Δίκτυα Υπολογιστών
Εργαστηριακή Άσκηση 4

Ονοματεπώνυμο: Κυριακόπουλος Γεώργιος – el18153
Ομάδα: 4
Όνομα PC/ΛΣ: George – Windows 10 Pro, 21H1
Ημερομηνία: 8/11/2021
Διεύθυνση IP: 192.168.1.8/192.168.1.7
Διεύθυνση MAC: 60-A4-B7-75-72-0F

Άσκηση 1:

1.1.
Χρησιμοποίησα την εντολή: *$ ping www.mit.edu -4 -n 3*, όπου με την παράμετρο -4 εξανάγκασα τη χρήση IPv4 και με την παράμετρο -n 3 όρισα τον αριθμό των πακέτων ίσο με 3.

1.2.
Λόγω του φίλτρου not multicast and not broadcast, εμφανίζονται στη λίστα μας πακέτα που δεν είναι ούτε multicast ούτε broadcast, δηλαδή που είναι unicast.

1.3.
Μέσω του παραθύρου εντολών λαμβάνω ότι: Packets lost = 0, άρα Packet Loss = 0%. Επίσης, λαμβάνω ότι: Average RTT = 46 ms.

1.4.
Μέσω του παραθύρου εντολών λαμβάνω ότι: RTT_1 = 47 ms, RTT_2 = 46 ms, RTT_3 = 43 ms.

1.5.
Μέσω του Wireshark λαμβάνω ότι: RTT_1 = 47.606 ms, RTT_2 = 46.278 ms, RTT_3 = 43.831 ms. Οι τιμές ουσιαστικά είναι ίσες με αυτές του παραθύρου εντολών.

1.6.
Για να παρατηρώ μόνο IPv4 πακέτα, αρκεί η εφαρμογή του φίλτρου απεικόνισης *ip*.

1.7.
Για να παρατηρώ μόνο την κίνηση ICMP που προκάλεσε η εντολή ping, αρκεί η εφαρμογή του φίλτρου απεικόνισης icmp σε συνδυασμό με ένα φίλτρο για τις ip.src ή ip.dst που θα περιλαμβάνει την IP στην οποία στείλαμε ping. Επομένως ένα τέτοιο φίλτρο είναι το εξής: *icmp and (ip.src == 184.30.212.47 or ip.dst == 184.30.212.47)*.

1.8.
Στάλθηκαν μηνύματα ICMP με Type 8, δηλαδή Echo (ping) requests από τον υπολογιστή μου.

1.9.
Η διεύθυνση IPv4 πηγής των παραπάνω μηνυμάτων είναι η 192.168.1.8 και η διεύθυνση προορισμού είναι η 184.30.212.47.

1.10.
Ελήφθησαν μηνύματα ICMP με Type 0, δηλαδή Echo (ping) replies από τον υπολογιστή μου.

1.11.
Η διεύθυνση IPv4 πηγής των παραπάνω μηνυμάτων είναι η 184.30.212.47 και η διεύθυνση προορισμού είναι η 192.168.1.8.

1.12.
Αυτό που άλλαξε σε σχέση με την καταγραφή του παρελθόντος είναι ότι ο DNS που χρησιμοποιεί ο υπολογιστής μου κάνει resolve το όνομα www.mit.edu σε άλλη IPv4, συγκεκριμένα στην 184.30.212.47 και όχι στην 18.7.22.83 που κάνει resolve ο DNS που χρησιμοποιήθηκε στο παλιό ping της εκφώνησης. Εκτός αυτού, έχει αλλάξει ο αριθμός των πακέτων (3 έναντι 4 στο παλιό), η ελάχιστη, μέγιστη και μέση καθυστέρηση σε millisecond, κάτι που είναι, φυσικά, εύλογο και το TTL από 242 σε 56.

Άσκηση 2:

2.1.
Χρησιμοποίησα την εντολή: *$ ping 192.168.1.1 -n 5 && ping 192.168.1.7 -n 5 && ping 127.0.0.1 -n 5*, όπου με την παράμετρο -n 5 όρισα τον αριθμό των πακέτων ίσο με 5.

2.2.
Έχει καταγράψει 5 ICMP Echo request μηνύματα.

2.3.
Ο προορισμός τους ήταν η προκαθορισμένη πύλη 192.168.1.1.

2.4.
Όχι, δεν υπήρχαν τέτοια μηνύματα. Ο λόγος είναι ότι όταν φτάνει το request αυτό στον οδηγό Ethernet, εξυπηρετείται από το «Προορισμός IPv4 = τοπική διεύθυνση IPv4» και επιστρέφει μέσω του οδηγού loopback στην είσοδο πακέτων IPv4 και πίσω στον υπολογιστή. Δεν περνάει, επομένως, ποτέ τον οδηγό Ethernet, άρα δεν πηγαίνει και στα επόμενα επίπεδα ώστε να το καταγράψει το Wireshark.

2.5.
Όχι, δεν υπήρχαν τέτοια μηνύματα. Ο λόγος είναι ότι το request αυτό πηγαίνει στον οδηγό loopback και επιστρέφει απευθείας μέσω της εισόδου πακέτων IPv4 πίσω στον υπολογιστή μας. Άρα, ομοίως με το παραπάνω ερώτημα, δεν το καταγράφει το Wireshark.

2.6.
Ουσιαστικά, όταν κάνουμε ping τη διεύθυνση loopback 127.0.0.1, το πακέτο αυτό δεν φτάνει ποτέ σε κάποιο τοπικό δίκτυο, καθώς εξυπηρετείται από την κάρτα δικτύου και μόνο. Σε αντίθετη περίπτωση, όταν κάνουμε ping τη διεπαφή του υπολογιστή μας, το πακέτο αυτό περνάει κανονικά στον οδηγό Ethernet και εξυπηρετείται μέσω αυτού. Επομένως, με ένα ping στη loopback διεύθυνση μπορούμε εύκολα να ελέγξουμε την κάρτα δικτύου μας και τους οδηγούς/λογισμικό της, χωρίς την ανάγκη ύπαρξης κάποιας σύνδεσης σε τοπικό δίκτυο.

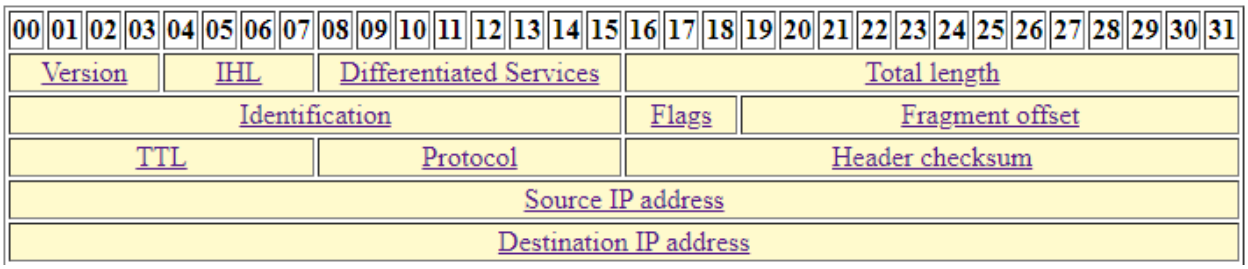
2.7.
Οι δύο σελίδες φορτώνουν κανονικά, όμως μόνο στα ping προς www.amazon.com δέχομαι απαντήσεις, ενώ σε αυτά προς www.netflix.com δεν υπάρχει απάντηση και τα request κάνουν time out. Αυτό συμβαίνει είτε λόγω κάποιου firewall που βρίσκεται κάπου στη διαδρομή προς το υποδίκτυο του www.netflix.com που μπλοκάρει τα ICMP πακέτα μου, είτε λόγω του ίδιου του υποδικτύου του www.netflix.com που μπλοκάρει τα ICMP πακέτα μου για λόγους ασφαλείας, πχ προστασία από ping floods ή άλλες μορφές DoS attacks.

Άσκηση 3:

3.1.
Χρησιμοποίησα το φίλτρο σύλληψης *host 147.102.40.15*, ώστε να καταγραφεί μόνο η κίνηση από ή προς τη διεύθυνση IPv4 του edu-dy.cn.ntua.gr.

3.2.
Χρησιμοποίησα το φίλτρο σύλληψης *ip.src == 192.168.1.7 and ip*, ώστε να εμφανίζονται στη λίστα μόνο τα πακέτα IPv4 που έστειλε ο υπολογιστής μου.

3.3.
Η επικεφαλίδα IPv4 έχει τα εξής πεδία (σε παρένθεση το μήκος τους): Version (4 bits), Header Length (4 bits), Differentiated Services Field (1 byte), Total Length (2 bytes), Identification (2 bytes), Flags (3 bits), Fragment Offset (13 bytes), Time to Live (1 byte), Protocol (1 byte), Header Checksum (2 bytes), Source Address (4 bytes), Destination Address (4 bytes). Η θέση τους στην επικεφαλίδα φαίνεται στο παρακάτω σχήμα:



3.4.
Τα μόνα πεδία που αλλάζουν τιμή είναι τα Total Length και Identification.

3.5.
Ναι είναι σταθερό και ίσο με 20 bytes σε όλα τα πακέτα.

3.6.
Το μικρότερο μήκος πακέτου IPv4 είναι 54 bytes και το μεγαλύτερο 80 bytes.

3.7.
Το πεδίο Differentiated Services Field έχει τιμή 0x00 και αντιστοιχεί στην Default τιμή, καθώς δεν έχουμε κάποια υπηρεσία που χρειάζεται ειδική αντιμετώπιση.

3.8.
Για τις τιμές του πεδίου Identification παρατηρώ ότι αλλάζουν και είναι μάλιστα ένας αύξων αριθμός όπως προχωράνε τα πακέτα.

3.9.
Η σημαία Don’t Fragment έχει τιμή 0x40, δηλαδή 01000000b, με τα 3 πρώτα bit 010 να είναι τα R, DF, MF του πεδίου Flag και το DF (Don’t Fragment) να είναι ίσο με 1.

3.10.
Το πεδίο Fragment Offset έχει παντού τιμή 0 (0000000000000b).

3.11.
Το πεδίο Protocol έχει τιμή 6 (0x06) και αντιστοιχεί στο πρωτόκολλο TCP.

3.12.
Η τιμή του Header Checksum θεωρητικά μπορεί να αλλάζει μεταξύ πακέτων, αφού αποτελεί το συμπλήρωμα ως προς ένα της επικεφαλίδας του πακέτου. Επομένως, είναι λογικό να διαφέρει από πακέτο σε πακέτο, εάν έχουν διαφορές, πχ στο μήκος. Στη καταγραφή μας, όμως είναι όλα ίσα με 0 (0x0000). Για αυτό ίσως φταίει η χρήση ενσύρματης σύνδεσης και ο τρόπος που χειρίζεται το checksum στο IP layer.

Άσκηση 4:

4.1.
Η ακριβής σύνταξη της εντολής είναι η εξής: *ping <IPv4> -n 1 -4 -f -l <length>*, όπου <IPv4> η διεύθυνση την οποία θέλουμε να κάνουμε ping και <length> το μήκος που θέλουμε να έχει το πακέτο.

4.2.
Η μέγιστη τιμή για πετυχημένη αποστολή είναι 1472 bytes.

4.3.
Η μικρότερη τιμή για την οποία χρειαζόμαστε θρυμματισμό είναι τα 1473 bytes.

4.4.
Χρησιμοποίησα το φίλτρο καταγραφής *not broadcast and not multicast,* ώστε να καταγραφεί μόνο η unicast κίνηση.

4.5.
Χρησιμοποίησα το φίλτρο απεικόνισης *(ip.src == 192.168.1.1 or ip.dst == 192.168.1.1) and ip,* ώστε να εμφανίζονται μόνο τα πακέτα από και προς τη διεύθυνση 192.168.1.1 που έκανα ping.

4.6.
Όχι, δεν παράγονται, αφού το πακέτο που προσπαθούμε να στείλουμε έχει μέγεθος μεγαλύτερο από το επιτρεπτό (MTU) και επομένως αποτυγχάνει η αποστολή του, άρα και η καταγραφή του από το Wireshark.

4.7.
Το πακέτο που στάλθηκε στην προκαθορισμένη πύλη 192.168.1.1 έχει μέγεθος 1514 bytes, με το Ethernet Header να αποτελεί τα 14 από αυτά. Επομένως, το μέγιστο μέγεθος IPv4 πακέτου που μπορούμε να στείλουμε είναι 1500 bytes. Αυτό μπορούμε να το επιβεβαιώσουμε και με την εντολή *netsh interface ipv4 show subinterfaces,* όπου θα δούμε την default τιμή 1500 στην στήλη MTU για την Ethernet διεπαφή μας.

4.8.
Γενικά, η τιμή μεγέθους δεδομένων ICMP που οδηγεί σε πακέτο IPv4 μέγιστου μήκους είναι η 65507. Αυτή προκύπτει εάν αφαιρέσουμε από το μέγιστο μήκος IPv4 πακέτου 65535 τα 20 bytes του IPv4 Header και τα 8 bytes του ICMP Header. Στην περίπτωση του τοπικού μας δικτύου, η τιμή αυτή είναι η τιμή του δεύτερου ερωτήματος, δηλαδή 1472 bytes. Αυτό προκύπτει εάν αφαιρέσουμε από τα 1500 bytes μήκος του μέγιστου IPv4 πακέτου (λόγω του MTU της διεπαφής του υπολογιστή μου) τα 20 bytes του IPv4 Header και τα 8 bytes του ICMP Header.

4.9.
Εάν κάνω ping στην IPv4 του υπολογιστή μου με μέγεθος ICMP δεδομένων 65507 αποτυγχάνει, καθώς τα windows θέτουν ως μέγιστο το 65500. Με 65500, ωστόσο πετυχαίνει

το ping, καθώς εξυπηρετείται μέσω του loopback που έχει πολύ μεγαλύτερο MTU (4294967295) σύμφωνα με την εντολή *netsh interface ipv4 show subinterfaces.*

4.10.
Το μεγαλύτερο πακέτο ping θα είναι με 65500 bytes μέγεθος ICMP δεδομένων, επομένως 65528 bytes μέγεθος πακέτου.

4.11.
Όχι, δεν έχει μεταφερθεί ως ένα πακέτο IPv4, όπως φαίνεται άλλωστε και από το μη μηδενικό Fragment Offset και από τα Flags που δεν έχουν την τιμή 0x01.

4.12.
Χρειάστηκαν 5 πακέτα IPv4, γιατί το σύνολο των 6000 bytes δεδομένων πρέπει να σπάσει σε κομμάτια των 1472 bytes που είναι το μέγιστο μέγεθος δεδομένων ενός μέγιστου πακέτου IPv4. Η διαίρεση 6000/1472 κάνει κάτι παραπάνω από 4, άρα θα χρειαστούν 5 πακέτα συνολικά.

4.13.
Το Identification είναι 0xf9db (63963) και το Don’t Fragment Bit είναι 0 για όλα τα πακέτα. Το More Fragments Bit είναι 1 για τα 4 πρώτα πακέτα, ενώ είναι 0 για το τελευταίο. Τέλος, το Fragment Offset είναι με τη σειρά των πακέτων 0, 1480, 2960, 4440, 5920.

4.14.
Ο συνδυασμός των Don’t fragment bit (0) και More fragments bit (1) στην επικεφαλίδα IPv4 μας δηλώνει ότι το πακέτο έχει θρυμματιστεί.

4.15.
Το πεδίο Fragment Offset (0) στην επικεφαλίδα IPv4 δηλώνει ότι αυτό είναι το πρώτο θραύσμα και όχι ένα μεταγενέστερο.

4.16.
Το πρώτο θραύσμα έχει μήκος 1514 bytes με 1472 bytes δεδομένα (εμφανίζονται ως 1480 μαζί με τα 8 του ICMP Header).

4.17.
Το πεδίο Fragment Offset (1480) στην επικεφαλίδα IPv4 δηλώνει ότι αυτό δεν είναι το πρώτο θραύσμα, αλλά κάποιο μεταγενέστερο.

4.18.
Ακολουθούν και άλλα θραύσματα.

4.19.
Αυτό φαίνεται από το More fragments bit (1) στην επικεφαλίδα IPv4.

4.20.
Μεταξύ του πρώτου και του δεύτερου θραύσματος αλλάζει μόνο το πεδίο Fragment Offset.

4.21.
Το offset βλέπουμε ότι μεγαλώνει σε πολλαπλάσια του 1480 που είναι τα 1472 bytes δεδομένων συν τα 8 bytes του ICMP Header. Επομένως, η τιμή 4440 του προτελευταίου (5920 του τελευταίου) σημαίνουν ότι έχουν ληφθεί πριν από αυτό το θραύσμα 4440 bytes (5920 αντίστοιχα), που είναι τα 3 θραύσματα που προηγήθηκαν επί τα 1480 (1472 + 8) bytes που μεταφέρει το καθένα.

4.22.
Μεταξύ των θραυσμάτων αλλάζουν τα πεδία Total Length, Flags και Fragment Offset της επικεφαλίδας IPv4.