# $4^o$ Εργαστήριο στα Δίκτυα Υπολογιστών
# Πρωτόκολλο IPv4 και θρυμματισμός

Όνοματεπώνυμο: **Αλέξανδρος Κυριακάκης (03112163)**
Ομάδα: **2**
Όνομα PC/ΛΣ: **MacBook Pro - Alexandros, macOS Catalina**
Ημερομηνία: **26/10/2020**
Διεύθυνση IP: **192.168.1.5**
Διεύθυνση MAC: **a4:83:e7:97:af:31**

## 1 Μετρήστε την καθυστέρηση

**1.1** Είναι:

```
ping -c 3 www.mit.edu
```

**1.2** Με αυτό το φίλτρο καταγράφουμε μόνο τα unicast πακέτα από και προς την συσκευή μας για μια πιο "καθαρή" καταγραφή.

**1.3**

- Packet Loss: 0.0%
- Average Delay: $86.548ms$

**1.4**

```
64 bytes from 23.12.110.203: icmp_seq=0 ttl=53 time=85.655 ms
64 bytes from 23.12.110.203: icmp_seq=1 ttl=53 time=86.235 ms
64 bytes from 23.12.110.203: icmp_seq=2 ttl=53 time=87.753 ms
```

**1.5** Οι αντίστοιχες τιμές απο το Wireshark είναι:

1. $time = 0.085535s$
2. $time = 0.086131s$
3. $time = 0.087683s$

Οι τιμές που διαβάζουμε στο Wireshark είναι ελάχιστα μικρότερες, αυτό οφείλεται στο γεγονός ότι η καταγραφή γίνεται κατά τη διάρκεια της μεταφοράς και όχι τη στιγμή της λήψης.

**1.6** Το φίλτρο "ip".

**1.7** Το φίλτρο είναι "**(ip.src == 23.12.110.203 or ip.dst == 23.12.110.203) and icmp**", όπου 23.12.110.203 είναι η διεύθυνση IPv4 του www.mit.edu. (Στην πράξη είναι αρκετό τουλάχιστον ένα απο τα σκέλη του and αλλά για λόγους πληρότητας βάλαμε και τα δύο.)

**1.8** Στάλθηκαν Echo (ping) requests.

**1.9**

- Source: 192.168.1.5
- Destination: 23.12.110.203

**1.10** Ελήφθησαν Echo (ping) replies.

**1.11**

- Source: 23.12.110.203
- Destination: 192.168.1.5

**1.12** Έχουν αλλάξει τα εξής:

1. Η διεύθυνση IPv4 του ονόματος www.mit.edu — $18.7.22.83 \neq 23.12.110.203$.
2. Το μέγεθος των πακέτων — $32bytes \neq 56bytes$
3. Ο χρόνος ζωής του πακέτου — $242 \neq 53$
4. Το γεγονός ότι οι χρόνοι απόκρισης στο παρελθόν είναι σταθεροί, παραπέμπει σε ντετερμινιστική διαδικασία ενώ δεδομένου ότι οι σύγχρονοι χρόνοι ενέχουν κάποια τυχαιότητα, παραπέμπει σε στοχαστική διαδικασία.

# 2 Περισσότερα για το Ping

**2.1** Η σύνταξη είναι:

```
ping -c 5 192.168.1.1 && ping -c 5 192.168.1.5 && ping -c 5 127.0.0.1
```

**2.2** Έχει καταγράψει μόνο 5 ICMP Echo requests.

**2.3** Η διεύθυνση της προκαθορισμένης πύλης 192.168.1.1

**2.4** Όχι δεν παρατηρώ. Αυτό συμβαίνει επειδή όπως βλέπουμε και στο σχήμα, στο στάδιο "Προορισμός IPv4 = τοπική διευθυνση IPv4?" η απάντηση είναι ΝΑΙ οπότε προωθέιται στον "Οδηγό loopback" και απο εκεί επιστρέφει στην είσοδο του υπολογιστή μας. Άρα δεν μπήκε ποτέ στο Τοπικό δίκτυο και γι αυτό δεν έγινε capture από το Wireshark.

**2.5** Ομοίως, με το **2.4**, δεν παρατηρώ διότι από την έξοδο του υπολογιστή μας θα πάει στον "Οδηγό loopback" και πίσω στην είσοδο άρα δεν θα καταγραφεί από το Wireshark.

**2.6** Η βασική διαφορά είναι ότι στην πράξη όταν κάνουμε ping στη διευθυνση loopback το πακέτο δε θα φύγει ποτέ από τον υπολογιστή μας άρα ακόμα και αν δε βρισκόμασταν σε τοπικό δίκτυο θα λειτουργούσε. Ενώ όταν κάνουμε ping στην ip μας αυτό θα δρομολογηθεί μέσω του "Οδηγού Ethernet" αρα θα μπεί στο τοπικό δίκτυο και απο κει θα προωθηθεί πίσω στον βρόγχο loopback.

**2.7** Το παράδοξο είναι ότι το www.netflix.com ενώ ανοίγει κανονικά στο browser, στα ping requests δεν απαντάει, ενώ με την ιδια διαδικασία το www.amazon.com απαντάει. Υποθέτω τις εξής εκδοχές:

- Επειδή από router μου μέχρι ζητούμενο υποδίκτυο μπορεί να μεσολάβησαν πολλοί server με τα δικά του firewall, μπορεί κάποιος από αυτούς να μπλοκάρει τα πακέτα ICMP. Οπότε να μην εφτασαν ποτέ στη δοσμένη διεύθυνση.
- Για λόγους ασφαλείας απο DoS Attacks μπορεί το ίδιο το υποδίκτυο του προορισμού να μπλοκάρει τα πακέτα ICMP.

# 3 Επικεφαλίδες IPv4

**3.1** Είναι:

```
host 192.168.1.5
```

**3.2** Είναι:

```
ip.src == 192.168.1.5
```

**3.3**

- Version: 4 bit
- Header Length: 4 bit
- Differentiated Services Codepoint: 6 bit
- Explicit Congestion Notification: 2 bit
- Total Length: 16 bit
- Identification: 16 bit
- Flags: 16 bit
- Time to live: 8 bit
- Protocol: 8 bit
- Header checksum: 16 bit
- Source: 32 bit
- Destination: 32 bit

**3.4** Βρήκα διαφορετικές τιμές στα εξής πεδία:

- Total Length: $64 \neq 52$
- Header checksum: $0xbd95 \neq 0xbda1$

**3.5** Ναι παραμένει το ίδιο.

**3.6** Το μικρότερο που παρατηρώ είναι 52 bytes ενώ το μεγαλύτερο 120 bytes.

**3.7** Έχει τιμή: Differentiated Services Field: 0x00 (Default) και χρησιμοποιείται για να κατηγοριοποιήσει το είδος της "κίνησης" του δικτύου έτσι ώστε να εξασφαλιζεται η σωστή λειτουργία του που στην προκειμένη άσκηση είναι Default συμπεριφορά χωρίς ιδιαίτερες απαιτήσεις.

**3.8** Ότι είναι σε όλα τα πακέτα ίδιες 0x0000.

**3.9** Έχει τιμή 0x4000.

**3.10** Έχει τιμή 0.

**3.11** Έχει τιμή 0x06 και αντιστοιχεί στο TCP.

**3.12** Το Checksum είναι ένας τρόπος ελέγχου για λάθη στα πακέτα που στέλνονται. Πρακτικά το Checksum είναι το συμπλήρωμα ως προς ένα του IPv4 πακέτου. Έτσι ώστε όταν φτάσει, το άθροισμα της επικεφαλίδας του πακέτου με το Checksum να κάνει 0. Άρα εξαρτάται άμεσα από τα περιεχόμενα του πακέτου, μέσω πχ του μεγέθους τους και άλλα, τα οποία προφανώς διαφέρουν άρα λογικό είναι να διαφέρουν και οι τιμές των Checksum.

# 4 Θρυμματισμός (Fragmentation) στο IPv4

**4.1** Είναι,

```
ping -D -s <size> -c 1 <dst IPv4>
```

**4.2** Είναι 1472 Bytes

**4.3** Είναι 1473 Bytes

**4.4** Χρησιμοποίησα,

```
not broadcast and not multicast
```

**4.5** Χρησιμοποίησα,

```
ip.src == 192.168.1.1 or ip.dst == 192.168.1.1
```

**4.6** Όχι δεν παράγονται, διότι αφού το πακέτο είναι μεγαλύτερο του MTU δεν μπορέι να ταξιδέψει άρα δεν θα μπει στον οδηγό Ethernet και αρα στο τοπικό δίκτυο και έτσι δεν θα το πιάσει το WireShark.

**4.7** Από το WireShark βλέπουμε ότι το μέγεθος του πακέτου συνολικά είναι 1514 Bytes ενώ το Ethernet Header είναι 14 Bytes άρα το μέγιστο IPv4 που μπορούμε να στείλουμε είναι $MTU = 1500\ Bytes$.

**4.8** Γνωρίζουμε ότι το μέγιστο μέγεθος πακέτου IPv4 είναι 65.535 Bytes εκ των οποίων τα 20 bytes αποτελούν το IPv4 Header και τα 8 bytes το ICMP Header άρα το μέγιστο μέγεθος δεδομένων είναι $65.535 - 20 - 8 = 65.507\ Bytes$. Στο δικό μας τοπικό δίκτυο είναι 1472 Bytes.

**4.9** Όχι, δεν επιτυγχάνει. Η μέγιστη τιμή που επιτυγχάνει είναι 8164 Bytes.

**4.10** Έχει 65.535 Bytes.

```
$ ping -D -s 100000 -c 1 192.168.1.5
ping: packet size too large: 100000 > 65507
```

**4.11** Όχι.

**4.12** Έχει θρυματιστεί σε 5 πακέτα IPv4. Γιατί το μέγιστο περιεχόμενο δεδομένων ενός πακέτου IPv4 στο δικτυό μας είναι 1472 Bytes και το συνολικό μέγεθος δεδομένων προς μετάφορά είναι 6.000 bytes, άρα θα χρειαστούν $\lceil \frac{6.000}{1472} \rceil = 5$ πακέτα.

**4.13**

| Identification | Don't Fragment Bit | More Fragments Bit | Fragment Offset |
|---|---|---|---|
| 0x00004df5 | 0 | 1 | 0 |
| 0x00004df5 | 0 | 1 | 1480 |
| 0x00004df5 | 0 | 1 | 2960 |
| 0x00004df5 | 0 | 1 | 4440 |
| 0x00004df5 | 0 | 0 | 5920 |

**4.14** Το δηλώνει το "Don't fragment: 0".

**4.15** Το δηλώνει το "Fragment Offset: 0".

**4.16** Είναι 13 bit.

**4.17** Το δηλώνει το "Fragment Offset: 1480"

**4.18** Ναι.

**4.19** Φαίνεται από το "More fragments: 1"

**4.20** Το "Fragment Offset" και το "Header Checksum".

**4.21** Στο προτελευταίο η τιμή του Fragment Offset είναι 4440 που σημαίνει ότι έχουν έχουν ληφθεί 4440 byte ως τώρα. Σε αυτά τα byte συμπεριλαμβάνονται και 8 για κάθε μια απο τις επικεφαλίδες των ICMP fragment που προηγήθηκαν. Δηλαδή περιέχει $(1472(data)+8(ICMP\ Header))\cdot 3 = 4440\ Bytes$. Ομοίος στο τελευταίο fragment που στάλθηκε, το Fragment Offset είναι $(1472(data) + 8(ICMP\ Header)) \cdot 4 = 5920\ Bytes$

**4.22** Όλα τα θραύσματα έχουν διαφορέτικά μεταξύ τους τα πεδία: "Fragment Offset" και "Header Checksum". Επίσης το τελευταίο θραύσμα έχει διαφορετικό το πεδίο "More Fragments: 0", σε σχέση με τα τέσσερα προηγούμενα που έχουν τιμή 1.