

ΓΕΩΡΓΑΚΟΠΟΥΛΟΣ ΓΕΩΡΓΙΟΣ 03120827

ΑΝΑΦΟΡΑ 11ΗΣ ΕΡΓΑΣΤΗΡΙΑΚΗΣ ΑΣΚΗΣΗΣ

Ομάδα: 1

Λογισμικό: Linux Ubuntu 22.04

Όνομα PC: glaptop

ΑΣΚΗΣΗ 1: Εισαγωγή στο IPv6

1.1 Θα επεξεργαστώ το αρχείο με την εντολή vi.

Θα προσθέσω την γραμμή: `ifconfig_em0_ipv6="inet6 accept_rtadv"`

1.2 Θα χρησιμοποιήσω τις εντολές: `service netif stop / service netif start`

1.3 Στην em0 του PC1 έχει αποδοθεί η: `fe80::a00:27ff:fe6c:de32%em0`

1.4 Στην em0 του PC2 έχει αποδοθεί η: `fe80::a00:27ff:feeb:c325%em0`

1.5 Οι διευθύνσεις αυτές είναι link-local. Προκύπτουν ως εξής:

Στην αρχή έχουμε το `fe80::`

Επειτα βάζουμε την MAC αλλά αντιστρέφοντας το 7ο bit του 1ου byte.

Στο μέσο της MAC παρεμβάλλουμε επίσης το `ff:fe`

1.6 Θα χρησιμοποιήσω την εντολή: `netstat -r6`

Παρατηρώ 9 εγγραφές στον πίνακα δρομολόγησης

1.7 Μόνο 1 εγγραφή αφορά την διεπαφή em0.

1.8 Οι εγγραφές σχετικές με αυτό το πρόθεμα δικτύου είναι:

`fe80::%em0/64`

`fe80::%lo0/64`

Οι αντίστοιχες διεπαφές εξόδου είναι em0 και lo0

1.9 Θα χρησιμοποιήσω την εντολή: `ping6 ::1`

Παρατηρώ ότι απαντάει το ίδιο το PC1 εφόσον η `::1` είναι η loopback διεύθυνση

1.10 Θα χρησιμοποιήσω την εντολή: `ping6 fe80::a00:27ff:fe6c:de32%em0`
(Για να δουλέψει πρέπει στο τέλος να προσθέσω και τον δείκτη ζώνης)

1.11 Πρέπει και πάλι στο τέλος να προσθέσουμε τον δείκτη ζώνης

Αρα η εντολή είναι: `ping6 fe80::a00:27ff:feeb:c325%em0`

1.12 Παρατηρώ ότι απαντάει το ίδιο το PC1, απο όπου κάνω το `ping6`, καθώς η διεύθυνση `ff01::1%1%em0` είναι όλοι οι κόμβοι στη διεπαφή em0

1.13 Παρατηρώ ότι απαντάει το PC2, καθώς η διεύθυνση `ff02::1%1%em0` είναι όλοι οι κόμβοι στην τοπική ζεύξη που είναι συνδεδεμένοι με την em0

1.14 Θα χρησιμοποιήσω την εντολή: `ifconfig em0 inet6 fd00:1::2/64`

- 1.15 Θα χρησιμοποιήσω την εντολή: `ifconfig em0 inet6 fd00:1::3/64`
- 1.16 Οι διευθύνσεις αυτές είναι ULA (Unique Local Address), δεν δρομολογούνται στο δημόσιο δίκτυο και είναι ανάλογες των 10.0.0.0/8, 172.16.0.0/12 και 192.168.0.0/16 στο IPv4
- 1.17 Πλέον στις διεπαφές em0 των PC υπάρχουν 2 διευθύνσεις
- 1.18 Παρατηρώ ότι έχουν προστεθεί 2 νέες εγγραφές
(Τις fd00:1::/64 και fd00:1::X)
- 1.19 Θα πρέπει να προσθέσω στα αρχεία /etc/hosts τα εξής:
fd00:1::3 PC2 (στο αρχείο του PC1)
fd00:1::2 PC1 (στο αρχείο του PC2)
- 1.20 Ναι, με την εντολή: `ping6 PC2`
- 1.21 Θα χρησιμοποιήσω την εντολή: `apr -a`
Παρατηρώ ότι δεν υπάρχει καμία εγγραφή
- 1.22 Για να μελετήσω την εντολή θα χρησιμοποιήσω το: `man ndp`
- 1.23 Θα χρησιμοποιήσω την εντολή: `ndp -a`
- 1.24 Στην Neighbor Cache του PC1 παρατηρώ 4 εγγραφές. Οι 2 εγγραφές που αφορούν το PC1 βρίσκονται σε κατάσταση R (Reachable) και οι 2 εγγραφές που αφορούν το PC2 βρίσκονται σε κατάσταση S (State)
- 1.25 Για την λίστα προθεμάτων θα χρησιμοποιήσω την εντολή: `ndp -p`
- 1.26 Θα χρησιμοποιήσω την εντολή: `ndp -c`
- 1.27 Θα χρησιμοποιήσω την εντολή: `tcpdump -v -n`
- 1.28 Παρατηρώ 6 πακέτα IPv6
- 1.29 Τα πακέτα αυτά μεταφέρουν ICMP μηνύματα.
Next Header = ICMPv6 (58)
- 1.30 Σειρά αποστολής και Τύπος μηνυμάτων:
PC1 to PC2: neighbor solicitation
PC2 to PC1: neighbor advertisement
PC1 to PC2: Echo Request
PC2 to PC1: Echo Reply
PC2 to PC1: neighbor solicitation
PC1 to PC2: neighbor advertisement
- 1.31 Το πρώτο πακέτο NS έχει για προορισμό την διεύθυνση ff02::111:ff00:3. Αυτή είναι multicast διεύθυνση Solicited Node που χρησιμοποιείται όταν δεν είναι γνωστή η εκ των προτέρων η διεύθυνση αποστολής. Προκύπτει από το πρόθεμα ff02:0:0:0:0:1:ff00:0/104 μαζί με τα 24 τελευταία bit της διεύθυνσης unicast που χρησιμοποιήθηκε (δηλαδή του PC2)
- 1.32 Το δεύτερο πακέτο NS έχει για προορισμό την διεύθυνση fd00:1::2. Αυτή είναι η unicast διεύθυνση του PC1
- 1.33 Η κατάσταση της εγγραφής είναι S και έχει διάρκεια ζωής 24 ώρες
- 1.34 Παρατηρώ τις καταστάσεις R και S.
- 1.35 Η κατάσταση R έχει διάρκεια ζωής 30 δευτερόλεπτα. Μόλις αυτός ο χρόνος λήξει, η κατάσταση της εγγραφής γίνεται S
- 1.36 Η κατάσταση S έχει διάρκεια ζωής 1 ημέρα. Ωστόσο την παρατηρούμε για περίπου 5 δευτερόλεπτα.
- 1.37 Τώρα όπου δεν τρέχει το `ping6`, η εγγραφή έχει μείνει στην κατάσταση S

- 1.38** Παρατηρώ μηνύματα NS και NA κάθε περίπου 10 με 15 δευτερόλεπτα. Τα NS μηνύματα παράγονται καθώς ζητείται να επιβεβαιωθεί η προσβασιμότητα του γείτονα και τα NA παράγονται ως απάντηση στα NS

ΑΣΚΗΣΗ 2: SLAAC και Στατική δρομολόγηση στο IPv6

- 2.1** Θα χρησιμοποιήσω τις εντολές:
sysrc ipv6_gateway_enable="YES"
service routing restart
- 2.2** Θα χρησιμοποιήσω τις εντολές:
ifconfig em0 inet6 fd00:1::3/64 delete
ifconfig em0 inet6 fd00:2::2/64
- 2.3** Θα χρησιμοποιήσω τις εντολές:
vtysh
configure terminal
interface em0
ip address fd00:1::1/64
- 2.4** Θα χρησιμοποιήσω τις εντολές:
interface em0
ip address fd00:3::1/126
- 2.5** Θα χρησιμοποιήσω τις εντολές:
vtysh
configure terminal
interface em1
ip address fd00:2::1/64
- 2.6** Θα χρησιμοποιήσω τις εντολές:
interface em0
ip address fd00:3::2/126
- 2.7** Θα χρησιμοποιήσω την εντολή: route -6 add default fd00:1::1
- 2.8** Θα χρησιμοποιήσω την εντολή: route -6 add default fd00:2::1
- 2.9** Θα χρησιμοποιήσω την εντολή: tcpdump -i em0 -v -n
- 2.10** Θα χρησιμοποιήσω τις εντολές:
ndp -c
ping6 -c 1 fd00:2::2/64
Παρατηρώ ότι το ping6 δεν είναι επιτυχές, καθώς ο R1 δεν έχει εγγραφή στον πίνακα του για το PC2 (δηλαδή για το LAN2)
- 2.11** Παρατηρώ ότι παράγονται μηνύματα NS από το PC1 για το R1, για τα οποία υπάρχει και η αντίστοιχη NS απάντηση. Επειτα παρατηρώ echo request από το PC1 και Destination Unreachable από τον R1. Τέλος παρατηρώ NS από τον R1 προς το PC1 και την αντίστοιχη NS απάντηση
- 2.12** Θα χρησιμοποιήσω την εντολή: ipv6 route fd00:2::0/64 fd00:3::2
- 2.13** Όχι, ακόμα το ping δεν δουλεύει καθώς δεν υπάρχει εγγραφή στον R2 για το LAN1
- 2.14** Θα χρησιμοποιήσω την εντολή: ipv6 route fd00:2::0/64 fd00:3::2

- 2.15** Ναι, τώρα το ping6 δουλεύει με επιτυχία
- 2.16** Θα χρησιμοποιήσω τις εντολές:
interface em0
no ipv6 nd suppress-ra
- 2.17** Θα χρησιμοποιήσω τις εντολές:
interface em0
ipv6 nd prefix fd00:1::/64
- 2.18** Θα χρησιμοποιήσω τις εντολές:
interface em1
no ipv6 nd suppress-ra
- 2.19** Θα χρησιμοποιήσω τις εντολές:
interface em0
ipv6 nd prefix fd00:2::/64
- 2.20** Θα χρησιμοποιήσω την εντολή: route -6 delete default fd00:1::1
- 2.21** Θα χρησιμοποιήσω την εντολή: tcpdump -i em0 -v -n -e
- 2.22** Θα χρησιμοποιήσω την εντολή: service netif restart
- 2.23** Παρατηρώ μηνύματα RS, RA και NS
- 2.24** Το μήνυμα NS παράγεται απο το PC1 διότι θέλει να επιβεβαιώσει οτι δεν έχει την ίδια IPv6 με κάποιο άλλο μηχάνημα σε αυτό το δίκτυο
- 2.25** Χρησιμοποιεί την ακαθόριστη διεύθυνση ::0 καθώς ακόμα δεν έχει οριστικοποιηθεί η διεύθυνση που του έχει αποδοθεί
- 2.26** Χρησιμοποιεί την link-local διεύθυνση του
- 2.27** Οι διευθύνσεις είναι:
NS: Solicited Node καθώς θέλει να ελέγξει αν αυτή χρησιμοποιείται και αλλού
RS: Την ff02:2 που είναι όλοι οι δρομολογητές αυτής της ζευξης
RA: Την ff02:1 που είναι όλοι οι κόμβοι της ζευξης
- 2.28** Οι διευθύνσεις MAC προορισμού των πλαισίων είναι:
NS: 33:33:ff:6c:de:32
RS: 33:33:00:00:00:02
RA: 33:33:00:00:00:01
Οι διευθύνσεις αυτές προκύπτουν απο το πρόθεμα 33:33 και τα 32 τελευταία bit της IPv6 διεύθυνσης προορισμού (εφόσον πρόκειται για multicast διευθύνσεις)
- 2.29** Θα χρησιμοποιήσω την εντολή: nrd -p
Οι σημαίες που παρατηρώ είναι οι LAO και απο αυτές έχουμε ότι το prefix αυτό μπορεί να χρησιμοποιηθεί για stateless address autoconfiguration και για on-link determination
- 2.30** Το PC1 λόγω του SLAAC έχει λάβει αυτόματα τις διευθύνσεις:
fe80::a00:27ff:fe6c:de32%em0
fe80:1::a00:27ff:fe6c:de32%em0
fe80::1%lo0
- 2.31** Ναι υπάρχει προκαθορισμένη διαδρομή, η οποία προέκυψε απο το SLAAC και είναι η link-local διεύθυνση του R1 στο LAN1
- 2.32** Απο το PC2 μπορώ να χρησιμοποιήσω την fe80::a00:27ff:fe6c:de32%em0

Απο το R1 μπορω τις fe80::a00:27ff:fe6c:de32%em0 και
fd00:1::a00:27ff:fe6c:de32%em0

ΑΣΚΗΣΗ 3: Δυναμική δρομολόγηση IPv6

3.1 Θα χρησιμοποιήσω τις εντολές:

R1 : no ipv6 route fd00:2::/64 fd00:3::2

R2 : no ipv6 route fd00:1::/64 fd00:3::1

3.2 Θα χρησιμοποιήσω τις εντολές:

router ripng

network em0

network em1

3.3 Θα χρησιμοποιήσω την εντολή: do show ipv6 route ripng

Παρατηρώ μια εγγραφή για το LAN2 μέσω του R2

3.4 Η διεύθυνση του επόμενου κόμβου είναι η link-local του R2 στο WAN και
έχει τιμή fe80::a00:27ff:feb1:8c30

3.5 Ναι το ping6 πετυχαίνει

3.6 Θα χρησιμοποιήσω την εντολή: tcpdump -i em1 -v -n -e

3.7 Παρατηρώ πακέτα ripng-resp 2 με διεύθυνση προορισμού την ff02::9
(multicast διεύθυνση για routers)

3.8 Το Hop Limit έχει τιμή 255 η οποία είναι η μέγιστη τιμή που μπορεί να
πάρει. Τα μηνύματα με hop limit μικρότερο του 255 αγνοούνται ώστε να
επιβεβαιώσουμε ότι δεν έχει περάσει από άλλον δρομολογητή

3.9 Το RIPng χρησιμοποιεί πρωτόκολλο UDP και την θύρα 521. Το RIP έχει
UDP αλλά χρησιμοποιεί την θύρα 520

3.10 Θα χρησιμοποιήσω την εντολή: no router ripng

3.11 Θα χρησιμοποιήσω την εντολή: do write file

3.12 Θα χρησιμοποιήσω την εντολή: service frr restart

3.13 Θα χρησιμοποιήσω τις εντολές:

router ospf6

ospf6 router-id 1.1.1.1

(και αντίστοιχα για το R2)

3.14 Θα χρησιμοποιήσω τις εντολές:

interface em0

ipv6 ospf6 area 0.0.0.0

exit

interface em1

ipv6 ospf6 area 0.0.0.0

3.15 Θα χρησιμοποιήσω τις εντολές:

interface em0

ipv6 ospf6 area 0.0.0.0

exit

interface em1

ipv6 ospf6 area 0.0.0.0

3.16 Θα χρησιμοποιήσω την εντολή: do show ip route ospf6

Παρατηρώ 3 εγγραφές. Το κόστος των εγγραφών προκύπτει από την διαίρεση του reference bandwidth με το interface bandwidth

3.17 Η διεύθυνση για τον επόμενο κόμβο είναι η link-local του R1 στο WAN1

3.18 Θα χρησιμοποιήσω την εντολή: `tcpdump ip6 -i em0 -v -n -e`

3.19 Παρατηρώ μηνύματα Hello με διεύθυνση προορισμού την ff02::5

3.20 Η τιμή του Hop Limit είναι 1

3.21 Ο αριθμός πρωτοκόλλου ανώτερου στρώματος είναι: Next Header = 89
(Ιδιο με αυτό του OSPFv2)

3.22 Ναι το ping πετυχαίνει

3.23 Θα χρησιμοποιήσω την εντολή: `no router ospf6`

3.24 Θα χρησιμοποιήσω την εντολή: `service frr restart`

3.25 Θα χρησιμοποιήσω τις εντολές:

`router-id 1.1.1.1`

`router bgp 65010`

3.26 Θα χρησιμοποιήσω την εντολή: `no bgp ebgp-requires-policy`

3.27 Θα χρησιμοποιήσω την εντολή: `no bgp default ipv4-unicast`

3.28 Θα χρησιμοποιήσω την εντολή: `neighbor fd00:3::2 remote-as 65020`

3.29 Θα χρησιμοποιήσω την εντολή: `address-family ipv6`

3.30 Θα χρησιμοποιήσω την εντολή: `network fd00:1::/64`

3.31 Θα χρησιμοποιήσω την εντολή: `neighbor fd00:3::2 activate`

3.32 Οι εντολές είναι οι αντίστοιχες με τις παραπάνω

3.33 Μπορούμε να το επιβεβαιώσουμε με την εντολή: `do show bgp neighbors`
Στην εγγραφή για το R2 παρατηρώ BGP State = Established

3.34 Θα χρησιμοποιήσω την εντολή: `do show ipv6 route bgp`

Παρατηρώ 1 δυναμική εγγραφή προς το LAN2

3.35 Η διεύθυνση του επόμενου κόμβου είναι η link-local του R2

3.36 Θα χρησιμοποιήσω την εντολή:

`do show ipv6 bgp neighbor fd00:3::2 advertised-routes`

Παρατηρώ ότι διαφημίζει 2 δίκτυα, τα LAN1 και LAN2 με επομένο βήμα το ::

	Network	Next Hop	Metric	LocPrf	Weight	Path
*>	fd00:1::/64	::	0		32768	i
*>	fd00:2::/64	::			0 65020	i

3.37 Θα χρησιμοποιήσω την εντολή: `tcpdump not icmp6 -i em1 -v -n`

3.38 Παρατηρώ μηνύματα Keep Alive. Αυτά χρησιμοποιούν πρωτόκολλο TCP και την θύρα 179. Παρατηρώ ότι είναι τα ίδια με το IPv4

3.39 Η τιμή του Hop Limit είναι 1

3.40 Ναι το ping6 είναι επιτυχές

3.41 Θα χρησιμοποιήσω τις εντολές:

`reboot`

`vttysh`

`configure terminal`

`router-id 1.1.0.0`

`interface em0`

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

- 3.42** Θα χρησιμοποιήσω την εντολή: `router bgp 65010`
- 3.43** Θα χρησιμοποιήσω την εντολή: `no bgp default ipv4-unicast`
- 3.44** Θα χρησιμοποιήσω την εντολή: `neighbor fd00:1::1 remote-as 65010`
- 3.45** Θα χρησιμοποιήσω τις εντολές:
`address-family ipv6`
`neighbor fd00:1::1 activate`
`exit`
- 3.46** Θα χρησιμοποιήσω την εντολή: `neighbor fd00:1::2 remote-as 65010`
- 3.47** Θα χρησιμοποιήσω τις εντολές:
`address-family ipv6`
`neighbor fd00:1::2 activate`
`neighbor fd00:1::2 next-hop-self`
- 3.48** Θα χρησιμοποιήσω την εντολή: `do show ipv6 bgp neighbors`
Στην εγγραφή για το PC1 παρατηρώ BGP State = Established
- 3.49** Θα χρησιμοποιήσω την εντολή: `do show ipv6 route bgp`
Παρατηρώ 2 εγγραφές (για τα LAN1 και LAN2). Επιλεγμένη είναι μόνο η εγγραφή για το LAN2
- 3.50** Η εγγραφή bgp για το LAN1 δεν είναι επιλεγμένη καθώς το PC1 είναι απευθείας συνδεδεμένο με το LAN1
- 3.51** Η διεύθυνση του επόμενου κόμβου για το LAN2 είναι η link-local διεύθυνση του R1 στο LAN1
- 3.52** Ναι, το ping6 είναι επιτυχές

ΑΣΚΗΣΗ 4: Μηχανισμός μετάβασης 464 XLAT

- 4.1** Θα χρησιμοποιήσω τις εντολές:
`interface em0`
`ip address 192.168.1.1/24`
- 4.2** Θα χρησιμοποιήσω τις εντολές:
`interface em0`
`ip address 192.168.2.1/24`
- 4.3** Θα χρησιμοποιήσω τις εντολές:
`interface em0`
`ip address 192.168.1.2/24`
`exit`
`ip route 0.0.0.0/0 192.168.1.1`
- 4.4** Θα χρησιμοποιήσω τις εντολές:
`interface em0`
`ip address 192.168.2.2/24`
`exit`
`ip route 0.0.0.0/0 192.168.2.1`
- 4.5** Χρησιμοποιώ την εντολή `vi /etc/rc.conf` για να τροποποιήσω το αρχείο
- 4.6** Θα χρησιμοποιήσω την εντολή: `service ipfw start`
- 4.7** Παρατηρώ 12 κανόνες (11 + default)
- 4.8** Ναι, το ping6 είναι επιτυχές

- 4.9** Θα χρησιμοποιήσω την εντολή:
`ipfw nat64clat nat64 create clat_prefix fd00:3:1::/96 plat_prefix 64:ff9b::/96 allow_private log`
- 4.10** Θα χρησιμοποιήσω την εντολή:
`ipfw add 2000 nat64clat nat64 ip4 from any to not me recv em0`
- 4.11** Θα χρησιμοποιήσω την εντολή:
`ipfw add 3000 nat64clat nat64 ip6 from 64:ff9b::/96 to fd00:3:1::/96 recv em1`
- 4.12** Θα χρησιμοποιήσω την εντολή: `ipn6 route 64:ff9b::/96 fd00:3::2`
- 4.13** Χρησιμοποιώ την εντολή `vi /etc/rc.conf` για να τροποποιήσω το αρχείο
- 4.14** Θα χρησιμοποιήσω την εντολή:
`ipfw nat64lsn nat64 create prefix4 2.2.2.0/24 prefix6 64:ff9b::/96 allow_private log`
- 4.15** Θα χρησιμοποιήσω την εντολή:
`ipfw add 2000 nat64lsn nat64 ip6 from fd00:3:1::/96 to 64:ff9b::/96 recv em0`
- 4.16** Θα χρησιμοποιήσω την εντολή:
`ipfw add 3000 nat64lsn nat64 ip4 from any to 2.2.2.0/24 recv em1`
- 4.17** Θα χρησιμοποιήσω την εντολή: `ip route fd00:3:1::/96 fd00:3::1`
- 4.18** Θα χρησιμοποιήσω την εντολή: `ip route 0.0.0.0/0 192.168.2.2`
- 4.19** Ναι, μπορώ να κάνουμε ping απο το PC1 στα R1 και PC2
- 4.20** Θα χρησιμοποιήσω τις εντολές:
`ifconfig ipfwlog0 create`
`tcpdump -i ipfwlog0`
- 4.21** Θα χρησιμοποιήσω τις εντολές:
`ifconfig ipfwlog0 create`
`tcpdump -i ipfwlog0`
- 4.22** Σε κάθε καταγραφή παρατηρώ 4 μηνύματα (2 Echo Request και 2 Echo Reply)
- 4.23** Θα χρησιμοποιήσω τις εντολές:
`interface em0`
`ip address 172.17.17.2/24`
`ip address 10.0.0.2/24`
- 4.24** Ναι, τα ping λειτουργουν
- 4.25** Το PC1 εμφανίζεται με την διεύθυνση 2.2.2.140 στα ping
- 4.26** Θα χρησιμοποιήσω την εντολή: `ipfw nat64lsn nat64 show states`
- 4.27** Ο πίνακας πριν και μετά τα ping είναι:

```

root@R2:~ # ipfw nat64lsn nat64 show states
root@R2:~ # ipfw nat64lsn nat64 show states
fd00:3:1::c0a8:102      2.2.2.140      ICMPv6        6      192.168.2.2
fd00:3:1::c0a8:102      2.2.2.140      ICMPv6        3      10.0.0.2
root@R2:~ #

```

Στην αρχή ο πίνακας είναι κενός. Μετά τα ping, παρατηρώ 2 εγγραφές (όσα και τα ping) οι οποίες μετα απο περίπου 1 λεπτό διαγράφονται

- 4.28** Η σύνδεση με ssh αποτυγχάνει λόγω του μεγέθους των πακέτων. Στην

- καταγραφή βλέπουμε μηνύματα Unreachable (need to fragment) και ICMP6 packet too big
- 4.29** Θα χρησιμοποιήσω την εντολή: `ifconfig em0 mtu 1480 up`. Τώρα η σύνδεση SSH πετυχαίνει

ΑΣΚΗΣΗ 5: Μηχανισμός μετάβασης Teredo

- 5.1** Θα χρησιμοποιήσω την εντολή: `dhclient em0`
Επιβεβαιώνω ότι υπάρχει πρόσβαση στο Internet καθώς το `ping www.google.com` λειτουργεί κανονικά
- 5.2** Θα χρησιμοποιήσω την εντολή: `pkg install miredo`
- 5.3** Θα προσθέσω συτή την γραμμή στο αρχείο `/etc/rc.conf` χρησιμοποιώντας την εντολή `sysrc`
- 5.4** Θα επεξεργαστώ αυτό το αρχείο με την εντολή `vi`. Έπειτα θα χρησιμοποιήσω την εντολή `service miredo start`
- 5.5** Παρατηρώ μια νέα διεπαφή με όνομα `teredo` και IPv6 διευθύνσεις: `fe80::ffff:ffff:ffff%teredo/64` και `2001:0:d911:c0d9:c3c:e3b6:920d:1748/128`
- 5.6** Θα χρησιμοποιήσω την εντολή: `tcpdump -n`
- 5.7** Απο την καταγραφή παρατηρώ ότι η διεύθυνση IPv4 του εξυπηρετητή `teredo` είναι `217.17.192.217`
- 5.8** Χρησιμοποιείται το πρωτόκολλο μεταφοράς UDP και στον εξυπηρετητή `teredo` αντιστοιχεί η θύρα `3544`
- 5.9** Παρατηρώ μηνύματα ICMPv6
- 5.10** Διαφημίζει το πρόθεμα Prefix: `2001:0:d911:c0d9::`
Το πρόθεμα αυτό ουσιαστικά είναι το Teredo Prefix (`2001:0::/32`) και η IPv4 διεύθυνση του εξυπηρετητή Teredo σε δεκαεξαδικό $(d9.11.c0.d9)16 = (217.17.192.217)10$
- 5.11** Τα πρώτα 2 `ping6` πετυχαίνουν. Το `ping6` με την `amazon` δεν επιστρέφει απάντηση
- 5.12** Θα χρησιμοποιήσω την εντολή: `ping6 www.ntua.gr`
- 5.13** Παρατηρώ επιπλέον μηνύματα Direct IPv6 Connectivity Test
- 5.14** Οχι, δεν παρατηρώ μηνύματα ICMP Echo Request/Reply
- 5.15** Το πρωτόκολλο ανωτέρου στρώματος είναι το UDP. Σχετικά με τον αναμεταδότη Teredo η διεύθυνση IPv4 είναι `216.66.86.178` και η θύρα είναι `3545`
- 5.16** Παρατηρώ πακέτα ICMP6 και συγκεκριμένα πακέτα ICMP6 echo Request και ICMP6 echo Reply
- 5.17** Ναι, το `ping6 fe80::ffff:ffff:ffff%teredo` λειτουργεί.
Ωστόσο το `ping6 2001:0:d911:c0d9:c3c:e3b6:920d:1748` δεν εμφανίζει απάντηση
- 5.18** Ναι, παρατηρώ μηνύματα ICMP6 echo request
- 5.19** Οχι, δεν παράγονται δεδομενογράμματα UDP
- 5.20** Οχι, δεν επιλέγεται ο ίδιος `teredo relay`