::/64 (1)
13:12:44.326072 IP6 (class 0xc0, hlim 255, next-header UDP (17) payload length: 52) fe80::a00:27ff:fe38:4bd.521 > ff02::9.521: [udp sum ok] ripng-resp 2:
    fd00:2::/64 (1)
    fd00:3::/64 (1)
```

3.7 Παρατηρούμε πακέτα ripng-response εκπεμπόμενα από τις WAN διεπαφές των R1, R2. Το μεν R1 στέλνει διαφημίσεις για το LAN1 και το WAN1, ενώ το R2 για το LAN2 και το WAN1. Διεύθυνση προορισμού στα παραπάνω είναι η ff02::9, η οποία αποτελεί multicast address για RIP routers.

3.8 Το hop_limit έχει τιμή 255, ώστε να διασφαλιστεί ότι δε θα περάσει από ενδιαμέσο δρομολογητή.

3.9 Χρησιμοποιείται το UDP και η θύρα 521, ενώ στο RIP είχαμε μεν UDP, αλλά χρησιμοποιούνταν η θύρα 520.

3.10 Εκτελούμε στα R1, R2 “no router ripng”.

3.11 Εκτελούμε στα R1, R2 σε GCM “do write memory”.

3.12 Εκτελούμε σε απλό terminal στα R1, R2 “service frr restart”.

3.13 Εκτελούμε τα παρακάτω στον R1 και αντίστοιχα στον R2:

```
R1(config)# router ospf6
R1(config-ospf6)# router-id 1.1.1.1
```

```

R2(config)# router ospf6
Warning: closing connection to ospf6d because of an I/O error!
success!
Warning: connecting to zebra...Warning: closing connection to zebra because of an I/O error!
success!
Warning: connecting to ripd...Warning: closing connection to ripd because of an I/O error!
success!
Warning: connecting to ripngd...Warning: closing connection to ripngd because of an I/O error!
success!
Warning: connecting to ospfd...Warning: closing connection to ospfd because of an I/O error!
success!
Warning: connecting to bgpd...Warning: closing connection to bgpd because of an I/O error!
success!
Warning: connecting to staticd...Warning: closing connection to staticd because of an I/O error!
success!
% [OSPF6] Unknown command: router ospf6

```

3.14

```

R1(config-ospf6)# interface em0 area 0.0.0.0
R1(config-ospf6)# interface em1 area 0.0.0.0

```

3.15 Εκτελούμε αντίστοιχα στον R2 “interface em0 area 0.0.0.0” και “interface em1 area 0.0.0.0”.

3.16 Βλέπουμε τις παρακάτω 3 εγγραφές. Το κόστος για το LAN1 (πρώτη εγγραφή) είναι 200 (100+100), ενώ για τα LAN2 και WAN1 (δεύτερη και τρίτη εγγραφή αντίστοιχα) είναι 100, καθώς το R2 είναι άμεσα συνδεδεμένο με αυτά:

```

R2(config)# do show ipv6 route ospf6
Codes: K - kernel route, C - connected, S - static, R - RIPng,
       O - OSPFv3, I - IS-IS, B - BGP, N - NHRP, T - Table,
       u - UNC, U - UNC-Direct, A - Babel, D - SHARP, F - PBR,
       f - OpenFabric,
       > - selected route, * - FIB route, q - queued, r - rejected, b - backup

O>* fd00:1::/64 [110/200] via fe80::a00:27ff:fe4d:fa0a, em0, weight 1, 00:02:14
O   fd00:2::/64 [110/100] is directly connected, em1, weight 1, 00:02:24
O   fd00:3::/64 [110/100] is directly connected, em0, weight 1, 00:02:19

```

3.17 Είναι η διεύθυνση fe80::a00:27ff:fe4d:fa0a, η οποία αποτελεί την link-local address της em1 του R1.

3.18 Εκτελούμε στο R2 “tcpdump -vvni em0”:

```

root@R2:~ # tcpdump -vvvni em0
tcpdump: listening on em0, link-type EN10MB (Ethernet), capture size 262144 byte
13:50:20.918720 IP6 (class 0xc0, hlim 1, next-header OSPF (89) payload length: 40) fe80::a00:27ff:fe4d:fa0a > ff02::5: OSPFv3, Hello, length 40
    Router-ID 1.1.1.1, Backbone Area
    Options [U6, External, Router]
    Hello Timer 10s, Dead Timer 40s, Interface-ID 0.0.0.2, Priority 1
    Designated Router 1.1.1.1, Backup Designated Router 2.2.2.2
    Neighbor List:
        2.2.2.2
13:50:23.022959 IP6 (class 0xc0, hlim 1, next-header OSPF (89) payload length: 40) fe80::a00:27ff:fe38:4bd > ff02::5: OSPFv3, Hello, length 40
    Router-ID 2.2.2.2, Backbone Area
    Options [U6, External, Router]
    Hello Timer 10s, Dead Timer 40s, Interface-ID 0.0.0.1, Priority 1
    Designated Router 1.1.1.1, Backup Designated Router 2.2.2.2
    Neighbor List:
        1.1.1.1
13:50:30.984182 IP6 (class 0xc0, hlim 1, next-header OSPF (89) payload length: 40) fe80::a00:27ff:fe4d:fa0a > ff02::5: OSPFv3, Hello, length 40
    Router-ID 1.1.1.1, Backbone Area
    Options [U6, External, Router]
    Hello Timer 10s, Dead Timer 40s, Interface-ID 0.0.0.2, Priority 1
    Designated Router 1.1.1.1, Backup Designated Router 2.2.2.2
    Neighbor List:
        2.2.2.2
13:50:33.026085 IP6 (class 0xc0, hlim 1, next-header OSPF (89) payload length: 40) fe80::a00:27ff:fe38:4bd > ff02::5: OSPFv3, Hello, length 40
    Router-ID 2.2.2.2, Backbone Area
    Options [U6, External, Router]
    Hello Timer 10s, Dead Timer 40s, Interface-ID 0.0.0.1, Priority 1
    Designated Router 1.1.1.1, Backup Designated Router 2.2.2.2
    Neighbor List:
        1.1.1.1
13:50:40.988687 IP6 (class 0xc0, hlim 1, next-header OSPF (89) payload length: 40) fe80::a00:27ff:fe4d:fa0a > ff02::5: OSPFv3, Hello, length 40
    Router-ID 1.1.1.1, Backbone Area
    Options [U6, External, Router]
    Hello Timer 10s, Dead Timer 40s, Interface-ID 0.0.0.2, Priority 1
    Designated Router 1.1.1.1, Backup Designated Router 2.2.2.2
    Neighbor List:
        2.2.2.2

```

3.19 Παρατηρούμε να εκπέμπονται OSPFv3 Hello πακέτα από τις διεπαφές em1 και em0 των R1 και R2 αντίστοιχα, με διεύθυνση προορισμού την ff02::5, η οποία αφορά multicast διευθύνσεις OSPFIGP.

3.20 Hop Limit = 1.

3.21 Βλέπουμε πως χρησιμοποιεί τον αριθμό πρωτοκόλλου 89, ίδιος δηλαδή με αυτόν του OSPFv2.

3.22 Μπορούμε να κάνουμε στην private address του PC1.

3.23 Εκτελούμε στα R1 και R2 “no router ospf6”.

3.24 Επανεκκινούμε την υπηρεσία frr στα R1/R2.

3.25 router-id 1.1.1.1, router bgp 65010

3.26 no bgp ebgp-requires-policy

3.27 no bgp default ipv4-unicast

3.28 neighbor fd00:3::2 remote-as 65020

3.29 address-family ipv6

3.30 network fd00:1::/64

3.31 neighbor fd00:3::2 activate

3.32 Εκτελούμε τις εντολές για τον R2:

```
root@R2:~ # vtysh
Hello, this is FRRouting (version 7.5.1).
Copyright 1996-2005 Kunihiro Ishiguro, et al.

R2# configure terminal
R2(config)# router-id 2.2.2.2
R2(config)# router bgp 65020
R2(config-router)# no bgp ebgp-requires-policy
R2(config-router)# no bgp default ipv4-unicast
R2(config-router)# neighbor fd00:3::1 remote-as 65010
R2(config-router)# address-family ipv6
R2(config-router-af)# network fd00:2::/64
R2(config-router-af)# neighbor
  A.B.C.D   Neighbor address
  WORD      Interface name or neighbor tag
  X:X::X:X  Neighbor IPv6 address
  fd00:3::1
R2(config-router-af)# neighbor fd00:3::1 activate
R2(config-router-af)# exit
```

3.33 show bgp ipv6 unicast summary

3.34 Με “do show ipv6 route bgp” βλέπουμε μία μόνο εγγραφή, για το LAN2:

```
R2>show ipv6 route bgp
B>* fd00:2::/64 [20/0] via fe80::a00:27ff:fe38:4bd, em1, weight 1, 00:03:05
```

3.35 Διεύθυνση του επόμενου κόμβου για το LAN2 είναι η fe80::a00:27ff:fe38:4bd, η οποία αποτελεί την link-local address της em0 του R2.

3.37 Εκτελούμε “tcpdump -vvni em1”.

3.38 Βλέπουμε μηνύματα BGP Keepalive μεταξύ των R1-R2. Χρησιμοποιείται το πρωτόκολλο TCP και η θύρα (προορισμού) 179, όπως στο IPv4. (θύρα πηγής δυναμική).

3.39 hlim=1

3.40 Μπορούμε στις παρακάτω 2 διευθύνσεις:

```
inet6 fd00:2::2 prefixlen 64
inet6 fd00:2::a00:27ff:fe1b:e6e6 prefixlen 64 autoconf
```

3.41, 3.42

```
root@PC1:~ # vtysh

Hello, this is FRRouting (version 7.5.1).
Copyright 1996-2005 Kunihiro Ishiguro, et al.

PC1# configure terminal
PC1(config)# interface em0
PC1(config-if)# ip address fd00:1::2/64

PC1(config)# router-id 1.1.0.0
PC1(config)# router bgp 65010
```

3.43

```
PC1(config-router)# no bgp default ipv4-unicast
```

3.44

```
PC1(config-router)# neighbor fd00:1::1 remote-as 65010
```

3.45

```
PC1(config-router)# address-family ipv6
PC1(config-router-af)# neighbor fd00:1::1 activate
```

3.46

```
R1(config-router)# neighbor fd00:1::2 remote-as 65010
```

3.47

```
R1(config-router-af)# neighbor fd00:1::2 activate
R1(config-router-af)# neighbor fd00:1::2 next-hop-self
```

3.48 Με την εντολή “do show ip bgp neighbors fd00:1::2” βλέπουμε πως αναφέρεται στην πρώτη σειρά η πληροφορία “internal link”:

```

BGP neighbor is fd00:1::2, remote AS 65010, local AS 65010, internal link
Hostname: R0
  BGP version 4, remote router ID 1.1.0.0, local router ID 1.1.1.1
  BGP state = Established, up for 00:04:59
  Last read 00:00:59, Last write 00:00:59
  Hold time is 180, keepalive interval is 60 seconds
  Neighbor capabilities:
    4 Byte AS: advertised and received
  AddPath:
    IPv6 Unicast: RX advertised IPv6 Unicast and received
    Route refresh: advertised and received(old & new)
    Address Family IPv6 Unicast: advertised and received
    Hostname Capability: advertised (name: R0, domain name: n/a) received (name:
R0, domain name: n/a)
    Graceful Restart Capability: advertised and received
      Remote Restart timer is 120 seconds
    Address families by peer:
      none
  Graceful restart information:
    End-of-RIB send: IPv6 Unicast
    End-of-RIB received: IPv6 Unicast
    Local GR Mode: Helper*
    Remote GR Mode: Helper
    R bit: False

```

3.49 Βλέπουμε τις παρακάτω 2 εγγραφές, για τα LAN1 και LAN2:

```

PC1(config)# do show ipv6 route bgp
Codes: K - kernel route, C - connected, S - static, R - RIPng,
       O - OSPFv3, I - IS-IS, B - BGP, N - MHRP, T - Table,
       v - UNC, U - UNC-Direct, A - Babel, D - SHARP, F - PBR,
       f - OpenFabric,
       > - selected route, * - FIB route, q - queued, r - rejected, b - backup
B   fd00:1::/64 [200/0] via fe80::a00:27ff:febc:47fc, em0, weight 1, 00:07:41
B>* fd00:2::/64 [200/0] via fd00:1::1, em0, weight 1, 00:06:11

```

3.50 Καθώς όντας άμεσα συνδεδεμένο, επιλέγει αυτή τη διαδρομή για το fd00:1::/64 (LAN1).

3.51 Είναι η fd00:1::1, η οποία είναι η private IPv6 address της em0 του R1.

3.52 Μπορούμε να κάνουμε στις παρακάτω 2 διευθύνσεις:

```

inet6 fd00:1::a00:27ff:fe88:285 prefixlen 64 autoconf
inet6 fd00:1::2 prefixlen 64

```

Άσκηση 4

4.1

```

R1(config)# interface em0
R1(config-if)# ip address 192.168.1.1/24

```

4.2

```
R2(config)# interface em1
R2(config-if)# ip address 192.168.2.1/24
```

4.3

```
PC1(config)# interface em0
PC1(config-if)# ip address 192.168.1.2/24
PC1(config-if)# exit
PC1(config)# ip route 0.0.0.0/0 192.168.1.1
```

4.4

```
root@PC2:~ # vtysh

Hello, this is FRRouting (version 7.5.1).
Copyright 1996-2005 Kunihiro Ishiguro, et al.

PC2# configure terminal
PC2(config)# ip route 0.0.0.0/0 192.168.2.1
PC2(config)# interface em0
PC2(config-if)# ip address 192.168.2.2/24
```

4.5

```
root@R1:~ # sysrc firewall_enable="YES"
firewall_enable: NO -> YES
root@R1:~ # firewall_nat64_enable="YES"
firewall_nat64_enable=YES: Command not found.
root@R1:~ # sysrc firewall_nat64_enable="YES"
firewall_nat64_enable: NO -> YES
root@R1:~ # sysrc firewall_type="open"
firewall_type: UNKNOWN -> open
root@R1:~ # sysrc firewall_logif="YES"
firewall_logif: NO -> YES
```

4.6 Εκκινούμε το ipfw στο R1.

```

root@R1:~ # service ipfw start
/etc/rc.conf: to: not found
/etc/rc.conf: to: not found
/etc/rc.conf: to: not found
/etc/rc.conf: to: not found
Flushed all rules.
00100 allow ip from any to any via lo0
00200 deny ip from any to 127.0.0.0/8
00300 deny ip from 127.0.0.0/8 to any
00400 deny ip from any to ::1
00500 deny ip from ::1 to any
00600 allow ipv6-icmp from :: to ff02::/16
00700 allow ipv6-icmp from fe80::/10 to fe80::/10
00800 allow ipv6-icmp from fe80::/10 to ff02::/16
00900 allow ipv6-icmp from any to any icmp6types 1
01000 allow ipv6-icmp from any to any icmp6types 2,135,136
65000 allow ip from any to any
Firewall rules loaded.
Firewall logging pseudo-interface (ipfw0) created.
/etc/rc.conf: to: not found

```

4.7 Περιέχει τους παρακάτω 12 κανόνες:

```

root@R1:~ # ipfw list
00100 allow ip from any to any via lo0
00200 deny ip from any to 127.0.0.0/8
00300 deny ip from 127.0.0.0/8 to any
00400 deny ip from any to ::1
00500 deny ip from ::1 to any
00600 allow ipv6-icmp from :: to ff02::/16
00700 allow ipv6-icmp from fe80::/10 to fe80::/10
00800 allow ipv6-icmp from fe80::/10 to ff02::/16
00900 allow ipv6-icmp from any to any icmp6types 1
01000 allow ipv6-icmp from any to any icmp6types 2,135,136
65000 allow ip from any to any
65535 deny ip from any to any

```

4.8 Μπορούμε αλλά μόνο στις 2 τελευταίες από τις παρακάτω διευθύνσεις:

```

inet6 fe80::a00:27ff:fe1b:e6e6%em0 prefixlen 64 scopeid 0x1
inet6 fd00:2::2 prefixlen 64
inet6 fd00:2::a00:27ff:fe1b:e6e6 prefixlen 64 autoconf

```

4.9

```

root@R1:~ # ipfw nat64clat nat64 create clat_prefix fd00:3:1::/96 plat_prefix 64
:ff9b::/96 allow_private log

```

4.10

```

root@R1:~ # ipfw add 2000 nat64clat nat64 ip4 from any to not me recv em0
02000 nat64clat nat64 ip4 from any to not me recv em0

```


4.11

```
root@R1:~ # ipfw add 3000 nat64clat nat64 ip6 from 64:ff9b::/96 to fd00:3:1::/96  
recv em1  
03000 nat64clat nat64 ip6 from 64:ff9b::/96 to fd00:3:1::/96 recv em1
```

4.12

```
R1(config)# ipv6 route 64:ff9b::/96 fd00:3::2
```

4.13

```
root@R2:~ # sysrc firewall_enable="YES"  
firewall_enable: NO -> YES  
root@R2:~ # sysrc firewall_nat64_enable="YES"  
firewall_nat64_enable: NO -> YES  
root@R2:~ # sysrc firewall_type="open"  
firewall_type: UNKNOWN -> open  
root@R2:~ # sysrc firewall_logif="YES"  
firewall_logif: NO -> YES
```

```
root@R2:~ # service ipfw start
```

4.14

```
root@R2:~ # ipfw nat64lsn nat64 create prefix6 64:ff9b::/96 prefix4 2.2.2.0/24 allow_private_log
```

4.15

```
root@R2:~ # ipfw add 2000 nat64lsn nat64 ip6 from fd00:3:1::/96 to 64:ff9b::/96  
recv em0  
02000 nat64lsn nat64 ip6 from fd00:3:1::/96 to 64:ff9b::/96 recv em0
```

4.16

```
root@R2:~ # ipfw add 3000 nat64lsn nat64 ip4 from any to 2.2.2.0/24 recv em1  
03000 nat64lsn nat64 ip4 from any to 2.2.2.0/24 recv em1
```

4.17

```
R2(config)# ipv6 route fd00:3:1::/96 fd00:3::1
```

4.18

```
R2(config)# ip route 0.0.0.0/0 192.168.2.2
```

4.19 Πετυχαίνουν:

```

root@PC1:~ # 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=1.422 ms

--- 192.168.1.1 ping statistics ---
1 packets transmitted, 1 packets received, 0.0% packet loss
round-trip min/avg/max/stddev = 1.422/1.422/1.422/0.000 ms
root@PC1:~ # ping -c 1 192.168.2.2
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=4.256 ms

--- 192.168.2.2 ping statistics ---
1 packets transmitted, 1 packets received, 0.0% packet loss
round-trip min/avg/max/stddev = 4.256/4.256/4.256/0.000 ms

```

4.20

```

root@R1:~ # ifconfig ipfwlog0 create
root@R1:~ # tcpdump -i ipfwlog0
tcpdump: WARNING: ipfwlog0: That device doesn't support promiscuous mode
(BIOCPRMISC: Invalid argument)
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on ipfwlog0, link-type PFLOG (OpenBSD pflog file), capture size 262144
bytes

```

4.21

```

root@R2:~ # ifconfig ipfwlog0 create
root@R2:~ # tcpdump -i ipfwlog0
tcpdump: WARNING: ipfwlog0: That device doesn't support promiscuous mode
(BIOCPRMISC: Invalid argument)
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on ipfwlog0, link-type PFLOG (OpenBSD pflog file), capture size 262144
bytes

```

4.22 Παρατηρούμε τα παρακάτω πακέτα στα R1 και R2 αντίστοιχα:

```

18:38:42.966813 IP 192.168.1.2 > 192.168.2.2: ICMP echo request, id 42244, seq 0, length 64
18:38:42.966882 IP6 fd00:3:1::c0a8:102 > 64:ff9b::c0a8:202: ICMP6, echo request, seq 0, length 64
18:38:43.001344 IP6 64:ff9b::c0a8:202 > fd00:3:1::c0a8:102: ICMP6, echo reply, seq 0, length 64
18:38:43.001356 IP 192.168.2.2 > 192.168.1.2: ICMP echo reply, id 42244, seq 0, length 64

```

```

18:38:30.254345 IP6 fd00:3:1::c0a8:102 > 64:ff9b::c0a8:202: ICMP6, echo request, seq 0, length 64
18:38:30.254393 IP 2.2.2.158 > 192.168.2.2: ICMP echo request, id 1024, seq 0, length 64
18:38:30.255862 IP 192.168.2.2 > 2.2.2.158: ICMP echo reply, id 1024, seq 0, length 64
18:38:30.255868 IP6 64:ff9b::c0a8:202 > fd00:3:1::c0a8:102: ICMP6, echo reply, seq 0, length 64

```

Βλέπουμε πως αρχικά το PC1 στέλνει ένα ICMP echo request με προορισμό το PC2, των οποίων οι IPv4 διευθύνσεις πηγής και προορισμού μετατρέπονται σε IPv6 καθώς διέρχονται από το WAN1, μέχρι να μετατραπούν ξανά σε IPv4 στο LAN2. Κατά την αντίστροφη πορεία έχουμε ξανά μετατροπή από IPv4 στο LAN2 σε IPv6 στο WAN1 και ξανά σε IPv4 στο LAN1.

4.23

```
PC2(config)# interface em0
PC2(config-if)# ip address 172.17.17.2/24
PC2(config-if)# ip address 10.0.0.2/24
```

4.24 Ναι

4.25 tcpdump -i em0

4.26

```
root@R2:~ # ipfw nat64lsn nat64 show states
fd00:3:1::c0a8:102      2.2.2.158      ICMPv6      59      172.17.17.2
fd00:3:1::c0a8:102      2.2.2.158      ICMPv6      55      10.0.0.2
root@R2:~ #
```

4.27 Παρατηρούμε πως ο χρόνος αναπαριστά τον χρόνο που απομένει μέχρι να διαγραφεί η εγγραφή από τον πίνακα και ανέρχεται σε περίπου 1 λεπτό.

4.28 Όχι δεν μπορούμε.

4.29 Ναι αυτή τη φορά επιτυγχάνει.

Άσκηση 5

5.1

```
root@PC:~ # dhclient em0
DHCPDISCOVER on em0 to 255.255.255.255 port 67 interval 3
May 31 22:29:27 PC dhclient[763]: send_packet: Network is down
DHCPDISCOVER on em0 to 255.255.255.255 port 67 interval 8
DHCPOFFER from 10.0.2.2
DHCPREQUEST on em0 to 255.255.255.255 port 67
DHCPACK from 10.0.2.2
bound to 10.0.2.15 -- renewal in 43200 seconds.
root@PC:~ # ping www.google.com
PING www.google.com (216.58.209.36): 56 data bytes
64 bytes from 216.58.209.36: icmp_seq=0 ttl=118 time=29.002 ms
64 bytes from 216.58.209.36: icmp_seq=1 ttl=118 time=30.243 ms
```

5.2 Εκτελούμε στα PC1, PC2 “pkg install miredo”

5.3 Εκτελούμε στα PC1, PC2 “sysrc miredo_enable=“YES””.

5.4 Εκτελούμε στα PC2 “service miredo start”.

5.5 Βλέπουμε την επιπλέον διεπαφή teredo με IPv6 address

2001:0:d911:c0d9:2c63:21dd:4dc4:826b/128

```
teredo: flags=43<UP,BROADCAST,RUNNING> metric 0 mtu 1500
options=80000<LINKSTATE>
inet6 fe80::ffff:ffff:ffff%teredo prefixlen 64 scopeid 0x3
inet6 2001:0:d911:c0d9:2c63:21dd:4dc4:826b prefixlen 128
groups: tun
nd6 options=21<PERFORMNUD,AUTO_LINKLOCAL>
Opened by PID 969
```

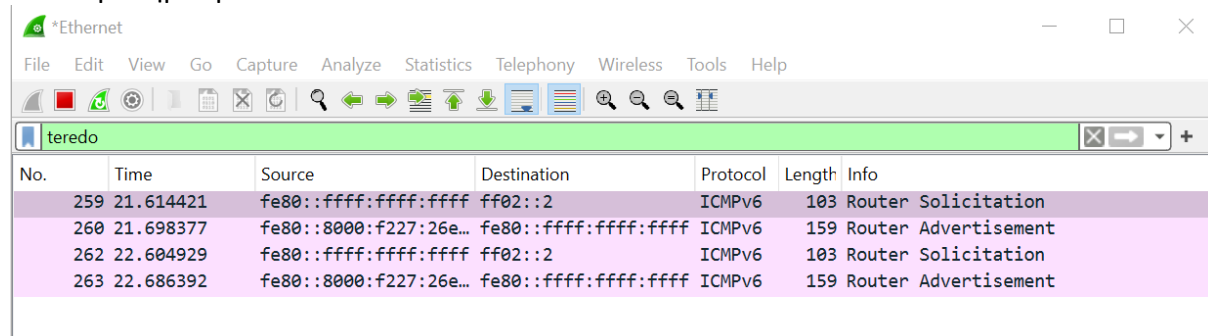
5.6 Εκτελούμε στο PC2 “tcpdump -ni em0”.

5.7 Είναι η 217.17.192.217:

```
root@PC:~ # tcpdump -ni em0
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on em0, link-type EN10MB (Ethernet), capture size 262144 bytes
20:49:27.974525 IP 10.0.2.15.47803 > 217.17.192.217.3544: UDP, length 61
20:49:28.052338 IP 217.17.192.217.3544 > 10.0.2.15.47803: UDP, length 117
```

5.8 Χρησιμοποιείται το UDP στο Transport Layer και η θύρα 3544.

5.9 Παρατηρούμε πακέτα ICMPv6:



The image shows a Wireshark packet capture window titled "*Ethernet". The interface selected is "teredo". The packet list shows four packets, all of which are ICMPv6. The first two are Router Solicitations (type 103) and the last two are Router Advertisements (type 159). The source and destination addresses are IPv6.

No.	Time	Source	Destination	Protocol	Length	Info
259	21.614421	fe80::ffff:ffff:ffff	ff02::2	ICMPv6	103	Router Solicitation
260	21.698377	fe80::8000:f227:26e...	fe80::ffff:ffff:ffff	ICMPv6	159	Router Advertisement
262	22.604929	fe80::ffff:ffff:ffff	ff02::2	ICMPv6	103	Router Solicitation
263	22.686392	fe80::8000:f227:26e...	fe80::ffff:ffff:ffff	ICMPv6	159	Router Advertisement

5.10 Ο διακομιστής Teredo διαφημίζει το πρόθεμα δικτύου IPv6 2001:0000::/32. Αυτό το πρόθεμα χρησιμοποιείται για τη δημιουργία διευθύνσεων Teredo IPv6, οι οποίες διευκολύνουν τη συνδεσιμότητα IPv6 μέσω δικτύων IPv4, ιδιαίτερα μέσω συσκευών NAT.

5.11 Μπορώ σε όλα εκτός του www.ibm.com.

5.12 κάνω ping6 www.ntua.gr

5.13 Παρατηρώ:

```
90 10.570755 fe80::8cb1:3f2c:b3fa:9f79 2001:0:d911:c0d9:a8:8af:4dc4:826b IPv6 90 IPv6 no next header
```

5.14 Όχι, δεν παρατηρώ

5.15 Χρησιμοποιείται το UDP στο Transport Layer και η θύρα 3544 με ipv4 216.66.84.238:

```
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on em0, link-type EN10MB (Ethernet), capture size 262144 bytes
20:45:34.654070 IP 10.0.2.15.56632 > 216.66.84.238.3545: UDP, length 56
20:45:35.694815 IP 10.0.2.15.56632 > 216.66.84.238.3545: UDP, length 56
20:45:36.746589 IP 10.0.2.15.56632 > 216.66.84.238.3545: UDP, length 56
20:45:37.783084 IP 10.0.2.15.56632 > 216.66.84.238.3545: UDP, length 56
20:45:38.793369 IP 10.0.2.15.56632 > 216.66.84.238.3545: UDP, length 56
20:45:39.865955 IP 10.0.2.15.56632 > 216.66.84.238.3545: UDP, length 56
20:45:40.894532 IP 10.0.2.15.56632 > 216.66.84.238.3545: UDP, length 56
20:45:41.611126 IP 216.66.84.238.3545 > 10.0.2.15.56632: UDP, length 56
20:45:41.921164 IP 10.0.2.15.56632 > 216.66.84.238.3545: UDP, length 56
20:45:42.204127 IP 216.66.84.238.3545 > 10.0.2.15.56632: UDP, length 56
20:45:42.978296 IP 10.0.2.15.56632 > 216.66.84.238.3545: UDP, length 56
20:45:44.016024 IP 10.0.2.15.56632 > 216.66.84.238.3545: UDP, length 56
20:45:45.044402 IP 10.0.2.15.56632 > 216.66.84.238.3545: UDP, length 56
20:45:46.117565 IP 10.0.2.15.56632 > 216.66.84.238.3545: UDP, length 56
20:45:46.824099 IP 10.0.2.15.56632 > 217.17.192.217.3544: UDP, length 61
20:45:46.906092 IP 217.17.192.217.3544 > 10.0.2.15.56632: UDP, length 117
20:45:47.141541 IP 10.0.2.15.56632 > 216.66.84.238.3545: UDP, length 56
20:45:47.230199 IP 216.66.84.238.3545 > 10.0.2.15.56632: UDP, length 56
20:45:48.164291 IP 10.0.2.15.56632 > 216.66.84.238.3545: UDP, length 56
20:45:48.361527 IP 216.66.84.238.3545 > 10.0.2.15.56632: UDP, length 56
```

5.16 Εκτελούμε στο PC1 “tcpdump -ni teredo”.

Βλέπουμε ICMPv6 Echo requests/replies:

```
root@PC:~ # tcpdump -ni teredo
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on teredo, link-type NULL (BSD loopback), capture size 262144 bytes
20:48:15.895108 IP6 2001:0:d911:c0d9:a8:8af:4dc4:826b > 2001:648:2000:329::101:
ICMP6, echo request, seq 354, length 16
20:48:16.871793 IP6 2001:648:2000:329::101 > 2001:0:d911:c0d9:a8:8af:4dc4:826b:
ICMP6, echo reply, seq 354, length 16
20:48:16.897402 IP6 2001:0:d911:c0d9:a8:8af:4dc4:826b > 2001:648:2000:329::101:
ICMP6, echo request, seq 355, length 16
20:48:17.958067 IP6 2001:0:d911:c0d9:a8:8af:4dc4:826b > 2001:648:2000:329::101:
ICMP6, echo request, seq 356, length 16
20:48:18.881728 IP6 2001:648:2000:329::101 > 2001:0:d911:c0d9:a8:8af:4dc4:826b:
ICMP6, echo reply, seq 356, length 16
20:48:18.963689 IP6 2001:0:d911:c0d9:a8:8af:4dc4:826b > 2001:648:2000:329::101:
ICMP6, echo request, seq 357, length 16
20:48:19.975398 IP6 2001:0:d911:c0d9:a8:8af:4dc4:826b > 2001:648:2000:329::101:
ICMP6, echo request, seq 358, length 16
```

5.17 Όχι, δε μπορούμε.

5.18 Ναι, παράγονται ICMPv6 Echo requests.

```
21:03:58.722396 IP6 2001:0:d911:c0d9:a8:8af:4dc4:826b > 2001:0:d911:c0d9:10f6:11
e6:4dc4:826b: ICMP6, echo request, seq 5, length 16
```

5.19 Παράγονται και στέλνονται στην 216.66.84.238, όπως είδαμε και πριν.

```

tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on em0, link-type EN10MB (Ethernet), capture size 262144 bytes
21:07:31.564271 IP 10.0.2.15.56632 > 216.66.84.238.3545: UDP, length 56
21:07:32.621323 IP 10.0.2.15.56632 > 216.66.84.238.3545: UDP, length 56
21:07:33.655444 IP 10.0.2.15.56632 > 216.66.84.238.3545: UDP, length 56
21:07:34.575873 IP 216.66.84.238.3545 > 10.0.2.15.56632: UDP, length 56
21:07:34.672005 IP 10.0.2.15.56632 > 216.66.84.238.3545: UDP, length 56
^C
5 packets captured
5 packets received by filter
0 packets dropped by kernel

```

5.20 Κάνοντας ping στο www.quad9.net βλέπουμε πως κάνουμε ping στη θύρα 3544 της 216.218.142.110, ενώ όταν κάνω στο www.f5.com στη θύρα 3544 της 217.17.192.217

```

root@PC:~ # tcpdump -ni em0
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on em0, link-type EN10MB (Ethernet), capture size 262144 bytes
21:11:13.600995 IP 216.218.142.110.3545 > 10.0.2.15.56632: UDP, length 56
21:11:14.443185 IP 10.0.2.15.56632 > 216.218.142.110.3545: UDP, length 56
21:11:14.644495 IP 216.218.142.110.3545 > 10.0.2.15.56632: UDP, length 56
21:11:15.463973 IP 10.0.2.15.56632 > 216.218.142.110.3545: UDP, length 56
21:11:15.686623 IP 216.218.142.110.3545 > 10.0.2.15.56632: UDP, length 56
21:11:16.504014 IP 10.0.2.15.56632 > 216.218.142.110.3545: UDP, length 56
21:11:16.732441 IP 216.218.142.110.3545 > 10.0.2.15.56632: UDP, length 56

```

```

root@PC:~ # tcpdump -ni em0
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on em0, link-type EN10MB (Ethernet), capture size 262144 bytes
21:12:00.027365 IP 10.0.2.15.51319 > 192.168.1.1.53: 20820+ AAAA? www.f5.com. (28)
21:12:00.242931 IP 192.168.1.1.53 > 10.0.2.15.51319: 20820 9/0/0 CNAME dubfwz8xn
cgmg.cloudfront.net., AAAA 2600:9000:203c:d400:14:232e:8a00:93a1, AAAA 2600:9000:203c:9e00:14:232e:8a00:93a1, AAAA 2600:9000:203c:4c00:14:232e:8a00:93a1, AAAA 2600:9000:203c:ea00:14:232e:8a00:93a1, AAAA 2600:9000:203c:a200:14:232e:8a00:93a1, AAAA 2600:9000:203c:1600:14:232e:8a00:93a1, AAAA 2600:9000:203c:5600:14:232e:8a00:93a1, AAAA 2600:9000:203c:b200:14:232e:8a00:93a1 (294)
21:12:00.245905 IP 10.0.2.15.56632 > 217.17.192.217.3544: UDP, length 66
21:12:00.833798 ARP, Request who-has 10.0.2.2 tell 10.0.2.15, length 28
21:12:00.835172 ARP, Reply 10.0.2.2 is-at 52:54:00:12:35:02, length 46
21:12:03.356379 IP 10.0.2.15.56632 > 217.17.192.217.3544: UDP, length 66
21:12:06.474443 IP 10.0.2.15.56632 > 217.17.192.217.3544: UDP, length 66
21:12:09.605264 IP 10.0.2.15.56632 > 217.17.192.217.3544: UDP, length 66
^C

```