

## Εργαστηριακή Άσκηση 4 Πρωτόκολλο Ipv4 και Θρυμματισμός

Θοδωρής Φρίξος Παπαρρηγόπουλος

el18040

Ομάδα 4η.

Ipv6: 2a02:587:4f0e:2b65:34b0:9e4a:d912:2f8a

MAC: 40-1C-83-38-F7-20

Λειτουργικό: Windows

DESKTOP-1403ER3

### Άσκηση 1

```
PS C:\Users\papar> ping www.mit.edu -n 3 -4

Pinging e9566.dscb.akamaiedge.net [23.201.181.139] with 32 bytes of data:
Reply from 23.201.181.139: bytes=32 time=97ms TTL=57
Reply from 23.201.181.139: bytes=32 time=44ms TTL=57
Reply from 23.201.181.139: bytes=32 time=44ms TTL=57

Ping statistics for 23.201.181.139:
    Packets: Sent = 3, Received = 3, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 44ms, Maximum = 97ms, Average = 61ms
```

1.1) ping [www.mit.edu](http://www.mit.edu) -n 3 -4

1.2) Δεν εμφανίζονται τα πλαίσια που απευθύνονται σε πολλούς ή όλους τους υπολογιστές του δικτύου (λογο πχ ARP).

1.3) 0% loss, Average = 61ms

1.4) Minimum = 44ms, Maximum = 97ms, Average = 61ms

1.5)

1o: 97ms

2o: 44ms

3o: 44ms

1.6) ip φίλτρο απεικόνισης

1.7) icmp

1.8) Στάλθηκαν request μηνυματα

1.9) source 192.168.1.7, 23.201.181.139

1.10) Στάλθηκαν reply μηνυματα

1.11) 23.201.181.139, 192.168.1.7

1.12) Έχουν αλλάξει οι IP διευθύνσεις, τα πακέτα και ο χρόνος αποστολής των πακέτων

## Άσκηση 2

2.1)

```
PS C:\Users\papar> ping 192.168.1.7 -n 5 -4

Pinging 192.168.1.7 with 32 bytes of data:
Reply from 192.168.1.7: bytes=32 time<1ms TTL=128
Reply from 192.168.1.7: bytes=32 time<1ms TTL=128
Reply from 192.168.1.7: bytes=32 time<1ms TTL=128
Reply from 192.168.1.7: bytes=32 time<1ms TTL=128
Reply from 192.168.1.7: bytes=32 time<1ms TTL=128

Ping statistics for 192.168.1.7:
    Packets: Sent = 5, Received = 5, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 0ms, Average = 0ms
```

```
PS C:\Users\papar> ping 192.168.1.1 -n 5 -4

Pinging 192.168.1.1 with 32 bytes of data:
Reply from 192.168.1.1: bytes=32 time=1ms TTL=64
Reply from 192.168.1.1: bytes=32 time=1ms TTL=64
Reply from 192.168.1.1: bytes=32 time=3ms TTL=64
Reply from 192.168.1.1: bytes=32 time=2ms TTL=64
Reply from 192.168.1.1: bytes=32 time=2ms TTL=64

Ping statistics for 192.168.1.1:
    Packets: Sent = 5, Received = 5, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 1ms, Maximum = 3ms, Average = 1ms
```

```
PS C:\Users\papar> ping 127.0.0.1 -n 5 -4

Pinging 127.0.0.1 with 32 bytes of data:
Reply from 127.0.0.1: bytes=32 time<1ms TTL=128
Reply from 127.0.0.1: bytes=32 time<1ms TTL=128
Reply from 127.0.0.1: bytes=32 time<1ms TTL=128
Reply from 127.0.0.1: bytes=32 time<1ms TTL=128
Reply from 127.0.0.1: bytes=32 time<1ms TTL=128

Ping statistics for 127.0.0.1:
    Packets: Sent = 5, Received = 5, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 0ms, Average = 0ms
```

## 2.2)

ip and icmp						
No.	Time	Source	Destination	Protocol	Length	Info
95	9.535188	192.168.1.7	192.168.1.1	ICMP	74	Echo (ping) request id=0x0001, seq=55/14080, ttl=128 (reply in 96)
96	9.536686	192.168.1.1	192.168.1.7	ICMP	74	Echo (ping) reply id=0x0001, seq=55/14080, ttl=64 (request in 95)
109	10.552902	192.168.1.7	192.168.1.1	ICMP	74	Echo (ping) request id=0x0001, seq=56/14336, ttl=128 (reply in 110)
110	10.557047	192.168.1.1	192.168.1.7	ICMP	74	Echo (ping) reply id=0x0001, seq=56/14336, ttl=64 (request in 109)
115	11.562943	192.168.1.7	192.168.1.1	ICMP	74	Echo (ping) request id=0x0001, seq=57/14592, ttl=128 (reply in 116)
116	11.565750	192.168.1.1	192.168.1.7	ICMP	74	Echo (ping) reply id=0x0001, seq=57/14592, ttl=64 (request in 115)
133	12.569512	192.168.1.7	192.168.1.1	ICMP	74	Echo (ping) request id=0x0001, seq=58/14848, ttl=128 (reply in 134)
134	12.571223	192.168.1.1	192.168.1.7	ICMP	74	Echo (ping) reply id=0x0001, seq=58/14848, ttl=64 (request in 133)
142	13.581417	192.168.1.7	192.168.1.1	ICMP	74	Echo (ping) request id=0x0001, seq=59/15104, ttl=128 (reply in 143)
143	13.584742	192.168.1.1	192.168.1.7	ICMP	74	Echo (ping) reply id=0x0001, seq=59/15104, ttl=64 (request in 142)

2.3) Προορισμός τους ήταν το default gateway

2.4) Όχι δεν καταγράφηκε αποστολή μηνυμάτων ICMP Echo request στο δίκτυο με πηγή και προορισμό τη διεύθυνση IPv4 του υπολογιστή μου. Αυτό συμβαίνει καθώς τα πακέτα που επιχείρησα να κάνω ping προς τον υπολογιστή μου δεν βγήκαν ποτέ στο τοπικό δίκτυο. Ο οδηγός Ethernet αναγνώρισε ότι αυτή η διεύθυνση είναι του υπολογιστή μου (τοπική διεύθυνση IPv4) με αποτέλεσμα να την στείλει στον οδηγό loopback.

2.5) Όπως και πριν, δεν παρατηρείται τέτοια κίνηση πακέτων στο δίκτυο, δεν καταγράφεται δηλαδή από το Wireshark. Αυτό συμβαίνει καθώς αναγνωρίστηκε αμέσως ότι αυτή η διεύθυνση είναι η διεύθυνση του βρόχου επιστροφής με αποτέλεσμα να σταλούν τα μηνύματα στον οδηγό loopback.

2.6) Δεν εμφανίζεται κάποια άμεση διαφορά στους χρόνους που χρειάστηκαν για να ληφθούν απαντήσεις (replies) στα πακέτα Echo request. Μάλιστα, σύμφωνα με το command prompt, τόσο η απάντηση από την IPv4 διεύθυνση του υπολογιστή μου όσο και από την 127.0.0.1 στάλθηκε ακαριαία, σε λιγότερο από 1ms. Η μόνη διαφορά που μπορεί να παρατηρηθεί είναι ότι ο οδηγός Ethernet πρώτα ελέγχει αν η ping address είναι η 127.0.0.1 και στην συνέχεια αν δεν είναι, κοιτάει αν είναι η IP address του υπολογιστή.

2.7) Το περιεργό είναι πως στο νετφλιξ εμφανίζεται η σελίδα αλλά το ping δεν καταφέρνει να λάβει απάντηση απο τα ping requests ενώ για το άμαζον δεν υπάρχει τέτοιο πρόβλημα. Μερικά πακέτα μπλοκάρονται για λόγους ασφάλειας απο το firewall όταν για να φτάσεις στο αντίστοιχο υποδίκτυο πρέπει να περάσεις σε πολλούς σερβερς.

### Άσκηση 3

3.1) host 192.168.1.7

3.2) ip.addr==192.168.1.7

3.3)

Version 4 bit, header length 4 bit, DS Field 1 byte, total length 2 byte,

Identification 2 byte, flags 3 bit, Fragment Offset 13 bit,

Time to Live 1 byte, Protocol 1 byte, Header Checksum 2 byte

Source Address 4 byte,

Destination Address 4 byte.

3.4) DS Field, total length, Identification, Time to Live, Header Checksum,

Source Address, Destination Address.

3.5) ναι, αφού το Header Length είναι σταθερά ίσο με 20 bytes

3.6) Μικρότερο 54, Μεγαλύτερο 251 Bytes

3.7) παίρνει τις τιμές 0x10 (για High-throughput data), 0x00 (Standard) , 0x08(Low-priority data)

3.8) Διαφέρουν σε κάθε πλαίσιο

3.9) Είναι πάντα Set, δηλαδή έχει τιμή 1.

3.10) Είναι σταθερά ίσο με 0 το fragment offset

3.11 Έχει τιμή 6 που αντιστοιχεί στο πρωτόκολλο TCP

3.12) Επειδή πολλά πεδία της επικεφαλίδας IPv4 έχουν διαφορετικές τιμές, το άθροισμα όλων των byte αυτών είναι διαφορετικό. Το Checksum, λοιπόν, (ως έλεγχος εγκυρότητας της πληροφορίας) βασίζεται στο άθροισμα των πεδίων της επικεφαλίδας IPv4, αλλάζει σε κάθε πακέτο.

## Άσκηση 4

4.1) `ping -f -l 1000 -n 1 -4 192.168.1.1`, όπου φυσικά το 192.168.1.1 είναι η διεύθυνση που γίνεται το ping (του δρομολογητή μου συγκεκριμένα) και το «1000» είναι το πλήθος σε byte που επέλεξα τυχαία να στείλω.

4.2) Η μέγιστη τιμή είναι 1472 bytes

4.3) Από τα 1473 bytes απαιτείται θρυμματισμός

4.4) not broadcast and not multicast

4.5) `ip.addr == 192.168.1.1 and ip.addr == 192.168.1.7`

4.6) Όχι, παράγεται μόνο το πακέτο που είχε μέγεθος 1472 bytes. Το άλλο πακέτο, επειδή είναι μεγαλύτερο από το όριο, δεν στέλνεται καν.

4.7)  $MTU = 1514\text{Bytes} - 14\text{Bytes (Ethernet Header)} = 1500\text{Bytes}$

4.8) το μέγιστο μέγεθος πακέτου IPv4 είναι 65.535 byte. Αν αφαιρέσουμε 20 bytes του IPv4 Header και τα 8 byte των πεδίων Type, Code, Checksum,... του ICMP μένουν 65507 bytes.

4.9) Όχι, στέλνονται μέχρι 65500 Bytes

4.10) 65.535 byte

4.11) Όχι δεν στάλθηκε σε ένα πακέτο

4.12) Στάλθηκαν σε 5 πακέτα. Δεδομένου ότι κάθε πακέτο μπορεί να είναι μέχρι 1472 bytes, έχουμε  $\text{ceil}(6000/1472) = 5$  πακέτα

4.13)

Πρώτο Πακέτο:

Identification: 0x5b01, Don't fragment = 0, More Fragments = 1, Fragment Offset = 0

Δεύτερο Πακέτο:

Identification: 0x5b01, Don't fragment = 0, More Fragments = 1, Fragment Offset = 1480

Τρίτο Πακέτο:

Identification: 0x5b01, Don't fragment = 0, More Fragments = 1, Fragment Offset = 2960

Τέταρτο Πακέτο:

Identification: 0x5b01, Don't fragment = 0, More Fragments = 1, Fragment Offset = 4440

Πέμπτο Πακέτο:

Identification: 0x5b01, Don't fragment = 0, More Fragments = 0, Fragment Offset = 5920

4.14) Το βλέπουμε από το More Fragments = 1, που δείχνει ότι ακολουθούν και άλλα fragments, άρα έχει θρυμματιστεί το πακέτο.

4.15) Το βλέπουμε από το Fragment Offset = 0

4.16) Είναι 1480 bytes

4.17) Το Fragment Offset, επειδή δεν είναι ίσο με 0 (Fragment Offset = 1480)

4.18) Ναι

4.19) More Fragments = 1

4.20) Μόνο το πεδίο Fragment Offset, που μας δείχνει από ποιο byte από τα 6000 ξεκινάει η πληροφορία του εκάστοτε πακέτου. Στο πρώτο πακέτο ξεκινάει το 1ο byte από τα 6000 (στην θέση 0), ενώ το δεύτερο πακέτο ξεκινάει από το 1481ο byte (το 1480ο δηλαδή αν μετρήσουμε από το 0).

4.21) Με παρόμοια λογική με το προηγούμενο ερώτημα, κάθε fragment offset ξεκινάει +1480 bytes μετά το προηγούμενο. Έτσι ξεκινώντας από το 0, το τέταρτο fragment έχει offset  $3 \times 1480 = 4440$ . Ομοίως το 5ο έχει  $4 \times 1480 = 5920$ .

4.22) Εκτός λοιπόν από το fragment offset, που όπως είδαμε παίρνει τιμές πολλαπλάσιες του 1480 σε κάθε πακέτο, αλλάζει και το , More Fragments, όπου στο τελευταίο πακέτο παίρνει τιμή 0, για να μας δείξει ότι δεν ακολουθούν άλλα πακέτα.