

Όνοματεπώνυμο: Ιωάννης Γιαννούκος	Όνομα PC: John John
Ομάδα: 1	Ημερομηνία: 5/6/2023

Εργαστηριακή Άσκηση 11

Το πρωτόκολλο IPv6

Προετοιμασία στο σπίτι

- 1) Κλείνω την υπηρεσία frr με `“service frr stop”`.
- 2) Δημιουργώ άδειο αρχείο παραμετροποίησης `/usr/local/etc/frr/ripngd.conf` για το RIPng με την εντολή `“touch /usr/local/etc/frr/ripngd.conf”`.
- 3) Δημιουργώ άδειο αρχείο παραμετροποίησης `/usr/local/etc/frr/ospf6d.conf` για το OSPFv3 με την εντολή `“touch /usr/local/etc/frr/ospf6d.conf”`.
- 4) Ορίζω ως frr την ταυτότητα ιδιοκτήτη και ομάδας των αρχείων `/usr/local/etc/frr/ripngd.conf` και `/usr/local/etc/frr/ospf6d.conf` με τις εντολές `“chown frr:frr /usr/local/etc/frr/ripngd.conf”` και `“chown frr:frr /usr/local/etc/frr/ospf6d.conf”`, αντίστοιχα.
- 5) Στο αρχείο παραμετροποίησης `/etc/rc.conf` προσθέτω το `ripngd` και `ospf6d` στη γραμμή `frr_daemons=“zebra staticd ripd ospfd bgpd”` ώστε αυτή να γίνει `frr_daemons=“zebra staticd ripd ripngd ospfd ospf6d bgpd”` (μπορώ να χρησιμοποιήσω και την εντολή `“sysrc”` για ευκολία και ασφάλεια ως εξής: `“sysrc -f /etc/rc.conf frr_daemons+=“foo”`).
- 6) Ξεκινώ την υπηρεσία frr ξανά με `“service frr start”`.
- 7) Κλείνω το εικονικό μηχάνημα με την εντολή `poweroff` και από τη διαδρομή *File* → *Export Appliance...* στο Virtual Box δημιουργώ ένα αρχείο `frr.ova`.
- 8) Αποθηκεύω το αρχείο `frr.ova` για να μπορώ να δημιουργώ στη συνέχεια εικονικά μηχανήματα και δρομολογητές.

Άσκηση 1

1.1) Επειδή η λειτουργία αυτόματης απόδοσης διευθύνσεων IPv6 είναι απενεργοποιημένη, και στα δύο PCs προσθέτω στο αρχείο `/etc/rc.conf` τη γραμμή `ifconfig_em0_ipv6="inet6 accept_rtadv"` για ενεργοποίηση της αποδοχής μηνυμάτων Router Advertisement στη διεπαφή `em0` εκτελώντας την εντολή `sysrc -f /etc/rc.conf ifconfig_em0_ipv6="inet6 accept_rtadv"`. Επιβεβαιώνω την αλλαγή αυτή με `cat /etc/rc.conf`.

1.2) Επανεκκινώ την υπηρεσία δικτύου `netif` με την εντολή `service netif restart`.

1.3) Με `ifconfig em0 inet6` βλέπω ότι στην διεπαφή `em0` του PC1 έχει αποδοθεί η διεύθυνση IPv6 `fe80::a00:27ff:fe53:8014`.

1.4) Με `ifconfig em0 inet6` βλέπω ότι στην διεπαφή `em0` του PC2 έχει αποδοθεί η διεύθυνση IPv6 `fe80::a00:27ff:fe3c:2b08`.

1.5) Οι διευθύνσεις αυτές είναι τύπου `link-local`, όπως και όλες οι διευθύνσεις που έχουν πρόθεμα `fe80::/64`. Παράγονται από τη διεύθυνση MAC της κάρτας δικτύου προσθέτοντάς της `fffe` στο μέσον της, αντιστρέφοντας το 7^ο bit (αρίθμηση από 1) και, τέλος, προσθέτοντας το πρόθεμα `fe80`.

1.6) Σε ένα από τα PCs εμφανίζω τον πίνακα δρομολόγησης με την εντολή `netstat -r6`. Εμφανίζονται σε αυτόν 9 εγγραφές.

```
root@R0:~ # netstat -r6
Routing tables

Internet6:
Destination      Gateway           Flags             Netif  Expire
:::/96            localhost         UGRS              lo0
localhost         link#3            UH                lo0
::ffff:0.0.0.0/96 localhost         UGRS              lo0
fe80::/10         localhost         UGRS              lo0
fe80::%em0/64     link#1            U                 em0
fe80::a00:27ff:fe5 link#1            UHS               lo0
fe80::%lo0/64     link#3            U                 lo0
fe80::1%lo0       link#3            UHS               lo0
ff02::/16         localhost         UGRS              lo0
root@R0:~ #
```

1.7) Στον πίνακα δρομολόγησης η στήλη `Netif` υποδεικνύει τη διεπαφή εξόδου των πακέτων για τον δεδομένο προορισμό. Τη διεπαφή `em0` αφορά μία από τις προηγούμενες εγγραφές.

1.8) Οι εγγραφές με πρόθεμα δικτύου `fe80::/64` είναι δύο και φαίνονται παραπάνω. Βλέπω ότι η μία αντιστοιχίζεται στη διεπαφή `em0` και η άλλη στην `lo0`.

1.9) Από το PC1 κάνω ping6 στη διεύθυνση ::1 με την εντολή “ping6 ::1”. Βλέπω ότι απαντά το ίδιο το PC1, καθώς η διεύθυνση ::1 είναι η loopback του.

1.10) Στο PC1, εκτελώντας “ping6 fe80::a00:27ff:fe53:8014”, βλέπω ότι δεν μπορώ να κάνω ping6 στη link-local διεύθυνση αυτού. Για να επιτύχει η προσπάθεια αυτή θα πρέπει να προσθέσω στο τέλος “%em0”.

1.11) Από το PC1 κάνω ping6 στην link-local διεύθυνση του PC2 με την εντολή “ping6 fe80::a00:27ff:fe3c:2b08” και βλέπω ότι αυτό αποτυγχάνει. Για να επιτύχει η προσπάθεια αυτή θα πρέπει να προσθέσω στο τέλος “%em0”.

1.12) Από το PC1 εκτελώ την εντολή “ping6 ff01::1%em0” και απαντά το PC1.

1.13) Από το PC1 εκτελώ την εντολή “ping6 ff02::1%em0”. Παρατηρώ ότι και τα δύο PCs απαντούν στο ping6 αυτό, καθώς η διεύθυνση στην οποία γίνεται το ping6 είναι διεύθυνση που ακούν όλοι οι κόμβοι της ζεύξης.

1.14) Ορίζω στη διεπαφή του PC1 στο LAN1 τη στατική διεύθυνση fd00:1::2/64 με την εντολή “ifconfig em0 inet6 fd00:1::2/64”.

1.15) Ορίζω στη διεπαφή του PC2 στο LAN1 τη στατική διεύθυνση fd00:1::3/64 με την εντολή “ifconfig em0 inet6 fd00:1::3/64”.

1.16) Οι διευθύνσεις των 2 προηγούμενων ερωτημάτων είναι μοναδικές τοπικές διευθύνσεις (Unique Local Addresses – ULA). Αντίστοιχες αυτών στο IPv4 είναι οι 192.168.1.2 και 192.168.1.3.

1.17) Πλέον σε κάθε διεπαφή em0 των PCs υπάρχουν 2 διευθύνσεις IPv6.

1.18) Εμφανίζω ξανά τον πίνακα δρομολόγησης μόνο για το IPv6 με την εντολή “netstat -r6”. Βλέπω ότι προστέθηκαν 2 νέες εγγραφές: fd00:1::/64 και fd00:1::2.

```
root@PC1:~ # netstat -r6
Routing tables

Internet6:
Destination      Gateway          Flags    Netif  Expire
::/96             localhost       UGRS     lo0
localhost        link#3          UH       lo0
::ffff:0.0.0.0/96 localhost       UGRS     lo0
fd00:1::/64       link#1          U        em0
fd00:1::2         link#1          UHS      lo0
fe80::/10         localhost       UGRS     lo0
fe80::%em0/64     link#1          U        em0
fe80::a00:27ff:fe5 link#1          UHS      lo0
fe80::%lo0/64     link#3          U        lo0
fe80::1%lo0       link#3          UHS      lo0
ff02::/16         localhost       UGRS     lo0
root@PC1:~ #
```

1.19) Για να μπορώ να χρησιμοποιώ τα ονόματα των μηχανημάτων αντί των IPv6 διευθύνσεων τους στις διάφορες δικτυακές εντολές θα πρέπει να προσθέσω δύο γραμμές, “fd00:1::2 PC1 PC1.my.domain” και “fd00:1::3 PC2 PC2.my.domain”, στο αρχείο /etc/hosts και των δύο PCs.

1.20) Ναι, μετά από την παραπάνω αλλαγή μπορώ να κάνω ping6 από το PC1 στο PC2 με το όνομά του.

1.21) Στο PC1 εμφανίζω τον πίνακα ARP με την εντολή “arp -a”. Παρατηρώ ότι αυτός είναι κενός.

1.22) Εμφανίζω τη βοήθεια της εντολής ndp με την εντολή “man ndp” και μελετώ τη χρήση της.

1.23) Για να εμφανίσω τον πίνακα γειτόνων (neighbor cache) του PC1 εκτελώ την εντολή “ndp -a”.

1.24) Στον πίνακα αυτό βλέπω 4 εγγραφές από τις οποίες οι δύο, που αφορούν το PC1, έχουν κατάσταση ‘R’ (Reachable) και οι άλλες δύο, που αφορούν το PC2, έχουν κατάσταση ‘S’ (Stale).

```
root@PC1:~ # ndp -a
Neighbor          Linklayer Address  Netif  Expire      S  Flags
fe80::a00:27ff:fe3c:2b08%em0 08:00:27:3c:2b:08  em0  23h30m48s  S
PC1                08:00:27:53:80:14  em0  permanent  R
PC2                08:00:27:3c:2b:08  em0  23h51m39s  S
fe80::a00:27ff:fe53:8014%em0 08:00:27:53:80:14  em0  permanent  R
root@PC1:~ #
```

1.25) Εμφανίζω τη λίστα προθεμάτων IPv6 στο PC2 με την εντολή “ndp -p”. Τα προθέματα για τα οποία υπάρχουν εγγραφές είναι τα fd00:1::/64, fe80::%em0/64 και fe80::%lo0/64. Η διάρκεια ζωής των εγγραφών αυτών είναι άπειρη.

```
root@PC1:~ # ndp -p
fd00:1::/64 if=em0
flags=L0 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
root@PC1:~ #
```

1.26) Από τα παραπάνω προθέματα μπορούν να χρησιμοποιηθούν από τον μηχανισμό αυτόματης απόδοσης διευθύνσεων (SLAAC) είναι τα δύο τελευταία, αυτά δηλαδή που έχουν ενεργοποιημένη τη σημαία ‘A’.

flags	The status of the prefix, expressed by a combination of the following letters:
A	This prefix can be used for stateless address autoconfiguration.
L, 0	This prefix can be used for on-link determination; that is, it can be used to determine whether a given destination address is on-link.
D	There are no reachable routers advertising this prefix.

1.27) Καθαρίζω τον πίνακα γειτόνων σε αμφότερα τα PC με την εντολή “ndp -c”.

1.28) Στο PC2 ξεκινώ μια καταγραφή πακέτων σε χωριστό παράθυρο (Alt+Fi) με εμφάνιση λεπτομερειών και απενεργοποιημένη την επίλυση ονομάτων και διευθύνσεων με την εντολή “tcpdump -vvv -n”.

1.29) Στο PC1 εκτελώ την εντολή “ping6 -c 1 PC2”. Σταματώ την καταγραφή στο PC2. Παρατηρώ ότι στάλθηκαν 6 πακέτα IPv6.

1.30) Τα πακέτα IPv6 αυτά μεταφέρουν μηνύματα ICMP6 των οποίων η τιμή του πεδίου Next header της επικεφαλίδας που τα προσδιορίζει είναι επίσης ICMPv6 (58).

1.31) Παρακάτω φαίνεται ένα διάγραμμα που δείχνει τη σειρά αποστολής και τον τύπο των μηνυμάτων που κατέγραψα προηγουμένως.

PC1	Φορά	Μήνυμα	Φορά	PC2
	→	Neighbor solicitation		
		Neighbor advertisement	←	
	→	Echo request		
		Echo reply	←	
	→	Neighbor solicitation		
		Neighbor advertisement	←	

1.32) Η διεύθυνση προορισμός του πρώτου πακέτου NS (neighbor solicitation) που κατέγραψα είναι ff02::1:ff00:3 και είναι διεύθυνση πολλαπλής διανομής (multicast) Solicited Node. Η διεύθυνση αυτή προκύπτει από τα τελευταία 24 bit της διεύθυνσης unicast προσθέτοντας σε αυτά το πρόθεμα ff02:0:0:0:0:1:ff00:0/104.

1.33) Η διεύθυνση προορισμός του δεύτερου πακέτου NS (neighbor solicitation) που κατέγραψα είναι fd00:1::2 και είναι η διεύθυνση unicast του PC1.

1.34) Εμφανίζοντας ξανά τον πίνακα γειτόνων του PC2 με την εντολή “ndp -a” βλέπω ότι η κατάσταση της εγγραφής για το PC1 είναι ‘S’ (Stale) και έχει διάρκεια ζωής 23 ώρες, 41 λεπτά και 5 δευτερόλεπτα.

1.35) Ξεκινώ ένα ping6 από το PC1 στο PC2 με “ping6 PC2” και το αφήνω να τρέχει. Παρατηρώ διαδοχικά αρκετές φορές για περίπου 1 λεπτό την κατάσταση της εγγραφής για το PC1 στον πίνακα γειτόνων του PC2. Παρατηρώ ότι η εγγραφή αυτή έχει κατάσταση ‘R’ (Reachable) για 23 δευτερόλεπτα και για 5 δευτερόλεπτα παίρνει την κατάσταση ‘S’ (Stale), και, έτσι, αλλάζει συνεχώς καταστάσεις.

1.36) Η διάρκεια της κατάστασης ‘R’ (Reachable) είναι 23 secs.

1.37) Η διάρκεια της κατάστασης ‘S’ (Stale) είναι 5 secs.

1.38) Σταματώ το ping6 και συνεχίζω να παρατηρώ διαδοχικά αρκετές φορές για περίπου άλλο 1 λεπτό την κατάσταση της εγγραφής για το PC1 στον πίνακα γειτόνων του PC2. Παρατηρώ ότι η κατάσταση της εγγραφής

αυτής συνεχίζει να αλλάζει όπως στο προηγούμενο ερώτημα, αλλά μόλις αλλάξει κατάσταση από Reachable σε Stale, δεν αλλάζει πλέον συνεχώς κατάσταση.

1.39) Ξεκινώ και πάλι ένα ping6 από το PC1 στο PC2 και το αφήνω να τρέχει. Στο PC2 ξεκινώ μια καταγραφή πακέτων με απενεργοποιημένη την επίλυση ονομάτων και διευθύνσεων με την εντολή “tcpdump -vnn -n”. Ναι, παρατηρώ παραγωγή και άλλων πακέτων πλην των ICMPv6 Echo {Request, Reply}, τα Neighbor {Solicitation, Advertisement}. Αυτά παράγονται κάθε περίπου 20 δευτερόλεπτα και έχουν σκοπό να επιβεβαιώνεται η προσβασιμότητα μεταξύ των.

Άσκηση 2

2.1) Προσθέτω στο αρχείο εκκίνησης /etc/rc.conf των R{1,2} την εντολή `ipv6_gateway_enable="YES"`, ώστε να ενεργοποιηθεί η προώθηση πακέτων IPv6, εκτελώντας την εντολή `sysrc -f /etc/rc.conf ipv6_gateway_enable="YES"`. Τέλος, επανεκκινώ την υπηρεσία routing με την εντολή `service routing restart`.

2.2) Διαγράφω τη διεύθυνση `fd00:1::3/64` στο PC2 με την εντολή `ifconfig em0 inet6 fd00:1::3/64 delete` και ορίζω στατική διεύθυνση `fd00:2::2/64` με την εντολή `ifconfig em0 inet6 fd00:2::2/64`.

2.3) Στον R1 μέσω vtysh ορίζω τη διεύθυνση `fd00:1::1/64` για τη διεπαφή του στο LAN1 με την εξής ακολουθία εντολών:

```
vtys  
configure terminal  
interface em0  
ipv6 address fd00:1::1/64
```

2.4) Στον R1 μέσω vtysh ορίζω τη διεύθυνση `fd00:3::1/126` για τη διεπαφή του στο WAN1 με την εξής ακολουθία εντολών (από global config. mode):

```
interface em1  
ipv6 address fd00:3::1/126
```

2.5) Στον R2 μέσω vtysh ορίζω τη διεύθυνση `fd00:2::1/64` για τη διεπαφή του στο LAN2 με την εξής ακολουθία εντολών:

```
vtys  
configure terminal  
interface em0  
ipv6 address fd00:2::1/64
```

2.6) Στον R2 μέσω vtysh ορίζω τη διεύθυνση fd00:3::2/126 για τη διεπαφή του στο WAN1 με την εξής ακολουθία εντολών (από global config. mode):
`interface em1`
`ipv6 address fd00:3::2/126`

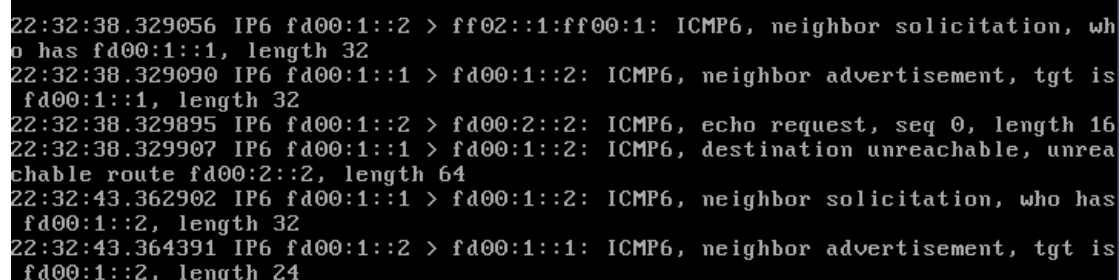
2.7) Ορίζω τη σωστή προεπιλεγμένη πύλη στο PC1 με την εντολή “route -6 add default fd00:1::1”.

2.8) Ορίζω τη σωστή προεπιλεγμένη πύλη στο PC2 με την εντολή “route -6 add default fd00:2::1”.

2.9) Ενεργοποιώ μια καταγραφή πακέτων στη διεπαφή του R1 στο LAN1 με την εντολή “tcpdump -i em0”.

2.10) Στο PC1 καθαρίζω τον πίνακα γειτόνων με “ndp -c” και εκτελώ την εντολή ping6 στέλνοντας ακριβώς ένα πακέτο προς το PC2, δηλαδή με “ping6 -c 1 fd00:2::2”. Το ping αποτυγχάνει, κάτι το οποίο περιμένω, καθώς ο R1 δεν έχει κάποια εγγραφή στον πίνακα δρομολόγησής του για την διεύθυνση του PC2.

2.11) Από την καταγραφή στον R1 βλέπω ότι παράγονται μηνύματα Neighbor {Solicitation, Advertisement}, ICMP6 Echo request και ICMP6 destination unreachable.



```
22:32:38.329056 IP6 fd00:1::2 > ff02::1:ff00:1: ICMP6, neighbor solicitation, who has fd00:1::1, length 32
22:32:38.329090 IP6 fd00:1::1 > fd00:1::2: ICMP6, neighbor advertisement, tgt is fd00:1::1, length 32
22:32:38.329895 IP6 fd00:1::2 > fd00:2::2: ICMP6, echo request, seq 0, length 16
22:32:38.329907 IP6 fd00:1::1 > fd00:1::2: ICMP6, destination unreachable, unreachable route fd00:2::2, length 64
22:32:43.362902 IP6 fd00:1::1 > fd00:1::2: ICMP6, neighbor solicitation, who has fd00:1::2, length 32
22:32:43.364391 IP6 fd00:1::2 > fd00:1::1: ICMP6, neighbor advertisement, tgt is fd00:1::2, length 24
```

Στην παραπάνω εικόνα φαίνονται οι διευθύνσεις προορισμού του καθενός πακέτου που καταγράφηκε.

2.12) Στον R1 μέσω vtysh προσθέτω την κατάλληλη στατική εγγραφή για το LAN2 με την εντολή “ipv6 route fd00:2::/64 fd00:3::2” από global config. mode:

2.13) Από το PC1 δεν μπορώ να κάνω ping6 στο PC2. Παρόλο που το μήνυμα ICMP6 echo request φτάνει στο PC2, αυτό με τη σειρά του δεν μπορεί να απαντήσει, καθώς ο R2 δεν γνωρίζει για πώς να δρομολογήσει πακέτα στο PC1.

2.14) Στον R2 μέσω vtysh προσθέτω στατική εγγραφή για το LAN1 με την εντολή “ipv6 route fd00:1::/64 fd00:3::1” από global config. mode.

2.15) Ναι, πλέον μπορώ να κάνω ping6 από το PC1 στο PC2.

2.16) Επειδή στο FRR η λειτουργία διαφήμισης δρομολογητή είναι απενεργοποιημένη για όλες τις διεπαφές, την ενεργοποιώ στη διεπαφή em0 του R1 με την ακολουθία εντολών (από global config. mode):

```
interface em0  
no ipv6 nd suppress-ra
```

2.17) Στον R1 για τη διεπαφή του στο LAN1 ορίζω ως πρόθεμα δικτύου για τη διαδικασία ανεύρεσης γειτόνων το fd00:1::/64 με την ακολουθία εντολών (από global config. mode):

```
interface em0  
ipv6 nd prefix fd00:1::/64
```

2.18) Στον R2 ενεργοποιώ τη διαφήμιση δρομολογητή για τη διεπαφή στο LAN2 με την ακολουθία εντολών (από global config. mode):

```
interface em0  
no ipv6 nd suppress-ra
```

2.19) Στον R2 για τη διεπαφή του στο LAN2 ορίζω ως πρόθεμα δικτύου για τη διαδικασία ανεύρεσης γειτόνων το fd00:2::/64 με την ακολουθία εντολών (από global config. mode):

```
interface em0  
ipv6 nd prefix fd00:2::/64.
```

2.20) Στο PC1 διαγράφω την προκαθορισμένη διαδρομή με την εντολή “route -6 delete default”.

2.21) Ξεκινώ μια καταγραφή πακέτων ICMPv6 στον R1 στη διεπαφή του στο LAN1, χωρίς επίλυση ονομάτων και εμφανίζοντας τις επικεφαλίδες Ethernet με την εντολή “tcpdump -i em0 -n -e”.

2.22) Επανεκκινώ την υπηρεσία δικτύου στο PC1 με την εντολή “service netif restart”.

2.23) Τα μηνύματα που ανταλλάσσονται κατά τις διαδικασίες αυτόματης απόδοσης διεύθυνσης (SLAAC) και ανίχνευσης ταυτόσημων διευθύνσεων (DAD) είναι Router {Solicitation, Advertisement} και Neighbor solicitation.

```
root@R0:~ # tcpdump -i em0 -n -e
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on em0, link-type EN10MB (Ethernet), capture size 262144 bytes
22:53:52.979445 08:00:27:53:80:14 > 33:33:00:00:00:02, ethertype IPv6 (0x86dd),
length 70: fe80::a00:27ff:fe53:8014 > ff02::2: ICMP6, router solicitation, length 16
22:53:52.979787 08:00:27:ab:a7:2b > 33:33:00:00:00:01, ethertype IPv6 (0x86dd),
length 110: fe80::a00:27ff:feab:a72b > ff02::1: ICMP6, router advertisement, length 56
22:53:53.544162 08:00:27:53:80:14 > 33:33:ff:53:80:14, ethertype IPv6 (0x86dd),
length 86: :: > ff02::1:ff53:8014: ICMP6, neighbor solicitation, who has fd00:1:a00:27ff:fe53:8014, length 32
```

2.24) Το PC1 παράγει το μήνυμα NS με σκοπό να μάθε αν κάποιο άλλο μηχανήμα στο υποδίκτυο έχει την IPv6 διεύθυνσή του.

2.25) Στο μήνυμα NS χρησιμοποιείται διεύθυνση πηγής η ::, επειδή δεν έχει λάβει ακόμα διεύθυνση.

2.26) Στο μήνυμα RS χρησιμοποιείται διεύθυνση πηγής η fe80::a00:27ff:fe53:8014, δηλαδή η διεύθυνση που το PC1 έχει.

2.27) Οι διευθύνσεις προορισμού των μηνυμάτων NS, RS και RA που στάλθηκαν είναι, αντίστοιχα, ff02::1:ff53:8014, ff02::2: και ff02::1:.

Το RS έχει ως διεύθυνση προορισμού την διεύθυνση πολλαπλής διανομής (multicast) στο ff02::2:/64, όπως και το RA στο ff02:1::/64. Το NS έχει ως διεύθυνση προορισμού τη διεύθυνση πολλαπλής διανομής Solicited-Node.

2.28) Οι διευθύνσεις MAC προορισμού των πλαισίων Ethernet που τα μεταφέρουν, όπως φαίνεται παραπάνω, είναι τα τελευταία 32 bits των IPv6 διευθύνσεων με πρόθεμα 33:33:.

2.29) Εμφανίζω πάλι τη λίστα προθεμάτων στο PC1 με την εντολή “ndp -p”.

```
root@PC1:~ # ndp -p
fd00:1::/64 if=em0
flags=LA0 vlttime=2592000, pltime=604800, expire=29d23h54m22s, ref=1
  advertised by
    fe80::a00:27ff:feab:a72b%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
root@PC1:~ #
```

Η διαφορά που παρατηρείται σε σχέση με το ερώτημα 1.25 είναι ότι στο πρόθεμα fd00:1::/64 οι τιμές των vlttime και pltime είναι πλέον πεπερασμένες και αναφέρεται ότι αυτό έχει διαφημιστεί από μία διεύθυνση.

2.30) Το PC1 έχει λάβει αυτόματα μέσω του SLAAC την διεύθυνση fe80::1 στη διεπαφή lo0 και την fe80::a00:27ff:fe53:8014 στην em0.

2.31) Εμφανίζω τον πίνακα δρομολόγησης για IPv6 στο PC1 με την εντολή “netstat -r6”. Παρατηρώ ότι έχει προστεθεί εγγραφή για προκαθορισμένη διαδρομή. Η προκαθορισμένη πύλη προέκυψε από το router advertisement που έστειλε ο R1.

2.32) Για να κάνω ping6 στο PC1 από το PC2 μπορώ να χρησιμοποιήσω τη διεύθυνση fd00:1::a00:27ff:fe53:8014. Από τον R1 μπορώ να χρησιμοποιήσω επίσης μόνο τη διεύθυνση fd00:1::a00:27ff:fe53:8014.

Άσκηση 3

3.1) Μέσω vtysh διαγράφω τις στατικές διαδρομές στους R{1,2} με την εντολή από global config. mode “no ipv6 route fd00:{2,1}::/64 fd00:3::{2,1}”, αντίστοιχα.

3.2) Στους R{1,2} εισέρχομαι σε RIPng router configuration mode με την εντολή “router ripng”. Ενεργοποιώ το RIPng στις διεπαφές των δρομολογητών στο WAN και τα αντίστοιχα LAN με τις εντολές “network em0” και “network em1”, και περιμένω λίγο.

3.3) Εμφανίζω στον R1 τον πίνακα δρομολόγησης IPv6 για το RIPng με την εντολή “do show ipv6 ripng” από global config. mode.

```
R0(config)# do show ipv6 ripng
Codes: R - RIPng, C - connected, S - Static, O - OSPF, B - BGP
Sub-codes:
      (n) - normal, (s) - static, (d) - default, (r) - redistribute,
      (i) - interface, (a/S) - aggregated/Suppressed

  Network      Next Hop          Via      Metric Tag Time
C(i) fd00:1::/64      ::                self        1    0
R(n) fd00:2::/64      fe80::a00:27ff:fe85:3f5e  em1        2    0  02:26
C(i) fd00:3::/64      ::                self        1    0
R(n) fd00:3::/126     fe80::a00:27ff:fe85:3f5e  em1        2    0  02:26
R0(config)#
```

Σε αυτόν φαίνονται να υπάρχουν 4 εγγραφές.

3.4) Η διεύθυνση του επόμενου κόμβου για το δίκτυο fd00:2::/64 είναι η fe80::a00:27ff:fe85:3f5e, η οποία είναι τύπου link-local.

3.5) Ναι, μπορώ να ξεκινήσω ping6 από το PC1 στο PC2 με “ping6 fd00:2::2”.

3.6) Ξεκινώ καταγραφή πακέτων IPv6 με το tcpdump στη διεπαφή του R1 στο WAN1, εμφανίζοντας λεπτομερείς πληροφορίες για τα πακέτα χωρίς επίλυση ονομάτων, με την εντολή “tcpdump -i em1 -vnn -n” και περιμένω τουλάχιστον ένα λεπτό.

3.7) Παρατηρώ ripng-resp πακέτα με προορισμό τη διεύθυνση ff02::9, η οποία είναι διεύθυνση πολλαπλής διανομής (multicast) για το πρωτόκολλο rip.

3.8) Το Hop Limit των πακέτων IPv6 που τα μεταφέρουν έχει τιμή 255, δηλαδή την μέγιστη που μπορεί να πάρει. Τίθεται αυτή η τιμή επειδή, κατ’ αρχήν, είναι η προεπιλεγμένη τιμή, και, επίσης, επειδή με την τιμή αυτή επιβεβαιώνεται ότι το πακέτο αυτό αρχικά δεν έχει προωθηθεί από άλλον δρομολογητή.

3.9) Το RIPng χρησιμοποιεί ως πρωτόκολλο μεταφοράς το UDP με θύρα την 521. Το πρωτόκολλο RIP χρησιμοποιεί και αυτό UDP, αλλά με θύρα την 520.

3.10) Απενεργοποιώ το RIPng στους R{1,2} με την παρακάτω ακολουθία εντολών από global config. mode:

```
router ripng
no network em0
no network em1
exit
no router ripng
```

3.11) Αποθηκεύω την παραμετροποίηση του FRR με την εντολή “write file” από privileged EXEC mode.

3.12) Επανεκκινώ την υπηρεσία FRR με “service frr restart”.

3.13) Στους R{1,2} εισέρχομαι σε OSPF6 router configuration mode με την εντολή “router ospf6”. Ορίζω router-id 1.1.1.1 και 2.2.2.2 στους R{1,2}, αντίστοιχα, με την εντολή “ospf6 router-id {1.1.1.1 , 2.2.2.2}”, αντίστοιχα.

3.14) Ενεργοποιώ το OSPF6 στον R1 δηλώνοντας τις διεπαφές του στα LAN1 και WAN1 στην περιοχή 0.0.0.0 με την εντολή “interface emX area 0.0.0.0” από ospf6 router config. mode.

3.15) Παρόμοια, στον R2 για τις διεπαφές του στα LAN2 και WAN1 στην περιοχή 0.0.0.0 με τις εντολές “`interface emX area 0.0.0.0`”, και περιμένω περίπου ένα λεπτό.

3.16) Εμφανίζω στον R2 τον πίνακα δρομολόγησης IPv6 για το OSPF6 με την εντολή “`do show ipv6 route ospf6`”.

```
R0(config-ospf6)# 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:fe81:a38b, em1, weight 1, 00:02:43
O   fd00:2::/64 [110/100] is directly connected, em0, weight 1, 00:02:53
O>* fd00:3::/64 [110/100] is directly connected, em1, weight 1, 00:02:47
O   fd00:3::/126 [110/100] is directly connected, em1, weight 1, 00:02:47
R0(config-ospf6)#
```

Βλέπω ότι εμφανίζονται 4 εγγραφές. Τα κόστη, 200 και 100, προέκυψαν έτσι επειδή κάθε ζεύξη υπολογίζεται ότι έχει κόστος 100 από προεπιλογή (by default).

3.17) Η διεύθυνση του επόμενου κόμβου για το δίκτυο fd00:1::/64 είναι η fe80::a00:27ff:fe81:a38b, και είναι τύπου link-local.

3.18) Ξεκινώ καταγραφή πακέτων IPv6 με το tcpdump στη διεπαφή του R2 στο WAN1, εμφανίζοντας λεπτομερείς πληροφορίες για τα πακέτα χωρίς επίλυση ονομάτων, με την εντολή “`tcpdump -i em1 -vvvv -n`”, και περιμένω τουλάχιστον ένα λεπτό.

3.19) Στην καταγραφή παρατηρώ πακέτα OSPFv4 τύπου Hello, με διεύθυνση προορισμού την ff02::5:.

3.20) Το Hop Limit των πακέτων IPv6 που τα μεταφέρουν έχει τιμή 1.

3.21) Το OSPFv3 χρησιμοποιεί αριθμό πρωτοκόλλου (next header) ανωτέρου στρώματος το 89, όπως ακριβώς και το OSPFv2.

3.22) Ναι, μπορώ να κάνω ping6 από το PC2 στο PC1 (με “`ping6 fd00:2::a00:27ff:fe3c:2b08`”).

3.23) Απενεργοποιώ το OSPF6 στους R{1,2} με την εντολή “`no router ospf6`” από global configuration mode.

3.24) Επανεκκινώ την υπηρεσία FRR με “`service frr restart`”.

3.25) Στον R1 ορίζω ως router-id 1.1.1.1 με “`router-id 1.1.1.1`” (global configuration mode) και εισέρχομαι σε router configuration mode για το

BGP δηλώνοντας αυτόνομο σύστημα AS 65010 με την εντολή “**router bgp 65010**” από global configuration mode.

3.26) Στην τυπική του χρήση για δρομολόγηση το FRR απαιτεί την εφαρμογή φίλτρων στις συνόδους eBGP για συμβατότητα με το RFC 8212. Χωρίς φίλτρο εισόδου καμία διαδρομή δεν γίνεται δεκτή και χωρίς φίλτρο εξόδου δεν ανακοινώνονται διαδρομές. Για τη συνέχεια ακυρώνω την απαίτηση αυτή με την εντολή “**no bgp ebgp-requires-policy**” (από το bgp router configuration mode που μόλις εισήλθα).

3.27) Στο BGP η σχέση γειτονίας από προεπιλογή (default) ενεργοποιείται για την οικογένεια διευθύνσεων IPv4. Στη συγκεκριμένη περίπτωση, όμως, δεν έχουμε ορίσει διευθύνσεις IPv4 στις διεπαφές του δρομολογητή, οπότε πρέπει να απενεργοποιηθεί η χρήση της. Αυτό, όμως, συνεπάγεται ότι θα πρέπει να ενεργοποιείται ρητά για την εκάστοτε χρησιμοποιούμενη οικογένεια διευθύνσεων με κάθε γείτονα. Δηλώνω ότι δεν θέλω να χρησιμοποιήσω την οικογένεια διευθύνσεων IPv4 unicast για τη δημιουργία γειτονίας με την εντολή “**no bgp default ipv4-unicast**”.

3.28) Δηλώνω τον R2 ως γείτονα στο αυτόνομο σύστημα AS 65020 με την εντολή “**neighbor fd00:3::2 remote-as 65020**”.

3.29) Εισέρχομαι στο υπο-μενού του R1 για την οικογένεια διευθύνσεων IPv6 προκειμένου να ορίσω τα διαφημιζόμενα δίκτυα και ενεργοποιώ τη σχέση γειτονίας με την εντολή “**address-family ipv6 unicast**”.

3.30) Διαφημίζω το δίκτυο του LAN1 με την εντολή “**network fd00:1::/64**”.

3.31) Ενεργοποιώ για IPv6 τη σχέση γειτονίας με τον R2 με την εντολή “**neighbor fd00:3::2 activate**” και εξέρχομαι με “**exit**”.

3.32) Επαναλαμβάνω τα προηγούμενα για τον R2 με router-id 2.2.2.2 στο αυτόνομο σύστημα AS 65020, απενεργοποιώντας την πολιτική φίλτρων eBGP και την οικογένεια διευθύνσεων IPv4 και ορίζοντας τον R1 ως γείτονα IPv6 στο αυτόνομο σύστημα AS 65010. Στην οικογένεια διευθύνσεων IPv6, διαφημίζω το δίκτυο LAN2 και ενεργοποιώ για IPv6 τη σχέση γειτονίας με τον R1. Όλα αυτά τα κάνω με την παρακάτω ακολουθία εντολών από global configuration mode:

```
router-id 2.2.2.2
router bgp 65020
no bgp ebgp-requires-policy
no bgp default ipv4-unicast
neighbor fd00:3::1 remote-as 65010
address-family ipv6
network fd00:2::/64
```

```
neighbor fd00:3::1 activate
exit
```

3.33) Περιμένω περίπου δύο λεπτά και εμφανίζω στον R1 τον πίνακα δρομολόγησης IPv6 για το BGP με “do show ipv6 route bgp”.

```
R0(config-router)# 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 - VNC, U - VNC-Direct, A - Babel, D - SHARP, F - PBR,
        f - OpenFabric,
        > - selected route, * - FIB route, q - queued, r - rejected, b - backup
B>* fd00:2::/64 [20/0] via fe80::a00:27ff:fe85:3f5e, em1, weight 1, 00:02:44
R0(config-router)#
```

Βλέπω μία δυναμική εγγραφή, αυτή για το δίκτυο του LAN2.

3.34) Η διεύθυνση του επόμενου κόμβου για το δίκτυο fd00:2::/64 είναι fe80::a00:27ff:fe85:3f5e και είναι τύπου link-local.

3.35) Ξεκινώ καταγραφή πακέτων με το tcpdump στη διεπαφή του R1 στο WAN1, εμφανίζοντας λεπτομερείς πληροφορίες για τα πακέτα, χωρίς επίλυση ονομάτων, χωρίς να συλλαμβάνω μηνύματα ICMPv6, με την εντολή “tcpdump -i em1 -nnnn -n not icmp6”, και περιμένω τουλάχιστον ένα λεπτό.

3.36) Παρατηρώ μηνύματα BGP Keepalive Message. Το πρωτόκολλο μεταφοράς που χρησιμοποιείται είναι το TCP στη θύρα 179, όπως ακριβώς και το BGP στο IPv4.

3.37) Το Hop Limit των πακέτων IPv6 που τα μεταφέρουν έχει τιμή 1.

3.38) Ναι, μπορώ να κάνω ping6 από το PC1 στο PC2 με “ping6 fd00:2::2”.

3.39) Αφού επανεκκινήσω το PC1 με την εντολή “reboot”, εισέρχομαι στο περιβάλλον του με vtysh και ορίζω στατική διεύθυνση fd00:1::2/64 για τη διεπαφή του στο LAN1 με “interface em0” και “ipv6 address fd00:1::2/64”, από global configuration mode.

3.40) Στο PC1 ορίζω ως router-id 1.1.0.0 με την εντολή “router-id 1.1.0.0” και εισέρχομαι σε router configuration mode για το BGP δηλώνοντας αυτόνομο σύστημα AS 65010 με την εντολή “router bgp 65010”.

3.41) Δηλώνω ότι δεν θα χρησιμοποιήσω την οικογένεια διευθύνσεων IPv4 unicast για τη δημιουργία γειτονίας με την εντολή “no bgp default ipv4-unicast”.

3.42) Δηλώνω (στο PC1) τον R1 ως γείτονα στο αυτόνομο σύστημα AS 65010 καθορίζοντας έτσι σύνοδο τύπου iBGP με την εντολή “neighbor fd00:1::1 remote-as 65010” από bgp router configuration mode.

3.43) Εισέρχομαι (στο PC1) στο υπο-μενού για την οικογένεια διευθύνσεων IPv6 με “address-family ipv6”, ενεργοποιώ για IPv6 τη σχέση γειτονίας με τον R1 με “neighbor fd00:1::1 activate” και εξέρχομαι με “exit”.

3.44) Στον R1 σε router configuration mode για το BGP δηλώνω το PC1 ως γείτονα στο ίδιο αυτόνομο σύστημα με την εντολή “neighbor fd00:1::2 remote-as 65010”.

3.45) Αφού εισέλθω (στον R1) στο υπο-μενού για την οικογένεια διευθύνσεων IPv6 με την εντολή “address-family ipv6”, ενεργοποιώ για IPv6 τη σχέση γειτονίας με το PC1 με “neighbor fd00:1::2 activate”, δηλώνω ότι για τις διαφημίσεις προς το PC1 το επόμενο βήμα είναι ο εαυτός του με την εντολή “neighbor fd00:1::2 next-hop-self”, και εξέρχομαι με “exit”.

3.46) Επιβεβαιώνω ότι έχει εγκατασταθεί σύνοδος iBGP μεταξύ του PC1 και R1 με την εντολή “do show ip bgp neighbors”, είτε από τον R1 είτε από το PC1, βλέποντας το ‘internal link’.

3.47) Εμφανίζω στο PC1 τον πίνακα δρομολόγησης IPv6 για το BGP με “do show ipv6 route bgp”.

```
R0(config-router)# 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 - VNC, 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:feab:a72b, em0, weight 1, 00:04:05
B>* fd00:2::/64 [200/0] via fd00:1::1, em0, weight 1, 00:04:00
R0(config-router)#
```

Εμφανίζονται οι δύο παραπάνω εγγραφές.

3.48) Η διαδρομή προς το δίκτυο fd00:1::/64 δεν είναι επιλεγμένη επειδή, εάν εμφανίσω ολόκληρο τον πίνακα δρομολόγησης με “do show ipv6 route” θα δω ότι υπάρχει (επιλεγμένη) στατική εγγραφή προς το δίκτυο αυτό.

3.49) Η διεύθυνση του επόμενου κόμβου για το δίκτυο fd00:2::/64 είναι η fd00:1::1, και είναι τύπου Unique Local Address (ULA).

3.50) Ναι, μπορώ να κάνω ping6 από το PC2 στο PC1, με “ping6 fd00:1::2”.

Άσκηση 4

4.1) Ορίζω μέσω vtysh του R1 την IPv4 διεύθυνση 192.168.1.1/24 για τη διεπαφή του στο LAN1 με την εξής ακολουθία εντολών:

```
vtysh
configure terminal
interface em0
ip address 192.168.1.1/24
```

4.2) Ορίζω μέσω vtysh του R2 την IPv4 διεύθυνση 192.168.2.1/24 για τη διεπαφή του στο LAN2 με την εξής ακολουθία εντολών:

```
vtysh
configure terminal
interface em0
ip address 192.168.2.1/24
```

4.3) Ορίζω μέσω vtysh του PC1 την IPv4 διεύθυνση 192.168.1.2/24 για τη διεπαφή του στο LAN1 και ως προκαθορισμένη διαδρομή την 192.168.1.1 με την εξής ακολουθία εντολών:

```
vtysh
configure terminal
interface em0
ip address 192.168.1.2/24
exit
ip route 0.0.0.0/0 192.168.1.1
```

4.4) Ορίζω μέσω vtysh του PC2 την IPv4 διεύθυνση 192.168.2.2/24 για τη διεπαφή του στο LAN2 και ως προκαθορισμένη διαδρομή την 192.168.2.1 με την εξής ακολουθία εντολών:

```
vtysh
configure terminal
interface em0
ip address 192.168.2.2/24
exit
ip route 0.0.0.0/0 192.168.2.1
```

4.5) Στο αρχείο εκκίνησης /etc/rc.conf του R1 προσθέτω τις ακόλουθες εντολές:

firewall_enable="YES" για να ενεργοποιηθεί το τείχος προστασίας ipfw,
firewall_nat64_enable="YES" για να ενεργοποιηθεί η ενσωματωμένη λειτουργία NAT64,
firewall_type="open" για ανοικτό τύπο τείχους προστασίας, και
firewall_logif="YES" για να ενεργοποιηθεί η δυνατότητα καταγραφής πακέτων της λειτουργίας NAT64.

Τα παραπάνω εισάγω με τις παρακάτω εντολές, αντίστοιχα:

```
sysrc -f /etc/rc.conf firewall_enable="YES"
sysrc -f /etc/rc.conf firewall_nat64_enable="YES"
```

```
sysrc -f /etc/rc.conf firewall_type="open"  
sysrc -f /etc/rc.conf firewall_logif="YES"
```

4.6) (R1) Εκκινώ την υπηρεσία του τείχους προστασίας με “`kldload ipfw`” και “`service ipfw start`”.

4.7) Με την εντολή “`ipfw list`” εμφανίζω τους κανόνες που περιέχει το τείχος προστασίας του R1.

```
root@R0:~ # 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  
root@R0:~ #
```

Βλέπω ότι εμφανίζονται 12 κανόνες σε αυτό.

4.8) Ναι, μπορώ να κάνω ping6 από το PC1 στο PC2 με “`ping6 fd00:2::2`” (σε διαφορετική περίπτωση, θα είχε γίνει κάποιο λάθος στο `/etc/rc.conf`).

4.9) Δημιουργώ πίνακα `nat64clat` με όνομα `nat64`, ώστε κίνηση με `clat_prefix fd00:3:1::/96` να μεταφράζεται σε `plat_prefix 64:ff9b::/96`, να επιτρέπεται η χρήση ιδιωτικών διευθύνσεων καθώς και η καταγραφή, με την εντολή “`ipfw nat64clat nat64 create clat_prefix fd00:3:1::/96 plat_prefix 64:ff9b::/96 allow_private log`”.

4.10) Προσθέτω στο τείχος προστασίας του R1 κανόνα με αύξοντα αριθμό 2000 ώστε να ωθείται προς μετάφραση στον πίνακα `nat64clat` με όνομα `nat64` η κίνηση IPv4, ανεξάρτητα διεύθυνσης πηγής και με προορισμό εκτός του R1, που λαμβάνεται από τη διεπαφή του στο LAN1, εκτελώντας την εντολή “`ipfw add 2000 nat64clat nat64 ipv4 from any to not me in via em0`”.

4.11) Προσθέτω κανόνα στο τείχος προστασίας του R1 με αύξοντα αριθμό 3000 ώστε να ωθείται προς μετάφραση στον πίνακα `nat64clat` με όνομα `nat64` η κίνηση IPv6 με πηγή το δίκτυο `64:ff9b::/96` και προορισμό το δίκτυο `fd00:3:1::/96`, που λαμβάνεται από τη διεπαφή του στο WAN1, εκτελώντας την εντολή “`ipfw add 3000 nat64clat nat64 ipv6 from 64:ff9b::/96 to fd00:3:1::/96 in via em1`”.

4.12) Μέσω vtysh του R1 προσθέτω διαδρομή προς το δίκτυο 64:ff9b::/96 μέσω του R2 με την εντολή “`ipv6 route 64:ff9b::/96 fd00:3::2`” από global config. mode.

4.13) Επαναλαμβάνω τις ρυθμίσεις της ερώτησης 4.5 στον R2 και εκκινώ την υπηρεσία του τείχους προστασίας με την παρακάτω ακολουθία εντολών:

```
sysrc -f /etc/rc.conf firewall_enable="YES"
sysrc -f /etc/rc.conf firewall_nat64_enable="YES"
sysrc -f /etc/rc.conf firewall_type="open"
sysrc -f /etc/rc.conf firewall_logif="YES"
kldload ipfw
service ipfw start
```

4.14) (R2) Δημιουργώ πίνακα nat64ln με όνομα nat64, ώστε κίνηση με πρόθεμα IPv4 2.2.2.0/24 να μεταφράζεται σε πρόθεμα IPv6 64:ff9b::/96, να επιτρέπεται η χρήση ιδιωτικών διευθύνσεων καθώς και η καταγραφή, εκτελώντας την εντολή “`ipfw nat64ln nat64 create prefix4 2.2.2.0/24 prefix6 64:ff9b::/96 allow_private log`”.

4.15) Προσθέτω κανόνα στο τείχος προστασίας του R2 με αύξοντα αριθμό 2000 ώστε να ωθείται προς μετάφραση στον πίνακα nat64ln με όνομα nat64 η κίνηση IPv6, με πηγή το δίκτυο fd00:3:1::/96 και προορισμό το δίκτυο 64:ff9b::/96, που λαμβάνεται από τη διεπαφή του στο WAN1, εκτελώντας την εντολή “`ipfw add 2000 nat64ln nat64 ipv6 from fd00:3:1::/96 to 64:ff9b::/96 in via em1`”.

4.16) Προσθέτω κανόνα στο τείχος προστασίας του R2 με αύξοντα αριθμό 3000 ώστε να ωθείται προς μετάφραση στον πίνακα nat64ln με όνομα nat64 η κίνηση IPv4, ανεξάρτητα διεύθυνσης πηγής και με προορισμό το δίκτυο 2.2.2.0/24, που λαμβάνεται από τη διεπαφή του στο LAN2, εκτελώντας την εντολή “`ipfw add 3000 nat64ln nat64 ipv4 from any to 2.2.2.0/24 in via em0`”.

4.17) Μέσω vtysh του R2 προσθέτω διαδρομή προς το δίκτυο fd00:3:1::/96 μέσω του R1 εκτελώντας “`ipv6 route fd00:3:1::/96 fd00:3::1`” από global config. mode.

4.18) (R2) Στη συνέχεια, προσθέτω ως προκαθορισμένη διαδρομή IPv4 την 192.168.2.2 εκτελώντας “`ip route 0.0.0.0/0 192.168.2.2`” από global config. mode.

4.19) Ναι, μπορώ να κάνω ping από το PC1 στα R1 και PC2 χρησιμοποιώντας τις IPv4 διευθύνσεις τους, δηλαδή με “`ping 192.168.1.1`” και “`ping 192.168.2.2`”.

4.20) Στο R1 δημιουργώ την ψευδο-διεπαφή ipfwlog0 και ξεκινώ μια καταγραφή σε αυτήν, εκτελώντας τις εντολές “ifconfig ipfwlog0 create” και “tcpdump -i ipfwlog0”.

4.21) Παρομοίως στο R2, εκτελώντας “ifconfig ipfwlog0 create” και “tcpdump -i ipfwlog0”.

4.22) Στο PC1 δίνω την εντολή “ping -c 1 192.168.2.2”. Στις καταγραφές των R{1,2} παρατηρώ πακέτα IPv4 και IPv6, echo request και echo reply.

R1:

```
root@R0:~ # 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
02:52:48.122292 IP 192.168.1.2 > 192.168.2.2: ICMP echo request, id 6660, seq 0,
length 64
02:52:48.122299 IP6 fd00:3:1::c0a8:102 > 64:ff9b::c0a8:202: ICMP6, echo request,
seq 0, length 64
02:52:48.124119 IP6 64:ff9b::c0a8:202 > fd00:3:1::c0a8:102: ICMP6, echo reply, s
eq 0, length 64
02:52:48.124122 IP 192.168.2.2 > 192.168.1.2: ICMP echo reply, id 6660, seq 0, l
ength 64
```

R2:

```
root@R0:~ # 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
02:53:24.061359 IP6 fd00:3:1::c0a8:102 > 64:ff9b::c0a8:202: ICMP6, echo request,
seq 0, length 64
02:53:24.061436 IP 2.2.2.175 > 192.168.2.2: ICMP echo request, id 1024, seq 0, l
ength 64
02:53:24.063047 IP 192.168.2.2 > 2.2.2.175: ICMP echo reply, id 1024, seq 0, l
ength 64
02:53:24.063051 IP6 64:ff9b::c0a8:202 > fd00:3:1::c0a8:102: ICMP6, echo reply, s
eq 0, length 64
```

4.23) Μέσω vtysh του PC2 ορίζω τις 172.17.17.2/24 και 10.0.0.2/24 ως δευτερεύουσες διευθύνσεις IPv4 στη διεπαφή του στο LAN2 με την εξής ακολουθία εντολών:

```
interface em0
ip address 172.17.17.2/24
ip address 10.0.0.2/24
```

4.24) Στο PC1 μπορώ να κάνω ping στις προηγούμενες διευθύνσεις IPv4.

4.25) Στον R2 εμφανίζω την κατάσταση του nat64lsn με “ipfw nat64lsn nat64 show states”.

```
root@R0:~ # ipfw nat64lsn nat64 show states
fd00:3:1::c0a8:102      2.2.2.175      ICMPv6          59      10.0.0.2
root@R0:~ #
```

4.26) Στο PC1 κάνω ping σε δύο από τις IPv4 διευθύνσεις του PC2 και στη συνέχεια ελέγχω την κατάσταση του nat64lsn με την παραπάνω εντολή. Παρατηρώ ότι, αφού κάνω τα pings, προστίθενται μία εγγραφή για κάθε ping στον πίνακα καταστάσεων του nat64lsn, οι οποίες εγγραφές έχουν διάρκεια ζωής περίπου 1 λεπτό.

```
root@R0:~ # ipfw nat64lsn nat64 show states
fd00:3:1::c0a8:102      2.2.2.175      ICMPv6          67      172.17.17.2
fd00:3:1::c0a8:102      2.2.2.175      ICMPv6          65      10.0.0.2
root@R0:~ # ipfw nat64lsn nat64 show states
fd00:3:1::c0a8:102      2.2.2.175      ICMPv6          68      172.17.17.2
fd00:3:1::c0a8:102      2.2.2.175      ICMPv6          66      10.0.0.2
```

Άσκηση 5

5.1) Ενεργοποιώ τον DHCP client στις διεπαφές των εικονικών μηχανημάτων και βεβαιώνομαι ότι έχω πρόσβαση στο Internet.

5.2) Εγκαθιστώ σε αυτά το teredo client κατεβάζοντας το πακέτο miredo με την εντολή “pkg install miredo”.

5.3) Προσθέτω την εντολή miredo_enable=“YES” στο αρχείο /etc/rc.conf, ώστε να ξεκινά η υπηρεσία teredo με την εντολή “sysrc -f /etc/rc.conf miredo_enable=“YES””.

5.4) Στο αρχείο /usr/local/etc/miredo/miredo.conf αφαιρώ τον χαρακτήρα # από τη γραμμή ServerAddress teredo.iks-jena.de (για να επιλεγθεί αυτός ο εξυπηρετητής) και το προσθέτω στη γραμμή ServerAddress teredo-remlab.de (που πλέον δεν λειτουργεί). Στη συνέχεια εκκινώ την υπηρεσία miredo με την εντολή “service miredo start”.

5.5) Στο PC1 βλέπω μια νέα διεπαφή δικτύου με όνομα teredo με διευθύνσεις IPv6 fe80::ffff:ffff:ffff και 2001:0:d911:c0d9:106d:ecf8:fa34:d77.

```

root@PC:~ # ifconfig
em0: flags=8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> metric 0 mtu 1500
    options=81009b<RXCSUM,TXCSUM,ULAN_MTU,ULAN_HWTAGGING,ULAN_HWCSUM,ULAN_HW
FILTER>
    ether 08:00:27:dc:a8:b8
    inet 10.0.2.15 netmask 0xfffff000 broadcast 10.0.2.255
    media: Ethernet autoselect (1000baseT <full-duplex>)
    status: active
    nd6 options=29<PERFORMNUD,IFDISABLED,AUTO_LINKLOCAL>
lo0: flags=8049<UP,LOOPBACK,RUNNING,MULTICAST> metric 0 mtu 16384
    options=680003<RXCSUM,TXCSUM,LINKSTATE,RXCSUM_IPV6,TXCSUM_IPV6>
    inet6 ::1 prefixlen 128
    inet6 fe80::1%lo0 prefixlen 64 scopeid 0x2
    inet 127.0.0.1 netmask 0xff000000
    groups: lo
    nd6 options=21<PERFORMNUD,AUTO_LINKLOCAL>
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:d911:c0d9:106d:ecf8:fa34:d77 prefixlen 128
    groups: tun
    nd6 options=21<PERFORMNUD,AUTO_LINKLOCAL>
    Opened by PID 951
root@PC:~ #

```

5.6) Ξεκινώ μια καταγραφή χωρίς επίλυση ονομάτων και διευθύνσεων στη διεπαφή em0 εκτελώντας “tcpdump -ni em0” και την αφήνω να τρέχει.

5.7) Η διεύθυνση IPv4 του εξυπηρετητή Teredo με τον οποίο επικοινωνεί το PC1 είναι η 217.17.192.217.

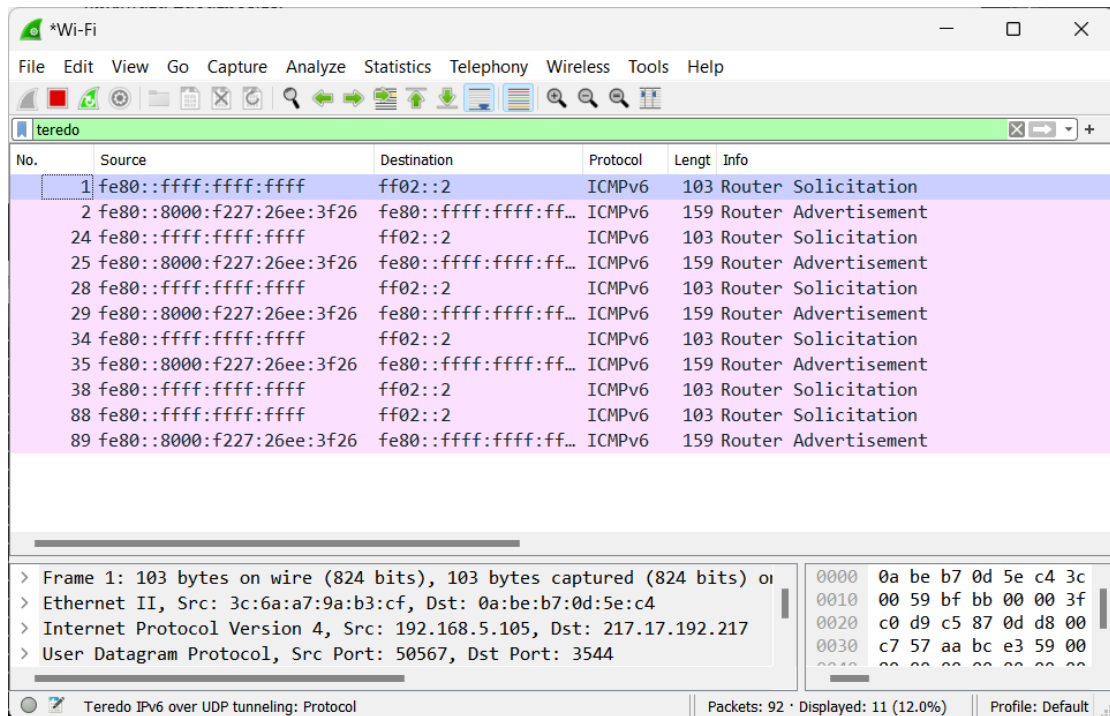
```

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
16:53:38.722611 IP 10.0.2.15.11613 > 217.17.192.217.3544: UDP, length 61
16:53:39.115421 IP 217.17.192.217.3544 > 10.0.2.15.11613: UDP, length 117
16:54:08.721650 IP 10.0.2.15.11613 > 217.17.192.217.3544: UDP, length 61
16:54:09.117145 IP 217.17.192.217.3544 > 10.0.2.15.11613: UDP, length 117
16:54:38.721486 IP 10.0.2.15.11613 > 217.17.192.217.3544: UDP, length 61
16:54:39.013428 IP 217.17.192.217.3544 > 10.0.2.15.11613: UDP, length 117

```

5.8) Για την επικοινωνία των PCs με τον εξυπηρετητή Teredo χρησιμοποιείται ως πρωτόκολλο μεταφοράς το UDP στην θύρα 3544.

5.9) Ξεκινώ με Wireshark μια καταγραφή πακέτων στη φυσική κάρτα του υπολογιστή μου εφαρμόζοντας φίλτρο απεικόνισης teredo και την αφήνω να τρέχει. Στην καταγραφή αυτή παρατηρώ μηνύματα πρωτοκόλλου ICMPv6.



5.10) Από το PC1 δεν μπορώ να κάνω ping6 σε κανένα από τα www.ibm.com και www.amazon.com, παρά μόνο στο www.ntua.gr.

5.11) Από νέο παράθυρο στο PC1 (Alt+Fi) κάνω ping6 στο www.ntua.gr και αφήνω να τρέχει.

5.12) Στην καταγραφή στο Wireshark παρατηρώ νέα μηνύματα τύπου Direct IPv6 Connectivity Test.

5.13) Όχι, δεν παρατηρώ μηνύματα ICMPv6 Echo {request,reply} στην καταγραφή στο Wireshark.

5.14) Στην καταγραφή στο PC1, παρατηρώ πακέτα πρωτοκόλλου UDP. Η θύρα που αντιστοιχεί στον αναμεταδότη teredo είναι η 3545.

```
File Machine View Input Devices Help
17:08:09.471061 IP 216.66.84.238.3545 > 10.0.2.15.11613: UDP, length 56
17:08:09.717052 IP 10.0.2.15.11613 > 216.66.84.238.3545: UDP, length 56
17:08:10.395364 IP 216.66.84.238.3545 > 10.0.2.15.11613: UDP, length 56
17:08:10.724667 IP 10.0.2.15.11613 > 216.66.84.238.3545: UDP, length 56
17:08:11.625932 IP 216.66.84.238.3545 > 10.0.2.15.11613: UDP, length 56
17:08:11.727163 IP 10.0.2.15.11613 > 216.66.84.238.3545: UDP, length 56
17:08:12.534681 IP 216.66.84.238.3545 > 10.0.2.15.11613: UDP, length 56
17:08:12.734957 IP 10.0.2.15.11613 > 216.66.84.238.3545: UDP, length 56
17:08:13.464879 IP 216.66.84.238.3545 > 10.0.2.15.11613: UDP, length 56
17:08:13.742709 IP 10.0.2.15.11613 > 216.66.84.238.3545: UDP, length 56
17:08:14.414301 IP 216.66.84.238.3545 > 10.0.2.15.11613: UDP, length 56
17:08:14.749475 IP 10.0.2.15.11613 > 216.66.84.238.3545: UDP, length 56
17:08:15.207586 IP 216.66.84.238.3545 > 10.0.2.15.11613: UDP, length 56
17:08:15.754861 IP 10.0.2.15.11613 > 216.66.84.238.3545: UDP, length 56
17:08:16.436507 IP 216.66.84.238.3545 > 10.0.2.15.11613: UDP, length 56
17:08:16.722399 IP 10.0.2.15.11613 > 217.17.192.217.3544: UDP, length 61
17:08:16.772223 IP 10.0.2.15.11613 > 216.66.84.238.3545: UDP, length 56
17:08:16.829292 IP 217.17.192.217.3544 > 10.0.2.15.11613: UDP, length 117
17:08:17.457591 IP 216.66.84.238.3545 > 10.0.2.15.11613: UDP, length 56
17:08:17.777654 IP 10.0.2.15.11613 > 216.66.84.238.3545: UDP, length 56
17:08:18.690408 IP 216.66.84.238.3545 > 10.0.2.15.11613: UDP, length 56
17:08:18.779275 IP 10.0.2.15.11613 > 216.66.84.238.3545: UDP, length 56
17:08:19.611057 IP 216.66.84.238.3545 > 10.0.2.15.11613: UDP, length 56
17:08:19.790111 IP 10.0.2.15.11613 > 216.66.84.238.3545: UDP, length 56
```

5.15) Σταματώ τις καταγραφές και ξεκινώ μόνο στο PC1 νέα καταγραφή χωρίς επίλυση ονομάτων και διευθύνσεων στη διεπαφή teredo εκτελώντας “tcpdump -ni teredo”.

5.16) Στην καταγραφή αυτή εμφανίζονται πακέτα ICMPv6, Echo {request,reply}.

5.17) Κάνοντας “ping6 fe80::ffff:ffff:ffff” βλέπω ότι δεν μπορώ να κάνω ping6 από το PC1 στο PC2 χρησιμοποιώντας τις διευθύνσεις IPv6 της διεπαφής teredo.

5.18) Ναι, παράγονται μηνύματα ICMPv6 στη διεπαφή teredo του PC1.

5.19) Σταματώ την καταγραφή στη διεπαφή teredo και ξεκινώ νέα στην em0 με “tcpdump -ni em0”. Παρατηρώ ότι δεν παράγονται δεδομενογράμματα UDP αντίστοιχα με τα ICMPv6 μηνύματα στη διεπαφή του PC1.

5.20) Κάνω “ping6 www.quad9.net” και μετά “ping6 www.f5.com”. Παρατηρώ ότι δεν επιλέγεται ο ίδιος teredo relay.