

<b>Ονοματεπώνυμο:</b> Πυλιώτης Αθανάσιος		<b>Όνομα PC:</b> DESKTOP-5DLG3IF
<b>Ομάδα:</b> 1	<b>Ημερομηνία:</b> 24/05/23	

## Εργαστηριακή Άσκηση 11

### *Το πρωτόκολλο IPv6*

### Άσκηση 1: Εισαγωγή στο IPv6

#### 1.1) PC1/2: **vim /etc/rc.conf**

Προσθέτουμε τη γραμμή που μας ζητείται: `ifconfig_em0_ipv6="inet6 accept_rtadv"`

#### 1.2) PC1/2: **service netif restart**

#### 1.3) PC1: **ifconfig em0**

`Inet6 fe80::a00:27ff:fec1:f33e%em0 prefixlen 64`

#### 1.4) PC2: **ifconfig em0**

`Inet6 fe80::a00:27ff:fece:fdc9%em0 prefixlen 64`

1.5) Αυτές οι διευθύνσεις είναι link-local λόγω του προθέματος `fe80::/10`. Προκύπτουν με το EUI-64 από την MAC των διεπαφών μετά το prefix, τα 3 πρώτα bytes της MAC, μετά το `fffe` και τέλος τα 3 τελευταία.

#### 1.6) PC1: **netstat -r6**

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

Internet6:
Destination      Gateway          Flags           Netif  Expire
:::/96           localhost       UGRS            lo0
localhost        link#5          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:fec link#1          UHS             lo0
fe80::%lo0/64    link#5          U               lo0
fe80::1%lo0      link#5          UHS             lo0
ff02::/16        localhost       UGRS            lo0
```

1.7) Παρατηρούμε πως μία εγγραφή αντιστοιχεί σε `em0` netif.

1.8) Σχετικές με το πρόθεμα `fe80::/64` υπάρχουν 2 εγγραφές:

- `fe80::%em0/64` → `em0`
- `fe80::a00:27ff:fec` → `lo0`

- `fe80::%lo0/64 → lo0`
- `fe80::1%lo0 → lo0`

**1.9) PC1: `ping ::1`**

Παρατηρούμε πως το PC2 δεν βλέπει το ping συνεπώς απαντάει η loopback (το PC1)

**1.10) PC1: `Ping6 fe80::a00:27ff:fec1:f33e`**

Παρατηρούμε πως δεν πετυχαίνει, UDP Connect: Network is unreachable. Για να πετύχει χρειάζεται το `%em0` στο τέλος χωρίς κενό σαν δείκτης ζώνης.

**1.11) PC1: `Ping6 fe80::a00:27ff:fece:fdc9%em0`**

Επιτυχία.

**1.12) PC1: `ping6 ff01::1%em0`**

Απαντάει το PC1 καθώς η διεύθυνση που κάνουμε ping είναι multicast και αφορά τη διεπαφή.

**1.13) PC1: `ping6 ff02::1%em0`**

Απαντάνε και τα δύο PC. Τώρα η διεύθυνση που κάνουμε το ping είναι πάλι multicast αλλά αφορά το link-local, δηλαδή το LAN1, και για αυτό απαντάνε όλα τα μηχανήματα που είναι συνδεδεμένα στο LAN1.

**1.14) PC1: `ifconfig em0 inet6 fd00:1::2/64`****1.15) PC2: `ifconfig em0 inet6 fd00:1::3/64`**

1.16) Οι παραπάνω διευθύνσεις είναι local unicast και προορίζονται για δρομολόγηση στο εσωτερικό των ιδιωτικών δικτύων. Στο `fd00::/8` οι διευθύνσεις είναι τυχαίες, μοναδικές και τοπικές. Συνεπώς είναι οι αντίστοιχες των `10.0.0.0/8`, `172.16.0.0/12` και `192.168.0.0/16` στο IPv4.

**1.17) PC1/2: `ifconfig em0`**

Παρατηρούμε πως υπάρχουν 2 διευθύνσεις IPv6 και μία MAC σε καθένα από αυτά.

**1.18) PC1: `netstat -r6`**

Εμφανίστηκαν 2 καινούριες εγγραφές σε σχέση με πριν.

1.19) PC1/2: `vim /etc/hosts` → προσθέτουμε τις γραμμές (και στα δύο μηχανήματα):

- `fd00:1::2`      PC1
- `fd00:1::3`      PC2

**1.20) PC1: `ping6 PC2`**

Ναι μπορούμε.

**1.21) PC1: `arp -a`**

Βλέπουμε 0 εγγραφές.

**1.22) PC1: `man ndp`****1.23) PC1: `ndp -a`**

1.24) Βλέπουμε συνολικά 3 εγγραφές. Παρατηρούμε πως 2 είναι σε κατάσταση reachable και 1 σε

κατάσταση stale.

```

root@R0:~ # ndp -a
Neighbor                               Linklayer Address  Netif  Expire      S  Flags
PC1                                     08:00:27:c1:f3:3e  em0    permanent  R
PC2                                     08:00:27:ce:fd:c9  em0    23h52m16s  S  R
fe80::a00:27ff:fec1:f33e%em0          08:00:27:c1:f3:3e  em0    permanent  R

```

#### 1.25) PC1: **ndp -p**

Υπάρχουν 3 προθέματα και είναι:

- fd00:1::/64 → pltime=infinity
- fe80::%em0/64 → pltime=infinity
- fe80::%lo0/64 → pltime=infinity

1.26) Μόνο όσα περιέχουν τη σημαία A μπορούν να χρησιμοποιηθούν από το SLAAC. Σε εμάς μόνο το δεύτερο και το τρίτο θα μπορούσαν να χρησιμοποιηθούν.

#### 1.27) PC1/2: **ndp -c**

Καθαρίζει τους πίνακες γειτόνων.

#### 1.28) PC2: **tcpdump -vvv -en -i em0**

#### 1.29) PC1: **ping6 -c 1 PC2**

Παρατηρούμε πως παράχθηκαν 6 πακέτα IPv6.

1.30) Μηνύματα ICMPv6 μεταφέρουν τα μηνύματα και στο πεδίο Next Header έχει τιμή 58.

1.31) Παρακάτω είναι σε bullets με τη σειρά που γίνεται η ανταλλαγή (Δεν έχω χρόνο για ζωγραφική λυπάμαι):

- PC1 → PC2: ICMPv6 Neighbor solicitation (LAN1) για να μάθει ποιος έχει την IPv6 του PC2
- PC2 → PC1: ICMPv6 Neighbor Advertisement
- PC1 → PC2: ICMPv6 Echo Request
- PC2 → PC1: ICMPv6 Echo Reply
- PC2 → PC1: ICMPv6 Neighbor Solicitation
- PC1 → PC2: ICMPv6 Neighbor Advertisement

1.32) Ο προορισμός της NS είναι ff02::1:ff0:3, που αποτελεί multicast διεύθυνση για κόμβους της τοπικής ζεύξης (LAN1). Προκύπτει από το πρόθεμα ff02::1:ff00:0/104 με τα 24 bits της unicast IPv6 του PC2.

1.33) Ο προορισμός του δεύτερου πακέτου είναι fd00:1::2. IPv6 ULA (Unicast) του PC1.

#### 1.34) PC2: **ndp -a**

PC1 Stale (ο μετρητής μετράει διαρκώς αντίστροφα 24 ώρες, τώρα είναι 23h34m42s)

#### 1.35) PC1: **ping6 PC2**

Παρατηρούμε πως ο μετρητής έφτασε στο 0, έγινε expired και μετά ξανά άρχισε. Όσο ήταν πολύ μικρός επίσης (δευτερόλεπτα) ήταν Reachable state. Γίνεται reachable για περίπου 15 δευτερόλεπτα, μετά stale για περίπου 5 και μετά πάλι reachable.

1.36) Η διάρκεια ήταν 35 δευτερόλεπτα και όταν λήξει γίνεται stale για 24 ώρες.

1.37) Η διάρκεια stale είναι 1 μέρα, αλλά λόγω του ping6 γίνεται 5 δευτερόλεπτα μέχρι να γίνει ξανά reachable.

1.38) Παρατηρούμε πως όταν γίνεται stale (μετά από reachable) δεν αλλάζει ξανά αυτή τη φορά.

1.39) PC1: **ping6 PC2**

PC2: **tcpdump -vvn -en -i em0**

**Ναι**, παρατηρούμε πως υπάρχουν κι άλλα πακέτα στην καταγραφή. Υπάρχουν πακέτα NS και NA μεταξύ των δύο PC. Παράγονται περίπου κάθε 40 seconds. Αυτός είναι ο χρόνος των 35+5 από παραπάνω. Παράγονται διότι κάθε 35 seconds λήγει η εγγραφή και χρειάζεται να σταλούν καινούρια NS/NA.

## Άσκηση 2: SLAAC και Στατική δρομολόγηση IPv6

2.1) R1/2: **vim /etc/rc.conf**

**ipv6\_gateway\_enable="YES"**

Αποθηκεύουμε και εκτελούμε:

R1/2: **service routing restart**

2.2) PC2: **ifconfig em0 inet6 fd00:1::3/64 delete → ifconfig em0 inet6 fd00:2::2/64**

2.3) R1: **vttysh → R1# configure terminal → R1(config)# interface em0 → R1(config-if)# ipv6 address fd00:1::1/64 → R1(config-if)# exit**

2.4) R1(config)# **interface em1 → R1(config-if)# ipv6 address fd00:3::1/126 → R1(config-if)# exit**

2.5) R2: **vttysh → R2# configure terminal → R2(config)# interface em1 → R2(config-if)# ipv6 address fd00:2::1/64 → R2(config-if)# exit**

2.6) R2(config)# **interface em0 → R2(config-if)# ipv6 address fd00:3::2/126 → R2(config-if)# exit**

2.7) PC1: **route -6 add default fd00:1::1**

2.8) PC2: **route -6 add default fd00:2::1**

2.9) R1: **tcpdump -vvv -en -i em0**

2.10) PC1: **ndp -c → ping6 -c 1 fd00:2::2**

Όχι δεν πετυχαίνει το ping. Ο R1 δεν έχει στο routing table διαδρομή για το LAN2 και δεν ξέρει που να προωθήσει τα πακέτα του PC1, οπότε τα απορρίπτει με destination unreachable.

2.11) Παράγονται μηνύματα NS με προορισμό ff02::1::ff00:1 αρχικά.

Στη συνέχεια υπάρχει πακέτο NA από τον R1 με προορισμό fd00:1::2.

Μετά παράγεται ICMPv6 Echo Request με προορισμό fd00:2::2.

Σαν απάντηση παρατηρούμε από τον R1 το ICMPv6 destination unreachable Με προορισμό fd00:1::2.

Τέλος υπάρχει το ICMPv6 NS στο fd00:1::2 και σαν απάντηση ICMPv6 NA προς το fd00:1::1.

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

2.13) PC1: **ndp -c → ping6 -c 1 fd00:2::2**

Όχι δεν πετυχαίνει το ping. Αυτή τη φορά συμβαίνει επειδή ο R2 δεν έχει στο routing table του εγγραφή για το LAN1 και συνεπώς δεν επιστρέφει τα ICMPv6 Echo Reply του PC2 και τα απορρίπτει γιατί δεν ξέρει μέσω ποιας διεπαφής να τα προωθήσει.

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

2.15) PC1: **ndp -c → ping6 -c 1 fd00:2::2**

Ναι πετυχαίνει το ping.

2.16) R1(config)# **interface em0 → R1(config-if)# 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) PC1: **route -6 delete default**
- 2.21) R1: **tcpdump -vvv -en -i em0**
- 2.22) PC1: **service netif restart**
- 2.23) Παρατηρούμε 3 μηνύματα, Router solicitation και Router advertisement και ένα μήνυμα NS από το PC1.
- 2.24) Παράγεται το NS για έλεγχο μήπως κάποιο άλλο μηχανήμα έχει ίδια IPv6 με αυτό στο LAN1. Αν κάποιο τελικά την έχει θα απαντήσει με NA και το DAD θα αποτύχει.
- 2.25) Η διεύθυνση πηγής που χρησιμοποιεί είναι η :: επειδή δεν έχει ακόμα οριστική IPv6 διεύθυνση γιατί πρέπει πρώτα να ελέγξει πως δεν χρησιμοποιείται από άλλο μηχανήμα στο υποδίκτυο.
- 2.26) Στα μηνύματα RS η διεύθυνση πηγής είναι η fe80::a00:27ff:fec1:f33e, η link-local του.
- 2.27) Οι διευθύνσεις προορισμού των μηνυμάτων είναι:
- NS: ff02::1:ffc1:f33e, η multicast του solicited-node του PC1, Λογικό αφού ψάχνει δια μηχανήμα με τη διεύθυνση που πάει να αποκτήσει στο LAN1
  - RS: ff02::2. Λογικό γιατί είναι προκαθορισμένη multicast που ακούνε όλοι οι δρομολογητές σε τοπικό δίκτυο
  - RA: ff02:1. Ακούνε όλα τα μηχανήματα σε τοπική ζεύξη. Λογικό αφού διαφημίζει την κάθε ζεύξη, πληροφορία που χρησιμοποιείται από τα μηχανήματα.
- 2.28) Οι MAC προορισμού είναι:
- NS: 33:33:ff:c1:f3:3e. Παράγεται από το 33:33 με την IPv6 διεύθυνση προορισμού να ακολουθεί (βλέπε πάνω)
  - RS: 33:33:00:00:00:02. Αντίστοιχα με παραπάνω.
  - RA: 33:33:00:00:00:01. Αντίστοιχα με παραπάνω.
- 2.29) PC1: **ndp -p**
- Υπάρχουν 3 προθέματα και είναι:
- fd00:1::/64 → pltime=604800
  - fe80::%em0/64 → pltime=infinity
  - fe80::%lo0/64 → pltime=infinity
- Η μοναδική διαφορά είναι πως το pltime για το prefix fd00:1::/64 έχει περιορισμένο χρόνο. Επίσης εμφανίζεται και το “advertised by fd80::a00:27ff:fe8a:ad2d%em0. Τέλος, και οι 3 εγγραφές έχουν το flag A (δηλαδή το έχει και η πρώτη τώρα)
- 2.30) PC1: **ifconfig em0**

fe80::a00:27ff:fec1:f33e%em0 interface em0

fd00:1::a00:27ff:fec1:f33e (ULA) από το R1

**2.31) PC1: netstat -r6**

Ναι υπάρχει προκαθορισμένη διαδρομή η οποία προέκυψε από την IPv6 που χρησιμοποίησε ο R1 στο LAN1 για RA.

**2.32) Από το PC2: fd00:1::a00:27ff:fec1:f33e**

Από τον R1: Μπορούμε και στις δύο διευθύνσεις που υπάρχουν διαθέσιμες (δηλαδή και στην fe80::a00:27ff:fec1:f33e%em0)

## Άσκηση 3: Δυναμική δρομολόγηση IPv6

3.1) R1: **vttysh** → R1# **configure terminal** → R1(config)# **no ipv6 route fd00:2::/64 fd00:3::2**

R2: **vttysh** → R2# **configure terminal** → R2(config)# **no ipv6 route fd00:1::/64 fd00:3::1**

3.2) R1/2(config)# **router ripng** → R1/2(config-router)# **network em0** → R1/2(config-router)# **network em1**

3.3) R1(config-router)# **do show ip ipv6 route ripng**

Μία εγγραφή βλέπουμε για το fd00:2::/64

3.4) Η διεύθυνση του επόμενου κόμβου είναι fe80::a00:27ff:feec:1d63 και είναι η link-local του R2 στο WAN1.

3.5) PC1: **ping6 fd00:2::2**

Ναι είναι επιτυχές.

3.6) R1: **tcpdump -vvv -en -i em1 ip6**

3.7) Παρατηρούμε πακέτα ripng-resp 2 και ripng-req με προορισμό ff02::9 και θύρα 521. Αποτελεί την διεύθυνση multicast IPv6 addresses στην οποία ακούνε όλοι οι δρομολογητές RIP για να στείλουν τα RIP updates τους. Και η θύρα είναι προκαθορισμένη (521). Πέρα από αυτά βλέπουμε και ένα πακέτο ripng-resp 2 με προορισμό το fe80::a00:27ff

3.8) Το hop limit έχει τιμή 255 επειδή πρέπει οι δρομολογητές να γνωρίζουν αν έχουν διέλθει μέσω άλλων δρομολογητών ή όχι (για κάθε δρομολογητή μειώνεται κατά 1)

3.9) Πρωτόκολλο στρώματος μεταφοράς το UDP και θύρα η 521. Δεν είναι ίδια με του RIP, που έχει θύρα 520.

3.10) R1/2(config)# **no router ripng**

3.11) R1/2(config)# **do write file**

3.12) R1/2: **service frr restart**

3.13) R1: **vttysh** → R1# **configure terminal** → R1(config)# **router ospf6** → R1(config-ospf6)# **ospf6 router-id 1.1.1.1**

R2: **vttysh** → R2# **configure terminal** → R2(config)# **router ospf6** → R2(config-ospf6)# **ospf6 router-id 2.2.2.2**

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

3.15) R2(config-ospf6)# **interface em0 area 0.0.0.0** → R2(config-ospf6)# **interface em1 area 0.0.0.0**

3.16) R2(config-ospf6)# **do show ipv6 route ospf6**

Βλέπουμε 3 εγγραφές. Οι 2 είναι directly connected με κόστος 100, αναμενόμενο με τον τύπο



κόστους του ospf6, ενώ η Τρίτη που είναι για το LAN1 βλέπουμε έχει κόστος 200 επειδή διέρχεται και από το R1 που δίνει παραπάνω κόστος.

3.17) Για το fd00:1::/64 η διεύθυνση που επόμενου κόμβου είναι fe80::a00:27ff:fe5e:88ef, η οποία είναι link-local διεύθυνση και συγκεκριμένα του R1 στο WAN1.

3.18) R2: tcpdump -vvv -en -i em0 ip6

3.19) OSPFv3 πακέτα που παρατηρούμε είναι OSPFv3 Hello και έχουν ως διεύθυνση προορισμού την ff02::5.

3.20) Hop Limit = 1

3.21) Το OSPFv3 χρησιμοποιεί τον αριθμό 89, το οποίο είναι ίδιο με το αντίστοιχο του OSPFv2

3.22) PC2: **ping6 fd00:1::a00:27ff:fec1:f33e**

3.23) R1/2(config)# **no router ospf6**

3.24) R1/2: **service frr restart**

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** → **exit**

3.32) Για το R2 είναι τα παρακάτω:

R2(config)# **router-id 2.2.2.2** → R1(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 fd00:3::1 activate** → **exit**

3.33) R1(config)# **do show ipv6 route bgp**

Βλέπουμε μία δυναμική εγγραφή.

3.34) Η διεύθυνση επόμενου κόμβου είναι fe80::a00:27ff:feec:1d63, link-local of R2 at WAN1

3.35) R1: **tcpdump -vvv -en -i em1 not icmp6**

3.36) Παρατηρούμε μηνύματα BGP keepalive. Πρωτόκολλο μεταφοράς είναι το TCP και θύρα που χρησιμοποιείται είναι η 179. Ναι είναι ίδια με του IPv4.

3.37) Hop Limit = 1

3.38) PC1: **ping6 fd00:2::2** → Επιτυχές

3.39) PC1: **reboot** → PC1: **vttysh** → PC1# **configure terminal** → PC1(config)# **interface em0** →

PC1(config-if)# **ipv6 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-as)# **neighbor fd00:1::1**

**activate** → PC1(config-router-ad)# **exit**

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

3.45) R1(config-router)# **address-family ipv6**

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

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

3.46) R1/PC1(config-router)# **do show ip bgp neighbors**

Βλέπουμε πως το BGP είναι established και πως έχουμε internal link

**3.47)** PC1(config)# **do show ipv6 route bgp**

Βλέπουμε 2 εγγραφές.

3.48) Δεν είναι επιλεγμένη η διαδρομή για το fd00:1::/64 (διαχειριστική απόσταση 200) επειδή είναι άμεσα συνδεδεμένη ήδη σε άλλη διαδρομή, η οποία εννοείται έχει μικρότερη διαχειριστική απόσταση και επιλέγεται αντί της εγγραφής BGP.

3.49) Η διεύθυνση επόμενου κόμβου για το fd00:2::64 είναι fd00:1::1, η unicast διεύθυνση IPv6 του R1 στο LAN1

3.50) PC2: **ping6 fd00:1::2**

Ναι είναι επιτυχές!

## Άσκηση 4: Μηχανισμός μετάβασης 464 XLAT

- 4.1) R1: **vttysh** → R1# **configure terminal** → R1(config)# **interface em0** → R1(config-if)# **ip address 192.168.1.1/24**
- 4.2) R2: **vttysh** → R2# **configure terminal** → R2(config)# **interface em1** → R2(config-if)# **ip address 192.168.2.1/24**
- 4.3) PC1: **vttysh** → PC1# **configure terminal** → 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) PC2: **vttysh** → PC2# **configure terminal** → PC2(config)# **interface em0** → PC2(config-if)# **ip address 192.168.2.2/24** → PC2(config-if)# **exit** → PC2(config)# **ip route 0.0.0.0/0 192.168.2.1**
- 4.5) R1: **vim /etc/rc.conf**  
    **firewall\_enable="YES"**  
    **firewall\_nat64\_enable="YES"**  
    **firewall\_type="open"**  
    **firewall\_logif="YES"**
- 4.6) R1: **kldload ipfw** → **service ipfw start**
- 4.7) R1: **ipfw show**  
    Περιέχει 12 κανόνες μαζί με τον προκαθορισμένο.
- 4.8) PC1: **ping6 fd00:2::2** → Ναι μπορούμε.
- 4.9) R1: **ipfw nat64clat nat64 create clat\_prefix fd00:3:1::/96 plat\_prefix 64:ff9b::/96 allow\_private log**
- 4.10) R1: **ipfw add 2000 nat64clat nat64 ip4 from any to not me recv em0**
- 4.11) R1: **ipfw add 3000 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) R2: **vim /etc/rc.conf**  
    **firewall\_enable="YES"**  
    **firewall\_nat64\_enable="YES"**  
    **firewall\_type="open"**  
    **firewall\_logif="YES"**  
    R2: **kldload ipfw** → **service ipfw start**
- 4.14) R2: **ipfw nat64lsn nat64 create prefix4 2.2.2.0/24 prefix6 64:ff9b::/96 allow\_private log**
- 4.15) R2: **ipfw add 2000 nat64lsn nat64 ip6 from fd00:3:1::/96 to 64:ff9b::/96 recv em0**
- 4.16) R2: **ipfw add 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) PC1: **ping 192.168.1.1/ ping 192.168.2.2**

Ναι είναι επιτυχής

4.20) R1: **ifconfig ipfwlog0 create → tcpdump -i ipfwlog0**

4.21) R2: **ifconfig ipfwlog0 create → tcpdump -i ipfwlog0**

4.22) PC1: **ping -c 1 192.168.2.2**

Παρατηρούμε ICMP Echo request, ICMP6 Echo request, ICMP6 Echo reply και ICMP echo reply στο R1 και ICMP6 Echo request, ICMP Echo request, ICMP Echo reply και ICMP6 echo reply στο R2, με αυτή τη σειρά κι από 1 το καθένα.

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) PC1: **ping 172.17.17.2 / ping 10.0.0.2**

Ναι μπορούμε, τα ping είναι επιτυχής.

4.25) R2: **ipfw nat64lsn nat64 show states** και δεν βλέπουμε τίποτα.

4.26) R2: **ipfw nat64lsn nat64 show states** και βλέπουμε 2 εγγραφές: (μετά το ping σε αυτές)

fd00:3:1::c0a8:102	2.2.2.252	ICMPv6	56	172.17.17.2
fd00:3:1::c0a8:102	2.2.2.252	ICMPv6	52	10.0.0.2

Διαρκούν 65 seconds από όταν γίνεται το ping, το βρίσκουμε με την παραπάνω εντολή που την εκτελούμε διαδοχικά. Εμφανίζονται μεταφράσεις και για τις δύο διευθύνσεις.

## Άσκηση 5: Μηχανισμός μετάβασης Teredo

5.1) PC1/2: **dhclient em0 → ping youtube.com**

Πετυχαίνει άρα έχουμε πρόσβαση στο internet.

5.2) PC1/2: **pkg install miredo**

5.3) PC1/2: **sysrc miredo\_enable="YES"**

5.4) PC1/2: **vim /usr/local/etc/miredo/miredo.conf**

Κάνουμε τις αλλαγές με τα # που ζητάει.

PC1/2: **service miredo start**

5.5) PC1: **ifconfig**

Βλέπουμε τη διεπαφή Teredo η οποία έχει ως IPv6 link-local την fe80::ffff:ffff:ffff%teredo και IPv6 Global address την 2001:0:d911:c0d9:2c:2bdb:a1b8:fb9f

5.6) PC1: **tcpdump -n -i em0**

5.7) Η διεύθυνση IPv4 του teredo είναι η 217.17.192.217

5.8) Πρωτόκολλο μεταφοράς είναι το UDP και θύρα στον εξυπηρετητή η 3544

5.9) Ξεκινάμε wireshark και κάνουμε καταγραφή με κατάλληλο φίλτρο teredo. Παρατηρούμε μηνύματα πρωτοκόλλου ICMPv6 RS/RA

5.10) PC1:

a. **ping6 [www.ntua.gr](http://www.ntua.gr)**

b. **ping6 [www.ibm.com](http://www.ibm.com)**

c. **ping6 [www.amazon.com](http://www.amazon.com)**

παρατηρούμε πως μόνο το δεύτερο ping6 αποτυγχάνει και τα άλλα 2 είναι επιτυχή.

5.11) PC1: **ping6 [www.ntua.gr](http://www.ntua.gr)** (σε νέο παράθυρο)

5.12) Στο wireshark παρατηρούμε Direct IPv6 Connectivity Test από το πρωτόκολλο Teredo

5.13) Όχι δεν παρατηρούμε ICMPv6 Echo Request/ Echo Reply.

5.14) Στη καταγραφή στο PC1 παρατηρούμε το πρωτόκολλο μεταφοράς UDP και η θύρα που αντιστοιχεί είναι η 3544 ακόμα

5.15) PC1: **tcpdump -n -i teredo**

5.16) Παρατηρούμε πακέτα ICMPv6 Echo Request και ICMPv6 Echo Reply

5.17) PC1: **ping6 2001:0:d911:c0d9:1cfc:7db9:a1b8:fb9f / ping6 fe80::ffff:ffff:ffff%teredo**

Ναι είναι και τα 2 επιτυχή.

5.18) Ναι παράγονται μηνύματα ICMPv6 ICMP Echo Request στην διεπαφή teredo.

5.19) PC1: **tcpdump -n -i em0**

Όχι δεν παράγονται για το link-local αλλά παράγονται UDP για το global address και

στέλλονται προς το 94.71.4.96

5.20) PC1: **ping6** [www.quad9.net](http://www.quad9.net) / **ping6** [www.f5.com](http://www.f5.com)

Όχι, δεν επιλέγεται ο ίδιος teredo relay. Για το πρώτο επιλέγεται ο 6to4.sjc2.he.net και για το δεύτερο επιλέγεται ο 6to4-2.zrh1.he.net (με άλλες IPv4)