



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ
ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

ΟΝΟΜΑΤΕΠΩΝΥΜΟ: ΓΚΟΥΜΕ ΛΑΟΥΡΕΝΤΙΑΝ

ΑΜ: 031 18 014

ΕΞΑΜΗΝΟ: 8^ο

ΟΜΑΔΑ: 3

ΕΡΓΑΣΤΗΡΙΟ ΔΙΚΤΥΩΝ ΥΠΟΛΟΓΙΣΤΩΝ



ΕΡΓΑΣΤΗΡΙΑΚΗ ΑΣΚΗΣΗ 11: ΤΟ ΠΡΩΤΟΚΟΛΛΟ IPv6

Προετοιμασία στο σπίτι

Δημιουργούμε το μηχάνημα:

```
root@R0:/usr/local/etc/frr # cat /etc/rc.conf
sshd_enable="YES" # to enable the ssh daemon
hostname="R0" #to assign the host name
gateway_enable="YES"
frr_enable="YES"
frr_daemons="zebra staticd ripd ripngd ospfd ospf6d bgpd"
```

Άσκηση 1: Εισαγωγή στο IPv6

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 7^ο 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 fe80:0000:0000:0000:0a00:27ff: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”.

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) Όμοια με πριν 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.

```

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:	Gateway	Flags	Netif
Destination			
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.

```

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) Καμία.

1.22) Εκτελούμε “man ndp”.

1.23) Εκτελούμε “ndp -a”.

```

root@PC1:~ # ndp -a
Neighbor                                Linklayer Address  Netif  Expire      S  Flags
fd00:1::2                              08:00:27:71:2a:cf  em0    permanent  R
fd00:1::3                              08:00:27:3c:8a:c5  em0    23h52m39s  S
fe80::a00:27ff:fe71:2acf%em0          08:00:27:71:2a:cf  em0    permanent  R
fe80::a00:27ff:fe3c:8ac5%em0          08:00:27:3c:8a:c5  em0    23h53m49s  S

```

1.24) 4 (ωστόσο είχαμε κάνει δοκιμαστικά ping6 στην 2^η διεύθυνση του PC2, για αυτό και εμφανίστηκε στον πίνακα).

1.25) Εκτελούμε “ndp -p” και βλέπουμε άπειρη διάρκεια ζωής (expire=Never). Από τα προθέματα αυτά, μπορούν να χρησιμοποιηθούν από τον μηχανισμό αυτόματης απόδοσης διευθύνσεων αυτές που έχουν το Flag A, επομένως τα προθέματα fe80::%em0/64 και fe80::%lo0/64.

```

root@PC1:~ # ndp -p
fd00:1::/64 if=em0
flags=LO vlttime=infinity, pltime=infinity, expire=Never, ref=1
No advertising router
fe80::%em0/64 if=em0
flags=LA0 vlttime=infinity, pltime=infinity, expire=Never, ref=0
No advertising router
fe80::%lo0/64 if=lo0
flags=LA0 vlttime=infinity, pltime=infinity, expire=Never, ref=0
No advertising router

```

1.26) Εκτελούμε “tcpdump -vvvni em0”.

1.27) “ndp -c”.

1.28) Βλέπουμε 4 IPv6 πακέτα, ένα Neighbor Solicitation με το οποίο ο PC1 ρωτάει για τη MAC του PC2, ένα Neighbor Advertisement που περιέχει την απάντηση στο παραπάνω και στη συνέχεια το ICMPv6 Request και Reply.

```

root@PC2:~ # tcpdump -vvvni em0
tcpdump: listening on em0, link-type EN10MB (Ethernet), capture size 262144 bytes
23:33:23.186024 IP6 (hlim 255, next-header ICMPv6 (58) payload length: 32) fe80::a00:27ff:fe71:2acf > ff02::1:ff3c:8ac5: [icmp6 sum ok] ICMP6, neighbor solicitation, length 32, who has fe80::a00:27ff:fe3c:8ac5
    source link-address option (1), length 8 (1): 08:00:27:71:2a:cf
    0x0000: 0800 2771 2acf
23:33:23.186147 IP6 (hlim 255, next-header ICMPv6 (58) payload length: 32) fe80::a00:27ff:fe3c:8ac5 > fe80::a00:27ff:fe71:2acf: [icmp6 sum ok] ICMP6, neighbor advertisement, length 32, tgt is fe80::a00:27ff:fe3c:8ac5, Flags [solicited, override]
    destination link-address option (2), length 8 (1): 08:00:27:3c:8a:c5
    0x0000: 0800 273c 8ac5
23:33:23.186873 IP6 (hlim 64, next-header ICMPv6 (58) payload length: 16) fe80::a00:27ff:fe71:2acf > fe80::a00:27ff:fe3c:8ac5: [icmp6 sum ok] ICMP6, echo request, seq 0
23:33:23.186937 IP6 (hlim 64, next-header ICMPv6 (58) payload length: 16) fe80::a00:27ff:fe3c:8ac5 > fe80::a00:27ff:fe71:2acf: [icmp6 sum ok] ICMP6, echo reply, seq 0

```

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)

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) Δε καταγράψαμε 2^ο πακέτο NS, αλλά NA, το οποίο έχει ως διεύθυνση προορισμού την IPv6 του PC1.

1.33) Η εγγραφή που αφορά το PC1 (2^η) είναι σε κατάσταση Stale ενώ η διάρκεια ζωής της είναι 24 ώρες (στην αρχή).

```

root@PC2:~ # ndp -a
Neighbor                               Linklayer Address  Netif Expire      S Flags
fd00:1::3                             08:00:27:3c:8a:c5  em0 permanent R
fe80::a00:27ff:fe71:2acf%em0          08:00:27:71:2a:cf  em0 22h38m23s S
fe80::a00:27ff:fe3c:8ac5%em0          08:00:27:3c:8a:c5  em0 permanent R

```

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 ξαναμπεί στην κατάσταση R.

1.36) Η διάρκεια ζωής της κατάστασης S είναι ένα 24ωρο.

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: SLAAC και Στατική δρομολόγηση IPv6

2.1)

```
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)

```
root@R1:~ # tcpdump -i em0
```

2.10) Το ring αποτυγχάνει, καθώς 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, neighbor 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, neighbor 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, neighbor 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)

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

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

2.14)

```
R2(config)# ipv6 route fd00:1::/64 fd00:3::1
```

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

2.16) Εκτελούμε 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  
delete net default
```

2.21) “tcpdump -eni em0”.

2.22) “service netif restart”.

2.23) Καταγράφουμε τα παρακάτω πακέτα:

```
root@R1:~ # tcpdump -eni em0  
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode  
listening on em0, link-type EN10MB (Ethernet), capture size 262144 bytes  
00:50:17.887674 08:00:27:71:2a:cf > 33:33:00:00:00:16, ethertype IPv6 (0x86dd),  
length 90: :: > ff02::16: HBH ICMP6, multicast listener report v2, 1 group recor  
d(s), length 28  
00:50:17.887693 08:00:27:71:2a:cf > 33:33:ff:71:2a:cf, ethertype IPv6 (0x86dd),  
length 86: :: > ff02::1:ff71:2acf: ICMP6, neighbor solicitation, who has fe80::a  
00:27ff:fe71:2acf, length 32  
00:50:17.887730 08:00:27:71:2a:cf > 33:33:00:00:00:16, ethertype IPv6 (0x86dd),  
length 110: :: > ff02::16: HBH ICMP6, multicast listener report v2, 2 group reco  
rd(s), length 48  
00:50:17.887735 08:00:27:71:2a:cf > 33:33:00:00:00:16, ethertype IPv6 (0x86dd),  
length 170: fe80::a00:27ff:fe71:2acf > ff02::16: HBH ICMP6, multicast listener r  
eport v2, 5 group record(s), length 108  
00:50:17.887771 08:00:27:71:2a:cf > 33:33:00:00:00:16, ethertype IPv6 (0x86dd),  
length 90: fe80::a00:27ff:fe71:2acf > ff02::16: HBH ICMP6, multicast listener re  
port v2, 1 group record(s), length 28  
00:50:18.356342 08:00:27:71:2a:cf > 33:33:00:00:00:02, ethertype IPv6 (0x86dd),  
length 70: fe80::a00:27ff:fe71:2acf > ff02::2: ICMP6, router solicitation, lengt  
h 16  
00:50:18.356982 08:00:27:ae:b1:aa > 33:33:00:00:00:01, ethertype IPv6 (0x86dd),  
length 110: fe80::a00:27ff:feae:b1aa > ff02::1: ICMP6, router advertisement, len  
gth 56  
00:50:18.607975 08:00:27:71:2a:cf > 33:33:ff:71:2a:cf, ethertype IPv6 (0x86dd),  
length 86: :: > ff02::1:ff71:2acf: ICMP6, neighbor solicitation, who has fd00:1:  
a00:27ff:fe71:2acf, length 32
```

Έχουμε τα παρακάτω πακέτα:

- 1) Το PC1 στέλνει multicast IPv6 over Ethernet στη MAC address 33:33:00:00:00:16, η οποία παράχθηκε βάζοντας στην αρχή το πρόθεμα 33:33 και στη συνέχεια τα τελευταία 32 bits της IPv6 προορισμού, η οποία εν προκειμένω είναι η ff02::16 (IPv6 source: 90: ::).
- 2) Το PC1 στέλνει NS (destination MAC 33:33:ff:71:2a:cf, IPv6 source: 86: :: , IPv6 destination: ff02::1:ff71:2acf το οποίο είναι το solicited node του προορισμού), με το οποίο ρωτάει αν έχει κανείς την IPv6 του.
- 3) Στέλνει ξανά πακέτο όπως το 1°, μόνο που IPv6 πηγή αυτή τη φορά είναι η 110: :: .
- 4) Ίδιο πακέτο με το 1, αλλά με Source IPv6 την link-local address που παίρνει, δηλαδή την fe80::a00:27ff:fe71:2acf.

- 5) Το PC1 στέλνει RS στο MAC/IPv6 destination 33:33:00:00:00:02 / ff02::2, ώστε να ζητήσει να λαμβάνει μηνύματα RA.
- 6) Ο R1 απαντάει στο destination MAC/IPv6 33:33:00:00:00:01 / ff02::1 με RA.
- 7) Το PC1 στέλνει NS στην MAC/IPv6 33:33:ff:71:2a:cf / ff02::1:ff71:2acf ρωτώντας ποιος έχει την fd00:1::a00:27ff:fe71:2acf (με source MAC 86: ::).

2.24) Παράγει 2 NS όπως είδαμε. Το μεν πρώτο χρησιμοποιείται για την εύρεση της mac του γείτονα ενώ το 2^ο για την ανίχνευση ταυτόσημων διευθύνσεων, ώστε να ορίσει σωστά τη διεύθυνσή του.

2.25) Βλέπουμε πως χρησιμοποιείται η undefined “::”, καθώς δεν έχει ακόμα διεύθυνση μέχρι και το 2^ο NS.

2.26) Την fe80::a00:27ff:fe71:2acf.

2.27) Έχουμε:

- NS → ff02::1:ff71:2acf (Solicited node)
- RS → ff02::2 (Multicast)
- RA → ff02::1 (Multicast)

2.28) Έχουμε:

- NS → 33:33:ff:71:2a:cf
- RS → 33:33:00:00:00:02
- RA → 33:33:00:00:00:01

Όπως είδαμε και στο 2.23 το πρόθεμα 33:33 χρησιμοποιείται για IPv6 Multicast Over Ethernet, ενώ ως τελευταία 32 bit μπαίνουν τα τελευταία 32 της IPv6 προορισμού.

2.29) Παρατηρούμε πως πλέον η πρώτη εγγραφή έχει επιπρόσθετα το flag A και πεπερασμένους χρόνους στα διάφορα πεδία.

```
root@PC1:~ # ndp -p
fd00:1::/64 if=em0
flags=LA0 vlttime=2592000, pltime=604800, expire=29d23h56m57s, ref=1
  advertised by
    fe80::a00:27ff:feae:b1aa%em0 (reachable)
fe80::%em0/64 if=em0
flags=LA0 vlttime=infinity, pltime=infinity, expire=Never, ref=0
  No advertising router
fe80::%lo0/64 if=lo0
flags=LA0 vlttime=infinity, pltime=infinity, expire=Never, ref=0
  No advertising router
```

2.30) Τις παρακάτω:

```
inet6 fe80::a00:27ff:fe71:2acf%em0 prefixlen 64 scopeid 0x1
inet6 fd00:1::a00:27ff:fe71:2acf 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       UGRS           lo0
default          fe80::a00:27ff:fea UG             em0
localhost        link#3          UH             lo0
::ffff:0.0.0.0/96 localhost       UGRS           lo0
fd00:1::/64      link#1          U              em0
fd00:1::a00:27ff:f link#1          UHS            lo0
fe80::/10        localhost       UGRS           lo0
fe80::%em0/64    link#1          U              em0
fe80::a00:27ff:fe7 link#1          UHS            lo0
fe80::%lo0/64    link#3          U              lo0
fe80::1%lo0      link#3          UHS            lo0
ff02::/16        localhost       UGRS           lo0
```

2.32) Από το PC2 μπορούμε να κάνουμε ping μόνο στην 2^η 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: Δυναμική δρομολόγηση IPv6

Σημείωση: Είχαμε ξεχάσει να προσθέσουμε την υπηρεσία *ripngd* στην αρχή με αποτέλεσμα να απαιτείται να ξαναφτιάξουμε τα μηχανήματα. Έχουμε πλέον το εξής *configuration*:

- **PC1:** MAC(08:00:27:88:02:85),
IPv6_Link-local(fe80::a00:27ff:fe88:285%em0),
IPv6_private(fd00:1::a00:27ff:fe88:285)
- **PC2:** MAC(08:00:27:1b:e6:e6),
IPv6_Link-local(fe80::a00:27ff:fe1b:e6e6%em0),
IPv6_private(fd00:2::2)
- **R1(em0):** MAC(08:00:27:bc:47:fc),
IPv6_Link-local(fe80::a00:27ff:febc:47fc%em0),
IPv6_private(fd00:1::1)
- **R1(em1):** MAC(08:00:27:4d:fa:0a),

- IPv6_Link-local(fe80::a00:27ff:fe4d:fa0a%em1),
IPv6_private(fd00:3::1)
- **R2(em0):** MAC(08:00:27:38:04:bd),
IPv6_Link-local(fe80::a00:27ff:fe38:4bd%em0),
IPv6_private(fd00:3::2)
- **R2(em1):** MAC(08:00:27:37:c8:2a),
IPv6_Link-local(fe80::a00:27ff:fe37:c82a%em1),
IPv6_private(fd00:2::1)

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) Εκτελούμε τα παρακάτω στα R1 και R2.

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

3.3) Βλέπουμε μία μόνο 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 - NHRP, T - Table,
        V - VNC, U - VNC-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 byte
s
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 ενδιάμεσο 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 a
n 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 a
n 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,
       v - 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 -vnni em0”. Παρουσιάζουμε ενδεικτικά κάποιες καταγραφές.

```

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 [V6, 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 [V6, 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 [V6, 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 [V6, 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 [V6, 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 OSPFIGRP.

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)

```
R1(config)# router-id 1.1.1.1
R1(config)# router bgp 65010
```

3.26)

```
R1(config-router)# no bgp ebgp-requires-policy
```

3.27)

```
R1(config-router)# no bgp default ipv4-unicast
```

3.28)

```
R1(config-router)# neighbor fd00:3::2 remote-as 65020
```

3.29)

```
R1(config-router)# address-family ipv6
```

3.30)

```
R1(config-router-af)# network fd00:1::/64
```

3.31)

```
R1(config-router-af)# neighbor fd00:3::2 activate
```

3.32) Εκτελούμε τις εντολές για τον R2, τις οποίες και δείχνουμε στην αρχή της επόμενης σελίδας.

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

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

```
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
```

3.34) 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.

3.35) Εκτελούμε “tcpdump -vnni em1”.

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

```
15:06:54.582090 IP6 (class 0xc0, flowlabel 0x868fc, hlim 1, next-header TCP (6)
payload length: 51) fd00:3::1.179 > fd00:3::2.33900: Flags [P.], cksum 0x3faf (c
orrect), seq 1131965319:1131965338, ack 3707604277, win 129, options [nop,nop,TS
val 876750729 ecr 2712640647], length 19: BGP
    Keepalive Message (4), length: 19

15:06:55.163137 IP6 (class 0xc0, flowlabel 0x520f0, hlim 1, next-header TCP (6)
payload length: 51) fd00:3::2.33900 > fd00:3::1.179: Flags [P.], cksum 0x5534 (c
orrect), seq 1:20, ack 19, win 129, options [nop,nop,TS val 2712700654 ecr 87675
0729], length 19: BGP
    Keepalive Message (4), length: 19
```

3.37) hlim=1.

3.38) Μπορούμε στις παρακάτω 2 διευθύνσεις:

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

3.39) Εκτελούμε στο PC1 αρχικά “reboot”. Το PC1 κρασάρει, οπότε το κλείνουμε και το ανοίγουμε απ’ την αρχή.

```
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
```

3.40)

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

3.41)

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

3.42)

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

3.43)

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

3.44)

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

3.45)

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

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

3.46) 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
```

3.47) 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 - NHRP, 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
```

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

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

3.50) 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: Μηχανισμός μετάβασης 464 XLAT

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  
2000 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  
3000 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 αμφότερα.

```
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
(BIOCPROMISC: 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
(BIOCPROMISC: 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, s
eq 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, l
ength 64
18:38:30.255862 IP 192.168.2.2 > 2.2.2.158: ICMP echo reply, id 1024, seq 0, len
gth 64
18:38:30.255868 IP6 64:ff9b::c0a8:202 > fd00:3:1::c0a8:102: ICMP6, echo reply, s
eq 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)


```

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:~ #

```

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

Άσκηση 5: Μηχανισμός μετάβασης Teredo

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:c38c:c38c:2474:1803:3c04:cbe8/128.

```

teredo: flags=43<UP,BROADCAST,RUNNING> metric 0 mtu 1500
options=800000<LINKSTATE>
inet6 fe80::ffff:ffff:ffff%teredo prefixlen 64 scopeid 0x3
inet6 2001:0:c38c:c38c:2474:1803:3c04:cbe8 prefixlen 128
groups: tun
nd6 options=21<PERFORMNUD,AUTO_LINKLOCAL>
Opened by PID 969

```

5.6) Εκτελούμε στο PC1 “tcpdump -ni em0”.

5.7) Είναι 195.140.195.140.

```

root@PC1:~ # 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
22:44:13.582561 IP 10.0.2.15.44502 > 195.140.195.140.3544: UDP, length 61
22:44:13.676519 IP 195.140.195.140.3544 > 10.0.2.15.44502: UDP, length 117
22:44:43.582860 IP 10.0.2.15.44502 > 195.140.195.140.3544: UDP, length 61
22:44:43.670581 IP 195.140.195.140.3544 > 10.0.2.15.44502: UDP, length 117

```

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

5.9) Παρατηρούμε πακέτα ICMP.

teredo						
No.	Time	Source	Destination	Protocol	Length	Info
43	17.833096	fe80::ffff:ffff:ffff	ff02::2	ICMP...	103	Router Solicitation
44	17.912502	fe80::8000:f227:3c73:3c73	fe80::ffff:ffff:ffff	ICMP...	159	Router Advertisement
53	22.958001	fe80::ffff:ffff:ffff	ff02::2	ICMP...	103	Router Solicitation
54	23.035181	fe80::8000:f227:3c73:3c73	fe80::ffff:ffff:ffff	ICMP...	159	Router Advertisement

5.10) Μπορώ να κάνω μόνο στο www.ibm.com.

5.11) Κάνουμε “ping6 www.ibm.com”.

5.12) Παρατηρούμε τα εξής:

126...	699.865...	fe80::d813:8b1e:1cba:d144	2001:0:c38c:c38c:2474:180...	IPv6	90	IPv6 no next header
--------	------------	---------------------------	------------------------------	------	----	---------------------

5.13) Όχι.

5.14) Βλέπουμε το πρωτόκολλο UDP και την θύρα 3545.

```

23:06:36.324359 IP 216.66.80.238.3545 > 10.0.2.15.44502: UDP, length 56
23:06:37.233478 IP 10.0.2.15.44502 > 216.66.80.238.3545: UDP, length 56
23:06:37.347712 IP 216.66.80.238.3545 > 10.0.2.15.44502: UDP, length 56
23:06:38.254853 IP 10.0.2.15.44502 > 216.66.80.238.3545: UDP, length 56

```

5.15) Εκτελούμε στο PC1 “tcpdump -ni teredo”.

5.16) Βλέπουμε ICMPv6 Echo requests/replies.

```

23:08:52.104444 IP6 2001:0:c38c:c38c:2474:1803:3c04:cbe8 > 2a02:26f0:11a:48d::1e89: ICMP6, echo request, seq 364, length 16
23:08:52.208770 IP6 2a02:26f0:11a:48d::1e89 > 2001:0:c38c:c38c:2474:1803:3c04:cbe8: ICMP6, echo reply, seq 364, length 16

```

5.17) Όχι, δε μπορούμε.

```

root@PC1:~ # ping6 -I teredo 2001:0:c38c:c38c:c5d:1801:3c04:cbe8
PING6(56=40+8+8 bytes) 2001:0:c38c:c38c:2474:1803:3c04:cbe8 --> 2001:0:c38c:c38c:c5d:1801:3c04:cbe8

```

```

root@PC1:~ # ping6 -I teredo 2001:0:c38c:c38c:c5d:1801:3c04:cbe8
ping6: UDP connect: Can't assign requested address
root@PC1:~ # ping6 2001:0:c38c:c38c:c5d:1801:3c04:cbe8
ping6: UDP connect: Can't assign requested address

```

5.18) Ναι, παράγονται ICMPv6 Echo requests.

```
23:16:00.676927 IP6 2001:0:c38c:c38c:2474:1803:3c04:cbe8 > 2001:0:c38c:c38c:1801:3c04:cbe8: ICMP6, echo request, seq 10, length 16
```

5.19) Παράγονται και στέλνονται στην 195.140.195.140, όπως είδαμε και πριν.

```
root@PC1:~ # 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
23:17:05.785077 IP 10.0.2.15.44502 > 195.140.195.140.3544: UDP, length 61
23:17:05.933621 IP 195.140.195.140.3544 > 10.0.2.15.44502: UDP, length 117
23:17:35.782945 IP 10.0.2.15.44502 > 195.140.195.140.3544: UDP, length 61
23:17:35.866260 IP 195.140.195.140.3544 > 10.0.2.15.44502: UDP, length 117
```

5.20) Κάνοντας ping στο www.hp.com βλέπουμε πως κάνουμε ping στη θύρα 3544 της 195.140.195.140, όπως και όταν κάνω στο www.f5.com.

```
root@PC1:~ # 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
23:30:19.677975 ARP, Request who-has 10.0.2.2 tell 10.0.2.15, length 28
23:30:19.679274 ARP, Reply 10.0.2.2 is-at 52:54:00:12:35:02, length 46
23:30:19.679346 IP 10.0.2.15.44112 > 62.217.126.164.53: 60902+ AAAA? www.hp
(28)
23:30:19.689682 IP 62.217.126.164.53 > 10.0.2.15.44112: 60902 4/0/0 CNAME w
.com.edgekey.net., CNAME e40715.dsca.akamaiedge.net., AAAA 2a02:26f0:ea::21
a, AAAA 2a02:26f0:ea::215:ac62 (157)
23:30:19.691656 IP 10.0.2.15.44502 > 195.140.195.140.3544: UDP, length 66
```

```
root@PC1:~ # 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
23:26:10.250804 IP 10.0.2.15.61407 > 62.217.126.164.53: 44573+ AAAA? www.f5
(28)
23:26:10.440591 IP 62.217.126.164.53 > 10.0.2.15.61407: 44573 9/0/0 CNAME d
Bxncgmg.cloudfront.net., AAAA 2600:9000:2045:4e00:14:232e:8a00:93a1, AAAA 2
000:2045:8a00:14:232e:8a00:93a1, AAAA 2600:9000:2045:5800:14:232e:8a00:93a1
h 2600:9000:2045:5a00:14:232e:8a00:93a1, AAAA 2600:9000:2045:5400:14:232e:8
9a1, AAAA 2600:9000:2045:2800:14:232e:8a00:93a1, AAAA 2600:9000:2045:aa00:1
e:8a00:93a1, AAAA 2600:9000:2045:5200:14:232e:8a00:93a1 (294)
23:26:10.443097 IP 10.0.2.15.44502 > 195.140.195.140.3544: UDP, length 66
23:26:10.597271 IP 195.140.195.140.3544 > 10.0.2.15.44502: UDP, length 48
23:26:10.597818 IP 10.0.2.15.44502 > 216.66.80.238.3545: UDP, length 40
23:26:10.643016 IP 216.66.80.238.3545 > 10.0.2.15.44502: UDP, length 66
23:26:10.643384 IP 10.0.2.15.44502 > 216.66.80.238.3545: UDP, length 56
23:26:10.735425 IP 216.66.80.238.3545 > 10.0.2.15.44502: UDP, length 56
```

5.21) Παρατηρούμε πως χρησιμοποιήθηκε διαφορετική θύρα αυτή τη φορά και συγκεκριμένα η 3544 στην αρχή και μετά η 43920.

```
root@PC1:~ # 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
23:33:00.909120 IP 10.0.2.15.43862 > 62.217.126.164.53: 54331+ AAAA? one.on
lone. (33)
23:33:00.919547 IP 62.217.126.164.53 > 10.0.2.15.43862: 54331 2/0/0 AAAA 26
00:4700::1111, AAAA 2606:4700:4700::1001 (89)
23:33:00.921630 IP 10.0.2.15.44502 > 195.140.195.140.3544: UDP, length 66
23:33:01.006152 IP 195.140.195.140.3544 > 10.0.2.15.44502: UDP, length 48
23:33:01.006862 IP 10.0.2.15.44502 > 195.140.195.140.43920: UDP, length 40
23:33:01.088073 IP 195.140.195.140.43920 > 10.0.2.15.44502: UDP, length 66
23:33:01.088469 IP 10.0.2.15.44502 > 195.140.195.140.43920: UDP, length 56
23:33:01.170904 IP 195.140.195.140.43920 > 10.0.2.15.44502: UDP, length 56
```