

Εργαστήριο Δικτύων Υπολογιστών

Εργαστηριακή Άσκηση 11 Το πρωτόκολλο IPv6

Όνοματεπώνυμο: Νικόλαος Παγώνας, el18175	Όνομα PC: nick-ubuntu
Ομάδα: 1 (Τρίτη 10:45)	Ημερομηνία Εξέτασης: Τρίτη 24/05/2022

Άσκηση 1: Εισαγωγή στο IPv6

Προετοιμασία στο σπίτι

1.

Εκτελούμε `service frr stop`.

2.

Εκτελούμε `touch /usr/local/etc/frr/ripngd.conf`.

3.

Εκτελούμε `touch /usr/local/etc/frr/ospf6d.conf`.

4.

Εκτελούμε `chown frr:frr /usr/local/etc/frr/{ripngd.conf,ospf6d.conf}`

5.

Αλλάζουμε τη ζητούμενη γραμμή σε:

```
frr_daemons="zebra staticd ripd ripngd ospfd ospf6d bgpd"
```

6.

Εκτελούμε:

```
reboot
service sshd status
service frr status
```

Τρέχουν κανονικά.

7.

Εκτελούμε `service frr start`.

8.

Εκτελούμε:

```
poweroff  
File - Export Appliance...
```

και δημιουργούμε ένα αρχείο `frr.ovn`.

9.

Αποθηκεύουμε το `frr.ovn` για μελλοντική χρήση.

1.1

Εκτελούμε:

```
### PC1 ###  
  
sysrc ifconfig_em0_ipv6="inet6 accept_rtadv"  
  
### PC2 ###  
  
sysrc ifconfig_em0_ipv6="inet6 accept_rtadv"
```

1.2

Εκτελούμε:

```
### PC1 ###  
  
service netif restart  
  
### PC2 ###  
  
service netif restart
```

1.3

Στο PC1 εκτελούμε `ifconfig em0`. Έχει αποδοθεί η διεύθυνση `fe80::a00:27ff:fe96:db`.

1.4

Στο PC2 εκτελούμε `ifconfig em0`. Έχει αποδοθεί η διεύθυνση `fe80::a00:27ff:fe25:d99f`.

1.5

Είναι είδους link-local (fe80::/10). Προκύπτει ως εξής:

PC1

08 00 27 96 00 db	(MAC address)
08 00 27 ff fe 96 00 db	(ff fe in the middle)
0a 00 27 ff fe 96 00 db	(Flip 7th bit of 1st byte)
fe 80 00 00 00 00 00 0a 00 27 ff fe 96 00 db	(Insert fe80::/10 prefix)
fe80:0000:0000:0000:0a00:27ff:fe96:00db	(IP address with leading 0s)
fe80::a00:27ff:fe96:db	(Final IP address)

PC2

08 00 27 25 d9 9f	(MAC address)
08 00 27 ff fe 25 d9 9f	(ff fe in the middle)
0a 00 27 ff fe 25 d9 9f	(Flip 7th bit of 1st byte)
fe 80 00 00 00 00 00 0a 00 27 ff fe 25 d9 9f	(Insert fe80::/10 prefix)
fe80:0000:0000:0000:0a00:27ff:fe25:d99f	(IP address with leading 0s)
fe80::a00:27ff:fe25:d99f	(Final IP address)

1.6

Στο PC1 εκτελούμε netstat -rn6. Εμφανίζονται 9 εγγραφές:

Destination	Gateway	Flags	Netif
::/96	::1	UGRS	lo0
::1	link#2	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:fe96:db%em0	link#1	UHS	lo0
fe80::%lo0/64	link#2	U	lo0
fe80::1%lo0	link#2	UHS	lo0
ff02::/16	::1	UGRS	lo0

1.7

Μία εγγραφή αφορά την em0:

Destination	Gateway	Flags	Netif
fe80::%em0/64	link#1	U	em0

1.8

Περιέχει τις εγγραφές:

Destination	Gateway	Flags	Netif
fe80::%em0/64	link#1	U	em0
fe80::a00:27ff:fe96:db%em0	link#1	UHS	lo0
fe80::%lo0/64	link#2	U	lo0
fe80::1%lo0	link#2	UHS	lo0

1.9

Στο PC1 εκτελούμε `ping6 ::1`. Απαντά το ίδιο το PC1.

1.10

Στο PC1 εκτελούμε `ping6 fe80::a00:27ff:fe96:db`. Εμφανίζεται μήνυμα λάθους "UDP connect: Network is unreachable". Πρέπει να προσθέσουμε τον δείκτη ζώνης %em0 στο τέλος της διεύθυνσης ως εξής:

```
ping6 fe80::a00:27ff:fe96:db%em0
```

Τώρα το ping είναι επιτυχές.

1.11

Στο PC1 εκτελούμε `ping6 fe80::a00:27ff:fe25:d99f`. Αποτυγχάνει όπως και πριν με μήνυμα λάθους "UDP connect: Network is unreachable". Προσθέτουμε τον δείκτη ζώνης %em0:

```
ping6 fe80::a00:27ff:fe25:d99f%em0
```

Το ping είναι επιτυχές.

1.12

Στο PC1 εκτελούμε `ping6 -I em0 ff01::1`. Απαντά το PC1, αφού η διεύθυνση ff01::1 παριστάνει όλους τους κόμβους στη διεπαφή.

1.13

Στο PC1 εκτελούμε `ping6 -I em0 ff02::1`. Απαντούν τα PC1 και PC2, αφού η διεύθυνση ff02::1 παριστάνει όλους τους κόμβους στην τοπική ζεύξη.

1.14

Στο PC1 εκτελούμε `ifconfig em0 inet6 fd00:1::2/64`.

1.15

Στο PC2 εκτελούμε `ifconfig em0 inet6 fd00:1::3/64`.

1.16

Είναι μοναδικές τοπικές διευθύνσεις (`fd00::/8`), οι ανάλογες των διευθύνσεων ιδιωτικής χρήσης `10.0.0.0/8`, `172.16.0.0/12`, `192.168.0.0/16` στο IPv4.

1.17

Στα PC1 και PC2 εκτελούμε `ifconfig em0`. Υπάρχουν 2 διευθύνσεις.

1.18

Στο PC1 εκτελούμε `netstat -rn6`. Προστέθηκαν 2 νέες εγγραφές (επισημαίνονται με `+++` στην αρχή της γραμμής):

Destination	Gateway	Flags	Netif
::/96	::1	UGRS	lo0
::1	link#2	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:fe96:db%em0	link#1	UHS	lo0
fe80::%lo0/64	link#2	U	lo0
fe80:1%lo0	link#2	UHS	lo0
ff02::/16	::1	UGRS	lo0

1.19

Πρέπει και στα δύο μηχανήματα να προσθέσουμε (μέσω `vi /etc/hosts`) τις γραμμές:

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

1.20

Στο PC1 εκτελούμε `ping6 PC2`. Το ping είναι επιτυχές.

1.21

Στο PC1 εκτελούμε `arp -a`. Δεν παρατηρούμε καμία εγγραφή.

1.22

Στο PC1 εκτελούμε `man ndp`.

1.23

Είναι npd -a.

1.24

Βλέπουμε 4 εγγραφές:

Neighbor	Linklayer Addr	Netif	Expire	S	Flags
PC1	08:00:27:96:00:db	em0	permanent	R	
PC2	08:00:27:25:d9:9f	em0	23h58m7s	S	
fe80::a00:27ff:fe96:db%em0	08:00:27:96:00:db	em0	permanent	R	
fe80::a00:27ff:fe25:d99f%em0	08:00:27:25:d9:9f	em0	22h7m44s	S	

1.25

Στο PC1 εκτελούμε ndp -p. Έχουμε:

```
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
```

Οι εγγραφές έχουν άπειρη διάρκεια ζωής. Από το SLAAC μπορούν να χρησιμοποιηθούν η 2η και η 3η εγγραφή, που έχουν τη σημαία "A" ενεργοποιημένη.

1.26

Σε νέο παράθυρο στο PC2 εκτελούμε tcpdump -vn.

1.27

Στα PC1 και PC2 εκτελούμε ndp -c.

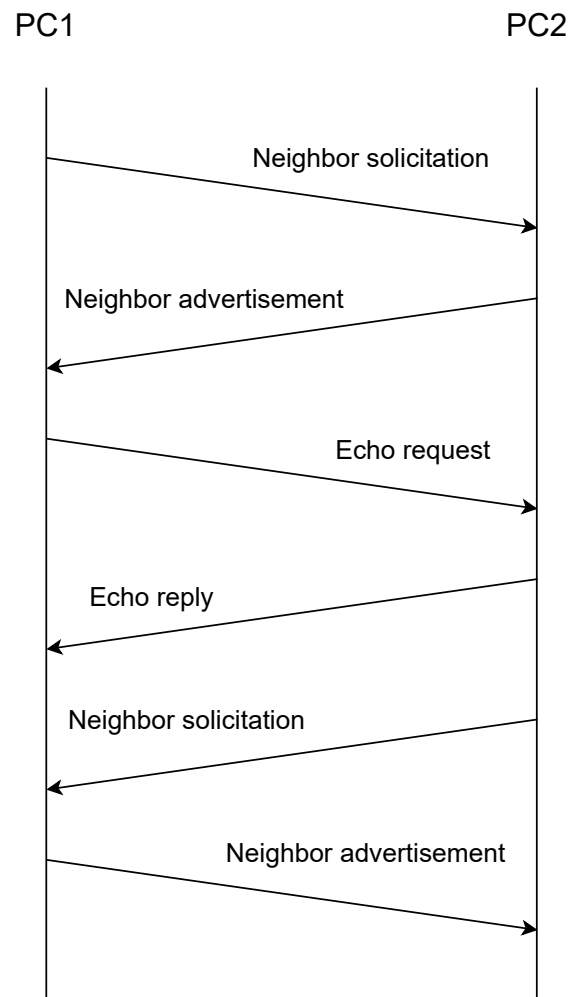
1.28

Στο PC1 εκτελούμε ping6 -c 1 PC2 και σταματάμε την καταγραφή. Βλέπουμε 6 πακέτα IPv6.

1.29

Μεταφέρουν μηνύματα του ICMPv6. Η τιμή του Next header είναι 58.

1.30



1.31

Η διεύθυνση προορισμού του πρώτου πακέτου NS (ff02::1:ff00:3) είναι multicast Solicited Node και προκύπτει από τα τελευταία 24 bit της διεύθυνσης unicast του PC2, αν σε αυτά προσθέσουμε το πρόθεμα ff02:0:0:0:0:1:ff00:0/104:

ff02:0:0:0:0:1:ff00:0000	(Special prefix)
+ 00:0003	(Last 24 bits of PC2 unicast address)

ff02:0:0:0:0:1:ff00:0003	(Multicast Solicited Node address with leading 0s)
ff02::1:ff00:3	(Final multicast Solicited Node address)

1.32

Η διεύθυνση προορισμού του δεύτερου πακέτου NS (fd00:1::2) είναι μοναδική τοπική διεύθυνση, και έχει προκύψει από ανάθεση (που έγινε σε προηγούμενο ερώτημα).

1.33

Στο PC2 εκτελούμε `ndp -a`. Η κατάσταση της εγγραφής για το PC1 είναι "S" (Stale). Η διάρκεια ζωής της σχετικής εγγραφής είναι 22h1m44s.

1.34

Στο PC1 εκτελούμε `ping6 PC2` και ύστερα με διαδοχικά `ndp -a` στο PC2 παρατηρούμε συνεχή εναλλαγή μεταξύ των καταστάσεων "R" και "S" (Reachable και Stale).

1.35

Φαίνεται να είναι ≈ 43 δευτερόλεπτα, και μόλις λήξει, η αντίστοιχη εγγραφή μπαίνει σε κατάσταση Stale.

1.36

Είναι 24 ώρες.

1.37

Σταματάμε το `ping` στο PC1 και ύστερα με διαδοχικά `ndp -a` στο PC2 παρατηρούμε αλλαγή από την κατάσταση "R" (Reachable) στην κατάσταση "S" (Stale), και αυτή τη φορά η κατάσταση παραμένει "S" (Stale).

1.38

Στο PC1 εκτελούμε `ping6 PC2`. Στο PC2 εκτελούμε `tcpdump -n`. Παρατηρούμε επιπλέον πακέτα ICMPv6 neighbor solicitation (NS), με σκοπό να επιβεβαιωθεί η προσβασιμότητα του γείτονα, και ICMPv6 neighbor advertisement (NA), ως απάντηση στα πακέτα NS. Παράγονται περίπου κάθε 23 δευτερόλεπτα.

Άσκηση 2: SLAAC και Στατική δρομολόγηση IPv6

2.1

Στους R1 και R2 εκτελούμε:

```
sysrc ipv6_gateway_enable="YES"  
service routing restart
```

2.2

Στο PC2 εκτελούμε:

```
ifconfig em0 inet6 fd00:1::3/64 delete  
ifconfig em0 inet6 fd00:2::2/64
```


2.3

Στον R1 εκτελούμε:

```
vttysh
configure terminal

interface em0

ipv6 address fd00:1::1/64
```

Σημείωση: Μπορούμε να δώσουμε και την εντολή "ip address" αντί για "ipv6 address". Και στις δύο περιπτώσεις η εντολή θα καταχωρηθεί ως "ipv6 address" στο running configuration. Εμείς προτιμούμε την εντολή "ipv6 address" για να είμαστε απολύτως ακριβείς.

2.4

Στον R1 εκτελούμε:

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

2.5

Στον R2 εκτελούμε:

```
vttysh
configure terminal

interface em1
ipv6 address fd00:2::1/64
```

2.6

Στον R2 εκτελούμε:

```
interface em0
ipv6 address fd00:3::2/126
```

2.7

Στο PC1 εκτελούμε route -6 add default fd00:1::1.

2.8

Στο PC2 εκτελούμε route -6 add default fd00:2::1.

2.9

Στον R1 εκτελούμε tcpdump -i em0.

2.10

Στο PC1 εκτελούμε:

```
ndp -c  
ping6 -c 1 fd00:2::2
```

Δε λαμβάνουμε απάντηση. Αυτό συμβαίνει διότι ο R1 δεν διαθέτει εγγραφή στον πίνακα δρομολόγησης σχετική με τη διεύθυνση fd00:2::2 και έτσι στέλνει μήνυμα λάθους ICMPv6 destination unreachable.

2.11

Παράγονται (καταγράφουμε και την διεύθυνση πηγής για πληρότητα):

ICMPv6 Message	IPv6 source address	IPv6 destination address
Neighbor solicitation	fd00:1::2	ff02::1:ff00:1
Neighbor advertisement	fd00:1::1	fd00:1::2
Echo request	fd00:1::2	fd00:2::2
Destination unreachable	fd00:1::1	fd00:1::2
Neighbor solicitation	fd00:1::1	fd00:1::2
Neighbor advertisement	fd00:1::2	fd00:1::1

2.12

Στον R1 εκτελούμε:

```
ipn6 route fd00:2::/64 fd00:3::2
```

Σημείωση: Όπως και πριν, επιλέγουμε την εντολή "ipn6", αλλά και η "ip" θα είχε τα ίδια αποτελέσματα (στο running configuration η εντολή καταχωρείται έτσι κι αλλιώς ως "ipn6").

2.13

Στο PC1 εκτελούμε ping6 -c 1 fd00:2::2. Δεν λαμβάνουμε απάντηση, διότι ο R2 δεν έχει εγγραφή στον πίνακα δρομολόγησης σχετική με τη διεύθυνση fd00:1::2 προκειμένου να στείλει το ICMPv6 echo reply.

2.14

Στον R2 εκτελούμε:

```
ipn6 route fd00:1::/64 fd00:3::1
```

2.15

Στο PC1 εκτελούμε ping6 -c 1 fd00:2::2. Το ping είναι επιτυχές.

2.16

Στον R1 εκτελούμε:

```
interface em0  
no ipn6 nd suppress-ra
```

2.17

Στον R1 εκτελούμε:

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

2.18

Στον R2 εκτελούμε:

```
interface em1
no ipv6 nd suppress-ra
```

2.19

Στον R2 εκτελούμε:

```
interface em1
ipv6 nd prefix fd00:2::/64
```

2.20

Στο PC1 εκτελούμε:

```
route -6 delete default
```

2.21

Στον R1 εκτελούμε `tcpdump -eni em0 icmp6`.

2.22

Στο PC1 εκτελούμε `service netif restart`.

2.23

Ανταλλάσσονται τα μηνύματα:

Type	MAC source	MAC destination	IPv6 source	IPv6 destination
NS	08:00:27:96:00:DB	33:33:FF:96:00:DB	::	FF02::1:FF96:DB
RS	08:00:27:96:00:DB	33:33:00:00:00:02	FE80::A00:27FF:FE96:DB	FF02::2
RA	08:00:27:03:C1:F6	33:33:00:00:00:01	FE80::A00:27FF:FE03:C1F6	FF02::1
NS	08:00:27:96:00:DB	33:33:FF:96:00:DB	::	FF02::1:FF96:DB

2.24

Παράγει 2 μηνύματα NS. Το πρώτο από αυτά χρειάζεται για την αυτόματη απόδοση διεύθυνσης και το δεύτερο για την ανίχνευση ταυτόσημων διευθύνσεων.

2.25

Χρησιμοποιεί την :: (Μη ορισμένη/Unspecified) και στα δύο, στο πρώτο NS επειδή ο PC1 δεν έχει αποκτήσει ακόμα διεύθυνση, και στο δεύτερο NS επειδή κατά τη διαδικασία DAD χρησιμοποιείται πάντα η ακαθόριστη διεύθυνση ::.

2.26

Χρησιμοποιεί την fe80::a00:27ff:fe96:db.

2.27

Είναι:

Type	IPv6 destination address
NS	ff02::1:ff96:db
RS	ff02::2
RA	ff02::1
NS	ff02::1:ff96:db

- Το NS έχει αυτή τη διεύθυνση επειδή αναφέρεται στην τοπική ζεύξη → στον PC1
- Το RS έχει αυτή τη διεύθυνση επειδή αναφέρεται σε όλους τους δρομολογητές στην τοπική ζεύξη.
- Το RA έχει αυτή τη διεύθυνση επειδή αναφέρεται σε όλους τους κόμβους της τοπικής ζεύξης.
- Τέλος, το δεύτερο NS έχει διεύθυνση ίδια με το παραπάνω NS.

2.28

Είναι:

Type	MAC destination address
NS	33:33:ff:96:00:db
RS	33:33:00:00:00:02
RA	33:33:00:00:00:01
NS	33:33:ff:96:00:db

Οι παραπάνω διευθύνσεις MAC προορισμού προκύπτουν από την παράθεση του προθέματος 33:33: να ακολουθείται από τα τελευταία 32 bit της IPv6 διεύθυνσης προορισμού, επομένως:

NS

ff 02 00 00 00 00 00 00 00 00 00 01 ff 96 00 db	(IPv6 destination address)
ff 96 00 db	(Last 32 bits)
33 33 ff 96 00 db	(Add 33 33 prefix)
33:33:ff:96:00:db	(Final MAC address)

RS

ff 02 00 00 00 00 00 00 00 00 00 00 00 00 02	(IPv6 destination address)
00 00 00 02	(Last 32 bits)
33 33 00 00 00 02	(Add 33 33 prefix)
33:33:00:00:00:02	(Final MAC address)

RA

ff 02 00 00 00 00 00 00 00 00 00 00 00 00 01	(IPv6 destination address)
00 00 00 01	(Last 32 bits)
33 33 00 00 00 01	(Add 33 33 prefix)
33:33:00:00:00:01	(Final MAC address)

NS

ff 02 00 00 00 00 00 00 00 00 01 ff 96 00 db	(IPv6 destination address)
ff 96 00 db	(Last 32 bits)
33 33 ff 96 00 db	(Add 33 33 prefix)
33:33:ff:96:00:db	(Final MAC address)

2.29

Στο PC1 εκτελούμε `ndp -p`. Έχουμε:

```
fd00:1::/64 if=em0
flags=LA0 vlttime=2592000, pltime=604800, expire=29d23h51m54s, ref=1
  advertised by
    fe80::a00:27ff:fe03:c1f6%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
```

Πλέον έχει μαθευτεί το πρόθεμα `fd00:1::/64` που διαφημίζει ο R1 (`fe80::a00:27ff:fe03:c1f6`).

2.30

Στο PC1 εκτελούμε `ifconfig em0`. Το PC1 έχει λάβει αυτόματα μέσω του SLAAC τη διεύθυνση `fd00:1::a00:27ff:fe96:db`.

2.31

Στο PC1 εκτελούμε `netstat -rn6`. Έχουμε:

Destination	Gateway	Flags	Netif
::/96	::1	UGRS	lo0
default	fe80::a00:27ff:fe03:c1f6%em0	UG	em0
::1	link#2	UH	lo0
::ffff:0.0.0.0/96	::1	UGRS	lo0
fd00:1::/64	link#1	U	em0
fd00:1::a00:27ff:fe96:db	link#1	UHS	lo0
fe80::/10	::1	UGRS	lo0
fe80::%em0/64	link#1	U	em0
fe80::a00:27ff:fe96:db%em0	link#1	UHS	lo0
fe80::%lo0/64	link#2	U	lo0
fe80::1%lo0	link#2	UHS	lo0
ff02::/16	::1	UGRS	lo0

Υπάρχει προκαθορισμένη διαδρομή (είναι η δεύτερη εγγραφή στον παραπάνω πίνακα). Προέκυψε από τα μηνύματα Router advertisement που έστειλε ο R1.

2.32

Μπορούμε να χρησιμοποιήσουμε τις διευθύνσεις:

```
### Ping PC2 --> PC1 ###
```

```
fd00:1::a00:27ff:fe96:db
```

```
### Ping R1 --> PC1 ###
```

```
fd00:1::a00:27ff:fe96:db
```

```
fe80::a00:27ff:fe96:db%em0
```

Άσκηση 3: Δυναμική δρομολόγηση IPv6

3.1

Εκτελούμε:

```
### R1 ###
```

```
no ipv6 route fd00:2::/64 fd00:3::2
```

```
### R2 ###
```

```
no ipv6 route fd00:1::/64 fd00:3::1
```

3.2

Εκτελούμε:

```
### R1 ###
```

```
router ripng  
network em0  
network em1
```

```
### R2 ###
```

```
router ripng  
network em0  
network em1
```

3.3

Στον R1 εκτελούμε `do show ipv6 route ripng`. Βλέπουμε μια εγγραφή:

```
R>* fd00:2::/64 [120/2] via fe80::a00:27ff:fed1:fb31, em1
```

3.4

Είναι `fe80::a00:27ff:fed1:fb31` και είναι τοπική στη ζεύξη διεύθυνση (link-local).

3.5

Στο PC1 εκτελούμε `ping6 fd00:2::2`. Το ping είναι επιτυχές.

3.6

Στον R1 εκτελούμε `tcpdump -vni em1 ip6`.

3.7

Παρατηρούμε πακέτα `ripng-resp`. Η διεύθυνση προορισμού τους είναι η multicast διεύθυνση `ff02::9` (η αντίστοιχη της `224.0.0.9` για το RIPv2).

3.8

Έχει τιμή 255, προκειμένου να μην μπορούν τα `ripng-resp` πακέτα να διέλθουν από δρομολογητές.

3.9

Χρησιμοποιεί το UDP και τη θύρα 521, ενώ το RIP χρησιμοποιεί πάλι το UDP, αλλά τη θύρα 520.

3.10

Εκτελούμε:

```
### R1 ###
```

```
no router ripng
```

```
### R2 ###
```

```
no router ripng
```

3.11

Εκτελούμε:

```
### R1 ###
```

```
do write file
```

```
### R2 ###
```

```
do write file
```

3.12

Εκτελούμε:

```
### R1 ###
```

```
service frr restart
```

```
### R2 ###
```

```
service frr restart
```

3.13

Εκτελούμε:

```
### R1 ###
```

```
router ospf6
```

```
router-id 1.1.1.1
```

```
ospf router-id 1.1.1.1
```

```
### R2 ###
```

```
router ospf6
```

```
router-id 2.2.2.2
```

```
ospf router-id 2.2.2.2
```


3.14

Στον R1 εκτελούμε:

```
router ospf6
interface em0 area 0.0.0.0
interface em1 area 0.0.0.0
```

3.15

Στον R2 εκτελούμε:

```
router ospf6
interface em0 area 0.0.0.0
interface em1 area 0.0.0.0
```

3.16

Στον R2 εκτελούμε `do show ipv6 route ospf`. Βλέπουμε 2 εγγραφές:

```
O>* fd00:1::/64 [110/200] via fe80::a00:27ff:febb:4046, em0
O   fd00:2::/64 [110/100] is directly connected, em1
```

Το κόστος τους προέκυψε ως το άθροισμα του κόστους όλων των ζεύξεων της εκάστοτε διαδρομής. Το κόστος μιας ζεύξης υπολογίζεται ως:

$$\frac{\text{Reference bandwidth}}{\text{Interface bandwidth}} = \frac{100 \text{ Mbps}}{1 \text{ Mbps}} = 100$$

όπου ως Reference bandwidth χρησιμοποιείται η προκαθορισμένη τιμή (εφόσον δεν την αλλάξαμε με κάποιο τρόπο), ενώ το Interface bandwidth μπορεί να βρεθεί από την εκτέλεση της `do show interface` στους R1 και R2. Έτσι, το κόστος προς το δίκτυο `fd00:1::/64` (πρώτη εγγραφή) είναι 200 (διαδρομή $R2 \rightarrow R1 \rightarrow PC1$, 2 ζεύξεις), ενώ το κόστος προς το δίκτυο `fd00:2::/64` (δεύτερη εγγραφή) είναι 100 (διαδρομή $R2 \rightarrow PC2$, 1 ζεύξη).

3.17

Είναι η `fe80::a00:27ff:febb:4046` (R1@em1), και είναι διεύθυνση τοπική στη ζεύξη (link-local).

3.18

Στον R2 εκτελούμε `tcpdump -vni em0 ip6`.

3.19

Παρατηρούμε πακέτο OSPFv3 Hello, με προορισμό τη διεύθυνση `ff02::5` (η αντίστοιχη της `224.0.0.5` για το OSPFv2).

3.20

Έχει τιμή 1.

3.21

Χρησιμοποιεί τον αριθμό 89 (OSPF), ίδιο με τον OSPFv2.

3.22

Στο PC2 εκτελούμε `ping6 fd00:1::a00:27ff:fe96:db`. Το ping είναι επιτυχές.

3.23

Εκτελούμε:

```
### R1 ###
```

```
no router ospf6
```

```
### R2 ###
```

```
no router ospf6
```

3.24

Εκτελούμε:

```
### R1 ###
```

```
service frr restart
```

```
### R2 ###
```

```
service frr restart
```

3.25

Στον R1 εκτελούμε:

```
router-id 1.1.1.1
```

```
router bgp 65010
```

3.26

Στον R1 εκτελούμε:

```
no bgp ebgp-requires-policy
```

3.27

Στον R1 εκτελούμε `no bgp default ipv4-unicast`.

3.28

Στον R1 εκτελούμε `neighbor fd00:3::2 remote-as 65020`.

3.29

Στον R1 εκτελούμε `address-family ipv6`.

3.30

Στον R1 εκτελούμε `network fd00:1::/64`.

3.31

Στον R1 εκτελούμε `neighbor fd00:3::2 activate` και εξερχόμαστε με `exit`.

3.32

Επαναλαμβάνουμε τα προηγούμενα για τον R2:

```
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

Στον R1 εκτελούμε `do show ipv6 route bgp`. Βλέπουμε 1 εγγραφή:

```
B>* fd00:1::/64 [20/0] via fe80::a00:27ff:febb:4046, em0
```

3.34

Είναι η `fe80::a00:27ff:fed1:fb31` (R2@em0), και είναι διεύθυνση τοπική στη ζεύξη (link-local).

3.35

Στον R1 εκτελούμε `tcpdump -vni em1`.

3.36

Παρατηρούμε μηνύματα BGP KEEPALIVE. Χρησιμοποιείται το TCP και η θύρα 179, όπως και στην περίπτωση του IPv4.

3.37

Έχει τιμή 1.

3.38

Στο PC1 εκτελούμε `ping6 fd00:2::a00:27ff:fe25:d99f`. Το ping είναι επιτυχές.

3.39

Στο PC1 εκτελούμε:

```
reboot
```

```
vtysh
```

```
configure terminal
```

```
interface em0
```

```
ipv6 address fd00:1::2/64
```

3.40

Στο PC1 εκτελούμε:

```
router-id 1.1.0.0
```

```
router bgp 65010
```

3.41

Στο PC1 εκτελούμε `no bgp default ipv4-unicast`.

3.42

Στο PC1 εκτελούμε `neighbor fd00:1::1 remote-as 65010`.

3.43

Στο PC1 εκτελούμε:

```
address-family ipv6
```

```
neighbor fd00:1::1 activate
```

```
exit
```

3.44

Στον R1 εκτελούμε:

```
router bgp 65010
```

```
neighbor fd00:1::2 remote-as 65010
```

3.45

Στον R1 εκτελούμε:

```
address-family ipv6
neighbor fd00:1::2 activate
neighbor fd00:1::2 next-hop-self
exit
```

3.46

Με την εντολή `do show ip bgp neighbors fd00:1::2`. Στο τέλος της πρώτης γραμμής αναγράφεται "internal link", ενώ το BGP state είναι Established.

3.47

Στο PC1 εκτελούμε `do show ipv6 route bgp`. Βλέπουμε 2 εγγραφές:

```
B   fd00:1::/64 [200/0] via fe80::a00:27ff:fe03:c1f6, em0
B>* fd00:2::/64 [200/0] via fd00:1::1, em0
```

3.48

Δεν είναι επιλεγμένη επειδή υπάρχει η προτιμότερη εγγραφή στον πίνακα δρομολόγησης:

```
C>* fd00:1::/64 [0/1] is directly connected, em0
```

3.49

Είναι η `fd00:1::1 (R1@em0)`, και είναι μοναδική τοπική διεύθυνση.

3.50

Στο PC2 εκτελούμε `ping6 fd00:1::2`. Το ping είναι επιτυχές.

Άσκηση 4: Μηχανισμός μετάβασης 464 XLAT

4.1

Στον R1 εκτελούμε:

```
vtysh
configure terminal

interface em0
ip address 192.168.1.1/24
```

4.2

Στον R2 εκτελούμε:

```
vttysh
configure terminal

interface em1
ip address 192.168.2.1/24
```

4.3

Στο PC1 εκτελούμε:

```
vttysh
configure terminal

interface em0
ip address 192.168.1.2/24

ip route 0.0.0.0/0 192.168.1.1
```

4.4

Στο PC2 εκτελούμε:

```
vttysh
configure terminal

interface em0
ip address 192.168.2.2/24

ip route 0.0.0.0/0 192.168.2.1
```

4.5

Στον R1 εκτελούμε:

```
sysrc firewall_enable="YES"
sysrc firewall_nat64_enable="YES"
sysrc firewall_type="open"
sysrc firewall_logif="YES"
```

4.6

Στον R1 εκτελούμε:

```
service ipfw start
```

4.7

Στον R1 εκτελούμε:

```
ipfw list
```

Εμφανίζονται 12 κανόνες:

```
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

Στο PC1 εκτελούμε `do ping ipv6 fd00:2::a00:27ff:fe25:d99f/64`. Το ping είναι επιτυχές.

4.9

Στον R1 εκτελούμε (όλο μαζί μία εντολή):

```
ipfw nat64clat nat64 create \
clat_prefix fd00:3:1::/96 \
plat_prefix 64:ff9b::/96 \
allow_private log
```

4.10

Στον R1 εκτελούμε:

```
ipfw add 2000 nat64clat nat64 ip4 from any to not me recv em0
```

4.11

Στον R1 εκτελούμε:

```
ipfw add 3000 nat64clat nat64 ip6 from 64:ff9b::/96 to fd00:3:1::/96 recv em1
```

4.12

Στον R1 εκτελούμε:

```
ipv6 route 64:ff9b::/96 fd00:3::2
```

4.13

Στον R2 εκτελούμε:

```
sysrc firewall_enable="YES"
sysrc firewall_nat64_enable="YES"
sysrc firewall_type="open"
sysrc firewall_logif="YES"
service ipfw start
```

4.14

Στον R2 εκτελούμε (όλο μαζί μία εντολή):

```
ipfw nat64lsn nat64 create \
prefix4 2.2.2.0/24          \
prefix6 64:ff9b::/96        \
allow_private log
```

4.15

Στον R2 εκτελούμε:

```
ipfw add 2000 nat64lsn nat64 ip6 from fd00:3:1::/96 to 64:ff9b::/96 recv em0
```

4.16

Στον R2 εκτελούμε:

```
ipfw add 3000 nat64lsn nat64 ip4 from any to 2.2.2.0/24 recv em1
```

4.17

Στον R2 εκτελούμε:

```
ipn6 route fd00:3:1::/96 fd00:3::1
```

4.18

Στον R2 εκτελούμε:

```
ip route 0.0.0.0/0 192.168.2.2
```

4.19

Στο PC1 εκτελούμε:

```
do ping 192.168.1.1
do ping 192.168.2.2
```

Τα ping είναι επιτυχή.

4.20

Στον R1 εκτελούμε:

```
ifconfig ipfwlog0 create
tcpdump -i ipfwlog0
```

4.21

Στον R2 εκτελούμε:

```
ifconfig ipfwlog0 create
tcpdump -i ipfwlog0
```

4.22

Στο PC1 εκτελούμε `ping -c 1 192.168.2.2`. Παρατηρούμε τα πακέτα:

R1 capture

Type	Source	Destination
ICMP echo request	192.168.1.2	192.168.2.2
ICMPv6 echo request	fd00:3:1::c0a8:102	64:ff9b::c0a8:202
ICMPv6 echo reply	64:ff9b::c0a8:202	fd00:3:1::c0a8:102
ICMP echo reply	192.168.2.2	192.168.1.2

R2 capture

Type	Source	Destination
ICMPv6 echo request	fd00:3:1::c0a8:102	64:ff9b::c0a8:202
ICMP echo request	2.2.2.158	192.168.2.2
ICMP echo reply	192.168.2.2	2.2.2.158
ICMPv6 echo reply	64:ff9b::c0a8:202	fd00:3:1::c0a8:102

4.23

Στο PC2 εκτελούμε:

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

4.24

Στο PC1 εκτελούμε:

```
do ping 172.17.17.2
do ping 10.0.0.2
```

Και τα δύο ping είναι επιτυχή.

4.25

Στον R2 εκτελούμε `ipfw nat64lsn nat64 show states`.

4.26

Εκτελούμε:

```
### PC1 ###
```

```
ping -c 1 172.17.17.2
```

```
ping -c 1 10.0.0.2
```

```
### R2 ###
```

```
ipfw nat64lsn nat64 show states
```

Εμφανίζονται οι εγγραφές:

```
fd00:3:1::c0a8:102 2.2.2.65 ICMPv6 <time> 172.17.17.2
```

```
fd00:3:1::c0a8:102 2.2.2.65 ICMPv6 <time> 10.0.0.2
```

όπου <time> είναι ο χρόνος που πέρασε από την καταχώρηση της εγγραφής. Αυτές οι εγγραφές αφορούν τις μεταφράσεις που σχετίζονται με τα δύο παραπάνω ping. Με διαδοχικές εκτελέσεις της:

```
ipfw nat64lsn nat64 show states
```

διαπιστώνουμε ότι οι εγγραφές διαρκούν 62 δευτερόλεπτα.

Άσκηση 5: Μηχανισμός μετάβασης Teredo

5.1

Στα PC1 και PC2 εκτελούμε:

```
dhclient em0 # Got 10.0.2.15
```

```
ping www.google.com # To test internet access
```

5.2

Στα PC1 και PC2 εκτελούμε:

```
pkg install miredo
```

5.3

Στα PC1 και PC2 εκτελούμε:

```
sysrc miredo_enable="YES"
```

5.4

Στα PC1 και PC2 εκτελούμε:

```
service miredo start
```

5.5

Στο PC1 εκτελούμε `ifconfig`. Εμφανίζεται η νέα διεπαφή:

```
Name: teredo
```

```
IPv6 address: 2001:0:c38c:c38c:14d3:5860:b07d:1bdc/128
```

5.6

Στο PC1 εκτελούμε `tcpdump -ni em0`.

5.7

Είναι η 195.140.195.140.

5.8

Χρησιμοποιείται το UDP, ενώ στον εξυπηρετητή Teredo αντιστοιχεί η θύρα 3544.

5.9

Ξεκινάμε μία καταγραφή με Wireshark στην φυσική κάρτα του υπολογιστή μας (διεπαφή `enp2s0`), με φίλτρο απεικόνισης `teredo`. Παρατηρούμε μηνύματα του πρωτοκόλλου ICMPv6, συγκεκριμένα Router Solicitation/Advertisement.

5.10

Στο PC1 εκτελούμε:

```
ping6 www.ntua.gr
```

```
ping6 www.ibm.com
```

```
ping6 www.google.com
```

Μπορούμε να κάνουμε `ping` μόνο στο `www.ibm.com`.

5.11

Στο PC1 εκτελούμε `ping6 www.ibm.com` και το αφήνουμε να τρέχει.

5.12

Παρατηρούμε μηνύματα IPv6 no next header.

5.13

Όχι, δεν παρατηρούμε.

5.14

Παρατηρούμε το IPv4, ενώ η θύρα που αντιστοιχεί στον αναμεταδότη Teredo είναι η 3545.

5.15

Σταματάμε τις καταγραφές και ύστερα στο PC1 εκτελούμε `tcpdump -ni teredo`.

5.16

Ναι, παράγονται ICMPv6 echo request/reply.

5.17

Στο PC1 εκτελούμε:

```
ping6 2001:0:c38c:c38c:2cdb:2a24:b07d:1bdc
```

Δεν λαμβάνουμε απάντηση.

5.18

Ναι, παράγονται ICMPv6 echo request.

5.19

Στο PC1 εκτελούμε `tcpdump -ni em0`. Δεν παράγονται δεδομενογράμματα UDP που να αντιστοιχούν στα ICMPv6 μηνύματα.

5.20

Στο PC1 εκτελούμε `ping6 www.hp.com` και μετά `ping6 www.f5.com`. Επιλέγεται διαφορετικός teredo relay κάθε φορά:

```
ping www.hp.com: 185.218.193.138  
ping www.f5.com: 216.66.80.238
```

5.21

Στο PC1 εκτελούμε `ping6 one.one.one.one`. Παρατηρούμε ότι ο teredo relay που επιλέγεται είναι ίδιος με τον teredo server που επιλέχθηκε στο ερώτημα 5.7.