ΤΕΙ ΚΡΗΤΗΣ
ΣΧΟΛΗ ΤΕΧΝΟΛΟΓΙΚΩΝ ΕΦΑΡΜΟΓΩΝ
ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΙΑΣ

Πτυχιακή εργασία

ΕΡΓΑΣΤΗΡΙΑΚΕΣ ΑΣΚΗΣΕΙΣ ΕΠΙΔΕΙΞΗΣ ΔΙΚΤΥΩΝ TCP/IP ΜΕ ΧΡΗΣΗ ΤΟΥ WIRESHARK

ΕΛΕΥΘΕΡΙΟΣ ΜΑΣΧΑΛΙΔΗΣ
ΑΜ 2769

Επιβλέπων Καθηγητής
Κώστας Βασιλάκης

Πρόλογος

Μέσα από την παρούσα πτυχιακή εργασία θα παρουσιαστεί μια πλήρης σειρά εργαστηριακών ασκήσεων με σκοπό την επίδειξη της λογικής και λειτουργίας των πρωτοκόλλων ενός δικτύου TCP/IP. Κάθε άσκηση αναφέρεται σε ένα συγκεκριμένο πρωτόκολλο και περιλαμβάνει μια εισαγωγική θεωρητική περιγραφή, τα βήματα της πειραματικής διαδικασίας και ένα σύνολο ερωτήσεων (και των αντίστοιχων απαντήσεων) που θα βοηθήσουν στην παρατήρηση και ερμηνεία συμβάντων δικτυακής φύσεως.
Το πρόγραμμα που θα χρησιμοποιηθεί ως εργαλείο επίδειξης και υλοποίησης των ανωτέρω ασκήσεων είναι το Wireshark. Το Wireshark αποτελεί ένα από τα διασημότερα προγράμματα παγκοσμίως για την ανάλυση δικτύων. Παρακολουθεί και καταγράφει σε πραγματικό χρόνο τη λειτουργία ενός δικτύου υπολογιστών, παρέχει όλες τις σχετικές πληροφορίες για τη κίνηση του δικτύου, τα πρωτόκολλα ανώτερου επιπέδου που το απαρτίζουν και τα δεδομένα που διακινούνται μέσα από αυτό. Παράλληλα, μπορεί να χρησιμοποιηθεί για εκπαιδευτικούς σκοπούς στα πλαίσια ενός εργαστηριακού μαθήματος "Δικτύων Υπολογιστών" για την επίδειξη και κατανόηση της λειτουργίας ενός δικτύου, την ανάλυση των πρωτοκόλλων του και την ανταλλαγή δεδομένων που πραγματοποιείται πάνω από αυτό.

-Στο 1ο κεφάλαιο γίνεται μια εισαγωγή στα δικτύα υπολογιστών και στα TCP/IP δικτύα ειδικότερα. Έπισης γίνεται αναφορά στα πρωτόκολλα που απαρτίζουν το διαδίκτυο, τα οποία θα μας απασχολήσουν στα παρακάτω κεφάλαια.
-Στο 2ο κεφάλαιο γίνεται μια παρουσίαση και ανάλυση των στοιχείων του Wireshark (περιβάλλον, δομή, φίλτρα, δυνατότητες κ.τ.λ.)
-Στο 3ο κεφάλαιο θα εξετάσουμε τα βασικά εργαλεία του Wireshark και θα δούμε βήμα προς βήμα μια πρώτη δοκιμαστική εκτέλεση του προγράμματος.
-Στο 4ο κεφ. θα εξετάσουμε το πρωτόκολλο HTTP, το πιο συνηθισμένο πρωτόκολλο στο World Wide Web. Θα δούμε τη βασική αλληλεπίδραση μεταξύ πελάτη και διανομέα, την ανάκτηση αρχείων HTML, την εξουσιοδότηση και την ασφάλεια στο HTTP.
-Στο 5ο κεφ. θα εξετάσουμε το πρωτόκολλο DNS, υπεύθυνο για την αντιστοίχηση μεταξύ των δυαδικών διευθύνσεων και των διευθύνσεων σε μορφή ASCII χαρακτήρων και θα χρησιμοποιήσουμε τα εργαλεία nslookup και ipconfig
-Στο 6ο κεφ. θα δούμε ένα από τα βασικότερα πρωτόκολλα, το TCP, που χρησιμοποιείται σχεδόν παντού και στόχος του είναι να επιβεβαιώνει την αξιόπιστη αποστολή και λήψη δεδομένων. Θα δούμε τα βασικα του χαρακτηριστικά, θα παρατηρήσουμε τον αλγόριθμο ελένχου συμφόρησης και θα εξετάσουμε το μηχανισμό ελένχου ροής του TCP
-Στο 7ο κεφ. θα ρίξουμε μια γρήγορη ματιά στο σχετικά απλό πρωτόκολλο μεταφοράς UDP που είναι γρήγορο και αποτελεσματικό, τουλάχιστον για τις εφαρμογές εκείνες που δεν απαιτούν αξιόπιστη επικοινωνία.
-Στο 8ο κεφ. θα δουμε το πρωτόκολλο IP, το κύριο πρωτόκολλο πάνω στο οποίο είναι βασισμένο το Διαδικτύο και είναι υπεύθυνο για τη δρομολόγηση των πακέτων

δεδομένων ανάμεσα στα διάφορα δίκτυα. Θα εξετάσουμε τα διάφορα πεδία ενός IP datagram και θα μελετήσουμε τον κατακερματισμό

-Στο 9ο κεφ. θα εξετάσουμε μερικά χαρακτηριστικά του ICMP. Το πρωτόκολλο που χρησιμοποιείται κυρίως για την ανταλλαγή μηνυμάτων λάθους στο διαδίκτυο, θα εξετάσουμε τα μυνήματα ICMP που προκύπτουν απο τα προγράμματα Ping και Traceroute

-Στο 10ο κεφ. θα ερευνήσουμε δυο πρωτόκολλα που συνήθως τα συναντάμε μαζί. Το Ethernet, το συνηθέστερο χρησιμοποιούμενο πρωτόκολλο ενσύρματης τοπικής δικτύωσης υπολογιστών, και το πρωτόκολλο ARP το οποίο χρησιμοποιείται για την απεικόνιση IP διευθύνσεων στις φυσικές διευθύνσεις που χρησιμοποιούνται από το επίπεδο σύνδεσης

-Στο 11ο κεφ. θα δούμε το πρωτόκολλο DHCP, ένα μηχανισμό διαχείρισης πρωτοκόλλων TCP/IP που χρησιμοποιείται ευρέως για τη παροχή διευθύνσεων IP στους πελάτες

-Στο 12ο κεφ. θα εξετάσουμε το πρωτόκολλο για ασύρματο δίκτυο 802.11. που χρησιμοποιείται για την παροχή ασύρματων δυνατοτήτων πρόσβασης στο Internet, τηλεφωνίας μέσω διαδικτύου και διασύνδεσης μεταξύ ηλεκτρονικών συσκευών και θα δούμε πως ανταλλάσσονται τα μυνήματα "στον αέρα"

-Στο 13ο κεφ. θα δούμε το πρωτόκολλο SSL που σχεδιάστηκε για να παρέχει ασφάλεια κατά την μετάδοση ευαίσθητων δεδομένων στο διαδίκτυο και θα εξετάσουμε τον τρόπο που κρυπτογραφούνται τα αρχεία

-Ακολουθούν στο 14ο κεφάλαιο τα Συμπεράσματα και η Βιβλιογραφία

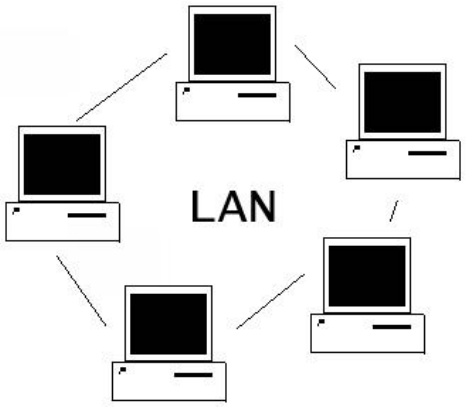
ΚΕΦΑΛΑΙΟ 1
ΕΙΣΑΓΩΓΗ ΣΤΑ ΔΙΚΤΥΑ TCP/IP

1.1 Δίκτυα Υπολογιστών

Η ανάγκη για επικοινωνία μεταξύ υπολογιστών ήταν σχεδόν ταυτόχρονη με την κατασκευή των πρώτων υπολογιστών. Έπρεπε να βρεθεί ένας τρόπος, ώστε άνθρωποι από διαφορετικά μέρη της χώρας να μπορούν να χρησιμοποιήσουν τις δυνατότητες των πανάκριβων αυτών υπολογιστών. Με την επικράτηση του προσωπικού υπολογιστή στις επιχειρήσεις, υπήρξε άμεση ανάγκη να συνδεθούν οι υπολογιστές μεταξύ τους, για να μπορούν οι χρήστες να ανταλλάζουν δεδομένα, μηνύματα και να μπορούν να χρησιμοποιήσουν τον κοινό εξοπλισμό, όπως εκτυπωτές, μόντεμ και άλλο υλικό. Έτσι, δημιουργήθηκαν τα πρώτα δίκτυα υπολογιστών. Το δίκτυο δηλαδή αποτελείται από ένα σύνολο Η/Υ, που συνδέονται μεταξύ τους με σκοπό την ανταλλαγή δεδομένων/πληροφοριών και την κοινή χρήση περιφερειακών συσκευών.

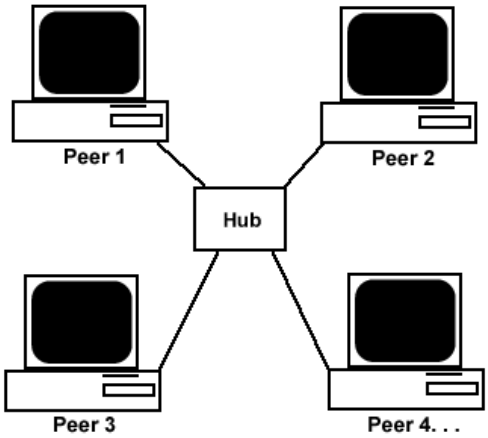
Τα βασικότερα πλεονεκτήματα της δικτύωσης υπολογιστών είναι
-Η **κοινή χρήση αρχείων**
Οι πληροφορίες, τις οποίες περιέχει ένα αρχείο, να είναι διαθέσιμες σε όλους όσους έχουν δικαίωμα πρόσβασης σε αυτό το αρχείο. Σημαντικό στοιχείο, επίσης, είναι η κοινή χρήση εφαρμογών λογισμικού, καθώς και η χρήση ειδικού λογισμικού, που επιτρέπει την συνεργασία και επικοινωνία των χρηστών του δικτύου
-Η **κοινή χρήση πόρων**
Ο όρος πόρος αναφέρεται στις συσκευές, οι οποίες είναι εγκατεστημένες για λειτουργία στο δίκτυο, πχ ένας εκτυπωτής. Αντί να υπάρχει εκτυπωτής συνδεδεμένος σε κάθε υπολογιστή, όλοι οι χρήστες του δικτύου μπορούν να εκτυπώσουν στον κοινόχρηστο εκτυπωτή.

Τα δίκτυα με βάση την γεωγραφική κάλυψη χωρίζονται σε :
ΤΟΠΙΚΑ ΔΙΚΤΥΑ
Ο όρος τοπικό δίκτυο (LAN- Local Area Network) περιγράφει ένα δίκτυο υπολογιστών όπου οι υπολογιστές βρίσκονται σε μικρή απόσταση μεταξύ τους (π.χ. μέσα σε ένα γραφείο ή σε ένα κτίριο) και μπορούν να επικοινωνήσουν με μεγάλες ταχύτητες (από 10 Mbit/sec μέχρι και 1 Gbit/sec). Υπάρχουν δυο είδη τοπικών δικτύων, με μεγάλες διαφορές στον τρόπο λειτουργίας τους.



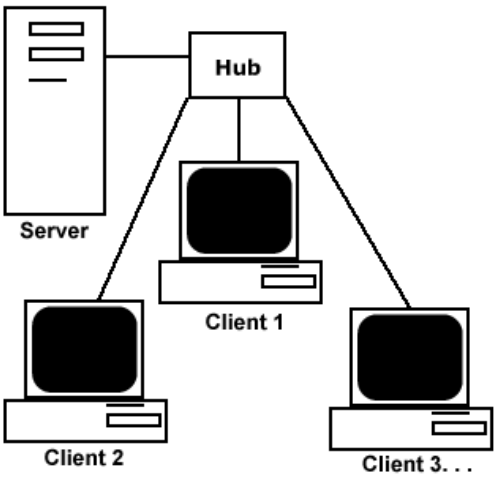
Σχ.1 Συνδεσμολογία δικτύου LAN

Ομότιμα δίκτυα (peer to peer)
Όλοι οι υπολογιστές είναι ισοδύναμοι. Κάθε υπολογιστής εργάζεται ανεξάρτητα από τους άλλους, μπορεί όμως να επιτρέψει την πρόσβαση στους πόρους του (αρχεία-περιφερειακά) και στους υπόλοιπους υπολογιστές του δικτύου. Τα ομότιμα δίκτυα δεν είναι τόσο εξελιγμένα σε θέματα ασφάλειας και κεντρικής διαχείρισης, έχουν όμως μικρό κόστος εγκατάστασης και συντήρησης.



Σχ.2 Συνδεσμολογία peer to peer δικτύου

Δίκτυα πελάτη/διακομιστή (Client/Server)
Στο μοντέλο αυτό, ένας ή περισσότεροι υπολογιστές (server) αφιερώνονται για να εξυπηρετούν τους υπόλοιπους (client). Τα αρχεία αποθηκεύονται στον κεντρικό υπολογιστή κάνοντας εύκολη τη διαχείριση τους (ακεραιότητα δεδομένων, δημιουργία εφεδρικών αντιγράφων, έλεγχος πρόσβασης και προστασία). Το κόστος όμως εγκατάστασης και συντήρησης είναι σημαντικά μεγαλύτερο. Σε αντίθεση με τα ομότιμα δίκτυα, όπου ο αριθμός των υπολογιστών είναι περιορισμένος, τα δίκτυα πελάτη/διακοσμητή έχουν πολύ μεγάλη επεκτασιμότητα.



Σχ.3 Συνδεσμολογία δικτύου πελάτη/διακοσμιτή

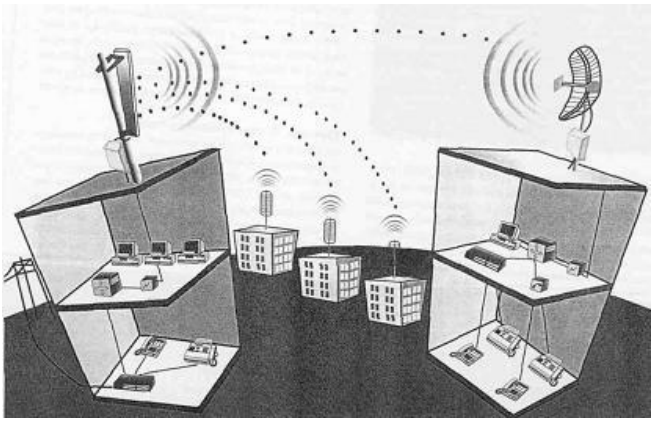
ΔΙΚΤΥΑ ΕΥΡΕΙΑΣ ΠΕΡΙΟΧΗΣ (WAN)

Τα δίκτυα αυτά χρησιμοποιούνται για τη διασύνδεση υπολογιστών, οι οποίοι βρίσκονται σε διαφορετικές πόλεις ή ακόμα και χώρες. Δηλαδή, ένα δίκτυο ευρείας περιοχής αποτελείται από γεωγραφικά απομακρυσμένα τοπικά δίκτυα, τα οποία συνδέονται μεταξύ τους μέσω τηλεφωνικού δικτύου ή μέσω δορυφόρων. Η δρομολόγηση δεδομένων μεταξύ τοπικών δικτύων, γίνεται με ειδικές συσκευές που λέγονται δρομολογητές (routers). Την χρήση των WAN κάνουν μεγάλες εταιρίες για να επικοινωνούν με τα υποκαταστήματα τους (π.χ. η διασύνδεση των υποκαταστημάτων μιας τράπεζας, τα οποία είναι διάσπαρτα στην Ελλάδα, αλλά και στο εξωτερικό). Το Internet είναι το μεγαλύτερο δίκτυο ευρείας περιοχής, στο οποίο είναι συνδεδεμένα χιλιάδες μικρότερα δίκτυα και προσωπικοί υπολογιστές σε όλο το κόσμο.


Σχ.4 Δίκτυο Ευρείας Περιοχής (WAN)

ΑΣΥΡΜΑΤΟ ΔΙΚΤΥΟ

Οι υπολογιστές συνδέονται με ραδιοκύματα και όχι με καλώδια.. Χρησιμοποιείται σαν επέκταση ή εναλλακτική λύση ενός ενσύρματου τοπικού δικτύου υπολογιστών και επιτρέπει στους χρήστες να λαμβάνουν και να μεταδίδουν δεδομένα καθώς μετακινούνται. Στα πλεονεκτήματα των ασύρματων δικτύων περιλαμβάνονται η κινητικότητα και η απουσία καλωδίων. Στα μειονεκτήματα περιλαμβάνονται μεταξύ άλλων η βραδύτερη σύνδεση σε σύγκριση με τα ενσύρματα δίκτυα και οι παρεμβολές από άλλες ασύρματες συσκευές, όπως για παράδειγμα τα ασύρματα τηλέφωνα



Σχ.5 Ασύρματο Δίκτυο

1.2 ΔΙΑΔΙΚΤΥΟ - INTERNET

Το Internet είναι το παγκόσμιο δίκτυο όπου είναι συνδεδεμένα χιλιάδες μικρότερα και μεγαλύτερα δίκτυα, από όλο το κόσμο. Υπολογιστές κάθε τύπου και μεγέθους, που βρίσκονται σε σπίτια, πανεπιστήμια, επιχειρήσεις, κυβερνητικούς οργανισμούς, συνδέονται σε δίκτυα. Συνδέοντας αυτά τα δίκτυα μεταξύ τους, σχηματίζεται το Internet.
Το Internet οφείλει, αναμφισβήτητα, τη μεγάλη διάδοση του, στο World Wide Web (Παγκόσμιος Ιστός). Αν και πολύ κόσμος ταυτίζει το Web με το Internet, στην ουσία είναι ένα υποσύνολο του. Όπως φανερώνει και το όνομα του (Ιστός), είναι ένα παγκόσμιο σύστημα από δικτυωμένους υπολογιστές (Web Servers), που παρέχουν στους χρήστες συνδεδεμένα μεταξύ τους έγγραφα.

Για να συνδεθούμε στο Internet, χρειαζόμαστε
-Προσωπικό υπολογιστή
-Modem (συσκευή που χρησιμοποιείται για την μετατροπή των σημάτων από ψηφιακά, τα οποία χρησιμοποιούν οι Η/Υ, σε αναλογικά, για την μεταφορά τους κυρίως με τις τηλεφωνικές γραμμές) ή συσκευή ISDN (Integrated Services Digital Network-Ψηφιακό Δίκτυο Ενοποιημένων Υπηρεσιών, αποτελεί την εξέλιξη του αναλογικού τηλεφωνικού δικτύου και επιτρέπει την ψηφιακή μετάδοση σήματος από άκρο σε άκρο. Παρέχει την δυνατότητα ταυτόχρονης χρήσης διαφόρων μορφών επικοινωνίας, όπως ήχου, εικόνας και δεδομένων, μέσα από μια τηλεφωνική σύνδεση).
-Τηλεφωνική σύνδεση (τυπική τηλεφωνική γραμμή ή ISDN)
-Συνδρομή σε μια εταιρία ή οργανισμό, που να μας παρέχει υπηρεσίες Internet (ISP-Internet Service Provider)
-Λογισμικό, που να μας επιτρέπει να συνδεθούμε μ' ένα διακοσμιτή και ειδικά προγράμματα, ανάλογα με την υπηρεσία, που θέλουμε να χρησιμοποιήσουμε.

ΥΠΗΡΕΣΙΕΣ ΤΟΥ INTERNET
-**Ο Παγκόσμιος Ιστός** (WWW-World Wide Web) είναι το δημοφιλέστερο τμήμα του Internet. Αποτελείται από ένα τεράστιο πλήθος εγγράφων, τα οποία συνδέονται μεταξύ τους με συνδέσμους υπερκειμένου. Οι σύνδεσμοι είναι κείμενο ή εικόνες που οδηγούν σε άλλα έγγραφα, όταν επιλέγονται με το δείκτη του ποντικιού. Για να επισκεφτούμε μια τοποθεσία του web και να εμφανίσουμε τις πληροφορίες, που μας παρέχει,

χρειαζόμαστε ένα ειδικό πρόγραμμα το οποίο λέγεται web browser (φυλλομετρητής ή πρόγραμμα περιήγησης ή πλοήγησης ιστού). Μερικά τέτοια είναι το Mozilla Firefox, Internet Explorer και άλλα. Κάθε έγγραφο του Ιστού έχει μια μοναδική διεύθυνση. Αποτελείται από το πρωτόκολλο μεταφοράς (http://), τη διεύθυνση του διακοσμητή (π.χ. www.in.gr) και το όνομα του εγγράφου (π.χ. index.html)

-Μηχανές Αναζήτησης
Μια μηχανή αναζήτησης είναι μια εφαρμογή που επιτρέπει την αναζήτηση κειμένων και αρχείων στο Διαδίκτυο. Αποτελείται από ένα πρόγραμμα υπολογιστή που βρίσκεται σε έναν ή περισσότερους υπολογιστές στους οποίους δημιουργεί μια βάση δεδομένων με τις πληροφορίες που συλλέγει από το διαδίκτυο. Ο χρήστης πληκτρολογεί μια λέξη- κλειδί και η μηχανή αναζήτησης θα εμφανίσει μια λίστα με συνδέσμους που οδηγούν σε τοποθεσίες σχετικές με το θέμα αναζήτησης. Γνωστές μηχανές αναζήτησης είναι η Google, Yahoo, Ask...

-Ηλεκτρονικό Ταχυδρομείο
Το ηλεκτρονικό ταχυδρομείο (E-mail) είναι η πιο διαδεδομένη υπηρεσία του διαδικτύου και αποτελεί έναν ταχύτατο τρόπο επικοινωνίας μεταξύ χρηστών του Internet σε ολόκληρο τον κόσμο. Είναι μια μορφή επικοινωνίας η οποία επιτρέπει στους χρήστες του διαδικτύου να στείλουν ένα μήνυμα σε άλλους χρήστες, που έχουν ηλεκτρονική διεύθυνση (e-mail address) με τρόπο που μοιάζει με αυτόν του κλασικού ταχυδρομείου. Μια ηλεκτρονική διεύθυνση αποτελείται από το όνομα του χρήστη, το σύμβολο @ και το όνομα της περιοχής (domain) στην οποία βρίσκεται ο διακομιστής αλληλογραφίας, όπου έχει λογαριασμό ο χρήστης.

-Μεταφορά Αρχείων
Τα αρχεία στο Internet μπορούν να μεταφέρονται από τον ένα υπολογιστή στον άλλο, επικοινωνώντας με μια κοινή γλώσσα (πρωτόκολλο) που ονομάζεται FTP (File Transfer Protocol). Στο Διαδίκτυο υπάρχει πλήθος από τοποθεσίες FTP (FTP sites) από τα οποία μπορείτε να «κατεβάσετε» (Download) αρχεία, δηλαδή, να τα μεταφέρετε από έναν απομακρυσμένο υπολογιστή στον υπολογιστή σας ή να «ανεβάσετε» αρχεία, δηλαδή να τα στείλετε σε έναν απομακρυσμένο υπολογιστή.

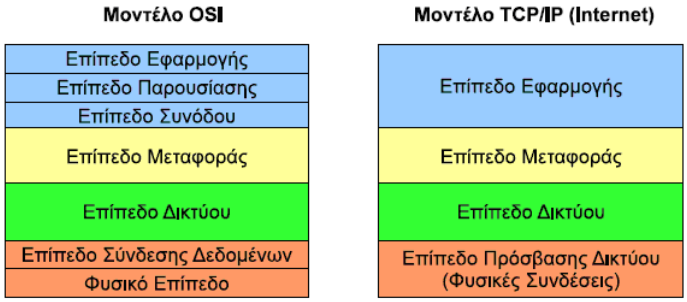
ΠΡΩΤΟΚΟΛΛΑ ΕΠΙΚΟΙΝΩΝΙΑΣ

Στην καθημερινή μας ζωή, πρωτόκολλο είναι ένα σύνολο από συμβάσεις που καθορίζουν το πώς πρέπει να πραγματοποιηθεί κάποια διαδικασία. Για παράδειγμα η αναγραφή των στοιχείων του αποστολέα και του παραλήπτη σε κάποιο φάκελο που πρόκειται να ταχυδρομηθεί είναι ένα είδος πρωτοκόλλου. Η διεύθυνση του παραλήπτη και η διεύθυνση του αποστολέα στο φάκελο είναι μηνύματα προς το ταχυδρομικό γραφείο, που περιγράφουν που θα πάει το γράμμα, σε διάφορες περιπτώσεις. Τα μηνύματα αυτά πρέπει να εμφανίζονται στις προβλεπόμενες θέσεις του φακέλου, και πρέπει να έχουν μια μορφή που να την καταλαβαίνει η ταχυδρομική υπηρεσία, αν θέλουμε να παραδοθεί σωστά ο φάκελος.
Στον κόσμο των δικτύων και του Internet, πρωτόκολλο είναι ένα σύνολο από συμβάσεις που καθορίζουν τον τρόπο που ανταλλάσσουν μεταξύ τους δεδομένα οι υπολογιστές που μπορεί να είναι διαφορετικού τύπου και να ανήκουν σε διαφορετικά δίκτυα. Τα πρωτόκολλα είναι αυτά που καθορίζουν το πώς διακινούνται τα δεδομένα, το πώς γίνεται ο έλεγχος, ο χειρισμός των λαθών, κλπ.

1.3 TCP/IP Δίκτυα

Το TCP/IP είναι μια συλλογή πρωτοκόλλων επικοινωνίας στα οποία βασίζεται το Διαδίκτυο αλλά και μεγάλο ποσοστό των εμπορικών δικτύων. Η ονομασία TCP/IP προέρχεται από τις συντομογραφίες των δυο κυριότερων πρωτοκόλλων που περιέχει, το TCP ή Transmission Control Protocol (Πρωτόκολλο Ελέγχου Μετάδοσης) και το IP ή Internet Protocol (Πρωτόκολλο Διαδικτύου).

Αυτή η συλλογή πρωτοκόλλων, είναι οργανωμένη σε 4 στρώματα ή επίπεδα (layers) και αποτελεί εξέλιξη του μοντέλου OSI, το οποίο παραμένει έως σήμερα μόνο θεωρητικό και προτείνει την κατάταξη των πρωτοκόλλων δικτύων σε έναν οργάνωση 7 στρωμάτων. Το καθένα τους απαντά σε συγκεκριμένα προβλήματα μεταφοράς δεδομένων και παρέχει μια καθορισμένη υπηρεσία στα υψηλότερα στρώματα. Τα ανώτερα επίπεδα είναι πιο κοντά στη λογική του χρήστη και εξετάζουν πιο αφηρημένα δεδομένα, στηριζόμενα σε πρωτόκολλα χαμηλότερων στρωμάτων για να μεταφράσουν δεδομένα σε μορφές που μπορούν να διαβιβαστούν με φυσικά μέσα. Στο παρακάτω σχήμα φαίνεται η σχέση των δυο μοντέλων



Σχ.6 Σχέση μοντέλου OSI και TCP/IP

Επίπεδο Εφαρμογής Το επίπεδο εφαρμογής παρέχει τις εφαρμογές (προγράμματα) που χρησιμοποιούν τα πρωτόκολλα του επιπέδου μεταφοράς και είναι το σημείο που ο τελικός χρήστης έρχεται σε επαφή με την στοίβα πρωτοκόλλων της τεχνολογίας TCP/IP. Το βασικό μοντέλο επικοινωνίας που χρησιμοποιείται στις περισσότερες εφαρμογές TCP/IP είναι το μοντέλο ***πελάτη– εξυπηρετητή***. Ο εξυπηρετητής είναι μια διεργασία (πρόγραμμα) η οποία εκτελείται σε ένα υπολογιστή (server) και ελέγχει τις εισερχόμενες αιτήσεις πελατών για να δει αν κάποια απευθύνεται προς αυτήν. Αν υπάρχει κάποια τέτοια αίτηση, ο εξυπηρετητής αναλαμβάνει να βρει τα δεδομένα που ζητούνται και να τα στείλει στον πελάτη. Ο πελάτης είναι πάλι αντίστοιχα το πρόγραμμα που χρησιμοποιείται (συνήθως από τον τελικό χρήστη) για να ζητήσει τα δεδομένα από τον εξυπηρετητή. Ο πελάτης στέλνει την αντίστοιχη αίτηση και περιμένει να λάβει τα δεδομένα που ζήτησε. Με το τέλος της εξυπηρέτησης ενός πελάτη, ο εξυπηρετητής επιστρέφει ξανά σε κατάσταση αναμονής, περιμένοντας νέα αίτηση (τυπικά ένας εξυπηρετητής είναι σε θέση να εξυπηρετήσει ταυτόχρονα περισσότερες από μια αιτήσεις).

Επίπεδο Μεταφοράς Το επίπεδο μεταφοράς είναι υπεύθυνο για την υλοποίηση των συνδέσεων μεταξύ των υπολογιστών ενός δικτύου. Η λειτουργία του στρώματος αυτού μπορεί να συγκριθεί με αυτή οποιουδήποτε μηχανισμού/μέσου μεταφοράς, π.χ. ένα όχημα που πρέπει να εξασφαλίζει την πλήρη και ασφαλή διακίνηση του φορτίου του. Το στρώμα μεταφοράς παρέχει αυτή την υπηρεσία σύνδεσης εφαρμογών μεταξύ τους, κάνοντας χρήση **θυρών (ports).** Το βασικό πρωτόκολλο εδώ είναι το TCP (πρωτόκολλο με σύνδεση) ενώ μπορεί να χρησιμοποιηθεί και το UDP (πρωτόκολλο χωρίς σύνδεση). Το TCP είναι υπεύθυνο για την αποκατάσταση αξιόπιστων ταυτόχρονων συνδέσεων διπλής κατεύθυνσης. Η έννοια του *αξιόπιστου* είναι ότι το TCP αναλαμβάνει να διορθώσει τα λάθη που τυχόν παρουσιάζονται στη μετάδοση (π.χ. μεταδίδοντας ξανά ένα πακέτο που χάθηκε ή αλλοιώθηκε). Το TCP παρέχει τις υπηρεσίες του στο αμέσως ανώτερο επίπεδο (Εφαρμογής). Καθώς θεωρείται ότι οι συνδέσεις που παρέχει είναι αξιόπιστες, τα προγράμματα στο επίπεδο εφαρμογής δεν κάνουν κανένα έλεγχο για ορθότητα των δεδομένων που προέρχονται από το TCP. Η έννοια του *ταυτόχρονου* είναι ότι ένας υπολογιστής μπορεί σε μια δεδομένη στιγμή να διατηρεί πλήθος διαφορετικών συνδέσεων TCP οι οποίες να λειτουργούν όλες μαζί αλλά καμιά να μην επηρεάζει την άλλη. *Επικοινωνία διπλής κατεύθυνσης* σημαίνει ότι μέσω μιας σύνδεσης μπορούν ταυτόχρονα να μεταδίδονται και να λαμβάνονται δεδομένα. Το πρωτόκολλο αυτοδύναμων πακέτων UDP δεν είναι ιδιαίτερα αξιόπιστο, αλλά επειδή είναι λιγότερο πολύπλοκο χρησιμοποιείται σε περιπτώσεις που η αξιοπιστία δεν είναι κρίσιμη και δεν είναι η επιθυμητή η χρήση του TCP.

Επίπεδο Δικτύου Το επίπεδο αυτό είναι υπεύθυνο για τη μετάδοση στο φυσικό δίκτυο των πακέτων που δημιουργούνται από τα πρωτόκολλα TCP και UDP που βρίσκονται στο αμέσως ανώτερο επίπεδο (Μεταφοράς). Το πρωτόκολλο IP είναι το βασικό πρωτόκολλο που χρησιμοποιείται σε αυτό το επίπεδο. Είναι αυτό που μας εξασφαλίζει την παγκόσμια διασυνδεσιμότητα και είναι υπεύθυνο για την παροχή λογικών διευθύνσεων (διευθύνσεων IP) στα σημεία διεπαφής του με το φυσικό δίκτυο (σε κάθε δηλ. συσκευή του δικτύου που διαθέτει δική της διεύθυνση). Είναι επίσης υπεύθυνο για την αντιστοίχηση των λογικών (IP) διευθύνσεων με τις φυσικές διευθύνσεις. Για τις εργασίες αυτές χρησιμοποιείται το πρωτόκολλο ARP (Address Resolution Protocol). Στο επίπεδο δικτύου λειτουργεί επίσης και το πρωτόκολλο ICMP ή Πρωτόκολλο Ελέγχου Μεταφοράς Μηνυμάτων. Αυτό χρησιμοποιείται για να αναφέρει προβλήματα και ασυνήθιστες καταστάσεις που σχετίζονται με το πρωτόκολλο IP.

Επίπεδο Ζεύξης Δεδομένων (ή επίπεδο πρόσβασης δικτύου). Παρέχει την πρόσβαση στο φυσικό μέσο στο οποίο η πληροφορία μεταδίδεται με την μορφή πακέτων. Το επίπεδο πρόσβασης δικτύου αντιπροσωπεύει το χαμηλότερο επίπεδο λειτουργικότητας που απαιτείται από ένα δίκτυο και περιλαμβάνει όλα τα στοιχεία της φυσικής σύνδεσης: καλώδια, κάρτες δικτύου, πρωτόκολλα πρόσβασης τοπικών δικτύων. Όπως κάθε επίπεδο στο TCP/IP (αλλά και στο OSI), το επίπεδο αυτό παρέχει τις υπηρεσίες του στο αμέσως ανώτερο επίπεδο, το επίπεδο δικτύου.

1.4 Επίπεδο Εφαρμογής

Το επίπεδο εφαρμογής είναι το μέρος όπου βρίσκονται οι εφαρμογές του δικτύου και τα πρωτόκολλα του επιπέδου εφαρμογής. Στο επίπεδο εφαρμογής τα κυριότερα πρωτόκολλα είναι τα εξής HTTP, FTP, SSL, DNS

1.4.1 Πρωτόκολλο HTTP

Το Πρωτόκολλο Μεταφοράς Υπερκειμένου **HTTP** (HyperText Transfer Protocol) λειτουργεί στο επίπεδο Εφαρμογής και αποτελεί το κύριο πρωτόκολλο που χρησιμοποιείται στους φυλλομετρητές του Παγκοσμίου Ιστού για να μεταφέρει δεδομένα (αρχεία κειμένου, γραφικών, εικόνας, ήχου, video ή οποιουδήποτε multimedia αρχείου) ανάμεσα σε έναν διακομιστή (server) και έναν πελάτη (client). Σήμερα το πρωτόκολλο αυτό είναι πλέον καθιερωμένο και διαδεδομένο σε σημείο που σχεδόν όλοι οι φυλλομετρητές να το θεωρούν δεδομένο και να το χρησιμοποιούν σε περίπτωση που ο χρήστης δεν καθορίσει ποιο πρωτόκολλο θέλει να χρησιμοποιήσει. Αν δηλαδή ο χρήστης δεν γράψει http://my.url αλλά γράψει σκέτο το: my.url σχεδόν όλοι οι φυλλομετρητές θεωρούν σαν δεδομένο το πρωτόκολλο http και όχι κάποιο άλλο.
Το HTTP υλοποιείται σε δυο προγράμματα: ένα πρόγραμμα πελάτη (φυλλομετρητής - browser) και ένα πρόγραμμα εξυπηρέτη. Το πρόγραμμα πελάτη και το πρόγραμμα εξυπηρέτη, που εκτελούνται σε διαφορετικά τερματικά συστήματα, συνομιλούν μεταξύ τους ανταλλάσσοντας μηνύματα HTTP. Το HTTP ορίζει την δομή αυτών των μηνυμάτων και το πώς ο πελάτης και ο εξυπηρέτης ανταλλάσσουν τα μηνύματα. Όταν ο χρήστης ζητάει μια ιστοσελίδα, το πρόγραμμα περιήγησης στέλνει μηνύματα αίτησης HTTP για τα αντικείμενα της σελίδας, στον εξυπηρέτη. Ο εξυπηρέτης δέχεται αιτήσεις και αποκρίνεται με μηνύματα απόκρισης HTTP, τα οποία περιέχουν τα αντικείμενα.
Το HTTP χρησιμοποιεί το TCP ως υποκείμενο πρωτόκολλο μεταφοράς που παρέχει μια αξιόπιστη μεταφορά δεδομένων στο HTTP. Αυτό σημαίνει ότι κάθε μήνυμα αίτησης HTTP που εκπέμπεται από μια διεργασία πελάτη φτάνει τελικά ανέπαφο στον εξυπηρέτη. Εδώ βλέπουμε ένα από τα μεγαλύτερα πλεονεκτήματα μιας αρχιτεκτονικής οργανωμένης σε διαδοχικά επίπεδα – το HTTP δεν χρειάζεται να ασχολείται για χαμένα δεδομένα ή για λεπτομέρειες περί του πώς το TCP επανορθώνει από την απώλεια ή από την αλλαγή σειράς δεδομένων στο δίκτυο.

1.4.2 Πρωτόκολλο DNS

Μια διεύθυνση IP (όπως θα δούμε αναλυτικότερα παρακάτω) έχει μέγεθος 32 bit και τυπικά την γράφουμε με τη μορφή τεσσάρων δεκαδικών αριθμών τα οποία χωρίζονται μεταξύ τους με τελείες. Για παράδειγμα μια έγκυρη διεύθυνση IP είναι: 94.69.78.90
Οι χρήστες όμως βρίσκουν αρκετά δύσκολο να απομνημονεύσουν αυτούς τους αριθμούς. Για το σκοπό αυτό, κρίθηκε σκόπιμο να χρησιμοποιούνται στους υπολογιστές συμβολικά ονόματα. Το **DNS** (Domain Name System ή Σύστημα Ονομάτων Τομέων ή Χώρων ή Περιοχών) είναι ένα ιεραρχικό σύστημα ονοματοδοσίας για δίκτυα υπολογιστών, που χρησιμοποιούν το πρωτόκολλο IP. Το σύστημα DNS μπορεί και αντιστοιχίζει ονόματα με διευθύνσεις IP ή άλλα ονόματα στο Διαδίκτυο ή κάποιο άλλο δίκτυο.
Ένας απλός τρόπος να γίνει αυτό είναι κάθε υπολογιστής να διαθέτει ένα αρχείο το οποίο να περιέχει την αντιστοιχία συμβολικών ονομάτων και IP διευθύνσεων. Για να δουλέψει αυτό το σύστημα, θα πρέπει αυτό το αρχείο να περιέχει τα ονόματα και τις διευθύνσεις όλων των υπολογιστών του δικτύου, να υπάρχει σε όλους τους υπολογιστές και να διατηρείται ενημερωμένο όταν γίνονται αλλαγές. Ο υπολογιστής

που ξεκινάει μια αποστολή δεδομένων, θα ψάξει μέσα σε αυτό το αρχείο να βρει το όνομα του υπολογιστή προορισμού και από την ίδια γραμμή θα διαβάσει την διεύθυνση IP που πρέπει να χρησιμοποιήσει. Η μέθοδος αυτή με το αρχείο αντιστοίχισης διευθύνσεων – ονομάτων δουλεύει καλά όταν το δίκτυο είναι μικρό. Τα βασικά προβλήματα για να το χρησιμοποιήσουμε σε ένα μεγάλο δίκτυο είναι:
• Κάθε υπολογιστής του δικτύου πρέπει να έχει ένα αντίγραφο αυτού του αρχείου.
• Το αρχείο πρέπει να διατηρείται ενημερωμένο κάθε φορά που γίνεται κάποια αλλαγή στο δίκτυο. Για παράδειγμα όταν προσθέτουμε ή αφαιρούμε ένα υπολογιστή, ή όταν αλλάζουμε ένα όνομα ή διεύθυνση. Επίσης πρέπει να κρατάμε ενημερωμένα όλα τα αντίγραφα του αρχείου.
• Αν το πλήθος των υπολογιστών είναι μεγάλο, η αναζήτηση σε ένα απλό αρχείο κειμένου θα είναι πολύ αργή. Σε κάθε περίπτωση θα σπαταλήσουμε πολύ χρόνο και κόπο για να ενημερώσουμε όλα τα αντίγραφα. Το DNS περιέχει ένα χώρο ονομάτων οργανωμένο ιεραρχικά και η λειτουργία του βασίζεται σε μια κατανεμημένη βάση δεδομένων. Ο χώρος διαιρείται σε ένα σύνολο περιοχών που μπορούν να διαιρεθούν ξανά σε περισσότερες περιοχές. Μια τέτοια δομή μοιάζει με δέντρο και φαίνεται στο σχήμα

Το πρώτο επίπεδο περιοχών ονομάζονται βασικές περιοχές και βρίσκονται στα δεξιά του ονόματος. Στις ΗΠΑ υπάρχουν επτά τέτοιες περιοχές οι οποίες έχουν καθιερωθεί ουσιαστικά παγκόσμια, και στις οποίες κατατάσσονται τα δίκτυα ανάλογα με τις δραστηριότητες του οργανισμού ή της επιχείρησης στην οποία ανήκουν. Οι περιοχές αυτές είναι οι παρακάτω:
- .arpa: Ειδικοί οργανισμοί διαδικτύου
- .com: Εταιρίες
- .edu: Εκπαιδευτικά ιδρύματα
- .gov: Κυβερνητικοί οργανισμοί
- .mil: Στρατιωτικοί οργανισμοί
- .net: Κέντρα διοίκησης δικτύου
- .org: Οτιδήποτε δεν μπορεί να καταταγεί σε κάποια από τις προηγούμενες κατηγορίες (τυπικά μη-κερδοσκοπικοί οργανισμοί)

Εκτός από τις παραπάνω κατηγορίες οι οποίες ισχύουν στις ΗΠΑ (αν και αυτό δεν σημαίνει ότι μια δικτυακή τοποθεσία που τελειώνει π.χ. σε .com θα βρίσκεται στις ΗΠΑ – μπορεί να βρίσκεται οπουδήποτε και γενικά αυτός ο διαχωρισμός χρησιμοποιείται διεθνώς) υπάρχει επίσης μια βασική περιοχή ανά χώρα. Ο προσδιορισμός τους γίνεται με βάση ένα μικρό τμήμα (δύο – τρία γράμματα) του ονόματος της χώρας. Για παράδειγμα, η περιοχή που αντιστοιχεί στην Ελλάδα ονομάζεται .gr, της Γερμανίας είναι .de και της Μεγάλης Βρετανίας είναι .uk. Κάτω από κάθε βασική περιοχή βρίσκεται ένα δεύτερο επίπεδο περιοχών το οποίο ονομάζεται domain. Το δεύτερο αυτό επίπεδο, τυπικά προσδιορίζει τον οργανισμό ή την επιχείρηση ο οποίος χρησιμοποιεί το όνομα. Κάθε μια από αυτές τις περιοχές είναι μοναδική. Τα ονόματα (domain names) που εκχωρούνται είναι συνήθως αντιπροσωπευτικά της εταιρίας ή οργανισμού στον οποίο ανήκουν. Τα domain names τοποθετούνται αριστερά του ονόματος της βασικής περιοχής και διαχωρίζονται με μια τελεία.
Για παράδειγμα, το ntua.gr αναφέρεται στο δίκτυο του Εθνικού Μετσόβειου Πολυτεχνείου. Το όνομα domain ntua έχει αποδοθεί στο ίδρυμα για αυτό το σκοπό και το .gr δείχνει ότι ανήκει στη βασική περιοχή που έχει εκχωρηθεί για την Ελλάδα (NTUA=National Technical University of Athens).

Η εταιρία ή οργανισμός στην οποία έχει εκχωρηθεί ένα domain name είναι ο αποκλειστικά υπεύθυνος για την διαχείριση του. Για παράδειγμα, αν ο διαχειριστής δικτύου της εταιρίας αποφασίσει ότι το δίκτυο θα χωριστεί σε μικρότερα τμήματα (υποδίκτυα) το ίδιο μπορεί να γίνει και με την περιοχή ονομάτων του οργανισμού. Κάθε νέο υποδίκτυο αντιστοιχεί σε περιοχή ονομάτων τρίτου επιπέδου και ονομάζεται subdomain. Στο όνομα, εμφανίζεται αριστερά του domain name και χωρίζεται πάλι με μια τελεία.
Για παράδειγμα: telecom.ntua.gr Το telecom είναι ένα subdomain του domain ntua που βρίσκεται στην περιοχή .gr (Ελλάδας). Το συγκεκριμένο όνομα έχει αποδοθεί στην περιοχή που ανήκει το εργαστήριο τηλεπικοινωνιών του Πολυτεχνείου.
Ένα όνομα μπορεί να αναφέρεται σε ένα συγκεκριμένο υπολογιστή αντί για μια ολόκληρη περιοχή διευθύνσεων. Για παράδειγμα, αν έχουμε το subdomain: telecom.ntua.gr και θέλουμε να αναφερθούμε στον υπολογιστή "pc01" που ανήκει σε αυτόν, το πλήρες όνομα θα ήταν: pc01.telecom.ntua.gr. Με αυτό τον τρόπο έχουμε φτιάξει ονόματα τετάρτου επιπέδου.

1.4.3 Πρωτόκολλο FTP

Το πρωτόκολλο μεταφοράς αρχείων **FTP** (File Transfer Protocol) επιτρέπει τη μεταφορά αρχείων μεταξύ υπολογιστών που χρησιμοποιούν την τεχνολογία TCP/IP. Το πρωτόκολλο αυτό χρησιμοποιεί το γνωστό μας μοντέλο πελάτη-εξυπηρετητή. Το πρωτόκολλο FTP χρησιμοποιεί, στο επίπεδο μεταφοράς, το πρωτόκολλο TCP για την μεταφορά των δεδομένων του. Ουσιαστικά το FTP μας επιτρέπει να δημιουργούμε αντίγραφα αρχείων ενός απομακρυσμένου εξυπηρετητή και μας επιτρέπει για παράδειγμα να δουλέψουμε στο σπίτι μας αφού λάβουμε το κατάλληλο αρχείο από τον εξυπηρετητή του γραφείου μας. Για να εξασφαλιστεί η ασφάλεια του συστήματος και να επιτρέπεται μόνο σε εξουσιοδοτημένους χρήστες η πρόσβαση στα αρχεία ενός εξυπηρετητή, υλοποιείται ένα σύστημα ελέγχου εξουσιοδότησης. Το σύστημα αυτό βασίζεται στην χρήση ονόματος πρόσβασης και κωδικού τα οποία πρέπει να πληκτρολογήσει ο χρήστης για την είσοδο του στο σύστημα. Τα στοιχεία αυτά δημιουργούνται από το διαχειριστή του εξυπηρετητή FTP και ελέγχονται κάθε φορά. Ο όρος "μεταφορά" στο FTP δηλώνει ότι το αρχείο μεταφέρεται από τον ένα υπολογιστή (εξυπηρετητή) στον άλλο (πελάτη). Ωστόσο το πρωτότυπο αρχείο δεν επηρεάζεται από αυτή τη διαδικασία (πρόκειται για αντιγραφή, και όχι για μετακίνηση).

1.4.4 Πρωτόκολλο SSL

Το πρωτόκολλο **SSL** λειτουργεί έχει σχεδιαστεί για την ασφαλή επικοινωνία ευαίσθητων πληροφοριών όπως προσωπικά στοιχεία και αριθμούς πιστωτικών καρτών μεταξύ δύο συστημάτων, από τα οποία το ένα λειτουργεί σαν client και το άλλο σαν server. Η εξασφάλιση του απορρήτου γίνεται με την κρυπτογράφηση όλων των μηνυμάτων στο επίπεδο SSL Record Protocol. Παρέχει, επιπλέον, υποχρεωτική πιστοποίηση της ταυτότητας του server και προαιρετικά της ταυτότητας του client, μέσω έγκυρων πιστοποιητικών από έμπιστες Αρχές Έκδοσης Πιστοποιητικών (Certificates Authorities). Υποστηρίζει πληθώρα μηχανισμών κρυπτογράφησης για την αντιμετώπιση όλων των διαφορετικών αναγκών και εξασφαλίζει την ακεραιότητα των δεδομένων ώστε κανείς να μην μπορεί να αλλοιώσει την πληροφορία χωρίς να γίνει

αντιληπτός. Όλα τα παραπάνω γίνονται με τρόπο διαφανές και απλό. Αυτό που ουσιαστικά κάνει το SSL είναι να παίρνει τις πληροφορίες από τις εφαρμογές υψηλότερων επιπέδων, να τις κρυπτογραφεί και στην συνέχεια να τις μεταδίδει στο Internet προς τον Η/Υ που βρίσκεται στην απέναντι πλευρά και τις ζήτησε. Κάθε σύνδεση SSL ξεκινά πάντα με την ανταλλαγή μηνυμάτων από τον server και τον client έως ότου επιτευχθεί η ασφαλής σύνδεση, πράγμα που ονομάζεται χειραψία (handshake). Η χειραψία επιτρέπει στον server να αποδείξει την ταυτότητά του στον client χρησιμοποιώντας τεχνικές κρυπτογράφησης δημοσίου κλειδιού και στην συνέχεια επιτρέπει στον client και τον server να συνεργαστούν για την δημιουργία ενός συμμετρικού κλειδιού που θα χρησιμοποιηθεί στην γρήγορη κρυπτογράφηση και αποκρυπτογράφηση των δεδομένων που ανταλλάσσονται μεταξύ τους.
-Στους συμμετρικούς αλγόριθμους το κλειδί κρυπτογράφησης μπορεί να υπολογιστεί από το κλειδί που χρησιμοποιείται για την αποκρυπτογράφηση και το ανάποδο. Μάλιστα στις περισσότερες περιπτώσεις τα κλειδιά κρυπτογράφησης και αποκρυπτογράφησης είναι τα ίδια. Αυτοί οι αλγόριθμοι χρειάζονται την συμφωνία μεταξύ του αποστολέα και του παραλήπτη για το κλειδί που θα χρησιμοποιηθεί, για να μπορέσουν να επικοινωνήσουν με ασφάλεια. Η ασφάλεια των αλγόριθμων βασίζεται στην μυστικότητα αυτού του κλειδιού.
-Οι ασύμμετροι αλγόριθμοι ή αλγόριθμοι δημόσιου κλειδιού είναι σχεδιασμένοι έτσι ώστε το κλειδί που χρησιμοποιείται για την κρυπτογράφηση να είναι διαφορετικό από το κλειδί που χρησιμοποιείται για την αποκρυπτογράφηση. Πέρα από αυτό, το κλειδί αποκρυπτογράφησης δεν μπορεί να υπολογιστεί από το κλειδί κρυπτογράφησης. Οι αλγόριθμοι αυτοί καλούνται και "δημόσιου κλειδιού" γιατί το κλειδί κρυπτογράφησης μπορεί να δημοσιοποιηθεί.
Ο καθένας μπορεί να κρυπτογραφήσει ένα μήνυμα με το δημόσιο κλειδί αλλά μόνο αυτός που διαθέτει το αντίστοιχο ιδιωτικό κλειδί μπορεί να το αποκρυπτογραφήσει. Οι συμμετρικοί αλγόριθμοι είναι πολύ πιο γρήγοροι από τους ασύμμετρους αλγόριθμους. Ως εκ τούτου οι συμμετρικοί αλγόριθμοι χρησιμοποιούνται για την κρυπτογράφηση του κυρίου μέρους των δεδομένων, ενώ οι αλγόριθμοι δημόσιου κλειδιού βρίσκουν κατάλληλη εφαρμογή σε πρωτόκολλα ανταλλαγής κλειδιών και ψηφιακών υπογραφών.

1.5 Επίπεδο Μεταφοράς

Το επίπεδο μεταφοράς είναι υπεύθυνο για τη μεταφορά μηνυμάτων εφαρμογής, δηλαδή παρέχει στο επίπεδο εφαρμογής έτοιμα μηνύματα όπως αυτά απεστάλησαν, σαν να μην είχε μεσολαβήσει το δίκτυο. Στο επίπεδο μεταφοράς τα βασικά πρωτόκολλα είναι τα TCP και UDP

1.5.1 Πρωτόκολλο TCP

Το πρωτόκολλο **TCP** (Transmission Control Protocol - Πρωτόκολλο Ελέγχου Μεταφοράς*)* αποτελεί το βασικό πρωτόκολλο που βρίσκεται στο επίπεδο μεταφοράς της τεχνολογίας TCP/IP. Οι περισσότερες σύγχρονες υπηρεσίες στο Διαδίκτυο βασίζονται στο TCP. Οι κύριοι στόχοι του πρωτοκόλλου TCP είναι να επιβεβαιώνεται η αξιόπιστη αποστολή και λήψη δεδομένων, επίσης να μεταφέρονται τα δεδομένα χωρίς λάθη μεταξύ του στρώματος δικτύου (network layer) και του στρώματος εφαρμογής (application layer) και, φτάνοντας στο πρόγραμμα του στρώματος

εφαρμογής, να έχουν σωστή σειρά. Το TCP μεταδίδει μόνο όταν το πλήθος των δεδομένων που έχει λάβει είναι επαρκές για να συμπληρωθεί το μέγεθος του πακέτου που έχει συμφωνηθεί κατά την εγκατάσταση της σύνδεσης. Από την άλλη όταν λάβει δεδομένα τα οποία υπερβαίνουν αυτό το μέγεθος πακέτου, τα σπάει σε μικρότερα.
Τα πακέτα του πρωτοκόλλου TCP καλούνται **segments (τμήματα)**. Ένα από τα κυριότερα μέρη ενός segment είναι η TCP **επικεφαλίδα** (TCP header) και τα προς μετάδοση Δεδομένα (Data), η οποία παρέχει συγκεκριμένες πληροφορίες για το πρωτόκολλο TCP. Η επικεφαλίδα γενικά αποτελείται από τα βοηθητικά δεδομένα που προσθέτει το TCP και είναι απαραίτητα για τη μετάδοση. Τα δεδομένα είναι φυσικά κομμάτι των πραγματικών δεδομένων του χρήστη που θα μεταφερθούν από το συγκεκριμένο τμήμα.

+	Bits 0 - 3	4 - 9	10 - 15	16 - 31
0	**Source Port** Θύρα Προέλευσης			**Destination Port** Θύρα Προορισμού
32	**Sequence Number** Αριθμός ακολουθίας			
64	**Acknowledgment Number** Αριθμός επιβεβαίωσης			
96	**Data Offset**	**Reserved**	**Flags** Σημαίες	**Window** Παράθυρο
128	**Checksum** Άθροισμα ελέγχου			**Urgent Pointer** Επείγοντα δεδομένα
160	**Options** Επιλογές (προαιρετικές)			
160/192+	**Data** Δεδομένα			

Σχ.7 TCP επικεφαλίδα

Source Port Αυτό το πεδίο προσδιορίζει την port (θύρα) του αποστολέα
Destination Port Αυτό το πεδίο προσδιορίζει την port (θύρα) του παραλήπτη
Sequence Number Ο sequence number (αριθμός ακολουθίας) έχει διπλό ρόλο:
-Εαν υπάρχει η SYN flag (SYN σημαία) τότε είναι ο αρχικός αριθμός ακολουθίας (ISN - initial sequence number) και η πρώτη octet δεδομένων του πακέτου είναι ο ISN+1.
-Αλλιώς, εάν δεν υπάρχει η SYN flag, τότε η πρώτη octet δεδομένων είναι ο αριθμός ακολουθίας.
Acknowledgment number Όταν υπάρχει η ACK flag η τιμή αυτού του πεδίου δείχνει τον επόμενο sequence number (αριθμό ακολουθίας) που αναμένει ο αποστολέας.
Data offset Είναι ο αριθμός από words μεγέθους 32 bit στην επικεφαλίδα TCP (TCP header). Καθορίζει το μέγεθος της επικεφαλίδας (πολλαπλάσιο του 32) και επομένως δείχνει και την αρχή των δεδομένων.
Reserved Πεδίο 6 bit "κρατημένων" (reserved) για μελλοντική χρήση. Η τιμή των bit πρέπει να είναι 0.
Flags (επίσης γνωστό ως bits ελέγχου - Control bits) Περιέχει 6 bit - σημαίες:

Σημαία	Σημασία	Προέλευση ονομασίας
URG	Το πεδίο urgent pointer είναι σημαντικό	**URG**ent
ACK	Το πεδίο επιβεβαίωσης είναι σημαντικό	**ACK**nowledgment
PSH	Λειτουργία ώθησης	**PuSH**
RST	Επαναρύθμιση σύνδεσης	**ReSeT**
SYN	Συγχρονισμός αριθμών ακολουθίας	**SYN**chronize
FIN	Ο αποστολέας δεν στέλνει άλλα δεδομένα	**FIN** (=τέλος)

Σχ.8 Πίνακας αντιστοιχίας σημαίας, σημασίας, πρέλευση ονομασίας

Window Ο αριθμός από octets δεδομένων (bytes) που επιθυμεί να δεχτεί ο αποστολέας του πακέτου, αρχίζοντας από εκείνη που δείχνει το πεδίο επιβεβαίωσης (acknowledgment field,ACK).
Checksum Το πεδίο checksum μεγέθους 16 bit χρησιμοποιείται για έλεγχο λαθών στην επικεφαλίδα και στα δεδομένα.
Options Μεταβλητή, η οποία καθορίζει ειδικές επιλεγόμενες ρυθμίσεις και μπορεί να καταλάβει χώρο στο τέλος της επικεφαλίδας TCP. Το μήκος τους είναι πολλαπλάσιο των 8 bit και σε το περιεχόμενο της επικεφαλίδας μετά την τελευταία επιλογή πρέπει να γεμίζει (πχ. με μηδενικά - 0). Με αυτόν τον τρόπο το data offset θα δείχνει σωστά την αρχή των δεδομένων.
Urgent pointer Εάν είναι ενεργοποιημένο το URG bit ελέγχου, τότε αυτό το πεδίο δείχνει τον αριθμό ακολουθίας (sequence number) της octet που βρίσκεται αμέσως μετά το τελευταίο byte από τα επείγοντα δεδομένα. Έτσι παρουσιάζει τη θέση του τελευταίου byte με επείγοντα δεδομένα.

Τα δεδομένα που έχουν χωρισθεί σε τμήματα πρέπει όταν φτάσουν στον προορισμό τους να ενωθούν ξανά για να δημιουργήσουν το αρχικό μεγαλύτερο πακέτο. Για να γίνει αυτό πρέπει να μπουν στη σωστή σειρά. Αυτή είναι και η λειτουργία του πεδίου που ονομάζεται **Αριθμός Σειράς**. Κάθε τμήμα έχει το δικό του αριθμό σειράς, ο οποίος δηλώνει σε ποια θέση πρέπει να μπει το συγκεκριμένο τμήμα μαζί με τα υπόλοιπα για να δημιουργηθεί ξανά το αρχικό πακέτο.
Καθώς η επικοινωνία βρίσκεται σε εξέλιξη, ο παραλήπτης πρέπει να μπορεί να επιβεβαιώνει στον αποστολέα ότι λαμβάνει δεδομένα. Για το σκοπό αυτό ο παραλήπτης στέλνει τμήματα επιβεβαίωσης λήψης χρησιμοποιώντας στην επικεφαλίδα τους τον **Αριθμό Επιβεβαίωσης**. Ο αριθμός επιβεβαίωσης δηλώνει ότι έχουν ληφθεί όλες οι οκτάδες (bytes) μέχρι και αυτό τον αριθμό. Για παράδειγμα, ο αριθμός επιβεβαίωσης 1500 σημαίνει ότι έχουμε λάβει όλα τα δεδομένα μέχρι τον αριθμό οκτάδας 1500. Αν ο αποστολέας δεν λάβει επιβεβαίωση μέσα σε ένα λογικό χρονικό διάστημα, θα επαναλάβει τη μετάδοση των δεδομένων.

Το πρωτόκολλο TCP ελέγχει επίσης την ποσότητα των δεδομένων που μεταδίδονται κάθε φορά. Η λειτουργία αυτή είναι γνωστή ως **έλεγχος ροής** και πραγματοποιείται με τη βοήθεια του πεδίου επικεφαλίδας τμήματος που ονομάζεται **Παράθυρο (Window size)**. Για να έχουμε την καλύτερη δυνατή απόδοση η μετάδοση είναι συνεχής, δηλ. ο αποστολέας δεν περιμένει να λάβει επιβεβαίωση λήψης ενός τμήματος για να στείλει το επόμενο (διαφορετικά θα είχαμε πολύ μικρό ρυθμό μετάδοσης). Από την άλλη βέβαια δεν μπορεί να γίνεται συνέχεια αποστολή χωρίς κάποιο είδος επιβεβαίωσης

λήψης. Αν στέλνουμε με ταχύτητα πολύ μεγαλύτερη από αυτή που μπορεί να δεχθεί ο απομακρυσμένος υπολογιστής, κάποια στιγμή θα γεμίσει η ενδιάμεση μνήμη που χρησιμοποιείται για την προσωρινή αποθήκευση των δεδομένων και ο παραλήπτης θα αρχίσει να απορρίπτει τα εισερχόμενα δεδομένα αφού δεν θα έχει που να τα αποθηκεύσει. Για το λόγο αυτό και τα δύο άκρα της σύνδεσης πρέπει να υποδεικνύουν πόσα δεδομένα μπορούν να δεχθούν κάθε φορά, βάζοντας τον αντίστοιχο αριθμό οκτάδων στο πεδίο "Παράθυρο" της επικεφαλίδας.
Η **θύρα TCP** είναι ένας αριθμός που χαρακτηρίζει πλέον μέσα στο μηχάνημα του αποστολέα (ή του παραλήπτη) την ίδια την εφαρμογή που πρόκειται να λάβει τα δεδομένα του συγκεκριμένου TCP τμήματος. Έτσι, όταν για παράδειγμα ανοίξουμε ένα φυλλομετρητή όπως το Firefox και αρχίσουμε να βλέπουμε μια σελίδα, τα τμήματα TCP που φεύγουν από τον υπολογιστή μας ως κομμάτι της συγκεκριμένης επικοινωνίας, χαρακτηρίζονται από ένα αριθμό ο οποίος είναι η **θύρα αφετηρίας**. Τα τμήματα αυτά περιέχουν επίσης και μια **θύρα προορισμού** η οποία εξασφαλίζει ότι όταν το τμήμα ληφθεί από το μηχάνημα προορισμού θα κατευθυνθεί στη σωστή εφαρμογή (στη συγκεκριμένη περίπτωση στον εξυπηρετητή ιστοσελίδων). Τα τμήματα που θα λάβουμε ως απάντηση, θα έχουν πλέον ως θύρα προορισμού την ίδια με την οποία ξεκινήσαμε την επικοινωνία, και άρα θα κατευθυνθούν στο ίδιο παράθυρο του Firefox.

1.5.2 Πρωτόκολλο UDP

Το TCP είναι σχετικά πολύπλοκο πρωτόκολλο στη λειτουργία του. Ένα αποτέλεσμα αυτής της πολυπλοκότητας είναι ότι εισάγει κάποιες καθυστερήσεις στην επικοινωνία. Υπάρχουν όμως εφαρμογές που ένα πιο απλό πρωτόκολλο θα μας εξυπηρετούσε καλύτερα. Τέτοια είδη εφαρμογών είναι:
• Εφαρμογές που τα μηνύματα τους χωράνε κάθε φορά σε ένα μόνο τμήμα και δεν χρειαζόμαστε τη λειτουργία τεμαχισμού που μας παρέχει το TCP.
• Εφαρμογές που δεν έχει σημασία αν χαθούν κάποια δεδομένα στη μετάδοση, ή δεν έχει νόημα η επαναμετάδοση τους αλλά μας ενδιαφέρει ωστόσο η μετάδοση να προχωράει όσο το δυνατόν πιο γρήγορα και χωρίς καθυστερήσεις. (πχ εφαρμογές φωνής)
• Γενικά εφαρμογές που έχει περισσότερη σημασία να μπορούμε να μεταδώσουμε με τις μικρότερες δυνατές καθυστερήσεις και μεγαλύτερη ταχύτητα παρά με ακρίβεια και αξιοπιστία.
Για τις περιπτώσεις αυτές, έχει σχεδιαστεί ένα ακόμα πρωτόκολλο στο επίπεδο μεταφοράς, το **UDP** (User Datagram Protocol ή Πρωτόκολλο Αυτοδύναμων Πακέτων Χρήστη).
Το UDP είναι πολύ απλούστερο από το TCP. Δεν διαθέτει τεμαχισμό και για το λόγο αυτό κάθε μήνυμα που μεταδίδεται από μια εφαρμογή μέσω UDP πρέπει να χωράει εξ' ολοκλήρου σε ένα τμήμα UDP. Είναι πρωτόκολλο αυτοδύναμου πακέτου χωρίς σύνδεση: Η αποστολή ξεκινάει αμέσως χωρίς να γίνει επικοινωνία με την άλλη μεριά και δεν έχει έτσι επιπλέον καθυστερήσεις. Δεν διαθέτει έλεγχο λαθών. Δεν κάνει επαναμετάδοση δεδομένων και δεν κρατάει αντίγραφο των δεδομένων που στάλθηκαν για επιβεβαίωση. Δεν εξασφαλίζει επίσης ότι τα τμήματα θα φτάσουν στον προορισμό τους με τη σωστή σειρά. Αν μια εφαρμογή που χρησιμοποιεί UDP χρειάζεται να εξασφαλίσει ότι τα δεδομένα της δεν έχουν επηρεαστεί από τα παραπάνω προβλήματα, θα πρέπει να τα ελέγξει η ίδια. Μεταφέρεται δηλ. ο έλεγχος λαθών από το επίπεδο

μεταφοράς στο επίπεδο εφαρμογής. Όπως και με το πρωτόκολλο TCP, το UDP χρησιμοποιεί θύρες (ports), τα **UDP ports.** Η χρήση τους είναι ακριβώς ίδια με του πρωτοκόλλου TCP και προσδιορίζονται από ένα ακέραιο αριθμό 16 bits (παίρνουν δηλ. τιμές από 0 – 65535). Ο αριθμός αυτός γράφεται στην επικεφαλίδα του UDP τμήματος. Το κάθε UDP τμήμα αποτελείται από δύο βασικά κομμάτια, την επικεφαλίδα και τα δεδομένα

+	Bits 0 - 15	16 - 31
0	Source Port	Destination Port
32	Length	Checksum
64	Data	

Σχ.9 UDP επικεφαλίδα

Source port Η πόρτα του αποστολέα από την οποία προήλθε το πακέτο. Εάν ο παραλήπτης επιθυμεί να στείλει κάποια απάντηση, θα πρέπει να την στείλει στην πόρτα αυτήν. Το συγκεκριμένο πεδίο δεν είναι υποχρεωτικό και στις περιπτώσεις που δεν χρησιμοποιείται θα πρέπει να έχει την τιμή μηδέν.
Destination port Η πόρτα του παραλήπτη στην οποία θα πρέπει να παραδοθεί το πακέτο.
Length Το πεδίο αυτό έχει μέγεθος 16-bit και περιλαμβάνει το μέγεθος του πακέτου σε bytes. Το μικρότερο δυνατό μέγεθος είναι 8 bytes, αφού η κεφαλίδα αυτή καθ' αυτή καταλαμβάνει τόσο χώρο. Θεωρητικά, το μέγεθος του UDP πακέτου δεν μπορεί να ξεπερνάει τα 65,527 bytes, αλλά πρακτικά το όριο μειώνεται στα 65,507 bytes λόγω διαφόρων περιορισμών που εισάγει το πρωτόκολλο IPv4 στο επίπεδο δικτύου.
Checksum Ένα πεδίο 16-bit το οποίο χρησιμοποιείται για επαλήθευση της ορθότητας του πακέτου στο σύνολό του, δηλαδή τόσο της κεφαλίδας όσο και των δεδομένων.

1.5.3 TCP/UDP Sockets

Το στρώμα μεταφοράς αντιπροσωπεύεται στο Internet από τα πρωτόκολλα TCP και UDP. Βάση λοιπόν κάθε δικτυακής εφαρμογής είναι το πώς χειριζόμαστε τα δύο αυτά πρωτόκολλα προγραμματιστικά μέσα στα τερματικά συστήματα. Η απάντηση εγκλείεται στη έννοια των **sockets**. Ένα socket (υποδοχή) περιγράφει το ένα άκρο (endpoint) μιας επικοινωνιακής σύνδεσης. Για κάθε νέα επικοινωνιακή (TCP) σύνδεση δημιουργείται ένα socket. Η εξαφάνιση της σύνδεσης συνεπάγεται την εξαφάνιση του socket και το αντίστροφο. Όλα τα sockets τα οποία χειρίζονται μία συγκεκριμένη διαδικασία έχουν βεβαίως το δικό τους όνομα για να ξεχωρίζον προγραμματιστικά. Το καθένα είναι συνδεδεμένο στην ‘άκρη’ μίας σύνδεσης και η άκρη αυτή αναγνωρίζεται με έναν απλό αριθμό ή τον αρ. θύρας. Περιλαμβάνει την τριάδα (πρωτόκολλο, διεύθυνση, αριθμό θύρας) για το ένα άκρο της επικοινωνιακής σύνδεσης. Η επικοινωνία με sockets απαιτεί την μετάδοση μηνυμάτων μεταξύ ενός socket στη μια διεργασία και ενός socket στην άλλη διεργασία.
TCP sockets
Το TCP socket είναι υπηρεσία προσανατολισμένη στην σύνδεση. Κατά την αρχικοποίηση έχουμε την εγκαθίδρυση μιας σύνδεσης μεταξύ 2 διεργασιών. Η

σύνδεση με TCP socket απαιτεί την ανταλλαγή 3 «πακετων χειραψίας» και είναι πιο χρονοβόρα στην αρχικοποιήση της από την αντίστοιχη με UDP datagrams. Εξασφαλίζουν μια αξιόπιστη μεταφορα πληροφορίας-ότι αποστέλλεται από ένα άκρο είναι σίγουρο ότι θα φτάσει στο άλλο. Είναι ανάλογη της τηλεφ υπηρεσίας, στην οποία, μετα την εγκαθύδριση μαις σύνδεσης μεταξύ 2 ομιλητών, αυτή χρησιμοποιήται μέχρι το πέρας της συζητήσεως τους.

UDP sockets

Το UDP socket είναι μια υπηρεσία χωρίς σύνδεση. Κατα την αρχικοποίηση δεν έχουμε την εγκαθίδρυση μιας σύνδεσης μεταξύ 2 διεργασιών. Ότι αποστέλλεται από το ένα άκρο δεν είναι σίγουρο ότι θα φτάσει στο άλλο. Είναι στην ευθύνη του αποστολέα να ελένξει ότι αυτό που έστειλε, το έλαβε τελικά ο παραλήπτης και δεν χάθηκε. Το ανάλογο εδώ είναι το ταχυδρομείο. Μπορούμε να στείλουμε πολλά πακέτα στον ίδιο παραλήπτη, αλλά δεν είναι σίγουρο ότι όλα θα ακολουθήσουν την ίδια διαδρομή-συνδεση-για να φτάσουν στον προορισμό τους.

Αυτές οι διαφορές καθορίζουν και την χρήση των 2 αυτών ειδών.

1.6 Επίπεδο Δικτύου

Το επίπεδο Διαδικτύου παραλαμβάνει τα πακέτα από το επίπεδο μεταφοράς (πρωτόκολλα TCP και UDP) και παρέχει την υπηρεσία παράδοσης του πακέτου στο επίπεδο μεταφοράς του προορισμού. Επίσης εδώ εκτελούνται τα πρωτόκολλα δρομολόγησης που καθορίζουν τα δρομολόγια που θα ακολουθήσουν τα πακέτα ανάμεσα στον αποστολέα και τον παραλήπτη. Το βασικό πρωτόκολλο σε αυτό το επίπεδο είναι το IP.

1.6.1 Πρωτόκολλο IP

Στο επίπεδο δικτύου της τεχνολογίας TCP/IP, συναντάμε το πρωτόκολλο **IP,** (Internet Protocol). Η λειτουργία του IP βασίζεται αποκλειστικά στην ιδέα του **αυτοδύναμου πακέτου ή datagram,** το οποίο σημαίνει ότι τα πακέτα μεταφέρονται από την πηγή στον προορισμό χωρίς να ακολουθούν συγκεκριμένη διαδρομή (το κάθε ένα μπορεί να ακολουθήσει διαφορετική). Οι έλεγχοι για αξιόπιστη μετάδοση γίνονται από το επίπεδο μεταφοράς, εφόσον χρησιμοποιείται το πρωτόκολλο TCP. Κάθε φορά που το TCP ή το UDP πρωτόκολλο από το επίπεδο μεταφοράς θέλει να μεταδώσει κάποιο τμήμα (θυμηθείτε ότι το TCP και το UDP παράγουν segments), το παραδίδει στο πρωτόκολλο IP. Η μόνη άλλη πληροφορία που χρειάζεται το IP (και η οποία του παρέχεται από το επίπεδο μεταφοράς) είναι η *διεύθυνση του υπολογιστή προορισμού*. Αυτό είναι και το μόνο στοιχείο που ενδιαφέρει το πρωτόκολλο IP. Το IP δεν ενδιαφέρεται καθόλου για το περιεχόμενο του τμήματος ή για το πως (και αν) σχετίζεται με το προηγούμενο ή επόμενο τμήμα που λαμβάνει. Απλώς τα προωθεί στον προορισμό τους. Για να γίνει αυτό βέβαια, θα πρέπει το IP αφού παραλάβει το τμήμα από το επίπεδο μεταφοράς να προσθέσει τη δική του επικεφαλίδα με τα απαραίτητα στοιχεία, σχηματίζοντας έτσι ένα αυτοδύναμο IP πακέτο. Το μέγιστο μήκος του πακέτου αυτού έχει ορισθεί στα 64 Kbytes. Μετά το σχηματισμό του πακέτου, αποστολή του IP είναι να βρει την κατάλληλη διαδρομή που θα το οδηγήσει στον προορισμό του. Μετά τον προσδιορισμό της διαδρομής του πακέτου, γίνεται η μετάδοση του μέσω των φυσικών δικτύων (που αντιστοιχούν στο επίπεδο πρόσβασης δικτύου της τεχνολογίας TCP/IP). Ένα φυσικό δίκτυο μπορεί ωστόσο να χρησιμοποιεί διαφορετικό μέγιστο μήκος μονάδας

μεταφοράς σε σχέση με τα 64 Kbyte που χρησιμοποιεί το IP. Στην περίπτωση που το IP πακέτο "δεν χωράει να περάσει" μέσα από το συγκεκριμένο φυσικό δίκτυο το πρωτόκολλο IP έχει τη δυνατότητα να διασπάσει τα αυτοδύναμα πακέτα σε μικρότερα τμήματα που ονομάζονται **κομμάτια ή fragments.** Το IP αναλαμβάνει να επανασυνθέσει αυτά τα κομμάτια στον προορισμό τους και να σχηματίσει ξανά το αρχικό αυτοδύναμο IP πακέτο. Η διάσπαση των πακέτων σε fragments γίνεται όταν το πακέτο φτάσει στον πρώτο δρομολογητή του δικτύου. Ο δρομολογητής αντιλαμβάνεται ότι το φυσικό δίκτυο που συνδέεται σε αυτόν δεν μπορεί να μεταδώσει ολόκληρο το πακέτο που έλαβε και το διασπά σε κομμάτια. (Μη ξεχνάμε ότι ο δρομολογητής είναι μια συσκευή που λειτουργεί στο επίπεδο δικτύου και άρα αντιλαμβάνεται τις πληροφορίες της επικεφαλίδας IP). Τα κομμάτια που δημιουργούνται είναι και αυτά εντελώς αυτοδύναμα και ανεξάρτητα μεταξύ τους και μπορεί πάλι το καθένα να ακολουθήσει διαφορετική διαδρομή μέχρι τον προορισμό

Έτσι κάθε πακέτο IP, αποτελείται από μια κεφαλίδα και στη συνέχεια ακολουθούν τα δεδομένα. Στη κεφαλίδα αυτή εμπεριέχονται πληροφορίες: πρώτον, για τα δεδομένα που εμπεριέχονται στο πακέτο και δεύτερον, οι διευθύνσεις αφετηρίας και προορισμού. Η διαδικασία προσθήκης της κεφαλίδας σε ένα πακέτο δεδομένων ονομάζεται **ενθυλάκωση.**

Σχ.10 IP επικεφαλίδα

Το πεδίο **Αναγνώριση** στην επικεφαλίδα του πακέτου IP χρησιμοποιείται ώστε το IP να αναγνωρίζει σε ποιο αυτοδύναμο IP πακέτο ανήκει το fragment που λαμβάνε τη δεδομένη στιγμή. Όλα τα κομμάτια που έχουν την ίδια τιμή σε αυτό το πεδίο, ανήκουν στο ίδιο αυτοδύναμο πακέτο.
Το πεδίο **Δείκτης Εντοπισμού Τμήματος** στην επικεφαλίδα του πακέτου IP χρησιμοποιείται ώστε το IP να αναγνωρίζει σε ποια θέση πρέπει να τοποθετηθεί το συγκεκριμένο fragment που λαμβάνεται για να δημιουργηθεί ξανά το αυτοδύναμο IP πακέτο. Η τιμή του δίνεται σε blocks των 8 bytes.
Προφανώς το IP χρειάζεται και ένα τρόπο να γνωρίζει αν το πακέτο που λαμβάνει τη δεδομένη στιγμή είναι ένα κανονικό ξεχωριστό αυτοδύναμο πακέτο ή αν αποτελεί τμήμα (fragment) κάποιου πακέτου. Για το σκοπό αυτό χρησιμοποιείται το πεδίο **More Fragments (MF) ή Ένδειξη Ύπαρξης Περισσότερων Κομματιών**. Αν αυτό το πεδίο έχει την τιμή 1, σημαίνει ότι τη δεδομένη στιγμή λαμβάνουμε ένα fragment ενός μεγαλύτερου πακέτου. Αν έχει την τιμή 0 σημαίνει είτε ότι λαμβάνουμε το τελευταίο fragment ή ότι το πακέτο είναι αυτοδύναμο. Σε κάθε πακέτο που έχει κομματιαστεί, όλα τα κομμάτια έχουν MF=1 εκτός από το τελευταίο. (στην πραγματικότητα τα πεδία

που χρησιμοποιούνται με αυτό τον τρόπο – με τιμές 0 ή 1 – ονομάζονται **flags ή σημαίες**). Είναι πιθανόν ο υπολογιστής προορισμού να μην μπορεί για οποιοδήποτε λόγο να δεχθεί δεδομένα τα οποία έχουν κομματιαστεί. Αν συμβαίνει αυτό, θέτει την τιμή του πεδίου **Don't Fragment, (DF), Ένδειξης Απαγόρευσης Διάσπασης Αυτοδύναμου Πακέτου** στην τιμή 1. Στην περίπτωση αυτή θα πρέπει να βρεθεί διαδρομή μέσα από το φυσικό δίκτυο η οποία να είναι ικανή να περάσει τα αυτοδύναμα IP πακέτα χωρίς να τα κομματιάσει. Αν δεν υπάρχει αυτή η δυνατότητα, το αυτοδύναμο πακέτο απορρίπτεται.

Τα υπόλοιπα πεδία που υπάρχουν στην επικεφαλίδα είναι τα εξής:

Έκδοση: Προσδιορίζει την έκδοση του πρωτοκόλλου που χρησιμοποιείται. Για να υπάρχει επικοινωνία μεταξύ πηγής και προορισμού πρέπει οπωσδήποτε να χρησιμοποιείται η ίδια έκδοση πρωτοκόλλου

Μήκος Επικεφαλίδας: Δηλώνει το μήκος της επικεφαλίδας του πακέτου σε λέξεις των 32 bits. Η μικρότερη τιμή που μπορεί να έχει το πεδίο αυτό είναι 5. Η μικρότερη δυνατή επικεφαλίδα έχει μήκος 5*32=160 bits, και αν διαιρέσουμε με το 8, 160/8=20 bytes.

Είδος Εξυπηρέτησης: Με το πεδίο αυτό δηλώνει ο υπολογιστής το είδος της υπηρεσίας που ζητάει από το επικοινωνιακό υποδίκτυο. Τα χαρακτηριστικά που προσδιορίζουν την υπηρεσία που προσφέρει το υποδίκτυο και που χρησιμοποιούνται από το IP για να περιγράψουν τις απαιτήσεις του είναι: Η ρυθμοαπόδοση, η αξιοπιστία και η καθυστέρηση.

Συνολικό Μήκος: Δίνει το συνολικό μήκος του συγκεκριμένου IP πακέτου, στο οποίο περιλαμβάνεται τόσο η επικεφαλίδα όσο και τα δεδομένα. Έχουμε ήδη πει ότι το μέγιστο μέγεθος είναι 64 Kbytes = 64*1024 = 65536 bytes. Ξέρουμε επίσης ότι η μικρότερη δυνατή επικεφαλίδα είναι 20 bytes. Άρα το μέγιστο μέγεθος για τα δεδομένα μας είναι 65536-20=65516 bytes.

Χρόνος Ζωής: Πρόκειται για ένα μετρητή που μειώνεται κατά 1 κάθε φορά που το πακέτο διέρχεται από ένα δρομολογητή. Όταν φτάσει την τιμή μηδέν, το πακέτο απορρίπτεται (το καταστρέφει ο δρομολογητής στον οποίο βρίσκεται εκείνη τη στιγμή). Με αυτό τον τρόπο αποφεύγεται να περιφέρονται στο δίκτυο "χαμένα" πακέτα που έχουν χάσει τον προορισμό τους και κάνουν κύκλους ή απλά έχουν καθυστερήσει πάρα πολύ να φτάσουν στον προορισμό τους λόγω λανθασμένης διαδρομής ή διεύθυνσης.

Αριθμός Πρωτοκόλλου: Πρόκειται για ένα αριθμό που χαρακτηρίζει το πρωτόκολλο του επιπέδου μεταφοράς στο οποίο θα πρέπει το IP να παραδώσει το εισερχόμενο αυτοδύναμο πακέτο. Για παράδειγμα, αν αυτό το πεδίο έχει την τιμή 6, το πακέτο θα παραδοθεί στο πρωτόκολλο TCP. Η τιμή αυτή προφανώς έχει τεθεί κατά την αποστολή (από το επίπεδο μεταφοράς του αποστολέα, όταν παρέδωσε το τμήμα στο IP)

Άθροισμα Ελέγχου: Επιτρέπει στο πρωτόκολλο IP στην απέναντι πλευρά (προορισμός) να ελέγξει την ορθότητα των δεδομένων της επικεφαλίδας. Αυτό είναι σημαντικό, καθώς η επικεφαλίδα τροποποιείται κάθε φορά που περνάει από κάποιο δρομολογητή αυξάνοντας έτσι την πιθανότητα να συμβεί κάποιο σφάλμα.

Διεύθυνση Πηγής: Πρόκειται για τη διεύθυνση IP του υπολογιστή πηγής.

Διεύθυνση Προορισμού: Πρόκειται για τη διεύθυνση IP του υπολογιστή προορισμού. Η διεύθυνση αυτή διαβάζεται από τους ενδιάμεσους δρομολογητές προκειμένου να προωθήσουν το πακέτο στον προορισμό του.

IP Επιλογές: Χρησιμοποιείται για ειδικές λειτουργίες του πρωτοκόλλου.

Συμπλήρωση: Χρησιμοποιείται ώστε το μέγεθος της επικεφαλίδας να είναι πάντα πολλαπλάσιο των 32 bits. (Στην πραγματικότητα ανήκει στις "IP Επιλογές")

1.6.2 Διευθυνσιοδότηση

Η **IP διεύθυνση** προορισμού είναι αυτή που υποδεικνύει σε ένα σύστημα, που πρέπει να παραδώσει ένα IP αυτοδύναμο πακέτο. Εκτός από τη διεύθυνση, χρησιμοποιούμε συχνά και τους όρους "όνομα" και "διαδρομή" οι οποίοι σχετίζονται επίσης με τη διαδικασία παράδοσης. **Η διεύθυνση** προσδιορίζει που βρίσκεται μια συσκευή, συνήθως τη λογική ή φυσική θέση της σε ένα δίκτυο. **Το όνομα** μπορεί επίσης να προσδιορίζει μια συσκευή ή ένα δίκτυο και χρησιμοποιείται κυρίως για λόγους ευκολίας (είναι πιο εύκολο να θυμόμαστε ένα όνομα από μια σειρά αριθμών). Όταν χρησιμοποιούμε όνομα, γίνεται τελικά αντιστοίχιση του σε μια διεύθυνση με τη βοήθεια κατάλληλης υπηρεσίας που θα δούμε αργότερα (DNS). **Η διαδρομή** είναι το μονοπάτι που πρέπει να ακολουθήσει ένα αυτοδύναμο IP πακέτο για να φτάσει στον προορισμό του. Μια συνηθισμένη διαδικασία είναι να προσδιορίσουμε τον παραλήπτη χρησιμοποιώντας ένα συμβολικό όνομα, το οποίο μετατρέπεται από το σύστημα στην αντίστοιχη IP διεύθυνση. Κατόπιν καθορίζεται η διαδρομή που πρέπει να ακολουθήσει ένα αυτοδύναμο πακέτο για να φτάσει στον προορισμό του.

1.6.3 IP Διευθύνσεις

Στο Πρωτόκολλο Internet (IP), κάθε δίκτυο και κάθε υπολογιστής που είναι συνδεδεμένος στο φυσικό δίκτυο έχει μια σταθερή διεύθυνση που επιτρέπει σε κάθε άλλο υπολογιστή που είναι συνδεδεμένος να τον προσδιορίσει απόλυτα. Σε κάθε υπολογιστή αντιστοιχεί μια μοναδική διεύθυνση, που ονομάζεται **διεύθυνση IP** (**IP address**) και η οποία αποτελεί την "ταυτότητα" του στο διαδίκτυο.
Μια διεύθυνση IP αποτελείται από 4 αριθμούς χωρισμένους με τελείες. Π.χ. ένας υπολογιστής που βρίσκεται στο Πανεπιστήμιο MIT έχει διεύθυνση 18.75.0.10, ένας άλλος που βρίσκεται στο ΕΜΠ 147.102.154.12 κι ένας τρίτος που βρίσκεται στο Πανεπιστήμιο Θεσσαλίας 194.177.200.6
Στην πραγματικότητα μία IP διεύθυνση είναι ένας δυαδικός αριθμός 32-bit που για να γίνει περισσότερο κατανοητός στους ανθρώπους, χωρίζεται σε 4 ομάδες των 8 bit και κατόπιν κάθε ομάδα μεταφράζεται στον αντίστοιχο δεκαδικό αριθμό. Π.χ.:

00010010 01001011 00000000 00001010 (δυαδικός αριθμός 32-bit)
18 . 75 . 0 . 10

Μια IP διεύθυνση περιέχει δύο κομμάτια πληροφορίας. Το πρώτο είναι ο **αριθμός δικτύου** στο οποίο ανήκει ο υπολογιστής. (Θυμηθείτε: το Internet αποτελείται από πολλά διαφορετικά δίκτυα. Κάθε δίκτυο χαρακτηρίζεται από έναν μοναδικό αριθμό που αποτελεί την "ταυτότητά" του στο Internet. Το δεύτερο είναι ένας τοπικός αριθμός υπολογιστή που προσδιορίζει τον υπολογιστή μέσα στο συγκεκριμένο δίκτυο)

1.6.4 Πρωτόκολλο DHCP

Για να συνδέσουμε έναν υπολογιστή σε ένα δίκτυο, στο συγκεκριμένο υπολογιστή πρέπει να ταξινομηθεί μια μοναδική διεύθυνση. Αυτό επιτυγχάνεται μέσω του **DHCP** (Πρωτοκόλλου Δυναμικής Διαμόρφωσης Κεντρικού Υπολογιστή Δικτύου ή Dynamic Host Configuration Protocol)**.** Το συγκεκριμένο πρωτόκολλο, γνωστό και ως αυτόματη διευθυνσιοδότηση δικτύου, αντιστοιχεί αυτόματα τις ρυθμίσεις παραμέτρων του δικτύου σε έναν εξυπηρετητή (Client). Με τον τρόπο αυτό δεν χρειάζεται να ασχοληθούμε εμείς με την παραμετροποίηση που θα καταστήσουν την επικοινωνία δυνατή. Η εφαρμογή του πρωτοκόλλου DHCP σε ένα δίκτυο, λαμβάνει χώρα ως μια αρχιτεκτονική πελάτη διακομιστή (client server) στην οποία ο υπολογιστής που ζητά να του χορηγηθεί μια δυναμική IP διεύθυνση είναι ο υπολογιστής πελάτης (DHCP client) ενώ ο υπολογιστής που θα του χορηγήσει τη διεύθυνση είναι ο διακομιστής (DHCP server) της υπηρεσίας .

1.6.5 Πρωτόκολλο NAT

Γενικά δεν μπορεί να αποδοθεί αυθαίρετα ένας οποιοσδήποτε 32-bit αριθμός σε έναν υπολογιστή – υπάρχουν διάφορες **κλάσεις IP διευθύνσεων** και ο κάθε υπολογιστής μπορεί να έχει ως διεύθυνση κάποιον αριθμό της κλάσης στην οποία ανήκει μόνο. Ειδικά για τα ιδιωτικά δίκτυα, όπου ο αριθμός τους ανά τον κόσμο είναι συντριπτικά μεγάλος (ας αναλογιστούμε για παράδειγμα ότι σε κάθε απλό εργαστήριο Πληροφορικής σε ένα σχολείο υπάρχει ένα τοπικό ιδιωτικό δίκτυο), οι IP διευθύνσεις που μπορούν να έχουν οι υπολογιστές μπορούν να ανήκουν μόνο σε κάποια από τις τρεις επόμενες κλάσεις:
-10.0.0.0 - 10.255.255.255 (Class A)
-172.16.0.0 - 172.31.255.255 (Class B)
-192.168.0.0 - 192.168.255.255 (Class C)
Έτσι, μπορούν ταυτόχρονα δύο διαφορετικοί υπολογιστές που ανήκουν σε διαφορετικά δίκτυα να έχουν την ίδια IP διεύθυνση (στο τοπικό ιδιωτικό του δίκτυο ο καθένας). Από την άλλη πλευρά όμως, αυτό το χαρακτηριστικό γίνεται μειονέκτημα όταν χρειαστεί να συνδεθούν δύο τέτοια δίκτυα – στο νέο μεγαλύτερο δίκτυο που δημιουργείται, είναι πιθανό να βρεθούν δύο υπολογιστές με την ίδια IP διεύθυνση - αυτό βέβαια δεν πρέπει να επιτραπεί να συμβεί. Το παραπάνω λοιπόν πρόβλημα αντιμετωπίζεται με το πρωτόκολλο **NAT** (Network Address Translation), ένα ειδικό πρωτόκολλο που εκτελούν οι πύλες (gateways) και έχει σαν αποτέλεσμα να αλλάζει την IP διεύθυνση ενός πακέτου που ξεκινά από έναν υπολογιστή εντός του τοπικού δικτύου και προωθείται εκτός του δικτύου. Πιο συγκεκριμένα, το NAT δουλεύει ως εξής: Κάθε υπολογιστής ενός ιδιωτικού δικτύου που ζητάει να συνδεθεί με κάποιον εκτός δικτύου, κάνει αίτηση στον NAT (που υπάρχει στην πύλη (gateway ή firewall)) για να πάρει μία νέα διεύθυνση. Ο NAT διαθέτει ένα σύνολο διαθέσιμων IP διευθύνσεων (address pool) και μία από αυτές τις αναθέτει στον υπολογιστή. Ταυτόχρονα, κρατάει μία βάση δεδομένων στην οποία καταγράφει τη διεύθυνση που απέδωσε σε κάθε υπολογιστή (διαδικασία MAP). Έτσι, κάθε πακέτο που φεύγει από τον υπολογιστή του ιδιωτικού δικτύου και «ταξιδεύει» στο Internet έχει σαν διεύθυνση αποστολέα τη νέα αυτή διεύθυνση. Αντίστροφα, κάθε υπολογιστής που θέλει να στείλει δεδομένα στον συγκεκριμένο υπολογιστή του ιδιωτικού δικτύου, στέλνει πακέτα με διεύθυνση παραλήπτη τη νέα διεύθυνση. Ο NAT είναι πάλι υπεύθυνος σε

αυτήν την περίπτωση για να παραλάβει ο υπολογιστής τα πακέτα που προορίζονται για αυτόν. Δηλαδή, ο NAT κοιτάει τη βάση δεδομένων και βλέπει ποια είναι η πραγματική IP διεύθυνση του υπολογιστή (δηλαδή η διεύθυνση που έχει στο ιδιωτικό του δίκτυο) και, με βάση αυτήν την πληροφορία, δρομολογεί τα εισερχόμενα πακέτα.

1.6.6 Δρομολόγηση

Ο αλγόριθμος δρομολόγησης ανήκει στο επίπεδο δικτύου και σκοπός του είναι να κατευθύνει ένα πακέτο από την πηγή στον προορισμό του. Ο όρος "δρομολόγηση" αναφέρεται στη διαδικασία εύρεσης της διαδρομής που πρέπει να ακολουθήσει ένα πακέτο για να φτάσει στον προορισμό του. Η διαδικασία αυτή δεν είναι πάντα εύκολη, τη στιγμή που γνωρίζουμε ότι ένα σύνθετο δίκτυο (όπως το Internet) μπορεί να διαθέτει πολλές εναλλακτικές διαδρομές που να οδηγούν το πακέτο στον ίδιο προορισμό.
Δρομολογητής (**router**) είναι μια ηλεκτρονική συσκευή η οποία αναλαμβάνει την αποστολή και λήψη πακέτων δεδομένων μεταξύ ενός ή περισσοτέρων διακομιστών, άλλων δρομολογητών και πελατών, κατά μήκος πολλαπλών δικτύων (δρομολόγηση). Η δρομολόγηση, κεντρική λειτουργία του επιπέδου δικτύου, γίνεται με βάση διάφορα κριτήρια και τελικώς επιλέγεται μία ανάμεσα σε διάφορες πιθανές διαδρομές. Οι δρομολογητές ανήκουν στο επίπεδο 3 , το επίπεδο δικτύου (Network Layer), του μοντέλου OSI.
Βασικό ρόλο στη διαδικασία δρομολόγησης έχει ο **πίνακας δρομολόγησης**. Το πρωτόκολλο IP χρησιμοποιεί αυτό τον πίνακα για να πάρει όλες τις αποφάσεις που έχουν να κάνουν με την δρομολόγηση πακέτων στον προορισμό τους. Η δρομολόγηση συνήθως βασίζεται στην διεύθυνση του δικτύου προορισμού. Κάθε υπολογιστής διαθέτει ένα πίνακα με διευθύνσεις δικτύων, για καθένα από τα οποία αντιστοιχεί ένας δρομολογητής. Όταν δημιουργείται ένα αυτοδύναμο πακέτο προς κάποιο δίκτυο, ο υπολογιστής συμβουλεύεται αυτό τον πίνακα για να τα στείλει στον αντίστοιχο δρομολογητή ο οποίος και θα τα προωθήσει τελικά στο δίκτυο προορισμού. Ο δρομολογητής θα αναλάβει να στείλει το πακέτο σε άλλο δρομολογητή κ.ο.κ. μέχρι να φτάσει σε ένα δρομολογητή ο οποίος να είναι συνδεδεμένος απευθείας με το δίκτυο προορισμού.
Ο αλγόριθμος δρομολόγησης που χρησιμοποιείται από το πρωτόκολλο IP για τη δρομολόγηση αυτοδύναμων πακέτων διακρίνει δύο περιπτώσεις:
• **Άμεση Δρομολόγηση (direct routing):** Στην περίπτωση αυτή ο υπολογιστής προορισμού βρίσκεται στο ίδιο δίκτυο με τον υπολογιστή αποστολής. Το πακέτο μπορεί να σταλεί απευθείας χωρίς άλλα βήματα, και άρα δεν γίνεται καμιά προώθηση του πακέτου. Πρόκειται για την απλούστερη μορφή δρομολόγησης.
• **Έμμεση Δρομολόγηση (indirect routing):** Στην περίπτωση αυτή ο υπολογιστής προορισμού βρίσκεται σε διαφορετικό δίκτυο από τον υπολογιστή αποστολής. Θα πρέπει το πακέτο να δρομολογηθεί μέσω των κατάλληλων δρομολογητών για να φτάσει στον προορισμό του. Προφανώς για το σκοπό αυτό θα χρησιμοποιηθούν οι πίνακες δρομολόγησης που αναφέραμε προηγουμένως.

1.6.7 Πρωτόκολλο ICMP

Το πρωτόκολλο Internet Control Message Protocol (**ICMP**) είναι ένα από τα βασικά πρωτόκολλα του διαδικτύου. Χρησιμοποιείται κυρίως από τα λειτουργικά συστήματα των ηλεκτρονικών υπολογιστών ενός δικτύου για την ανταλλαγή μηνυμάτων λάθους, όπως για παράδειγμα την έλλειψη κάποιας υπηρεσίας από έναν server ή την απουσία ενός υπολογιστή από το δίκτυο.
Το πρωτόκολλο ICMP διαφέρει από τα πρωτόκολλα TCP και UDP διότι συνήθως δεν χρησιμοποιείται από τις εφαρμογές που εκτελούνται σε κάποιον υπολογιστή, αλλά από το λειτουργικό του σύστημα. Εξαίρεση σε αυτό τον κανόνα αποτελεί το εργαλείο **ping,** το οποίο στέλνει μηνύματα ICMP Echo Request σε κάποιον υπολογιστή του δικτύου για να διαπιστώσει εάν ο υπολογιστής αυτός υπάρχει ή όχι και επίσης πόσο χρόνο χρειάζεται το μήνυμα να φτάσει σε αυτόν. Εάν ο υπολογιστής αυτός υπάρχει, θα απαντήσει με μηνύματα Echo Response.
Τα μηνύματα ICMP κατασκευάζονται στο επίπεδο δικτύου και αποτελούν κανονικά πακέτα IP. Όπως και το πρωτόκολλο UDP, το ICMP δεν εγγυάται ότι το πακέτο θα φτάσει αξιόπιστα στον προορισμό του. Μερικές από τις πιο συνηθισμένες δικτυακές εφαρμογές χρησιμοποιούν πακέτα ICMP, όπως για παράδειγμα η εντολή **traceroute**. Η εντολή αυτή χρησιμοποιείται για την εύρεση όλων των κόμβων ενός δικτύου από τους οποίους πρέπει να περάσει ένα πακέτο για να φτάσει στον τελικό προορισμό του. Αυτό που κάνει ουσιαστικά είναι να στέλνει πακέτα UDP με συγκεκριμένο χρόνο ζωής (TTL - Time To Live) και να περιμένει πακέτα ICMP που να περιέχουν μήνυμα σφάλματος "ο χρόνος ζωής τελείωσε" (Time To Live exceeded in transit) ή "ο προορισμός δεν βρέθηκε" (Destination unreachable). Στο σημείο αυτό αξίζει να αναφερθεί ότι ο χρόνος ζωής (TTL - Time To Live) ενός πακέτου είναι ο μέγιστος αριθμός των κόμβων του δικτύου από τους οποίους θα πρέπει να περάσει έως ότου φτάσει στον προορισμό του. Εάν ένα πακέτο κατά την πορεία του στο δίκτυο περάσει από περισσότερους κόμβους απ' ότι αναγράφεται στο πεδίο TTL, τότε το πακέτο αυτομάτως απορρίπτεται και ο υπολογιστής ο οποίος διαπίστωσε το σφάλμα στέλνει ένα ICMP μήνυμα σφάλματος στον υπολογιστή που δημιούργησε το πακέτο. Τέλος, η εντολή ping χρησιμοποιεί επίσης το πρωτόκολλο ICMP για την λειτουργία της και συγκεκριμένα τα ICMP μηνύματα "Echo request" και "Echo reply".

1.7 Επίπεδο Ζεύξης Δεδομένων

Ο ρόλος του επιπέδου ζεύξης δεδομένων είναι είναι η διακίνηση πακέτων του επιπέδου δικτύου μεταξύ δυο οντοτήτων. Το φυσικό επίπεδο, που αποτελείται από τα φυσικά στοιχεία του δικτύου (π.χ. hubs, καλώδια δικτύου, οπτικές ίνες, ομοαξονικά καλώδια, κάρτες δικτύων) και τις προδιαγραφές χαμηλού επιπέδου των σημάτων (τάση, συχνότητα, κλπ.), θεωρείται συχνά ως μέρος του επιπέδου ζεύξης δεδομένων. Εδώ συναντάμε τα πρωτόκολλα Ethernet, MAC,ARP, WIFI

1.7.1 Πρωτόκολλο Ethernet

Το **Ethernet** είναι το συνηθέστερο χρησιμοποιούμενο πρωτόκολλο ενσύρματης τοπικής δικτύωσης υπολογιστών. Αναπτύχθηκε από την εταιρεία Xerox κατά τη δεκαετία του '70 και το 1985 το Ethernet έγινε αποδεκτό επίσημα από τον οργανισμό IEEE ως το πρότυπο 802.3 για ενσύρματα τοπικά δίκτυα (LAN).

Πρακτικά, το Ethernet χρησιμοποιεί τη μέθοδο μετάδοσης δεδομένων σε μορφή πακέτων (packet switching) μέγιστου μεγέθους (Maximum Transmission Unit, MTU) 1500 bytes και ελάχιστου 46 bytes. Για το σκοπό αυτό, δεδομένα με μήκος μεγαλύτερο των 1500 bytes κατατέμνονται σε πακέτα των 46-1500 bytes (το λεγόμενο payload) τα οποία αποστέλλονται διαδοχικά στη γραμμή επικοινωνίας. Αν το payload έχει μήκος μικρότερο των 46 bytes, προστίθενται επιπλέον κενά bytes ώστε αυτό να αποκτήσει το επιθυμητό ελάχιστο μήκος. Επιπλέον του payload, προστίθενται πληροφορίες όπως ο σειριακός αριθμός της κάρτας Ethernet, οι φυσικές διευθύνσεις (MAC addresses) αποστολέα και παραλήπτη, το μήκος του payload, καθώς και δεδομένα για έλεγχο σφαλμάτων κατά τη μετάδοση.

1.7.2 Διεύθυνση MAC

Κάθε συσκευή που επικοινωνεί σε ένα δίκτυο διαθέτει δύο διευθύνσεις: Η μία είναι η διεύθυνση IP η οποία αποδίδεται από το πρωτόκολλο δικτύου και η άλλη είναι η φυσική διεύθυνση γνωστή και ως διεύθυνση υλικού (hardware address).
Οι φυσικές διευθύνσεις είναι μοναδικές, γιατί διαφορετικά δεν θα μπορούσαμε να προσδιορίσουμε τις συσκευές στο δίκτυο. Για παράδειγμα μια κάρτα δικτύου (που είναι από τις πλέον κοινές συσκευές σε ένα δίκτυο) διαθέτει μια φυσική διεύθυνση που της έχει αποδοθεί από τον κατασκευαστή της στο στάδιο της κατασκευής της. Σύμφωνα με το μοντέλο OSI, η φυσική διεύθυνση βρίσκεται στο υποεπίπεδο πρόσβασης στο μέσο γνωστό και ως Media Access Control, για το λόγο αυτό ονομάζεται και **διεύθυνση MAC**.
Μια **διεύθυνση MAC** (Media Access Control - έλεγχος πρόσβασης σε μέσα) είναι ένας δεκαεξαδικός σειριακός αριθμός (ως προς την αναπαράσταση) ο οποίος είναι μοναδικός για κάθε δικτυακή συσκευή. Ο αριθμός έχει τη μορφή `xx:xx:xx:xx:xx:xx`, για παράδειγμα `0A:12:A1:B2:AE:04` για την 16-δική αναπαράσταση.

1.7.3 Πρωτόκολλο ARP

Έχουμε πλέον μιλήσει και για τα δύο είδη διευθύνσεων: Τη **φυσική (MAC) διεύθυνση** που δίνει ο κατασκευαστής του δικτυακού υλικού στις συσκευές του (π.χ. στις κάρτες δικτύου). Τη **διεύθυνση IP** που ανήκει στην τεχνολογία TCP/IP και αποδίδεται στις συσκευές του δικτύου από τον διαχειριστή του δικτύου.
Η μετατροπή μιας IP διεύθυνσης στην αντίστοιχη φυσική σε ένα περιβάλλον τοπικού δικτύου επιτυγχάνεται με το πρωτόκολλο **ARP** (Address Resolution Protocol, Πρωτόκολλο Μετατροπής Διεύθυνσης) που ανήκει στα πρωτόκολλα του χαμηλότερου επιπέδου (Σύνδεσης Δικτύου) στην ιεραρχία του TCP/IP. Αυτό θα μπορούσε και να επιτευχθεί αποθηκεύοντας την αντιστοίχιση αυτή σε ένα πίνακα δύο στηλών (IP Διεύθυνση, Φυσική Διεύθυνση). Ωστόσο αν κάνουμε αυτή τη διαδικασία χειροκίνητα, σύντομα θα έχουμε μεγάλο πρόβλημα ειδικά για μεγάλα δίκτυα. Το πρωτόκολλο ARP κάνει ακριβώς αυτή την αντιστοίχιση από IP διεύθυνση στην φυσική και διατηρεί τον πίνακα *δυναμικά*. Αυτό σημαίνει ότι αναλαμβάνει να ανακαλύψει αρχικά ποια διεύθυνση IP αντιστοιχεί σε ποια διεύθυνση MAC αλλά και να ενημερώνει αυτές τις καταχωρίσεις όταν προστίθενται νέα μηχανήματα (ή όταν γίνεται αλλαγή διευθύνσεων σε υπάρχοντα). Κεντρικό στοιχείο στη λειτουργία του πρωτοκόλλου ARP είναι ο

πίνακας δύο στηλών (IP Διεύθυνση, Φυσική Διεύθυνση). Κάθε γραμμή του πίνακα αντιστοιχεί σε μια εγγραφή δηλ. σε μια συσκευή.

IP Διεύθυνση	Ethernet Διεύθυνση
223.1.2.1	08-00-39-00-2F-C3
223.1.2.3	08-00-5A-21-A7-22
223.1.2.4	08-00-10-99-AC-54

Όταν το πρωτόκολλο ARP λάβει μια διεύθυνση IP, αρχικά θα διερευνήσει τον πίνακα ARP για να δει αν υπάρχει η αντίστοιχη εγγραφή:
• Αν βρεθεί η εγγραφή στον πίνακα ARP, το πρωτόκολλο θα επιστρέψει την αντίστοιχη φυσική διεύθυνση που αναφέρει ο πίνακας.
• Αν δεν βρεθεί εγγραφή, το πρωτόκολλο θα δημιουργήσει μια **αίτηση ARP**. Η αίτηση αυτή είναι ένα μήνυμα το οποίο απευθύνεται σε όλα τα μηχανήματα του τοπικού δικτύου. Περιέχει την διεύθυνση IP του υπολογιστή προορισμού. Αν μια συσκευή στο δίκτυο αναγνωρίσει αυτή την IP ως δική της, θα στείλει τη φυσική της διεύθυνση ως απάντηση στη συσκευή που δημιούργησε την αίτηση. Αμέσως μετά τη λήψη της απάντησης, η συσκευή που δημιούργησε την αίτηση θα ενημερώσει τον πίνακα ARP, δημιουργώντας μια νέα εγγραφή με τη διεύθυνση IP και τη φυσική διεύθυνση της συσκευής που μόλις έλαβε. Όταν χρειαστεί ξανά να βρει τη φυσική διεύθυνση αυτής της συσκευής, θα διαβάσει απλώς τον πίνακα και δεν θα χρειαστεί να δημιουργηθεί νέο αίτημα ARP.
Όταν μια ARP εγγραφή δεν έχει χρησιμοποιηθεί για μεγάλο χρονικό διάστημα, στα περισσότερα συστήματα διαγράφεται αυτόματα.

1.7.4 WIFI

Το IEEE 802.11 είναι μια οικογένεια προτύπων της IEEE για ασύρματα τοπικά δίκτυα (WLAN) που είχαν ως σκοπό να επεκτείνουν το 802.3 (Ethernet, το συνηθέστερο πρωτόκολλο ενσύρματης δικτύωσης υπολογιστών) στην ασύρματη περιοχή. Τα πρότυπα 802.11 είναι ευρύτερα γνωστά ως «WiFi» επειδή η WiFi Alliance, ένας οργανισμός ανεξάρτητος της IEEE, παρέχει την πιστοποίηση για τα προϊόντα που υπακούν στις προδιαγραφές του 802.11. Αυτή η οικογένεια πρωτοκόλλων αποτελεί το καθιερωμένο πρότυπο της βιομηχανίας στο χώρο των ασύρματων τοπικών δικτύων.
Ο όρος WiFi (Wireless Fidelity) χρησιμοποιείται για να προσδιορίσει τις συσκευές που βασίζονται στην προδιαγραφή IEEE 802.11 b/g/n και εκπέμπουν σε συχνότητες 2.4GHz. Ωστόσο το WiFi (ασύρματη πιστότητα) έχει επικρατήσει και ως όρος αναφερόμενος συνολικά στα ασύρματα τοπικά δίκτυα. Συνήθεις εφαρμογές του είναι η παροχή ασύρματων δυνατοτήτων πρόσβασης στο Internet, τηλεφωνίας μέσω διαδικτύου (VoIP) και διασύνδεσης μεταξύ ηλεκτρονικών συσκευών όπως τηλεοράσεις, ψηφιακές κάμερες, DVD Player και ηλεκτρονικοί υπολογιστές. Σε φορητές ηλεκτρονικές συσκευές το 802.11 βρίσκει εφαρμογές ασύρματης μετάδοσης, όπως π.χ. στη μεταφορά φωτογραφιών από ψηφιακές κάμερες σε υπολογιστές για περαιτέρω επεξεργασία και εκτύπωση, αν και σε αυτόν τον τομέα έχει υποσκελιστεί από το πρωτόκολλο Bluetooth για τα πολύ μικρότερης εμβέλειας ασύρματα προσωπικά δίκτυα.

Κεφάλαιο 2
Εισαγωγή στο wireshark

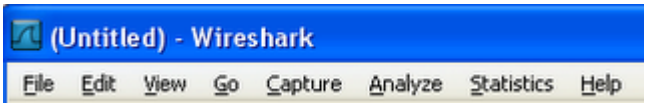
Το Wireshark είναι ένα ελεύθερο και ανοιχτού κώδικα λογισμικό. Χρησιμοποιείται για ανάλυση δικτύου, παρακολούθηση δικτύου, εντοπισμό και αντιμετώπιση προβλημάτων στα δίκτυα και για εκπαίδευση. Ουσιαστικά παρακολουθεί την κίνηση των πακέτων. Παρέχει πληροφορίες για το δίκτυο και τα πρωτόκολλα ανώτερου επιπέδου σχετικά με τα δεδομένα που διακινούνται σ' αυτό. Το Wireshark χρησιμοποιεί τη δικτυακή βιβλιοθήκη libpcap για την σύλληψη (ανάλυση) των πακέτων. Προσφέρει τη δυνατότητα χρήσης φίλτρων για διαφορετικούς τύπους πακέτων που θέλουμε να ανιχνεύσουμε καθώς επίσης και στατιστική ανάλυση και ανάλυση με γράφους. Το Wireshark είναι εργαλείο ανίχνευσης. Σε καμιά περίπτωση δεν μπορεί ο χρήστης να στείλει πακέτα με αυτό ούτε μπορεί να χρησιμοποιηθεί ως προειδοποίηση για κάποιο εισβολέα του δικτύου. Το Wireshark ονομαζόταν Ethereal μέχρι το 2006, όταν ο επικεφαλής προγραμματιστής, αποφάσισε την αλλαγή του ονόματός του λόγω δικαιωμάτων χρήσης που προϋπήρχαν για το όνομα Ethereal.

Τα πακέτα δεδομένων μπορούν να συλληφθούν τόσο από ενσύρματο όσο και ασύρματο δίκτυο και αυτές οι πληροφορίες μπορούν να προβληθούν ζωντανά καθώς συλλαμβάνονται ή ενώ αναλύονται εν ευθέτω χρόνω.
Επίσης υποστήριζει πολλά plugins που σημαίνει ότι το εργαλείο μπορεί να επεκταθεί ώστε να προστεθούν νέα πρωτόκολλα και χαρακτηριστικά.
Το Wireshark είναι διαθέσιμο για όλα τα κύρια λειτουργικά συστήματα όπως τα Windows, Linux και Mac, κάτι που το καθιστά ιδανικό για δίκτυα με διαφορετικές πλατφόρμες.

Η **δύναμη του Wireshark** πηγάζει από:
-την ευκολία εγκατάστασής του.
-την απλότητα της χρήσης του μέσω του γραφικού περιβάλλοντος (GUI) που το καθιστά ικανό να αναλύσει όλη την κυκλοφορία ενός δικτύου που χρησιμοποιεί ποικίλα πρωτόκολλα.
- το μεγάλο εύρος της λειτουργικότητας του.

Ακολουθούν κάποια screenshots για την κατανόηση των χαρακτηριστικών και του περιβάλλοντος του Wireshark

ΜΕΝΟΥ



Σχ. 11 Το μενου του Wireshark

Τα οχτώ μενού στην κορυφή της πλατφόρμας χρησιμεύουν στη ρύθμιση του Wireshark:

- "File" Ανοίγει ή αποθηκεύει μία σύλληψη(ανάλυση).
- "Edit" Βρίσκει ή σημειώνει πακέτα. Ρυθμίζει τις γενικές προτιμήσεις.
- "View" Ρυθμίζει την προβολή της πλατφόρμας του Wireshark.

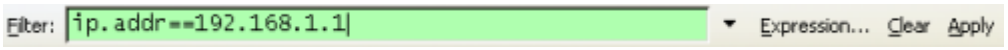
- "Go" Πηγαίνει σε δεδομένα εντός της σύλληψης.
- "Capture" Ορίζει τις επιλογές των φίλτρων της σύλληψης και εκκινεί τη σύλληψη.
- "Analyze" Ορίζει τις επιλογές Ανάλυσης.
- "Statistics" Απεικονίζει στατιστικά για το Wireshark.
- "Help" Βρίσκει διαθέσιμη υποστήριξη μέσω διαδικτύου ή τοπικά.

ΣΥΝΤΟΜΕΥΣΕΙΣ

Σχ.12 Συντομεύσεις στο Wireshark

Χρήσιμες συντομεύσεις είναι διαθέσιμες κάτω ακριβώς από τα μενού. Μπορείτε να βλέπετε πληροφορίες για τις συντομεύσεις καθώς μετακινείτε τον κέρσορα του ποντικιού σας πάνω από τα εικονίδια.

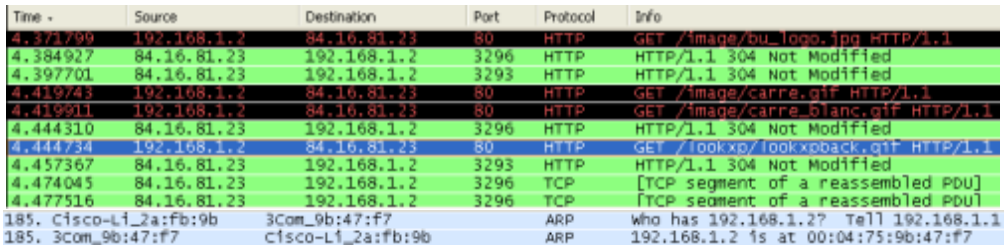
ΠΡΟΒΟΛΗ ΦΙΛΤΡΟΥ



Σχ.13 Φίλτρα στο Wireshark

Η προβολή Φίλτρου χρησιμοποιείται για την αναζήτηση μέσα στα καταγεγραμμένα αρχεία συλλήψεων.

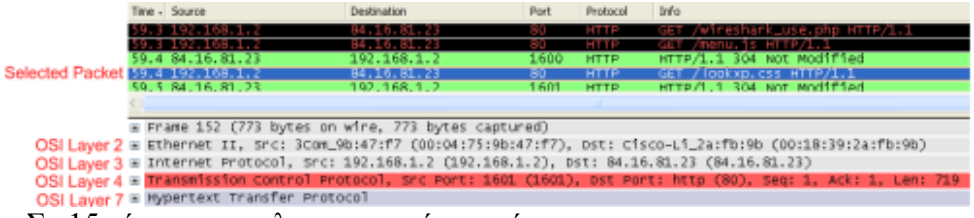
ΠΙΝΑΚΑΣ ΛΙΣΤΑΣ ΠΑΚΕΤΩΝ



Σχ.14 Πίνακας της λίστας πακέτων

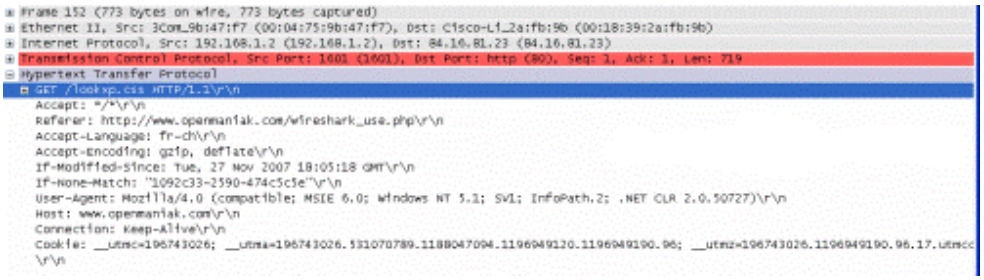
Ο πίνακας της λίστας πακέτων απεικονίζει όλα τα συλλαμβανόμενα πακέτα. Μπορείτε να πάρετε πληροφορίες όπως τις διευθύνσεις MAC/IP προορισμού ή εκκίνησης, τους αριθμούς των θυρών TCP/UDP, το πρωτόκολλο ή τα περιεχόμενα του πακέτου. Μπορείτε να προσθαφαιρέσετε στήλες ή να αλλάξετε κάποια από τα χρώματα του πίνακα ως ακολούθως:
Edit menu (Edit μενού) > Preferences (Προτιμήσεις)

ΠΙΝΑΚΑΣ ΛΕΠΤΟΜΕΡΕΙΩΝ ΠΑΚΕΤΩΝ



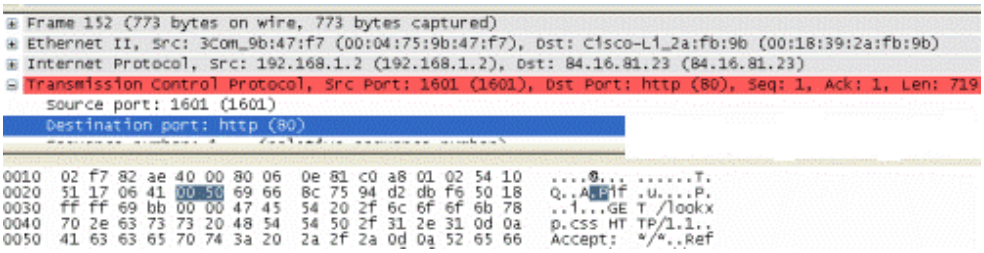
Σχ.15 πίνακας των λεπτομερειών πακέτων

Ο πίνακας των λεπτομερειών πακέτων δίνει εις βάθους πληροφορίες σχετικά με το επιλεγμένο πακέτο στον πίνακα με τη λίστα των πακέτων. Οι πληροφορίες προβάλλονται ανά πρωτόκολλο και μπορούν να επεκταθούν ή να συμπτυχθούν. Στη φωτογραφία παρακάτω, οι πληροφορίες του πρωτοκόλλου HTTP εκτεταμένες.



Σχ.16 οι πληροφορίες του πρωτοκόλλου HTTP εκτεταμένες

ΠΙΝΑΚΑΣ ΑΝΑΤΟΜΙΑΣ



Σχ.17 πίνακας ανατομίας ή πίνακας των bytes πακέτων

Ο πίνακας ανατομίας αποκαλείται επίσης και ως «πίνακας των bytes πακέτων» από το Wireshark, απεικονίζει τις ίδιες πληροφορίες όπως εκείνες που εμφανίζονται στον πίνακα των λεπτομερειών πακέτου αλλά σε δεκαεξαδική μορφή. Στο παραπάνω παράδειγμα, επιλέξαμε τον αριθμό της θύρας TCP (80) στον πίνακα των λεπτομερειών του πακέτου και η αντίστοιχη δεκαεξαδική του μορφή εμφανίζεται αυτομάτως στον πίνακα ανατομίας (0050).

ΔΙΑΦΟΡΑ

Σχ.18 διάφορες πληροφορίες

Στο τέλος της πλατφόρμας, μπορείτε να βρείτε τις ακόλουθες πληροφορίες:
- την κάρτα δικτύου η οποία χρησιμοποιείται για τη σύλληψη.
- εάν η σύλληψη εκτελείται ή είναι σταματημένη.
- που αποθηκεύεται η σύλληψη στο σκληρό δίσκο.
- το μέγεθος της σύλληψης.
- τον αριθμό των πακέτων που συλλαμβάνονται. (P)
- τον αριθμό των εικονιζόμενων πακέτων. (D) (Πακέτα τα οποία συμφωνούν με το φίλτρο προβολής) - τον αριθμό των επιλεγμένων πακέτων. (M)

2.1 Φίλτρα Στο WIRESHARK

Ένα κοινό πρόβλημα κατά την εκκίνηση του Wireshark με τις προεπιλεγμένες ρυθμίσεις είναι ότι λαμβάνεται μεγάλη ποσότητα πληροφοριών στην οθόνη με αποτέλεσμα να μη μπορείτε να βρείτε την πληροφορία που αναζητάτε. Γι' αυτό το λόγο τα φίλτρα είναι τόσο σημαντικά, θα μας βοηθήσουν να αναζητήσουμε, στα χρήσιμα αρχεία καταγραφών, τα δεδομένα που μας ενδιαφέρουν.

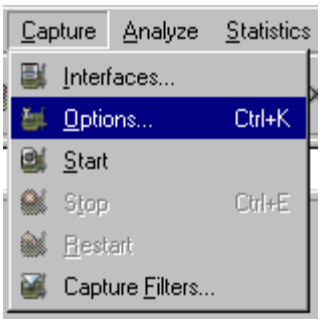
Φίλτρα σύλληψης: Χρησιμοποιούνται για την επιλογή των δεδομένων που θα καταγραφούν στα αρχεία καταγραφής. Καθορίζονται πριν την εκκίνηση της σύλληψης.
Φίλτρα απεικόνισης: Χρησιμοποιούνται για την αναζήτηση μέσα στα αρχεία καταγραφών. Μπορούν να τροποποιηθούν ενόσω τα δεδομένα συλλαμβάνονται.

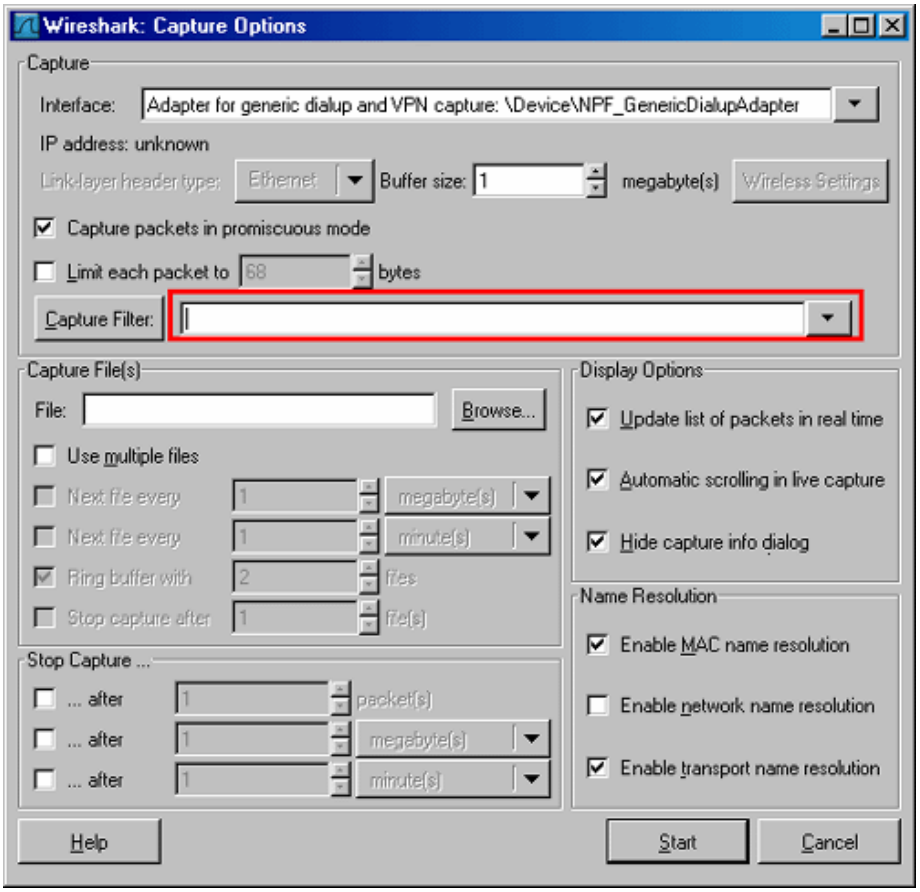
Οι στόχοι των δυο φίλτρων είναι διαφορετικοί.
Τα φίλτρα σύλληψης χρησιμοποιούνται ως μιας πρώτης μορφής φιλτράρισμα για τον περιορισμό του μεγέθους των συλληφθέντων δεδομένων με σκοπό την αποφυγή δημιουργίας μεγάλων αρχείων καταγραφής.
Τα φίλτρα απεικόνισης είναι περισσότερο δυνατά (και σύνθετα) και μας επιτρέπουν να αναζητήσουμε τα ακριβή δεδομένα που επιθυμούμε

2.1.1 Φίλτρα Σύλληψης

Η σύνταξη του φίλτρου σύλληψης είναι η ίδια που χρησιμοποιείται και στα προγράμματα τη βιβλιοθήκη Lipcap (Linux) ή τη Winpcap (Windows) όπως το δημοφιλές TCPdump. Το φίλτρο σύλληψης θα πρέπει να οριστεί πριν την έναρξη της σύλληψης με Wirshark, ενώ τα φίλτρα απεικόνισης τα οποία μπορούν να τροποποιηθούν οποιαδήποτε στιγμή κατά τη διάρκεια της σύλληψης.

Τα βήματα για τη ρύθμιση ενός φίλτρου σύλληψης είναι τα επόμενα:
- επιλέγουμε capture (Σύλληψη) > options (Επιλογές).
-δίνουμε το όνομα που θέλουμε στο πεδίο "capture filter" ή πατάμε στο κουμπί "capture filter" για να δώσουμε ένα όνομα στο φίλτρο μας και να το χρησιμοποιήσουμε και για τις ακόλουθες συλλήψεις μας.
- Πατάμε στο Start για τη σύλληψη των δεδομένων.





Σχ.19 Βήματα για τη ρύθμιση ενός φίλτρου σύλληψης

Σύνταξη:	**Πρωτόκολλο**	**Κατεύθυνση**	**Host(s)**	**Τιμή**	**Λογικές Πράξεις**	**Άλλες εκφράσεις**
Παράδειγμα:	tcp	dst	10.1.1.1	80	and	tcp dst 10.2.2.2 3128

Πρωτόκολλο::
Τιμές: ether, fddi, ip, arp, rarp, decnet, lat, sca, moprc, mopdl, tcp and udp.
Εάν δεν δηλωθεί κανένα πρωτόκολλο, χρησιμοποιούνται όλα τα πρωτόκολλα.
Κατεύθυνση::
Τιμές: src, dst, src and dst, src or dst
Εάν δεν δηλωθεί ούτε πηγή ούτε προορισμός, οι λέξεις κλειδιά "src ή dst" εφαρμόζονται.
Για παράδειγμα, το "host 10.2.2.2" είναι ίσο με το "src ή dst host 10.2.2.2".
Host(s):
Τιμές: net, port, host, portrange.
Εάν δεν δηλωθεί host(s), χρησιμοποιείται η λέξη κλειδί "host".
Για παράδειγμα, το "src 10.1.1.1" είναι ίσο με το "src host 10.1.1.1".
Λογικές Πράξεις:
Τιμές: not, and, or.
Η άρνηση ("not") έχει μεγαλύτερη προτεραιότητα. Η εναλλαγή ("or") και η αλληλουχία ("and") έχουν ίση προτεραιότητα και συνδέουν τα αρίστερα με τα δεξιά.
Για παράδειγμα,
το "not tcp port 3128 and tcp port 23" είναι ίσο με το "(not tcp port 3128) and tcp port 23".

το "not tcp port 3128 and tcp port 23" ΔΕΝ είναι ίσο με το "not (tcp port 3128 and tcp port 23)".
Παραδείγματα:

tcp dst port 3128

Δείχνει τα πακέτα με προορισμό την θύρα TCP 3128.

ip src host 10.1.1.1

Δείχνει τα πακέτα με τη διεύθυνση πηγής IP που ισούται με το 10.1.1.1.

host 10.1.2.3

Δείχνει τα πακέτα με πηγή ή προορισμό τη διεύθυνση IP η οποία είναι ίση με το 10.1.2.3.

src portrange 2000-2500

Δείχνει τα πακέτα με θύρες πηγής UDP ή TCP μεταξύ του εύρους 2000-2500.

not imcp

Δείχνει τα πάντα εκτός από τα πακέτα του icmp . (το icmp χρησιμοποιείται τυπικά από το εργαλείο ping)

src host 10.7.2.12 and not dst net 10.200.0.0/16

Δείχνει τα πακέτα με τη διεύθυνση IP πηγής η οποία ισούται με το 10.7.2.12 και την ίδια στιγμή όχι με τη διεύθυνση IP προορισμού του δικτύου 10.200.0.0/16.

(src host 10.4.1.12 or src net 10.6.0.0/16) and tcp dst portrange 200-10000 and dst net 10.0.0.0/8

Δείχνει τα πακέτα με τη διεύθυνση IP πηγής 10.4.1.12 ή το δίκτυο πηγής 10.6.0.0/16, το αποτέλεσμα είναι η αλυσιδωτή σύνδεση των πακέτων τα οποία έχουν προορισμό το εύρος θυρών TCP από 200 έως 10000 και διεύθυνση IP του δικτύου προορισμού 10.0.0.0/8.

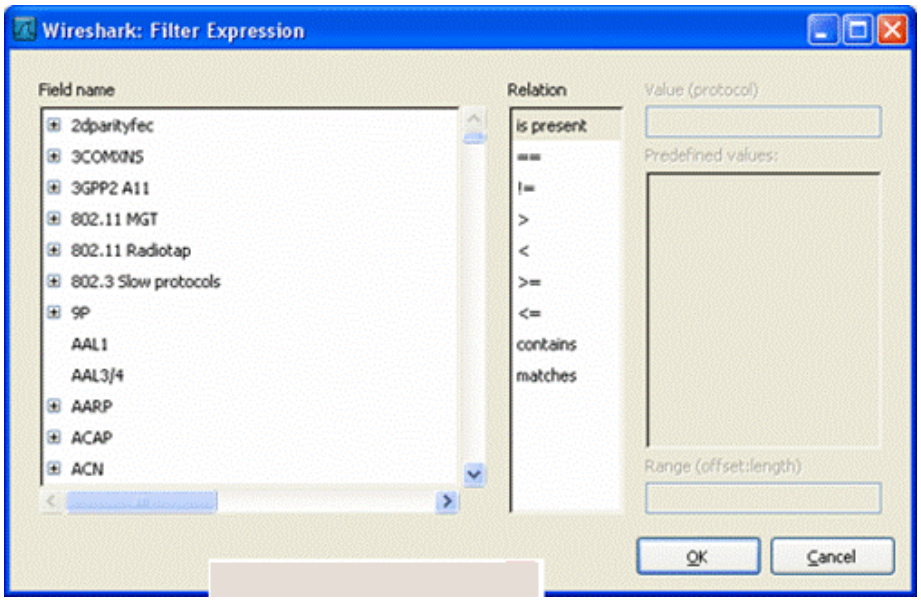
Σημειώσεις:
Η αριστερή πλάγιος "\" χρησιμοποιείται όταν μια λέξη κλειδί χρησιμοποιείται ως τιμή.
Το "ether proto \ip" (ισούται με το "ip"). Αυτό έχει ως στόχο όλα τα πρωτόκολλα IP.
Το "ip proto \icmp" (ισούται με το "icmp"). Αυτό έχει ως στόχο τα icmp πακέτα τα οποία τυπικά χρησιμοποιούνται από το εργαλείο ping.
Οι λέξεις κλειδιά "multicast" και "broadcast" μπορούν επίσης να χρησιμοποιηθούν μετά τις λέξεις "ip" ή "ether". Η λέξη κλειδί "no broadcast" είναι χρήσιμη όταν θέλουμενα αποκλείσετε τις αιτήσεις μετάδοσης (broadcast requests).

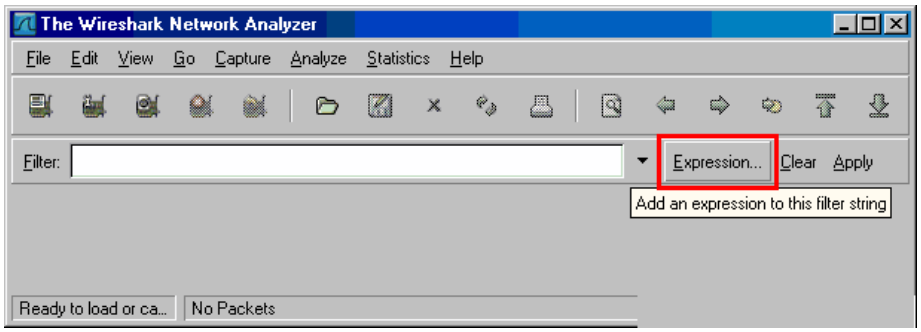
2.1.2. Φίλτρα Απεικόνισης

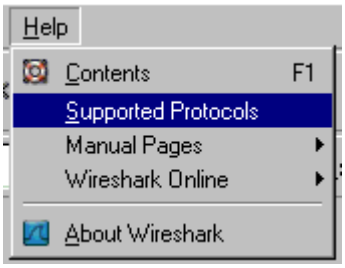
Τα φίλτρα απεικόνισης χρησιμοποιούνται για την αναζήτηση μέσα σε δεδομένα τα οποία έχουν συλληφθεί με κάποιο φίλτρο σύλληψης. Οι δυνατότητες αναζήτησης μπορούν να είναι μεγαλύτερες από εκείνες ενός φίλτρου σύλληψης και δεν είναι απαραίτητο να γίνει η επανεκκίνηση της σύλληψης όταν επιθυμήσουμε την αλλαγή του φίλτρου μας.

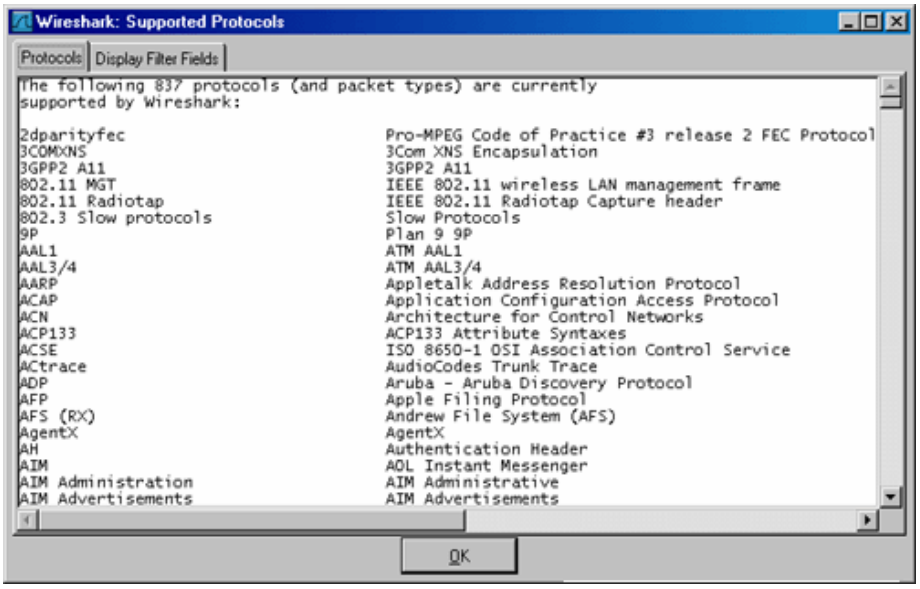
Σύνταξη:	Πρωτόκολλο	Ακολουθία 1	Ακολουθία 2	Πράξη σύγκρισης	Τιμή	Λογικές Πράξεις	Άλλες εκφράσεις
Παράδειγμα:	ftp	passive	ip	==	10.2.3.4	xor	icmp.type

Πρωτόκολλο:
Ένας μεγάλος αριθμός πρωτοκόλλων, τα οποία βρίσκονται μεταξύ του 2ου και του 7ου του μοντέλου OSI, είναι διαθέσιμος. Μπορείτε να τα δείτε όταν πατήσετε πάνω στο κουμπί "Expression..." στο κυρίως παράθυρο.
Μερικά παραδείγματα : IP, TCP, DNS, SSH









Σχ.20 Λίστα διαθέσιμων πρωτοκόλλων

Πράξεις σύγκρισης::
Έξι πράξεις σύγκρισης είναι διαθέσιμες:

Αγγλική μορφή:	Μορφή γλώσσας C:	Σημασία:
eq	==	Ίσο
ne	!=	Διάφορο
gt	>	Μεγαλύτερο από
lt	<	Μικρότερο από
ge	>=	Μεγαλύτερο ή ίσο
le	<=	Μικρότερο ή ίσο

Λογικές εκφράσεις:

Αγγλική μορφή:	Μορφή γλώσσας C:	Σημασία:
and	&&	Λογικό AND
or	||	Λογικό OR
xor	^^	Λογικό XOR
not	!	Λογικό NOT

Η λογική έκφραση XOR, πολύ γνωστή από τους προγραμματιστές, χρησιμοποιείται ως αποκλειστική αλλαγή. Όταν χρησιμοποιείται μεταξύ δυο συνθηκών μέσα σ' ένα φίλτρο, το αποτέλεσμα θα εμφανιστεί στην οθόνη μόνο αν μια από τις δύο συνθήκες ισχύει, αλλά όχι όταν ισχύουν και οι δύο όπως γίνεται με τη λογική έκφραση OR.

Ας πάρουμε ως παράδειγμα το παρακάτω φίλτρο απεικόνισης:
"tcp.dstport 80 xor tcp.dstport 1025"
Μόνο τα πακέτα με προορισμό TCP στη θύρα 80 ή πηγή TCP στη θύρα 1025 (αλλά όχι κατά 2 ταυτόχρονα!) θα εμφανιστούν στην οθόνη σαν αποτέλεσμα.

Παράδειγμα:

snmp || dns || icmp Εμφάνιση μεταφορών SNMP ή DNS ή ICMP.

ip.addr == 10.1.1.1
Εμφανίζει τα πακέτα με διεύθυνση IP πηγής ή προορισμού η οποία ισούται με 10.1.1.1.

ip.src != 10.1.2.3 or ip.dst != 10.4.5.6
Εμφανίζει τα πακέτα με διεύθυνση IP πηγής διαφορετική της 10.1.2.3 ή με διεύθυνση IP προορισμού διαφορετική της 10.4.5.6.
Με άλλα λόγια, τα εμφανιζόμενα πακέτα θα έχουν:
διεύθυνση IP πηγής: οποιαδήποτε εκτός της 10.1.2.3 ,διεύθυνση IP προορισμού: οποιαδήποτε και διεύθυνση IP πηγής: οποιαδήποτε, διεύθυνση IP προορισμού: οποιαδήποτε εκτός της 10.4.5.6

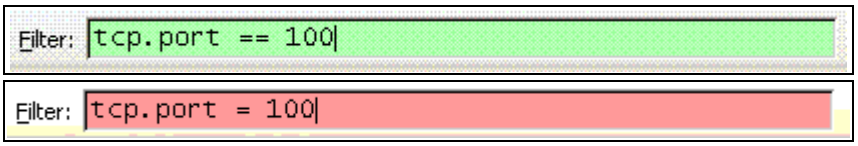
ip.src != 10.1.2.3 and ip.dst != 10.4.5.6
Εμφανίζει τα πακέτα με διεύθυνση IP πηγής διαφορετική από την 10.1.2.3 και την ίδια στιγμή με διεύθυνση IP προορισμού διαφορετική της 10.4.5.6
Με άλλα λόγια, τα εμφανιζόμενα πακέτα θα έχουν:
διεύθυνση IP πηγής: οποιαδήποτε εκτός της 10.1.2.3 και διεύθυνση IP προορισμού: οποιαδήποτε εκτός της 10.3.4.5.6

tcp.port == 25 Εμφανίζει τα πακέτα πηγής TCP ή προορισμό την θύρα 25.

tcp.dstport == 25 Εμφανίζει τα πακέτα προορισμού TCP με θύρα προορισμού 25.

tcp.flags Εμφανίζει τα πακέτα με σημαία TCP.

tcp.flags.syn == 0x02 Εμφανίζει τα πακέτα με σημαία TCP SYN.
Εάν η σύνταξη του φίλτρου είναι σωστή, θα υπογραμμιστεί με πράσινο χρώμα, ειδάλλως εάν υπάρχει λάθος στην σύνταξή του θα υπογραμμιστεί με κόκκινο χρώμα.



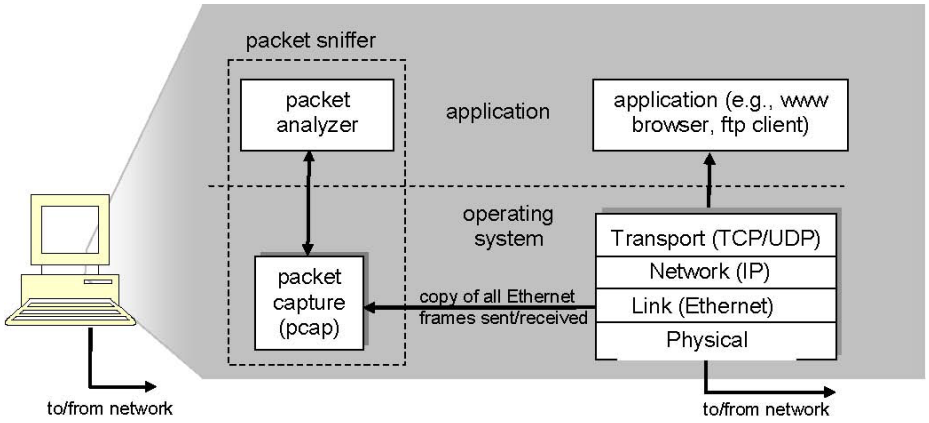
Σχ.21 Σωστή και λάθος σύνταξη φίλτρου

Κεφάλαιο 3
Ασκήσεις επίδειξης του περιβάλλοντος εργασίας του wireshark

Συχνά, μπορούμε να κατανοήσουμε καλύτερα τα δικτυακά πρωτοκόλλα "παρατηρώντας τα σε δράση", δηλαδή παρατηρώντας την ακολουθία των μηνυμάτων που ανταλλάσσονται μεταξύ δύο οντοτήτων πρωτοκόλλων. Έτσι, εισχωρώντας στις λεπτομέρειες της λειτουργίας των πρωτοκόλλων και αναγκάζοντάς τα να εκτελέσουν συγκεκριμένες ενέργειες, τις συνέπειες των οποίων μπορούμε να παρακολούθησουμε μέσω του Wireshark, μπορούμε να βγάλουμε πολύτιμα συμπεράσματα για την κατανόηση της λειτουργίας τους. Αυτό μπορεί να γίνει σε προσομοιωμένα σενάρια ή σε ένα περιβάλλον "πραγματικού" δικτύου όπως το διαδίκτυο. Στα εργαστήρια Wireshark θα τρέξουμε διάφορες δικτυακές εφαρμογές σε διαφορετικά σενάρια χρησιμοποιώντας έναν υπολογιστή στο γραφείο, στο σπίτι ή σε ένα εργαστήριο. Θα παρατηρήσουμε τα δικτυακά πρωτόκολλα στον υπολογιστή μας "σε δράση" να αλληλεπιδρούν και να ανταλλάσσουν μηνύματα με οντότητες πρωτοκόλλων που εκτελούνται αλλού στο διαδίκτυο.

Το βασικό εργαλείο για την παρατήρηση των μηνυμάτων που ανταλλάσσονται μεταξύ των εκτελούμενων οντοτήτων πρωτοκόλλων καλείται **packet sniffer.** Όπως υπονοεί και το όνομα, ο packet sniffer συλλαμβάνει ("sniffs") τα μηνύματα τα οποία στέλνονται ή λαμβάνονται από τον υπολογιστή μας. Επίσης, ο packet sniffer συνήθως αποθηκεύει και απεικονίζει τα περιεχόμενα διαφόρων πεδίων πρωτοκόλλων που περιέχονται στα μηνύματα που συλλαμβάνονται. Ο ίδιος ο packet sniffer είναι παθητικός. Παρατηρεί τα μηνύματα που στέλνονται και λαμβάνονται από τις εφαρμογές και τα πρωτόκολλα που τρέχουν στον υπολογιστή μας αλλά ο ίδιος δεν στέλνει ποτέ πακέτα. Ο packet sniffer λαμβάνει ένα αντίγραφο των πακέτων που στέλνονται ή λαμβάνονται από τις εφαρμογές και τα πρωτόκολλα που εκτελούνται στον υπολογιστή μας.

Στο Σχ.22 φαίνεται η δομή ενός packet sniffer. Στο δεξί μέρος του Σχ.22 φαίνονται τα πρωτόκολλα (στην προκειμένη περίπτωση τα πρωτόκολλα του Διαδικτύου) και οι εφαρμογές (όπως ένας web browser ή ένας ftp client) που τρέχουν κανονικά στον υπολογιστή μας. Ο packet sniffer, ο οποίος φαίνεται μέσα στο πλαίσιο των διακικομένων γραμμών του Σχ.22, είναι μια προσθήκη στο συνήθες λογισμικό του υπολογιστή μας και αποτελείται από δυο μέρη. Η **βιβλιοθήκη σύλληψης πακέτων (packet capture library)** λαμβάνει ένα αντίγραφο κάθε πλαισίου επιπέδου ζεύξης που στέλνεται ή λαμβάνεται από τον υπολογιστή μας. Υπενθυμίζεται ότι τα μυνήματα που ανταλλάσσονται από τα πρωτόκολα ανωτέρων επιπέδων, όπως το HTTP, FTP, TCP, UDP ή το IP, τελικά ενθυλακώνονται όλα μέσα σε πλαίσια επιπέδου ζεύξης τα οποία μεταδίδονται μέσω φυσικών μέσων όπως ένα καλώδιο Ethernet. Επομένως, η σύλληψη όλων των πλαισίων ζεύξης μας παρέχει όλα τα μυνήματα που στέλνονται και λαμβάνονται από όλα τα πρωτόκολλα και όλες τις εφαρμογές που εκτελούνται στον υπολογιστή μας.



Σχ.22 Δομή packet sniffer

Το δεύτερο συστατικό στοιχείο ενός packet sniffer είναι **ο αναλυτής πακέτων (packet analyzer)**, ο οποίος απεικονίζει τα περιεχόμενα όλων των πεδίων μέσα στο μήνυμα ενός πρωτοκόλλου. Για το σκοπό αυτό, ο αναλυτής πακέτων πρέπει να "καταλαβαίνει" τη δομή όλων των μηνυμάτων που ανταλλάσσονται από τα πρωτόκολλα. Για παράδειγμα, έστω ότι ενδιαφερόμαστε να απεικονήσουμε τα διάφορα πεδία των μηνυμάτων που ανταλλάσσονται από το πρωτόκολλο HTTP στο Σχ.22. Ο αναλυτής πακέτων καταλαβαίνει τη μορφή των πλαισίων Ethernet και επομένως μπορεί να αναγνωρίσει ένα αυτοδύναμο πακέτο IP (IP datagram) μέσα σε πλαίσιο Ethernet. Επίσης, καταλαβαίνει τη μορφή ενός IP datagram, ώστε να είναι σε θέση να εξάγει ένα TCP segment που περιέχεται μέσα σε ένα IP datagram. Επιπλέον, καταλαβαίνει τη δομή ενός TCP segment οπότε μπορεί να εξάγει το μήνυμα HTTP που περιέχεται στο TCP segment. Τέλος, καταλαβαίνει το πρωτόκολλο HTTP και έτσι, για παράδειγμα, γνωρίζει ότι τα πρώτα bytes ενός μηνύματος HTTP θα περιέχουν τις ακολουθίες χαρακτήρων GET, POST ή HEAD

Στα εργαστήρια αυτά θα χρησιμοποιήσουμε τον packet sniffer Wireshark (http://www.wireshark.org/) ο οποίος θα μας δώσει την δυνατότητα να απεικονήσουμε τα περιεχόμενα των μηνυμάτων που στέλνονται ή λαμβάνονται από τα πρωτόκολλα σε διαφορετικά επίπεδα της στοίβας πρωτοκόλλων. (Σε τεχνική γλώσσα, το Wireshark είναι ένας αναλυτής πακέτων που χρησιμοποιεί μία βιβλιοθήκη σύλληψης πακέτων στον υπολογιστή σας. Το Wireshark είναι ένας αναλυτής δικτυακών πρωτοκόλλων που προσφέρεται δωρεάν για Windows, Linux/Unix και Mac. Έχει μία μεγάλη βάση χρηστών και καλά στοιχειοθετημένη υποστήριξη που περιλαμβάνει έναν οδηγό χρήστη (http://www.wireshark.org/docs/wsug_html_chunked/), σελίδες εγχειριδίου χρήστη (man pages) (http://www.wireshark.org/docs/man-pages/) και ένα λεπτομερή κατάλογο συχνών ερωτημάτων (Frequently Asked Questions, FAQ) (http://www.wireshark.org/faq.html), είναι πλούσιος σε λειτουργίες που περιλαμβάνουν τη ικανότητα να αναλύει περισσότερα απο εκατοντάδες πρωτοκόλλων και έχει μία καλά σχεδιασμένη διεπαφή χρήστη (user interface). Λειτουργεί σε υπολογιστές που χρησιμοποιούν Ethernet, Token-Ring, FDDI, σειριακές συνδέσεις (PP και SLIP), ασύρματα τοπικά δίκτυα 802.11 και συνδέσεις ATM (ανάλογα με το λειτουργικό σύστημα).

3.1 Πως να λάβουμε το Wireshark

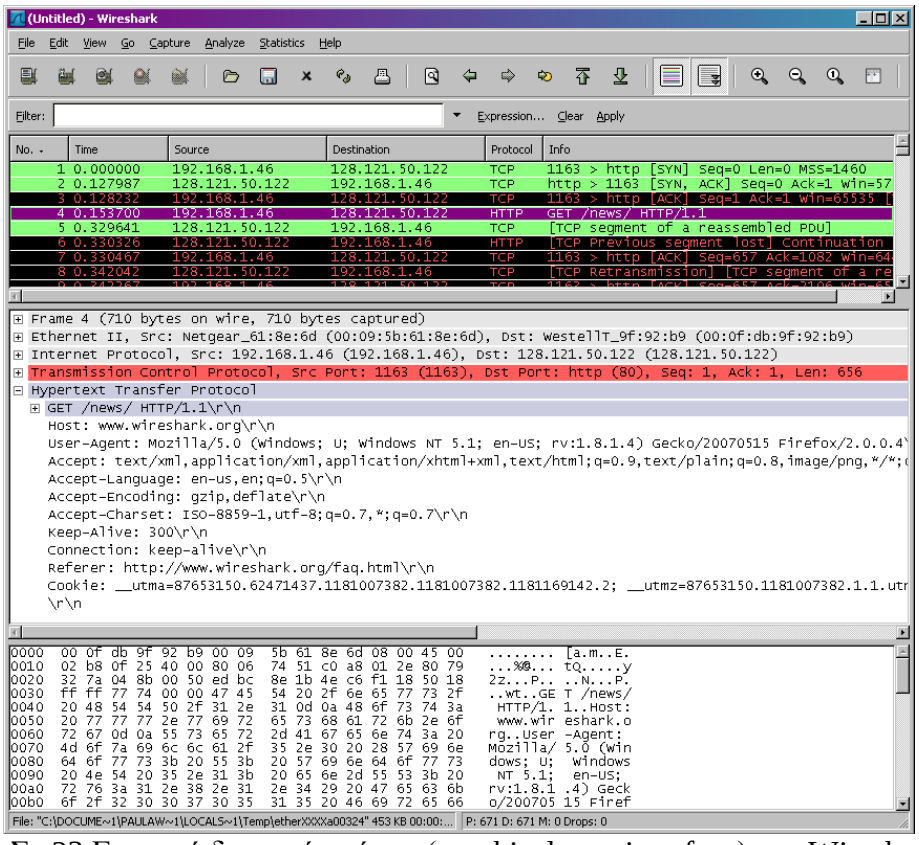
Για να τρέξουμε το Wireshark χρειαζόμαστε ένα υπολογιστή που να υποστηρίζει και το Wireshark και τη βιβλιοθήλη σύλληψης πακέτων libcap ή WinPCap. Εάν το

λογισμικό libcap δεν είναι ήδη εγκατεστημένο στο λειτουργικό μας σύστημα θα εγκατασταθεί όταν εγκαταστήσουμε το Wireshark.

•Πηγαίνουμε στην ιστοσελίδα http://www.wireshark.org/download.html, φορτώνουμε και εγκαταστουμε το δυαδικό αρχείο Wireshark που είναι κατάλληλο για τον υπολογιστή μας.
• Φορτώνουμε τον οδηγό χρήστη του Wireshark.
Ο κατάλογος FAQ του Wireshark περιλαμβάνει έναν αριθμό από χρήσιμες υποδείξεις και ενδιαφέρουσες πληροφορίες, ειδικά εάν αντιμετωπίσουμε προβλήματα με την εγκατάσταση ή την εκτέλεση του Wireshark.

3.2 Εκτέλεση του Wireshark

Κατά την εκτέλεση του προγράμματος Wireshark εμφανίζεται στην οθόνη η γραφική διεπαφή χρήστη (graphical user interface, GUI) του Wireshark που φαίνεται στο Σχ.23



Σχ.23 Γραφική διεπαφή χρήστη (graphical user interface) του Wireshark

Η διεπαφή του Wireshark περιλαμβάνει πέντε κύρια στοιχεία:
• Τα **μενού των εντολών (command menus)** είναι συνηθισμένα πτυσσόμενα (pull down) μενού που βρίσκονται στο επάνω μέρος του παραθύρου. Προς το παρόν μας ενδιαφέρουν τα μενού File και Capture. Το μενού File επιτρέπει την αποθήκευση δεδομένων για πακέτα που έχουν συλληφθεί ή το άνοιγμα ενός αρχείου που περιέχει δεδομένα πακέτων που είχαν συλληφθεί προηγουμένως και την έξοδο από το Wireshark. Το μενού Capture μας επιτρέπει να ξεκινήσουμε τη σύλληψη πακέτων.

• Το **παράθυρο καταλόγου πακέτων (packet-listing window)** παρουσιάζει μία περίληψη της μιας γραμμής για κάθε πακέτο που συλλαμβάνεται η οποία περιλαμβάνει τον αριθμό πακέτου, τον χρόνο σύλληψης του πακέτου, τις διευθύνσεις πηγής και προορισμού του πακέτου, το είδος του πρωτοκόλλου και πληροφορίες σχετικες με το πρωτόκολλο οι οποίες περιέχονται στο πακέτο. Στο πεδίο είδος πρωτοκόλλου (protocol type) αναφέρεται το ανωτάτο επιπέδο πρωτόκολλου το οποίο έστειλε ή έλαβε ένα πακέτο, δηλαδή, το πρωτόκολλο που είναι η πηγή ή ο τελικός αποδέκτης αυτού του πακέτου.

• Το **παράθυρο λεπτομερειών επικεφαλίδας πακέτου (packet-header details window)** παρέχει λεπτομέρειες σχετικά με το επιλεγμένο στο παράθυρο packet-listing πακέτο. Οι λεπτομέρειες αυτές περιλαμβάνουν πληροφορίες σχετικά με το πλαίσιο Ethernet και το IP datagram που περιέχουν αυτό το πακέτο. Τέλος, λεπτομέρειες παρέχονται επίσης για το ανωτάτο επιπέδο πρωτόκολλου το οποίο έστειλε ή έλαβε αυτό το πακέτο.

• Το παράθυρο **περιεχομένων πακέτου (packet-contents window)** παρουσιάζει ολόκληρο το περιεχόμενο ενός συλλαμβανόμενου πλαισίου και σε μορφή ASCII και σε δεκαεξαδική μορφή.

•Στο επάνω μέρος της διεπαφής Wireshark βρίσκεται **το πεδίο του φίλτρου παρουσίασης πακέτων (packet display filter field)** στο οποίο μπορούμε να εισάγουμε το όνομα ενός πρωτοκόλλου ή άλλη πληροφορία έτσι ώστε να φιλτράρουμε την πληροφορία που παρουσιάζεται στο παράθυρο packet-listing (άρα και στα παράθυρα packet-header και packet-contents). Στο παράδειγμα που ακολουθεί θα χρησιμοποιήσουμε το πεδίο packet display filter ώστε να κάνουμε το Wireshark να μην παρουσιάσει όλα τα πακέτα εκτός από εκείνα που αντιστοιχούν σε μηνύματα HTTP.

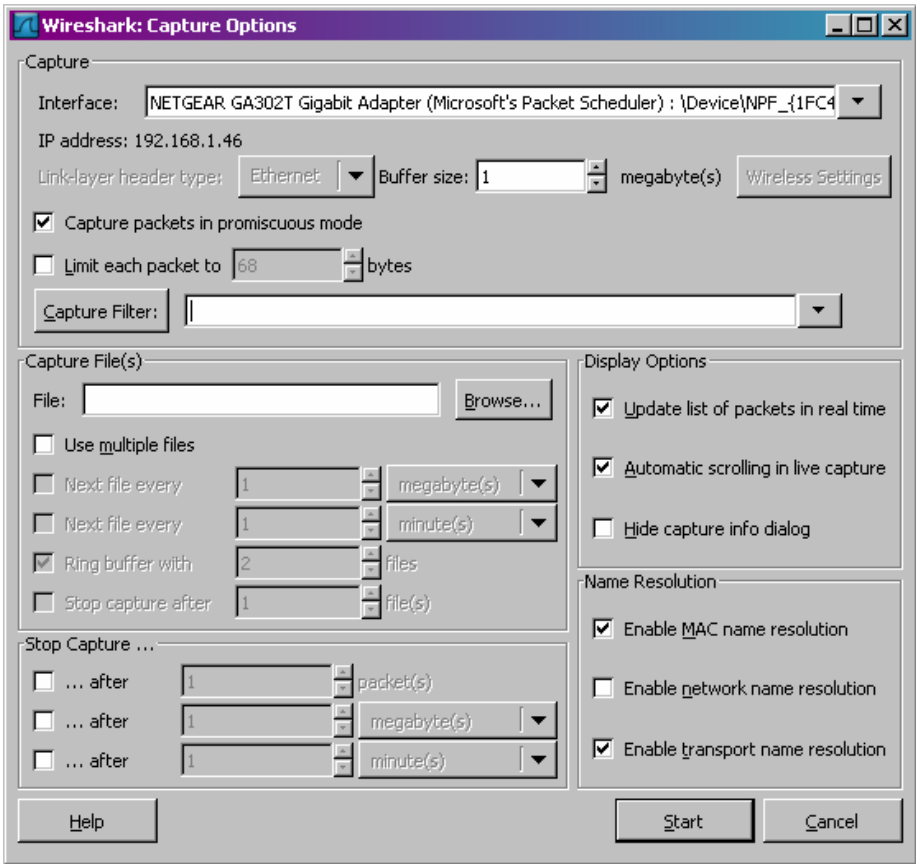
3.3 Δοκιμαστική εκτέλεση του Wireshark

Συνδέουμε τον υπολογιστή μας στο διαδίκτυο μέσω μίας ενσύρματης διεπαφής Ethernet και ακολουθούμε τα παρακάτω βήματα:

1.Ξεκινάμε τον web browser της αρεσκείας μας, ο οποίος θα εμφανίσει την αρχική σελίδα που έχουμε επιλέξει.

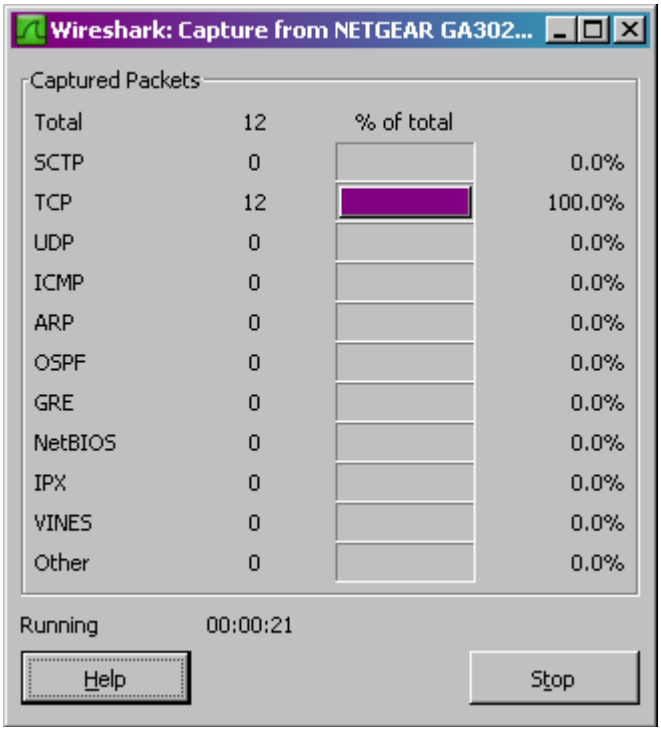
2.Ξεκινάμε το λογισμικό Wireshark. Θα δούμε αρχικά ένα παράθυρο παρόμοιο με αυτό του Σχ.23 με τη διαφορά ότι δε θα εμφανίζονται πακέτα στα παράθυρα packet-listing, packet-header, ή packet-contents, αφού το Wireshark δεν έχει αρχίσει ακόμη να συλλαμβάνει πακέτα.

3. Για να αρχίσει η σύλληψη πακέτων, επιλέγουμε Options στο μενού Capture. Αυτό θα έχει ως αποτέλεσμα την εμφάνιση του παραθύρου “Wireshark: Capture Options” όπως φαίνεται στο Σχ.24.



Σχ.24 Παράθυρο Capture Options του Wireshark

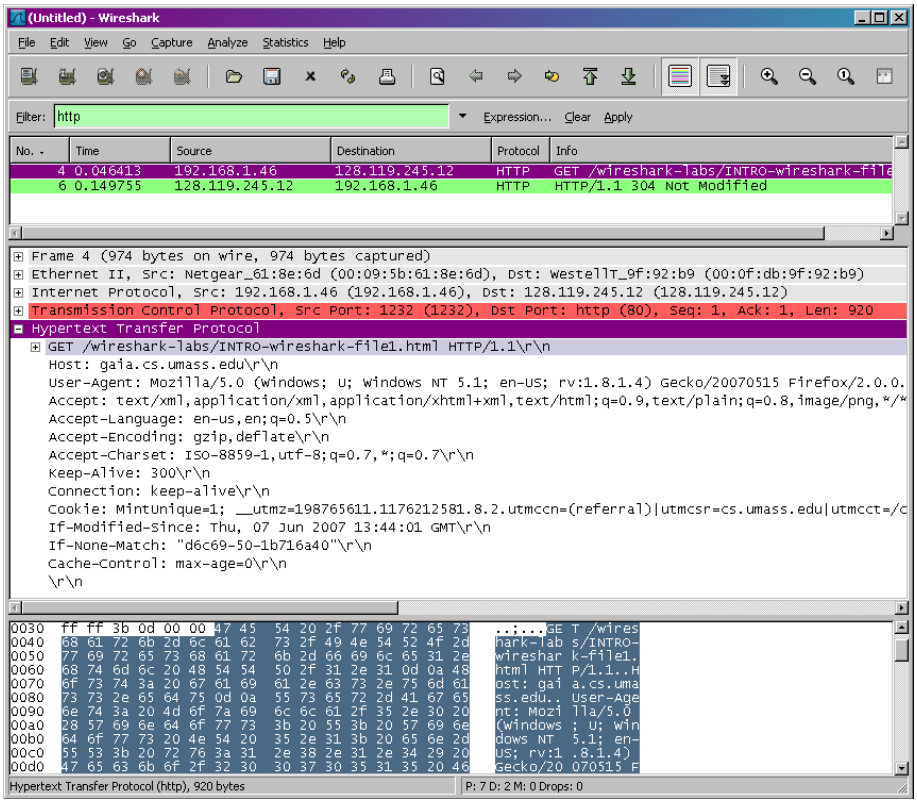
4.Μπορουμε να χρησιμοποιήσουμε όλες τις προεπιλεγμένες τιμές αυτού του παραθύρου. Οι διεπαφές δικτύου (δηλαδή οι φυσικές συνδέσεις) του υπολογιστή μας με το δίκτυο θα εμφανίζονται στο μενού Interface στο επάνω μέρος του παραθύρου Capture Options. Σε περίπτωση που ο υπολογιστής μας έχει περισσότερες από μία ενεργές διεπαφές δικτύου (π.χ. εάν έχουμε αμφότερες μία ασύρματη και μία ενσύρματη σύνδεση Ethernet), θα χρειαστεί να επιλέξουμε μία διεπαφή την οποία θα χρησιμοποιήσουμε για να στέλνουμε και να λαμβάνουμε πακέτα (το πιθανότερο την ενσύρματη διεπαφή). Αφού επιλέξουμε τη διεπαφή δικτύου (ή χρησιμοποιήσουμε την προεπιλεγμένη διεπαφή που επιλέγει το Wireshark), κάνουμε κλικ στο Start. Στο σημείο αυτό αρχίζει η σύλληψη των πακέτων: όλα τα πακέτα που στέλνονται ή λαμβάνονται από τον υπολογιστή μας συλλαμβάνονται από το Wireshark.
5.Μόλις αρχίσει η σύλληψη πακέτων θα εμφανισθεί ένα παράθυρο περίληψης σύλληψης πακέτων (packet capture summary window) όπως φαίνεται στο Σχ.25. Το παράθυρο αυτό συνοψίζει τον αριθμό των διαφόρων ειδών πακέτων που συλλαμβάνονται και περιέχει το κουμπί Stop το οποίο μας επιτρέπει να διακόψουμε τη σύλληψη πακέτων. Δεν σταματάμε τη σύλληψη πακέτων ακόμη.



Σχ.25: Παράθυρο Packet Capture του Wireshark

6. Ενώ το Wireshark τρέχει, εισάγουμε το URL: http://gaia.cs.umass.edu/wireshark-labs/INTRO-wireshark-file1.html ώστε ο browser να παρουσιάσει αυτήν την ιστοσελίδα. Για να παρουσιάσει αυτή τη σελίδα, ο browser μας θα επικοινωνήσει με τον HTTP server στο gaia.cs.umass.edu και θα ανταλλάξει μηνύματα HTTP με τον server. Τα πλαίσια Ethernet που περιέχουν αυτά τα μηνύματα HTTP θα συλληφθούν από το Wireshark.
7.Αφού ο browser μας παρουσιάσει τη σελίδα INTRO-wireshark-file1.html, σταματαμε τη σύλληψη πακέτων επιλέγοντας stop στο παράθυρο capture του Wireshark. Αυτό θα έχει ως αποτέλεσμα να εξαφανισθεί το παράθυρο capture του Wireshark και το κύριο παράθυρο του Wireshark να εμφανίζει όλα τα πακέτα που συνελήφθησαν από τότε που αρχίσαμε τη σύλληψη πακέτων. Το κύριο παράθυρο του Wireshark θα πρέπει τώρα να μοιάζει με αυτό του Σχ.23. Έχουμε τώρα στη διάθεσή μας "ζωντανά" δεδομένα πακέτων τα οποία περιέχουν όλα τα μηνύματα πρωτοκόλλων που ανταλλάχθηκαν μεταξύ του υπολογιστή μας και άλλων δικτυακών οντοτήτων. Οι ανταλλαγές μηνυμάτων HTTP με τον web server gaia.cs.umass.edu θα πρέπει να εμφανίζονται κάπου στον κατάλογο πακέτων που συνελήφθησαν. Όμως θα εμφανίζονται επίσης και πολλά άλλα είδη πακέτων (για παράδειγμα, ο μεγάλος αριθμός διαφορετικών ειδών πρωτοκόλλων που φαίνονται στη στήλη Protocol του Σχήματος). Αν και η μόνη δική μας ενέργεια ήταν να φορτώσουμε μία ιστοσελίδα, προφανώς στον υπολογιστή μας έτρεχαν πολλά άλλα πρωτόκολλα χωρίς να τα αντιλαμβάνεται ο χρήστης.
8.Πληκτρολογούμε "http" (χωρίς τα εισαγωγικά και με μικρά γράμματα - στο Wireshark όλα τα ονόματα πρωτοκόλλων είναι με μικρά γράμματα) στο παράθυρο προσδιορισμού του φίλτρου παρουσίασης, στο επάνω μέρος του κυρίου παραθύρου του Wireshark. Στη συνέχεια επιλέγουμε Apply (δεξιά από εκεί όπου εισάγαμε "http"). Αυτό θα έχει ως αποτέλεσμα στο παράθυρο packet-listing να εμφανίζονται μόνο τα μηνύματα HTTP.
9.Επιλέγουμε το πρώτο μήνυμα HTTP που εμφανίζεται στο παράθυρο packet-listing. Αυτό θα πρέπει να είναι το μήνυμα HTTP GET το οποίο στάλθηκε από τον υπολογιστή

μας στον HTTP server gaia.cs.umass.edu. Όταν επιλέξουμε το μήνυμα HTTP GET, οι πληροφορίες για το πλαίσιο Ethernet, το IP datagram, το TCP segment και την επικεφαλίδα του μηνύματος HTTP θα εμφανισθούν στο παράθυρο packet header (1). Κάνοντας κλικ στα κουτάκια με τα σύμβολα συν (+) και πλην (-) στην αριστερή πλευρά του παραθύρου packet-header details, ελαχιστοποιούμε την πληροφορία που εμφανίζεται για το πλαίσιο, το Ethernet, το πρωτόκολλο Internet και το πρωτόκολλο TCP. Μεγιστοποιούμε την πληροφορία που εμφανίζεται για το πρωτόκολλο HTTP. Το κύριο παράθυρο του Wireshark θα πρέπει τώρα να μοιάζει σε γενικές γραμμές με αυτό που φαίνεται στο Σχ.26.



Σχ.26: Το παράθυρο του Wireshark μετά το βήμα 9

10. Έξοδος από το Wireshark

Στο σημείο αυτό έχετε ολοκληρώσει το πρώτο εργαστήριο.

Ο πρωταρχικός στόχος αυτού του πρώτου εργαστηρίου ήταν η εισαγωγή στο Wireshark. Βασιζόμενοι στον πειραματισμό σας με το Wireshark, απαντήστε στις ακόλουθες ερωτήσεις:

ΕΡΩΤΗΣΕΙΣ

1.Αναφέρατε έως 10 διάφορα πρωτόκολλα που εμφανίζονται στη στήλη Protocol στο αφιλτράριστο παράθυρο packet-listing στο βήμα 7 παραπάνω.

2.Πόσος χρόνος πέρασε από τότε που στάλθηκε το μήνυμα HTTP GET μέχρι να ληφθεί η απόκριση HTTP OK; (Η τιμή της στήλης Time στο παράθυρο packet-listing είναι, εκ προεπιλογής, το χρονικό διάστημα από την έναρξη σύλληψης πακέτων σε δευτερόλεπτα. Για να δείτε το πεδίο Time με τη μορφή ώρα της ημέρας (time-of-day), επιλέξτε το μενού View, μετά επιλέξτε Time Display Format και μετά επιλέξτε Time-of-day.)

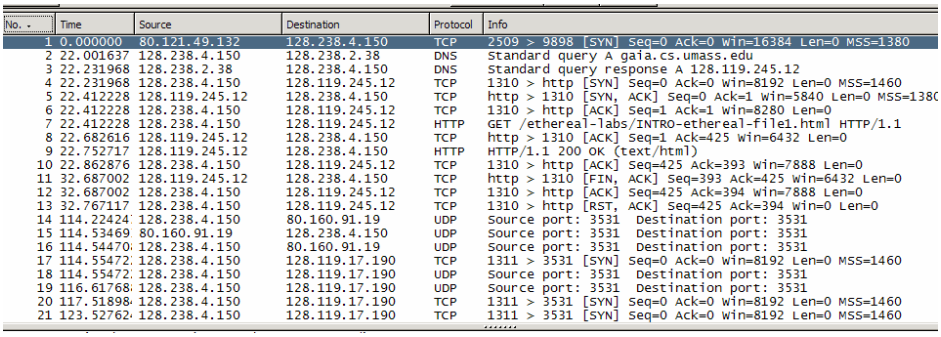
3.Ποιά είναι η διεύθυνση IP του gaia.cs.umass.edu (επίσης γνωστή ως www.net.cs.umass.edu); Ποιά η IP διεύθυνση του υπολογιστή σας;

4.Εκτυπώστε τα δύο μηνύματα HTTP που απεικονίζονται στο βήμα 9 παραπάνω. Για να το κάνετε αυτό, επιλέξτε Print από το menu εντολών File του Wireshark, επιλέξτε "Selected Packet Only" και "Print as displayed" και μετά κάντε κλικ στο OK.

(1)Υπενθυμίζεται ότι το μήνυμα HTTP GET που στέλνεται στον web server gaia.cs.umass.edu περιέχεται μέσα σε ένα TCP segment, το οποίο περιέχεται μέσα σε ένα IP datagram, το οποίο είναι ενθυλακωμένο μέσα σε ένα πλαίσιο Ethernet.

ΑΠΑΝΤΗΣΕΙΣ

1.Τα πρωτόκολλα που εμφανίζονται είναι τα εξής TCP, UDP, HTTP, DNS.



No.	Time	Source	Destination	Protocol	Info
1	0.000000	80.121.49.132	128.238.4.150	TCP	2509 > 9898 [SYN] Seq=0 Ack=0 Win=16384 Len=0 MSS=1380
2	22.001637	128.238.4.150	128.238.2.38	DNS	Standard query A gaia.cs.umass.edu
3	22.231968	128.238.2.38	128.238.4.150	DNS	Standard query response A 128.119.245.12
4	22.231968	128.238.4.150	128.119.245.12	TCP	1310 > http [SYN] Seq=0 Ack=0 Win=8192 Len=0 MSS=1460
5	22.412228	128.119.245.12	128.238.4.150	TCP	http > 1310 [SYN, ACK] Seq=0 Ack=1 Win=5840 Len=0 MSS=138(
6	22.412228	128.238.4.150	128.119.245.12	TCP	1310 > http [ACK] Seq=1 Ack=1 Win=8280 Len=0
7	22.412228	128.238.4.150	128.119.245.12	HTTP	GET /ethereal-labs/INTRO-ethereal-file1.html HTTP/1.1
8	22.682616	128.119.245.12	128.238.4.150	TCP	http > 1310 [ACK] Seq=1 Ack=425 Win=6432 Len=0
9	22.752717	128.119.245.12	128.238.4.150	HTTP	HTTP/1.1 200 OK (text/html)
10	22.862876	128.238.4.150	128.119.245.12	TCP	1310 > http [ACK] Seq=425 Ack=393 Win=7888 Len=0
11	32.687002	128.119.245.12	128.238.4.150	TCP	http > 1310 [FIN, ACK] Seq=393 Ack=425 Win=6432 Len=0
12	32.687002	128.238.4.150	128.119.245.12	TCP	1310 > http [ACK] Seq=425 Ack=394 Win=7888 Len=0
13	32.767117	128.238.4.150	128.119.245.12	TCP	1310 > http [RST, ACK] Seq=425 Ack=394 Win=0 Len=0
14	114.22424	128.238.4.150	80.160.91.19	UDP	Source port: 3531 Destination port: 3531
15	114.53469	80.160.91.19	128.238.4.150	UDP	Source port: 3531 Destination port: 3531
16	114.54470	128.238.4.150	80.160.91.19	UDP	Source port: 3531 Destination port: 3531
17	114.55472	128.238.4.150	128.119.17.190	TCP	1311 > 3531 [SYN] Seq=0 Ack=0 Win=8192 Len=0 MSS=1460
18	114.55472	128.238.4.150	128.119.17.190	UDP	Source port: 3531 Destination port: 3531
19	116.61768	128.238.4.150	128.119.17.190	UDP	Source port: 3531 Destination port: 3531
20	117.51898	128.238.4.150	128.119.17.190	TCP	1311 > 3531 [SYN] Seq=0 Ack=0 Win=8192 Len=0 MSS=1460
21	123.52762	128.238.4.150	128.119.17.190	TCP	1311 > 3531 [SYN] Seq=0 Ack=0 Win=8192 Len=0 MSS=1460

2.Αν κοιτάξουμε στο τμήμα της GET request θα δούμε ότι η στιγμή που έφτασε το πακέτο είναι 11:43:13.422848000

```
▽ Frame 109 (492 bytes on wire, 492 bytes captured)
    Arrival Time: Sep 17, 2004 11:43:13.422848000
    Time delta from previous packet: 6.826032000 seconds
    Time since reference or first frame: 9.263432000 seconds
    Frame Number: 109
    Packet Length: 492 bytes
    Capture Length: 492 bytes
```

Στο τμήμα HTTP OK βλέπουμε ότι η στιγμή που έφτασε το πακέτο είναι 11:43:13.43960400

```
▽ Frame 110 (444 bytes on wire, 444 bytes captured)
    Arrival Time: Sep 17, 2004 11:43:13.439604000
    Time delta from previous packet: 0.016756000 seconds
    Time since reference or first frame: 9.280188000 seconds
    Frame Number: 110
    Packet Length: 444 bytes
    Capture Length: 444 bytes
```

Η διαφορά των δύο χρόνων είναι 0.43960400-0.426032000=0.013572

3. Στο IP τμήμα της GET request βλέπουμε την πηγή και τον προορισμό

```
Source: 128.238.244.28 (128.238.244.28)
Destination: 128.119.245.12 (128.119.245.12)
```

Η πηγή είναι η διεύθυνση του υπολογιστή μας 128.238.244.28
Και η διεύθυνση προορισμού είναι το IP της www.net.cs.umass.edu =128.119.245.12

4.

HTTP GET:

```
Frame 4 (862 bytes on wire, 862 bytes captured)
Ethernet II, Src: Netgear_61:8e:6d (00:09:5b:61:8e:6d), Dst: WestellT_9f:92:b9
(00:0f:db:9f:92:b9)
Internet Protocol, Src: 192.168.1.46 (192.168.1.46), Dst: 128.119.245.12
(128.119.245.12)
Transmission Control Protocol, Src Port: 1474 (1474), Dst Port: http (80), Seq: 1,
Ack: 1, Len: 808
Hypertext Transfer Protocol
    GET /wireshark-labs/INTRO-wireshark-file1.html HTTP/1.1\r\n
    Host: gaia.cs.umass.edu\r\n
    User-Agent: Mozilla/5.0 (Windows; U; Windows NT 5.1; en-US; rv:1.8.1.4)
    Gecko/20070515 Firefox/2.0.0.4\r\n
    Accept:
    text/xml,application/xml,application/xhtml+xml,text/html;q=0.9,text/plain;q=0.8,im
    age/png,*/*;q=0.5\r\n
    Accept-Language: en-us,en;q=0.5\r\n
    Accept-Encoding: gzip,deflate\r\n
    Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.7\r\n
    Keep-Alive: 300\r\n
    Connection: keep-alive\r\n
```

HTTP OK:

```
Frame 6 (439 bytes on wire, 439 bytes captured)
Ethernet II, Src: WestellT_9f:92:b9 (00:0f:db:9f:92:b9), Dst: Netgear_61:8e:6d
(00:09:5b:61:8e:6d)
Internet Protocol, Src: 128.119.245.12 (128.119.245.12), Dst: 192.168.1.46
(192.168.1.46)
Transmission Control Protocol, Src Port: http (80), Dst Port: 1474 (1474), Seq: 1,
Ack: 809, Len: 385
Hypertext Transfer Protocol
    HTTP/1.1 200 OK\r\n
    Date: Thu, 07 Jun 2007 18:09:01 GMT\r\n
    Server: Apache/2.0.52 (CentOS)\r\n
    Last-Modified: Thu, 07 Jun 2007 18:08:01 GMT\r\n
    ETag: "d6c69-50-cb94a240"\r\n
    Accept-Ranges: bytes\r\n
    Content-Length: 80
    Keep-Alive: timeout=10, max=100\r\n
    Connection: Keep-Alive\r\n
    Content-Type: text/html; charset=ISO-8859-1\r\n
    \r\n
Line-based text data: text/html
```

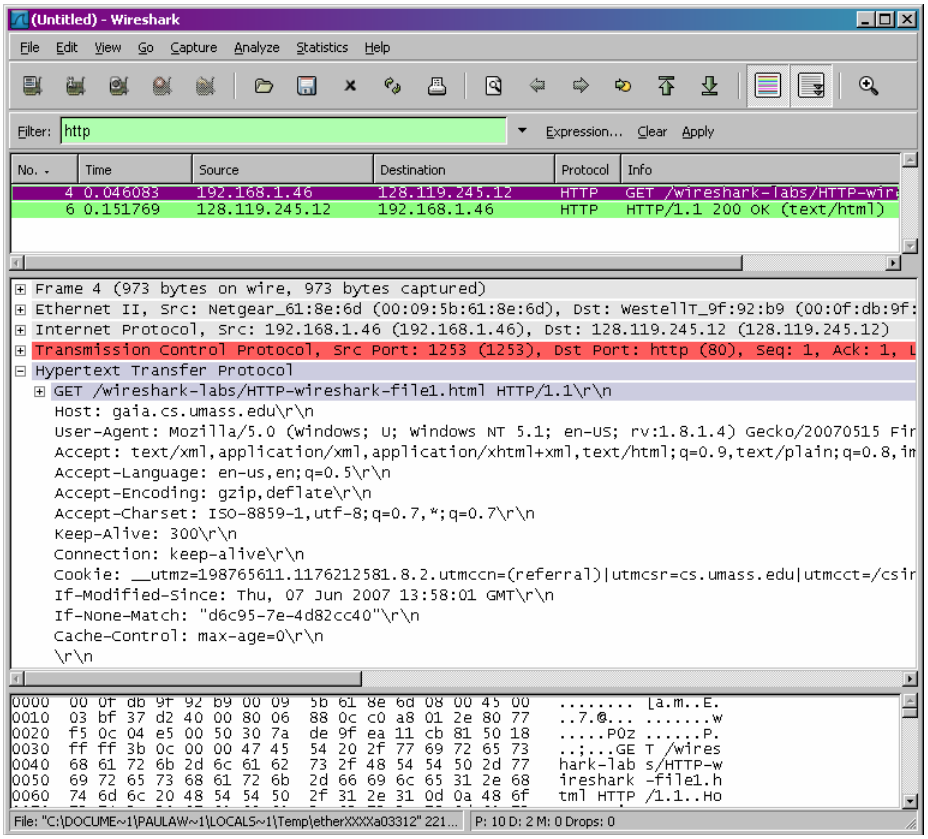
Κεφάλαιο 4
Ασκήσεις επίδειξης του πρωτοκόλλου HTTP

Το Πρωτόκολλο Μεταφοράς Υπερκειμένου (HyperText Transfer Protocol, HTTP) είναι το πιο συνηθισμένο πρωτόκολλο στο World Wide Web. Είναι ένα σύνολο κανόνων που καθορίζει τον τρόπο με τον οποίο θα γίνει η μεταφορά του υπερκειμένου (hypertext) (το οποίο μπορεί να αντιστοιχεί σε αρχεία κειμένου, γραφικών, εικόνας, ήχου, video ή οποιουδήποτε multimedia αρχείου) μεταξύ δύο ή περισσοτέρων υπολογιστών.
Μετά την πρώτη γεύση του packet sniffer στο εισαγωγικό εργαστήριο, είμαστε έτοιμοι να χρησιμοποιήσουμε το Wireshark για να εξετάσουμε τα πρωτόκολλα σε λειτουργία. Στο εργαστήριο αυτο θα δουμε κάποιες πλευρές του πρωτοκόλλου HTTP –τη βασική αλληλεπίδραση GET/απόκριση, τις μορφές των μυνημάτων HTTP, την ανάκτηση μεγάλων αρχείων HTML, την ανάκτηση μεγάλων αρχείων HTML με ενσωματωμένα αντικείμενα, την εξουσιοδότηση και την ασφάλεια στο HTTP.

4.1 Η Βασική Αλληλεπίδραση GET/Απόκριση στο HTTP

Ας ξεκινήσουμε την διερεύνηση του HTTP με ένα πολύ απλό αρχείο HTML, πολύ μικρό, που δεν περιέχει ενσωματωμένα αντικείμενα.
Ακολουθούμε τα παρακάτω βήματα
1.Ξεκινάμε τον web browser μας
2.Ξεκινάμε τον packet sniffer, όπως στο εισαγωγικό εργαστήριο, χωρίς να ξεκινήσουμε την σύλληψη πακέτων ακόμη. Εισάγουμε ‘http’(χωρίς τα εισαγωγικά) στο παράθυρο προδιαγραφών του φίλτρου παρουσίασης ώστε στο παράθυρο καταλόγου πακέτων να παρουσιάζονται αργότερα μόνο τα συλλαμβανόμενα μηνύματα HTTP.
3.Περιμένουμε λίγο περισσότερο από ένα λεπτό και μετά αρχίζουμε τη σύλληψη πακέτων απο το Wireshark
4.Εισάγουμε το ακόλουθο URL στον browser
http://gaia.cs.umass.edu/wireshark-labs/HTTP-wireshark-file1.html .
Στον browser θα πρέπει να εμφανιστεί ένα πολύ απλό αρχείο HTML με μια γραμμή μόνο.
5.Διακόπτουμε τη σύλληψη πακέτων από το Wireshark

Το παράθυρο wireshark θα πρέπει να μοιάζει με το παράθυρο του Σχ.27. Εάν δεν είστε σε θέση να τρέξετε το wireshark σε μια ζωντανή σύνδεση δικύου, μπορείτε να φορτώσετε ενα trace πακέτων το οποίο δημιουργήθηκε ακολούθωντας τα παρακάτω βήματα(1)



Σχ.27 Το παράθυρο wireshark μετά την ανάκτηση από τον browser σας του trace.

Στο Σχ.27 από το παράθυρο κατάλογου πακέτων, φαίνεται ότι συνελήφθησαν δύο μυνήματα http, το μήνυμα GET (από τον browser στον web server gaia.cs.umass.edu) και το μήνυμα απόκρισης απο τον server στον browser. Στο παράθυρο περιεχομένων πακέτου φαίνονται λεπτόμερειες του επιλεγμένου μυνήματος HTTP GET. Υπενθυμίζεται οτι, αφού το μυνημα μεταφέρθηκε μέσα σε ενα TCP segment (τα πακέτα TCP καλούνται τμήματα, τομείς ή τεμάχια), το οποίο μεταφέρθηκε μέσα σε ένα IP datagram μέσα σε πλαίσιο Ethernet, το wireshark παρουσιάζει πληροφορίες και για τα πακέτα Frame, Ethernet, IP και TCP. Επειδή θέλουμε να ελαχιστοποιήσουμε το ποσό των πληροφοριών που παρουσιάζονται για δεδομένα που δεν σχετίζονται με το HTTP (στο εργαστήριο αυτό ενδιαφερόμαστε για το HTTP, θα εξετάσουμε τα άλλα πρωτόκολλα σε επόμενα εργαστήρια, βεβαιωθήτε ότι τα κουτάκια που βρίσκονται αριστερά των πληροφοριών για τα πακέτα Frame, Ethernet, IP και TCP έχουν το σύμβολο (+) (που σημαίνει ότι υπάρχει κρυμμένη πληροφορία η οποία δεν εμφανίζεται) και ότι η γραμμη HTTP έχει το σύμβολο (-) (που σημαίνει πως εμφανίζονται όλες οι πληροφορίες σχετικά με το μήνυμα HTTP. (Σημείωση-αγνοείστε τα μηνύματα HTTP GET και αποκρίσεων για το Favicon.ico)

Εξετάζοντας τις πληροφορίες των μηνυμάτων HTTP GET και απόκριση, απαντήστε στις ακόλουθες ερωτήσεις. Εκτυπώστε πρώτα το μήνυμα GET και το μήνυμα απόκρισης. Σε κάθε απάντηση σας να υποδεικνύετε το σημείο του μυνήματος που περιέχει την πληροφορία που την αιτιολογεί.

ΕΡΩΤΗΣΕΙΣ

1.Ποιά έκδοση του HTTP τρέχει στον browser σας και ποιά τρέχει στον server;

2.Ποιές γλώσσες υποδεικνύει ο browser στον server οτι μπορεί να αποδεχθεί;

3.Ποιά είναι η διεύθυνση IP του υπολογιστή σας; Ποιά είναι η διεύθυνση του server gaia.cs.umass.edu;

4.Ποιος είναι ο κώδικας κατάστασης (status code) που επιστρέφει ο server στον browser σας;

5.Πότε τροποποιήθηκε για τελευταία φορά στον server το αρχείο HTML το οποίο ανακτήσατε;

6.Πόσα bytes περιεχομένου επιστρέφονται στον browser σας;

7.Εξετάζοντας τα ανεπεξέργαστα δεδομένα στο παραθύρο περιεχομένων πακέτου, βλέπεται στα δεδομένα να περιλαμβάνονται επικεφαλίδες που δεν εμφανίζονται στο παράθυρο καταλόγου πακέτων; Εάν υπάρχουν, κατανομάστε μια.

Στην απάντηση σας στην ερώτηση 5, θα παρατηρήσετε οτι το έγγραφο που μόλις ανακτήσατε τροποποιήθηκε για τελευταία φορά μέσα στο τελευταίο λεπτό πρίν το φορτώσετε. Αυτό οφείλεται στο οτι για το συγκέκριμενο αρχείο ο server gaia.cs.umass.edu θέτει το χρόνο τελευταίας τροποποιήσης του αρχείου στον τρέχοντα χρόνο μια φορά ανα λεπτό. Έτσι, ο browser σας θα φορτώνει ενα 'νέο' αντίγραφο του εγγράφου

4.2 Η Αλληλεπίδραση υπό Συνθήκη GET/Απόκριση στο HTTP

Για να δούμε την αλληλεπίδραση υπό συνθήκη GET/απόκριση θα ακολουθήσουμε τα παρακάτω βήματα αφού πρώτα σιγουρευτούμε οτι η cache (προσωρινή αποθήκευση των αντικειμένων) του browser μας ειναι άδεια (για firefox επιλέξτε tools>clear private data ή για τον internet explorer επιλέξτε tools>internet options>delete file.)
>Ξεκινάμε τον web browser μας
>Ξεκινάμε τον packet sniffer wireshark
>Εισάγουμε το ακόλουθο url στον browser http://gaia.cs.umass.edu/wireshark-labs/HTTP-wireshark-file2.html στον browser θα πρέπει να εμφανιστεί ένα πολύ μικρό αρχείο html με 5 γραμμές
>Γρήγορα εισάγουμε για άλλη μια φορά το ίδιο url στον browser μας (ή απλά επιλέγουμε το refresh του browser)
>Σταματάμε τη σύλληψη πακέτων απο το wireshark και εισάγουμε 'http' στο παράθυρο προδιαγράφων του φίλτρου παρουσιάσης ώστε στο παράθυρο καταλόγου πακέτων να παρουσιάζονται μόνο τα συλλαμβανόμενα μηνύματα http.
(ΣΗΜΕΙΩΣΗ-αν δεν είστε σε θέση να τρέξετε το wireshark σε μια ζωντάνη σύνδεση δικτύου, μπορείτε να χρησιμοποιήσετε το trace πακέτων http-ethernal-trace-2 για να απαντήσετε στις παρακάτω ερώτησεις)

ΕΡΩΤΗΣΕΙΣ

8.Ελέγξτε τα περιεχόμενα της πρώτης αίτησης HTTP GET από τον browser σας στον server. Υπάρχει η γραμμή 'if-modified-since' στην αίτηση HTTP GET;

9.Ελέγξτε τα περιέχομενα της πρώτης απόκρισης του server. Επέστρεψε ο server τα περιεχόμενα του αρχείου; Πού βασίζεται το συμπέρασμα σας;

10.Ελέγξτε τα περιέχομενα της δεύτερης αίτησης HTTP GET απο τον browser σας στον server. Υπάρχει η γραμμή 'if-modified-since' στην αίτηση HTTP GET; Εαν υπάρχει η γραμμή αυτή, τι είδους πληροφόρια ακολουθεί την επικεφαλίδα 'if-modified-since';

11.Τι κώδικα και φράση κατάστασης HTTP επιστρέφει ο server ως απόκριση στην δεύτερη αίτηση HTTP GET; Επέστρεψε ο server τα περιεχόμενα του αρχείου; Εξηγείστε

4.3 Ανάκτηση Μεγάλων Εγγράφων

Τα έγγραφα που ανακτήθηκαν στα προηγούμενα π.χ. ήταν απλά και μικρά αρχεία HTML. Στη συνέχεια θα δούμε τι συμβαίνει όταν φορτώνουμε ένα μεγάλο αρχείο HTML. Ακολουθήστε τα παρακάτω βήματα
>ξεκινάμε το browser μας αφού βεβαιωθούμε οτι η cache του είναι άδεια
>ξεκινάμε τον packet sniffer
>εισάγουμε το ακόλουθο url στον browser μας
http://gaia.cs.umass.edu/wireshark-labs/HTTP-wireshark-file3.html.
Στον browser μας τώρα θα πρέπει να εμφανίζεται το καταστατικό των ανθρώπινων δικαιωμάτων των ΗΠΑ
>Σταματάμε τη σύλληψη πακέτων απο το Wireshark και εισάγουμε 'http' στο παράθυρο προδιαγράφων του φίλτρου παρουσιάσης ώστε στο παράθυρο καταλόγου πακέτων να παρουσιάζονται μόνο τα συλλαμβανόμενα μηνύματα HTTP
(ΣΗΜΕΙΩΣΗ-αν δεν είστε σε θέση να τρέξετε το wireshark σε μια ζωντάνη σύνδεση δικτύου,μπορείτε να χρησιμοποιήσετε το trace πακέτων http-ethernal-trace-3 για να απαντήσετε στις παρακάτω ερώτησεις

Στο παράθυρο καταλόγου πακέτων θα πρέπει να δούμε το μήνυμα HTTP GET, ακολουθούμενο απο μια απόκριση πολλαπλών πακέτων. Στην αίτηση HTTP GET το μύνημα απόκρισης GET αποτελείται από μια γραμμή κατάστασης (status line), ακολουθούμενη απο γραμμές επικεφαλίδας (header lines), ακολουθούμενες απο μια κενή γραμμή, ακολουθούμενη απο το σώμα οντότητας (entity body). Στην δική μας αίτηση HTTP GET, το σώμα της οντότητας της απόκρισης είναι ολόκληρο το αιτούμενο HTML. Σ'αυτήν την περίπτωση, το αρχείο HTML είναι σχετικά μεγάλο και με 4500 bytes είναι πολύ μεγάλο για να χωρέσει σε ενα TCP segment. Έτσι, το ένα μήνυμα απόκρισης HTTP τεμάχιζεται σε αρκέτα κομμάτια από το TCP και κάθε κομμάτι περιέχεται σε ξεχώριστο TCP segment. Κάθε TCP segment καταγράφεται ως ξεχωριστό πακέτο απο το wireshark και το γεγονός οτι η μια απόκριση HTTP τεμαχίστηκε σε πολλαπλά πακέτα TCP υποδεικνύεται με την φράση 'continuation' που εμφανίζει το Wireshark. Τονίζουμε στο σημείο αυτό ότι δεν υπάρχει μήνυμα "Continuation" στο HTTP!

ΕΡΩΤΗΣΕΙΣ

12.Πόσα μηνύματα αιτήσεων HTTP GET στάλθηκαν από τον browser σας;

13.Πόσα TCP segments που περιείχαν δεδομένα χρειάστηκαν για την μεταφορά μιας απόκρισης HTTP;

14.Τι κώδικας και ποιά φράση κατάστασης σχετίζοναι με την απόκριση στην αίτηση HTTP GET;

15.Υπάρχουν γραμμές κατάστασης HTTP στα δεδόμενα που να σχετίζονται με τον τεμαχίσμο του σώματος οντότητας από το TCP;

4.4 Έγγραφα HTML με Ενσωματωμένα Αντικείμενα

Αφού είδαμε τον τρόπο με τον οποίο το wireshark παρουσιάζει την συλλαμβανόμενη κίνηση πακέτων για μεγάλα αρχεία HTML, μπορούμε να εξετάσουμε τι συμβαίνει όταν ο browser μας φορτώνει ένα μεγάλο αρχείο με ενσωματωμένα αντικείμενα, δηλ. ένα αρχείο που περιλαμβάνει άλλα αντικείμενα (π.χ. αρχεία εικόνων) τα οποία είναι αποθηκευμένα σε έναν ή περισσότερους servers
Ακολουθήστε τα παρακάτω βήματα
>Ξεκινάμε τον browser αφού βεβαιωθούμε οτι η cache του ειναι άδεια
>Ξεκινάμε τον packet sniffer
>Εισάγουμε τον ακόλουθο URL στον browser
http://gaia.cs.umass.edu/wireshark-labs/HTTP-wireshark-file4.html.
Στον browser θα πρέπει να εμφανιστεί ένα μικρό αρχείο HTML με δυο εικόνες. Αυτό σημαίνει οτι το αρχείο HTML δεν περιέχει τις ίδιες τις εικόνες αλλά τα URL για τις εικόνες. Ο browser θα πρέπει να ανακτήσει αυτά τα λογότυπα απο τους υποδεικνυόμενους ιστιότοπους. Το λογότυπο του εκδότη του βιβλίου θα ανακτηθεί από τον ιστιότοπο www.awbc.com. Η εικόνα του εξωφύλλου του βιβλίου είναι αποθηκευμένη στον server manic.cs.umass.edu
>Σταματάμε τη σύλληψη πακέτων απο το wireshark και εισάγουμε ‘http’ στο παράθυρο προδιαγράφων του φίλτρου παρουσιάσης ώστε στο παράθυρο καταλόγου πακέτων να παρουσιάζονται μόνο τα συλλαμβανόμενα μηνύματα http
(ΣΗΜΕΙΩΣΗ-αν δεν είστε σε θέση να τρέξετε το wireshark σε μια ζωντάνη σύνδεση δικτύου, μπορείτε να χρησιμοποιήσετε το trace πακέτων http-ethernal-trace-4 για να απαντήσετε στις παρακάτω ερώτησεις

ΕΡΩΤΗΣΕΙΣ

16.Πόσα μηνύματα αιτήσεων HTTP GET στάλθηκαν από τον browser σας; Σε ποιές διευθύνσεις IP στάλθηκαν αυτές οι αιτήσεις GET;

17.Μπορείτε να διακρίνετε αν ο browser σας φόρτωσε τις δυο εικόνες σειριακά ή αν οι εικόνες φορτώθηκαν παράλληλα από τους δυο ιστιότοπους; Εξηγειστε

4.5 Εξουσιοδότηση στο HTTP

Τελος, ας προσπαθήσουμε να επισκεφθούμε ενα site με κωδικό πρόσβασης και ας εξετάσουμε την ακολουθία μηνυμάτων HTTP που ανταλλάσσονται για ενα site αυτού του είδους. Το url http://gaia.cs.umass.edu/wireshark-labs/protected_pages/HTTP-ethereal-file5.html προστατεύται με κωδικό πρόσβασης. Το όνομα χρήστη (username) είναι 'wireshark-students' και ο κωδικός πρόσβασης (password) είναι 'network'. Για να προσπεράσουμε αυτό το προστατευόμενο με κώδικα πρόσβασης site ακολουθούμε τα παρακάτω βήματα
>Ξεκινάμε το browser μας αφού βεβαιωθούμε οτι η cache του είναι άδεια
>Ξεκινάμε τον packet sniffer
>Εισάγουμε το ακόλουθο url στον browser μας
http://gaia.cs.umass.edu/wireshark-labs/protected_pages/HTTP-wireshark-file5.html
Πληκτολογούμε το όνομα χρήστη και τον κωδικό πρόσβασης που μας ζητούνται
>Σταματάμε τη σύλληψη πακέτων απο το wireshark και εισάγουμε 'http' στο παράθυρο προδιαγράφων του φίλτρου παρουσιάσης ώστε στο παράθυρο καταλόγου πακέτων να παρουσιάζονται μόνο τα συλλαμβανόμενα μηνύματα http
(ΣΗΜΕΙΩΣΗ- αν δεν είστε σε θέση να τρέξετε το wireshark σε μια ζωντάνη σύνδεση δικτύου, μπορείτε να χρησιμοποιήσετε το trace πακέτων http-ethernal-trace-5 για να απαντήσετε στις παρακάτω ερώτησεις

ΕΡΩΤΗΣΕΙΣ

18.Ποιά η απόκριση του server (κωδικός κατάστασης και φράση) στο αρχικό μήνυμα HTTP GET από τον browser σας;

19.Όταν ο browser σας στελνει το μήνυμα HTTP GET για δεύτερη φορά, ποιό νέο πεδίο περιλαμβάνεται στο μήνυμα HTTP GET;

Το όνομα χρήστη (wireshark-students) και ο κωδικός πρόσβασης (network) τα οποία εισάγουμε είναι κωδικόποιημενα στη σειρά χαρακτήρων (d2lyZXNoYXJrLXN0dWRlbnRzOm51dHdvcms=) που ακολουθεί την επικεφαλίδα 'Authorization':Basic' στο μήνυμα HTTP GET του client. Αν και μπορεί να φαίνεται οτι το όνομα του χρήστη και ο κωδικός πρόσβασης είναι κρυπτογραφημένα, στην πραγματικότητα είναι κωδικοποιημένα σε μια μορφή γνωστή ως Base64. Το όνομα και ο κωδικός πρόσβασης δεν είναι κρυπτογραφημένα! Για εξακρίβωση, πηγαίνουμε στο http://www.motobit.com/util/base64-decoder-encoder.asp και εισάγουμε την κωδικοποίηση σε base64 ακολουθία χαρακτήρων d2lyZXNoYXJrLXN0dWRlbnRz και πατάμε αποκωδικοποίηση (decode). Ορίστε! Έχουμε μεταφράσει κώδικα Base64 σε κώδικα ASCII και επομένως θα πρέπει να δούμε και το όνομα χρήστη μας. Για να δούμε και τον κώδικα πρόσβασης, εισάγουμε το υπόλοιπο της σειράς χαρακτήρων Om51dHdvcms= και πατάμε decode. Αφού οποιοσδήποτε μπορεί να φορτώσει ενα εργαλείο όπως το wireshark και να συλλαμβάνει τα πακέτα που διέρχονται απο το δίκτυο του και αφού οποιοσδήποτε μπορεί να μεταφράσει απο Base64 σε ASCII, είναι προφανές οτι οι απλοί κωδικοί πρόσβασης σε sites δεν είναι ασφαλείς εκτός αν ληφθούν πρόσθετα μέτρα. Όπως θα δούμε σε επόμενο κεφάλαιο, υπάρχουν τρόποι να κάνουμε ασφαλέστερη την πρόσβαση στον παγκόσμιο ιστό. Ωστόσο είναι προφανές οτι απαιτούνται πιο προηγμένες μέθοδοι απο τη βασική εξουσιοδότηση του HTTP.

(1)Φορτώστε το αρχείο http://gaia.cs.umass.edu/wireshark-labs/wireshark-traces.zip και εξάγετε το αρχείο http-wireshark-trace-1.Τα traces που περιέχονται σε αυτό το αρχείο zip συλλέχθηκαν απότο wireshark ενώ εκτελούνταν τα παραπάνω βήματα στον υπολόγιστη του συγγραφέα. Αφού λάβουμε το trace,το φορτώνουμε στο wireshark,και απο το μενού file επιλέγουμε open και στη συνέχεια επιλλέγουμε το αρχείο http://gaia.cs.umass.edu/wireshark-labs/wireshark-traces.zip.Το παράθυρο του wireshark πρέπει να ναι το ίδιο με το σχ.27

ΑΠΑΝΤΗΣΕΙΣ

1.Η Βασική Αλληλεπίδραση GET/Απόκριση στο HTTP

```
No.     Time        Source          Destination     Protocol Info
4       0.048291    192.168.1.46    128.119.245.12  HTTP    GET /wireshark-
labs/HTTP-wireshark-file1.html HTTP/1.1

Hypertext Transfer Protocol
    GET /wireshark-labs/HTTP-wireshark-file1.html HTTP/1.1\r\n
    Host: gaia.cs.umass.edu\r\n
    User-Agent: Mozilla/5.0 (Windows; U; Windows NT 5.1; en-US; rv:1.8.1.4)
Gecko/20070515 Firefox/2.0.0.4\r\n
    Accept:
text/xml,application/xml,application/xhtml+xml,text/html;q=0.9,text/plain;q=0.8
,image/png,*/*;q=0.5\r\n
    Accept-Language: en-us,en;q=0.5\r\n
    Accept-Encoding: gzip,deflate\r\n
    Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.7\r\n
    Keep-Alive: 300\r\n
    Connection: keep-alive\r\n
    Cookie: MintUnique=1;
__utmz=198765611.1176212581.8.2.utmccn=(referral)|utmcsr=cs.umass.edu|utmcct=/c
sinfo/news.html|utmcmd=referral;
__utma=198765611.821901841.1145892528.1176212581.1179945703.9;
__utma=267820956.1666738513.1163587262.118
    If-Modified-Since: Thu, 07 Jun 2007 22:07:01 GMT\r\n
    If-None-Match: "d6c95-7e-224fab40"\r\n
    \r\n


No.     Time        Source          Destination  Protocol Info
6       0.155044    128.119.245.12 192.168.1.46 HTTP     HTTP/1.1 200 OK
(text/html)

Hypertext Transfer Protocol
    HTTP/1.1 200 OK\r\n
    Date: Thu, 07 Jun 2007 22:09:08 GMT\r\n
    Server: Apache/2.0.52 (CentOS)\r\n
    Last-Modified: Thu, 07 Jun 2007 22:09:01 GMT\r\n
    ETag: "d6c95-7e-2976b940"\r\n
    Accept-Ranges: bytes\r\n
    Content-Length: 126
    Keep-Alive: timeout=10, max=100\r\n
    Connection: Keep-Alive\r\n
    Content-Type: text/html; charset=ISO-8859-1\r\n
    \r\n
Line-based text data: text/html
    <html>\n
    Congratulations.  You've downloaded the file \n
    http://gaia.cs.umass.edu/ethereal-labs/HTTP-ethereal-file1.html!\n
    </html>\n
```

1.Και οι δυο τρέχουν την HTTP 1,1 έκδοση

2.Αγγλικά

3.Η IP διεύθυνση μου είναι `192.168.1.46` και του server είναι `128.119.245.12`

4. HTTP/1.1 200 OK (text/html)

5. Τροποποιήθηκε για τελευταία φορά Thu,07 Jun 2007 22:09:01 GMT

6.126

7 .Όχι,εμφανίζονται όλες οι επικεφαλίδες

2.Η Αλληλεπίδραση υπό συνθήκη GET/απόκριση στο HTTP

```
No.     Time        Source                Destination           Protocol Info
      4 0.046887    192.168.1.46          128.119.245.12        HTTP     GET
/wireshark-labs/HTTP-wireshark-file2.html HTTP/1.1

Hypertext Transfer Protocol
    GET /wireshark-labs/HTTP-wireshark-file2.html HTTP/1.1\r\n
        Request Method: GET
        Request URI: /wireshark-labs/HTTP-wireshark-file2.html
        Request Version: HTTP/1.1
    Host: gaia.cs.umass.edu\r\n
    User-Agent: Mozilla/5.0 (Windows; U; Windows NT 5.1; en-US; rv:1.8.1.4)
Gecko/20070515 Firefox/2.0.0.4\r\n
    Accept:
text/xml,application/xml,application/xhtml+xml,text/html;q=0.9,text/plain;q=0.8
,image/png,*/*;q=0.5\r\n
    Accept-Language: en-us,en;q=0.5\r\n
    Accept-Encoding: gzip,deflate\r\n
    Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.7\r\n
    Keep-Alive: 300\r\n
    Connection: keep-alive\r\n
    Cookie:
__utmz=198765611.1176212581.8.2.utmccn=(referral)|utmcsr=cs.umass.edu|utmcct=/c
sinfo/news.html|utmcmd=referral;
__utma=198765611.821901841.1145892528.1176212581.1179945703.9;
__utmz=267820956.1180131637.4.1.utmccn=(organic)|utmcsr=
    \r\n
```

```
No.     Time        Source                Destination           Protocol Info
      7 0.151515    128.119.245.12        192.168.1.46          HTTP
HTTP/1.1 200 OK (text/html)

Hypertext Transfer Protocol
    HTTP/1.1 200 OK\r\n
        Request Version: HTTP/1.1
        Response Code: 200
    Date: Thu, 07 Jun 2007 16:29:06 GMT\r\n
    Server: Apache/2.0.52 (CentOS)\r\n
    Last-Modified: Thu, 07 Jun 2007 16:29:01 GMT\r\n
    ETag: "d6c96-173-69876d40"\r\n
    Accept-Ranges: bytes\r\n
    Content-Length: 371
    Keep-Alive: timeout=10, max=100\r\n
    Connection: Keep-Alive\r\n
    Content-Type: text/html; charset=ISO-8859-1\r\n
    \r\n
Line-based text data: text/html
    \n
    <html>\n
    \n
    Congratulations again!  Now you've downloaded the file lab2-2.html. <br>\n
    This file's last modification date will not change.  <p>\n
    Thus  if you download this multiple times on your browser, a complete copy
<br>\n
    will only be sent once by the server due to the inclusion of the IN-
MODIFIED-SINCE<br>\n
    field in your browser's HTTP GET request to the server.\n
    \n
    </html>\n
```

```
No.     Time        Source                Destination           Protocol Info
     12 2.932093    192.168.1.46          128.119.245.12        HTTP     GET
/wireshark-labs/HTTP-wireshark-file2.html HTTP/1.1

Hypertext Transfer Protocol
    GET /wireshark-labs/HTTP-wireshark-file2.html HTTP/1.1\r\n
        Request Method: GET
        Request URI: /wireshark-labs/HTTP-wireshark-file2.html
        Request Version: HTTP/1.1
    Host: gaia.cs.umass.edu\r\n
    User-Agent: Mozilla/5.0 (Windows; U; Windows NT 5.1; en-US; rv:1.8.1.4)
Gecko/20070515 Firefox/2.0.0.4\r\n
    Accept:
text/xml,application/xml,application/xhtml+xml,text/html;q=0.9,text/plain;q=0.8
,image/png,*/*;q=0.5\r\n
    Accept-Language: en-us,en;q=0.5\r\n
    Accept-Encoding: gzip,deflate\r\n
    Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.7\r\n
    Keep-Alive: 300\r\n
    Connection: keep-alive\r\n
    Cookie:
__utmz=198765611.1176212581.8.2.utmccn=(referral)|utmcsr=cs.umass.edu|utmcct=/c
sinfo/news.html|utmcmd=referral;
__utma=198765611.821901841.1145892528.1176212581.1179945703.9;
__utmz=267820956.1180131637.4.1.utmccn=(organic)|utmcsr=
    If-Modified-Since: Thu, 07 Jun 2007 16:29:01 GMT\r\n
    If-None-Match: "d6c96-173-69876d40"\r\n
    Cache-Control: max-age=0\r\n
    \r\n
```

```
No.     Time        Source                Destination           Protocol Info
     13 3.030398    128.119.245.12        192.168.1.46          HTTP
HTTP/1.1 304 Not Modified

Hypertext Transfer Protocol
    HTTP/1.1 304 Not Modified\r\n
        Request Version: HTTP/1.1
        Response Code: 304
    Date: Thu, 07 Jun 2007 16:29:09 GMT\r\n
    Server: Apache/2.0.52 (CentOS)\r\n
    Connection: Keep-Alive\r\n
    Keep-Alive: timeout=10, max=98\r\n
    ETag: "d6c96-173-69876d40"\r\n
    \r\n
```

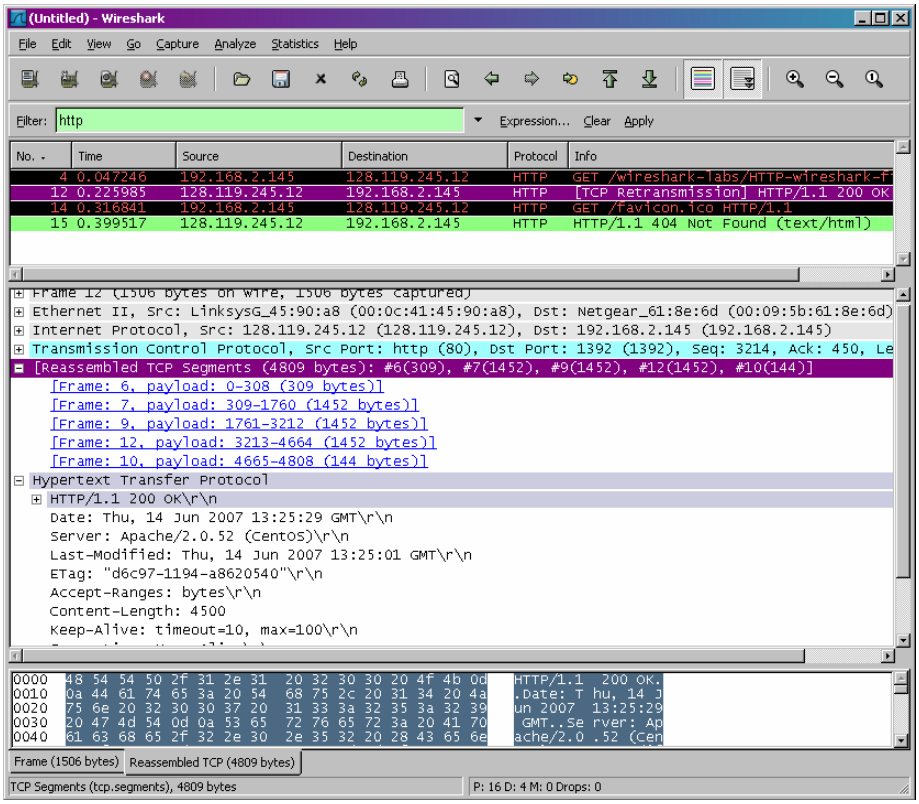
8.Όχι

9.Ναι, γιατι μπορούμε να δούμε τα περιεχόμενα στο Line-based text data πεδίο

10. Ναι, η πληροφορία που ακολουθεί είναι Thu,07 Jun 2007 16:29:01 GMT που είναι η ημερομηνία της τελευταίας τροποποίησης του αρχείου από την προηγούμενη αίτηση GET

11.Ο κώδικας και η φράση κατάστασης που επιστρέφει ο server είναι HTTP/1.1 304 Not Modified (μη τροποποιημένα).Ο server δεν επιστρέφει τα περιεχόμενα του αρχείου.

3.Ανάκτηση μεγάλων εγγράφων



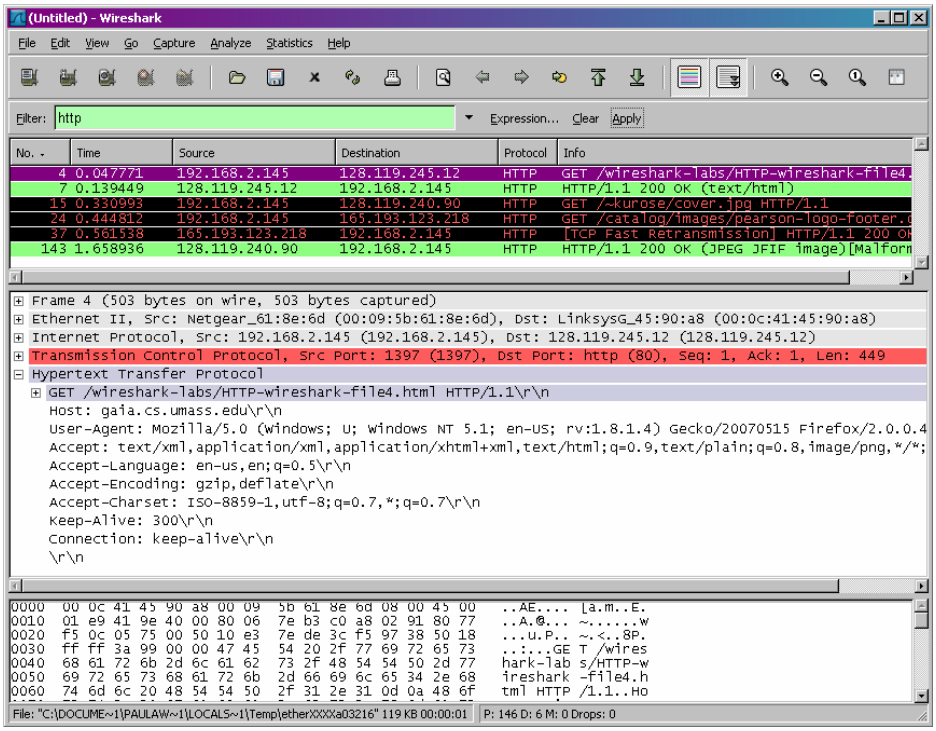
12.Στάλθηκε 1 μήνυμα όπως φένεται στην παραπάνω εικόνα

13. Χρειάστηκαν 5.Το μέγεθος τους είναι 309 ,1452 ,1452, 1452 και 144 bytes αντίστοιχα. Σύνολο 4500 bytes

14. 200 OK

15.Όχι

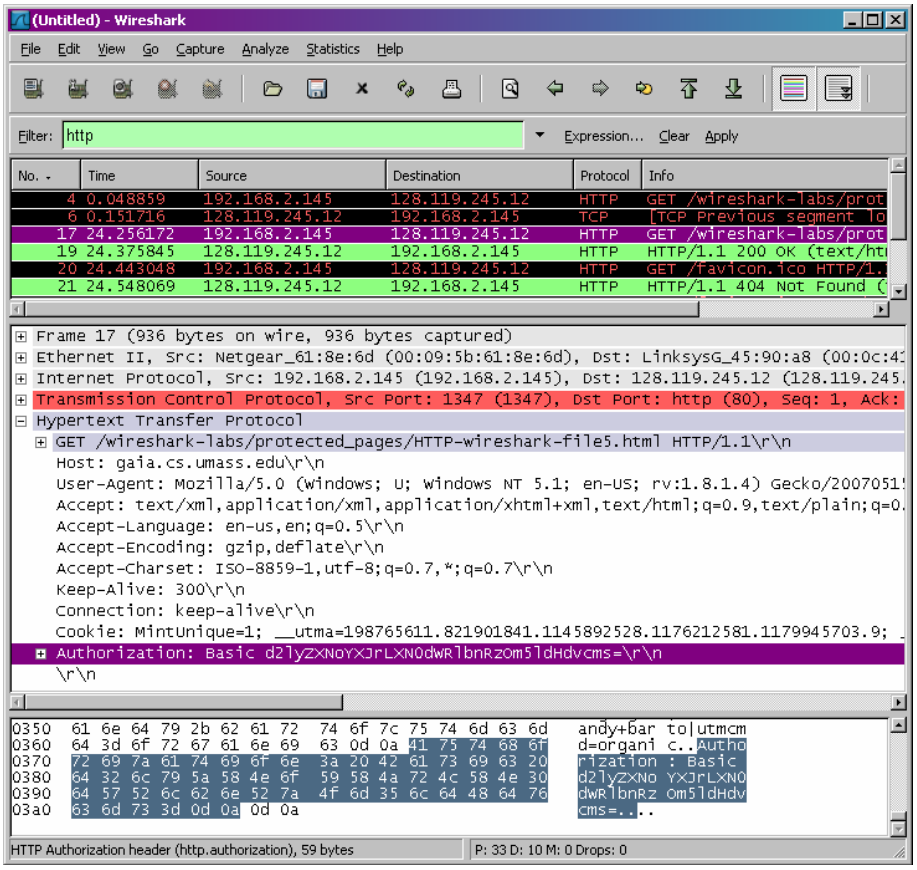
4.Έγγραφα HTML με ενσωματωμένα αντικείμενα



16.Όπως βλέπουμε στο παραπάνω screenshot στάλθηκαν 3 HTTP GET αιτήσεις στις ακόλουθες διευθύνσεις
α. 128.119.245.12
β. 128.119.240.90
γ. 165.193.123.218

17.Βλέποντας στο TCP port μπορούμε να διακρίνουμε πως φορτώθηκαν. Και οι 2 φορτώθηκαν σειριακά.

5.Εξουσιοδότηση στο HTTP



18.Κώδικας κατάστασης 401
Φράση Authorization Required

19.Όπως βλέπουμε στο screenshot το νέο πεδίο είναι το Authorization.
Authorization: Basic d2lyZXNoYXJrLXN0dWRlbnRzOm5ldHdvcms=\r\n

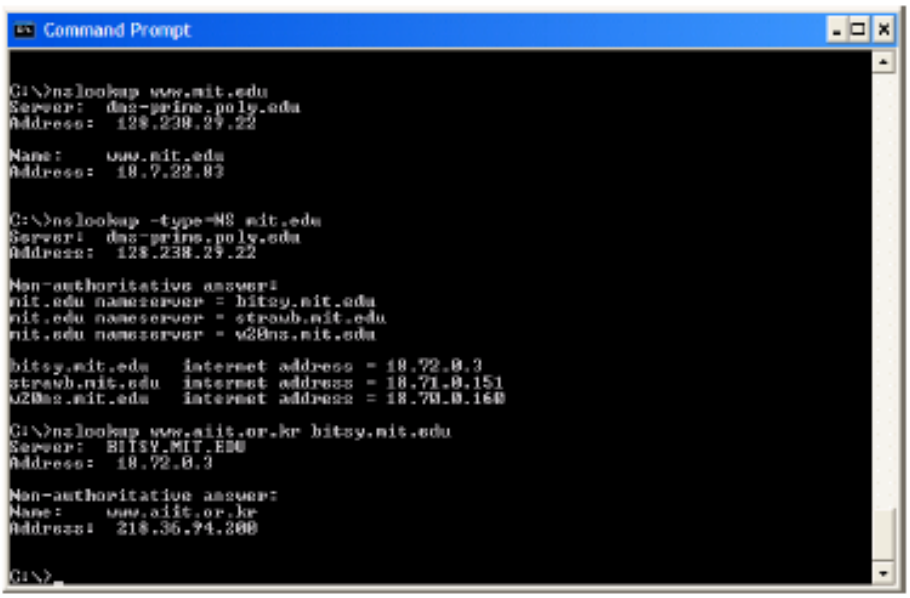
Κεφάλαιο 5
Ασκήσεις επίδειξης της λειτουργίας του DNS

Κάθε μηχάνημα - host του Διαδικτύου μπορεί να αναγνωριστεί από την IP διεύθυνσή του, που είναι ένας δυαδικός αριθμός των 32 bits. Το ίδιο το δίκτυο καταλαβαίνει μόνο τις IP διευθύνσεις. Οι άνθρωποι, όμως, μπορούν πιο εύκολα να θυμούνται ονόματα και όχι δυαδικά νούμερα. Για την αντιστοίχιση μεταξύ των δυαδικών διευθύνσεων και των διευθύνσεων σε μορφή ASCII χαρακτήρων χρησιμοποιείται το πρωτόκολλο του στρώματος εφαρμογών, το DNS (Domain Name System). Πιο απλά, το DNS μεταφράζει ονόματα τερματικών συστημάτων (hostnames) σε διευθύνσεις IP, παίζοντας έτσι ένα σημαντικό ρόλο στην υποδομή του Διαδικτύου.

Στο εργαστήριο αυτό θα εξετάσουμε την πλευρά του πελάτη (client) στο DNS. Υπενθυμίζεται οτι ο ρόλος του client στο DNS είναι σχετικά απλός- ο client στέλνει ένα ερώτημα (query) στον τοπικό DNS server από τον οποίο λαμβάνει μια απόκριση (response). Πολλά από τα μηνύματα που ανταλλάσσονται, καθώς οι ιεραερχημένοι DNS servers επικοινώνουν μεταξύ τους είτε αναδρομικά (recursively) είτε επαναληπτικά (iteratively) για να απαντήσουν στο ερώτημα DNS του client, δεν γίνονται αντιληπτά από τον DNS client. Από τη σκοπιά του DNS client το πρωτόκολλο είναι πολύ απλό – διατυπώνεται ένα ερώτημα στον τοπικό DNS server από τον οποίο λαμβάνεται μια απόκριση.

5.1 Το Εργαλείο Nslookup

Στο εργαστήριο αυτό θα χρησιμοποιήσουμε εκτενώς το εργαλείο nslookup το οποίο ειναι διαθέσιμο στα περισσότερα συστήματα Linux/Unix και Microsoft. Για να τρέξουμε το nslookup σε περιβάλλον Linux/Unix, πληκτρολογούμε την εντολή nslookup στη γραμμή εντολών. Για να το τρέξουμε σε περιβάλλον Windows, ανοίγουμε το παράθυρο Command Prompt και τρέχουμε το nslookup στη γραμμή εντολών.
Στην απλούστερη λειτουργία του το nslookup επιτρέπει στο τερματικό σύστημα στο οποίο τρέχει να στείλει ένα ερώτημα DNS για μια εγγραφή DNS σε οποιοδήποτε καθορισμένο DNS server. Ο DNS server στον οποίο στέλνεται το ερώτημα μπορεί να είναι ένας root DNS server (εξυπηρετητής κορυφής), ένας top-level-domain DNS server (εξυπηρετητής ανώτατου επιπέδου), ένας authoritative DNS server (επίσημος εξυπηρετητής) ή ένας ενδιάμεσος DNS server. Για να γίνει αυτό, το nslookup στέλνει ένα ερώτημα DNS στον καθορισμένο DNS server, λαμβάνει μια απόκριση DNS απο τον ίδιο DNS server και απεικονίζει το αποτέλεσμα.



Σχ.28 Screenshot με αποτελέσματα τριών ανεξάρτητων εντολών nslookup

Το παραπάνω screenshot δείχνει τα αποτελέσματα τριών ανεξάρτητων εντολών nslookup όπως απεικονίζονται στο παράθυρο Command Prompt των Windows. Στο παράδειγμα αυτό ο client βρίσκεται στην πανεπιστημιούπολη του Polytechnic University των ΗΠΑ και ο τοπικός του DNS server έχει το όνομα dnw-prime.poly.edu. Έαν δεν έχει καθοριστεί ο DNS server, το nslookup στέλνει το ερώτημα στον προεπιλεγμένο DNS server που στην προκειμένη περίπτωση είναι ο dns-prime.poly.edu. Θεωρούμε την πρώτη εντολή

Nslookup www.mit.edu

Με την παραπάνω εντολή ο DNS client ζητά να του αποσταλεί η διεύθυνση IP του host www.mit.edu. Όπως φαίνεται στο screenshot, η απόκριση στην εντολή αυτή περιλαμβάνει δύο κομμάτια πληροφορίας (α) το όνομα και τη διεύθυνση IP του DNS server που παρέχει την απάντηση, και (β) την ίδια απάντηση που αποτελείται από το όνομα και τη διεύθυνση IP του www.mit.edu. Αν και η απόκριση προήλθε από τον τοπικό DNS server του Polytechnic University, είναι αρκετά πιθανό ο τοπικός αυτός DNS server να έχει επικοινωνήσει με τρόπο επαναληπτικό με αρκετούς DNS προκειμένου να λάβει την απάντηση

Θεωρούμε τώρα τη δεύτερη εντολή
Nslookup -type=NS mit.edu

Στο παράδειγμα αυτό παρέχουμε την προαιρετική επιλογή '-type=NS' και την διεύθυνση 'mit.edu'. Αυτό έχει ως αποτέλεσμα το nslookup να στείλει ένα ερώτημα για μια εγγραφή τυπου NS στον προεπιλεγμένο τοπικό DNS server. Με το ερώτημα αυτό ο DNS client ζητά να του αποσταλούν τα ονόματα των authoritative DNS servers για το domain name 'mit.edu'. (Όταν δεν χρησιμοποιείται η επιλογή –type, το nslookup χρησιμοποιεί εκ προεπιλογής type=A). Στην απάντηση, η οποία απεικονίζεται στο παραπάνω screnshot, υποδεικνύεται ο DNS server που την παρείχε (δηλαδή ο προεπιλεγμένος τοπικός DNS server) καθώς και τα ονόματα τριών nameservers στο MIT. Καθένας από τους τρείς αυτούς servers είναι ένας authoritative

DNS server για τους hosts της πανεπιστημιούπολης του MIT. Το nslookup υποδεικνύει επίσης ότι η απάντηση είναι ‘non-authoritative’, δηλαδή οτι προήλθε από την cache κάποιου server και όχι από ένα authoritative DNS server του MIT. Τέλος, η απάντηση περιλαμβάνει επίσης τις διευθύνσεις IP των authoritative DNS servers στο MIT. (Αν και το ερώτημα τύπου NS που έστειλε το nslookup δεν ζητούσε με ρήτο τρόπο τις διευθύνσεις IP, ο τοπικός DNS server παρείχε ‘δωρεάν’ αυτή την πληροφορία την οποία απεικόνισε το nslookup)

Θεωρείστε, τέλος, την τρίτη εντολή
Nslookup www.aitt.or.kr bitsy.mit.edu

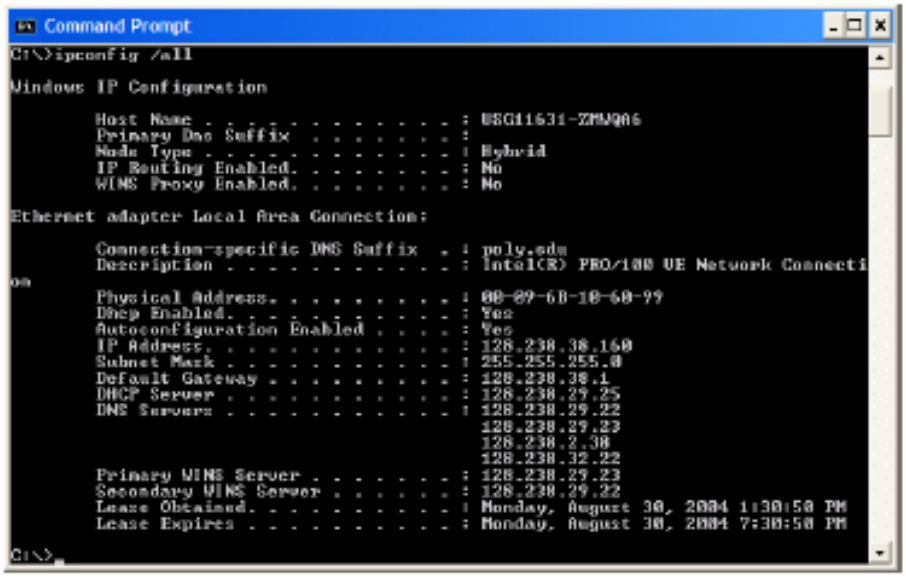
Στο παράδειγμα αυτό υποδεικνύουμε οτι θέλουμε το ερώτημα να σταλεί στον DNS server bitsy.mit.edu αντί του τοπικού DNS server (dns-prime.poly.edu). Έτσι η ανταλλαγή ερωτήματος και απόκρισης γίνεται απευθείας μεταξύ του host που στέλνει το ερώτημα και του server bitsy.mit.edu. Ο DNS server bitsy.mit.edu παρέχει τη διεύθυνση IP του host www.aiit.or.kr, ενός web server στο Advanced Institute of Information Technology της Κορέας.

Μετά από τα παραπάνω ενδεικτικά παραδείγματα ακολουθεί η γενική σύνταξη των εντολών nslookup που έχει ως εξής
nslookup –option1 –option2 host-to-find dns-server

Το nslookup μπορεί να εκτελεσθεί χωρίς καμία, με μία, δύο ή περισσότερες προαιρετικές επιλογές. Όπως φαίνεται από τα παραπάνω παραδείγματα, το όνομα του DNS server είναι επίσης προαιρετικό- εάν δεν το παρέχουμε, το ερώτημα στέλνεται στον προεπιλεγμένο τοπικό DNS server. Μετά από αυτή την ανασκόπηση του nslookup, ας πειραματισθούμε εκτελώντας τα ακόλουθα
1.Τρέχουμε το nslookup ώστε να αποκτήσετε τη διεύθυνση IP ενός web server που βρίσκεται στην Ασία
2.Τρέχουμε το nslookup ώστε να προσδιορίσετε τους authoritative DNS servers για κάποιο πανεπιστήμιο της Ευρώπης
3.Τρέχουμε το nslookup ώστε ένας από τους DNS servers της απάντησης στην ερώτηση 2 να ερωτηθεί σχετικά με τους mail servers του Yahoo!mail.

5.2 Το Εργαλείο Ipconfig

Το ipconfig στα (Windows) και το ifconfig (στα Linux/Unix) είναι από τα πιο χρήσιμα εργαλεία στο τερματικό μας σύστημα. Θα περιγράψουμε μόνο το ipconfig, αν και το ifconfig σε Linux/Unix είναι παρόμοιο. Το ipconfig μπορεί να χρησιμοποιηθεί για να δείξει πληροφορίες σχετικές με το TCP/IP που τρέχει στο τερματικό μας σύστημα, π.χ. διεύθυνση IP τερματικού συστήματος, διεύθυνση IP DNS server, είδος adapter κ.α. Μπορούμε να δούμε όλες αυτές τις πληροφορίες εισάγοντας την εντολή ipconfig/all στο παράθυρο Command Prompt όπως φαίνεται στο ακόλουθο screenshot



Σχ.29 Screenshot για την εντολή ipconfig/all

Το ipconfig είναι επίσης πολύ χρήσιμο για πληροφοριές DNS που είναι αποθηκευμένες στο τερματικό μας σύστημα. Ένας host μπορεί να αποθηκεύσει έγγραφες DNS που έχουν αποκτηθεί πρόσφατα. Για να δούμε αυτές τις εγγραφές που είναι στην προσωρινή μνήμη (cache), εισάγουμε την ακόλουθη εντολή στο παράθυρο Command Prompt

ipconfig/displaydns

Σε κάθε καταχώρηση φαίνεται ο εναπομένων χρόνος ζωής (time to live-TTL) σε δευτερόλεπτα. Για να αδειάσουμε την cache, εισάγουμε στο Command Prompt ipconfig/flushdns
Με την εντολή αυτή διαγράφονται όλες οι καταχωρήσεις από το αρχείο hosts του τερματικού συστήματος

5.3 Παρακολούθηση του DNS με το Wireshark

Μετά την εξοικείωση με το nslookup και το ipconfig, θα χρησιμοποιήσουμε το wireshark για να συλλάβουμε τα πακέτα DNS που δημιουργούνται κατά τη συνηθισμένη δραστηριότητα πλοήγησης του Παγκόσμιου Ιστου.
>Χρησιμοποιούμε το ipconfig για να αδειάσουμε την DNS cache του host μας
>Ανοίγουμε το web browser μας και αδειάζουμε την cache του. (Στην περίπτωση του Internet Explorer, πηγαίνουμε στο μενού Tools και επιλέγουμε Internet Options, στη συνέχεια επιλέγουμε Delete Files στην καρτέλα General)
>Ανοίγουμε το Wireshark και εισάγουμε 'ip.addr == διεύθυνση_IP_host' στο φίλτρο. Χρησιμοποιούμε το ipconfig για να βρούμε τη διεύθυνση IP του host μας (διεύθυνση _IP_host). Το φίλτρο αυτό απομακρύνει όλα τα πακέτα που δεν προέρχονται από και δεν προορίζονται για τον host μας.
>Ξεκινάμε τη σύλληψη πακέτων από το Wireshark
>Με τη βοήθεια του browser, πάμε στην ιστιοσελιδα http//www.ietf.org

>Σταματάμε τη σύλληψη πακέτων

Εάν δεν είστε σε θέση να τρέξετε το Wireshark σε μια ζωντανή σύνδεση δικτύου, μπορείτε να φορτώσετε ένα trace πακέτων το οποίο δημιουργηθηκε ακολουθώντας τα παραπάνω βήματα (1)

ΕΡΩΤΗΣΕΙΣ

4.Εντοπίστε τα μηνύματα ερωτημάτων (query) και αποκρίσεων (response) του DNS. Ποιό πρωτόκολλο μεταφοράς χρησιμοποιείται για τη μεταφορά τους, UDP ή TCP;

5.Ποιά η θύρα προορισμού (destination port) του μηνύματος ερωτήσης; Ποιά η θύρα πηγής (source port) του μηνύματος απόκρισης;

6.Σε ποιά διεύθυνση IP στάλθηκε το μήνυμα ερωτήματος; Χρησιμοποιήστε το ipconfig για να προσδιορίσετε τη διεύθυνση IP του τοπικού σας DNS server. Τι σχέση έχουν μεταξύ τους οι δυο διευθύνσεις IP;

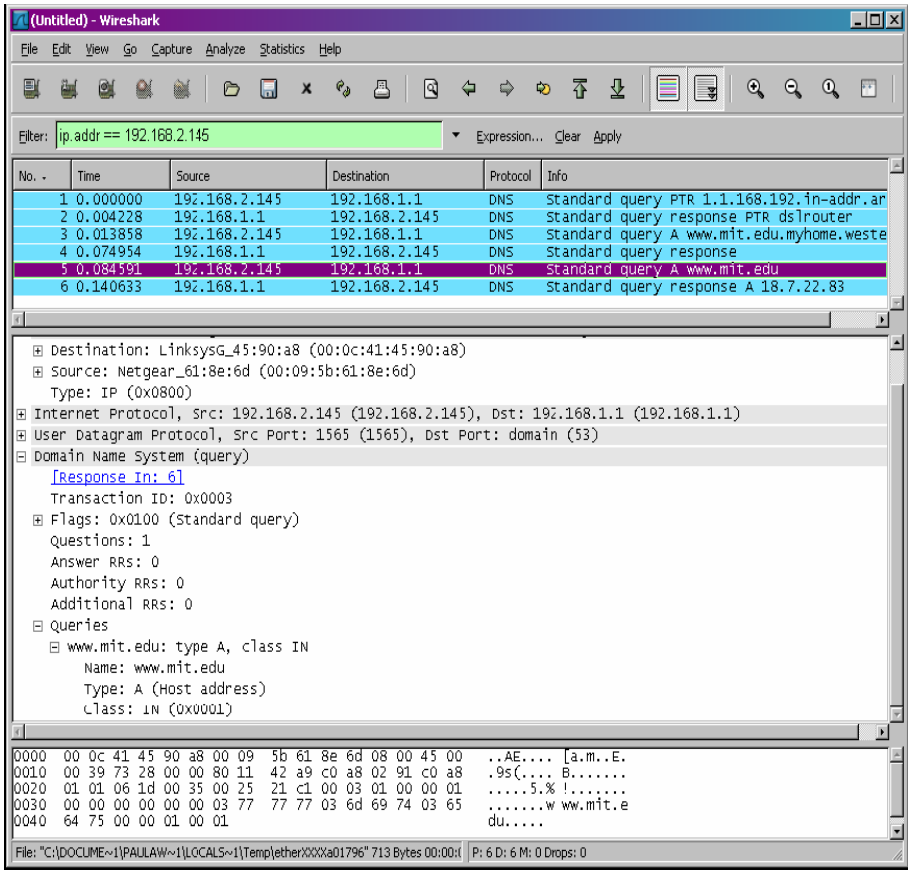
7.Εξετάστε το μήνυμα ερωτήσης. Ποιό το ‘είδος’ (‘Type’) της ερώτησης; Περιέχονται ‘απαντήσεις’(‘answers’) στο μήνυμα ερώτησης;

8.Εξετάστε το μήνυμα απόκρισης. Πόσες ‘απαντήσεις’ περιέχονται στο μήνυμα αυτό; Τι περιέχει καθεμιά απο τις απαντήσεις αυτές;

9.Θεωρείστε το επακόλουθο πακέτο SYN (SYN packet ή SYN segment) που στέλνει το TCP που τρέχει στον host σας. Η διεύθυνση προορισμού αυτού του πακέτου αντιστοιχεί σε καμιά από τις διευθύνσεις IP που παρέχονται στο μήνυμα απόκρισης του DNS;

10.Η ιστιοσελίδα http//www.ietf.org περιέχει εικόνες. Χρειάζεται ο host σας να στείλει νέα ερωτήματα DNS πρίν από την ανάκτηση κάθε εικόνας;

Ας ασχοληθούμε τώρα με το nslookup (2)
>Ξεκινάμε τη σύλληψη πακέτων
>Τρέχουμε το nslookup για το όνομα host www.mit.edu
>Σταματάμε τη σύλληψη πακέτων
Η ακολουθία των πακέτων (trace) που απεικονίζεται στο wireshark θα πρέπει να μοιάζει με αυτήν που φαίνεται στο παρακάτω screenshot



Σχ.30 Screenshot με την ακολουθία των πακέτων για την εντολή nslookup

Όπως φαίνεται, το nslookup έστειλε στην πραγματικότητα τρία ερωτήματα DNS και έλαβε τρεις αποκρίσεις αντίστοιχα. Για να απαντήσετε στις παρακάτω ερωτήσεις αγνοείστε τα δυο πρώτα ζεύγη ερωτημάτων/αποκρίσεων καθώς αφορούν το nslookup και κατα κανόνα δεν δημιουργούνται από συνήθεις διαδικτυακές εφαρμογές. Επικεντρωθείτε στο τελευταίο ερώτημα και την τελευταία απόκριση DNS

ΕΡΩΤΗΣΕΙΣ

11.Ποιά η θύρα προορισμού (destination port) του μηνύματος ερωτήματος; Ποιά η θύρα πηγής (source port) του μηνύματος απόκρισης;

12.Σε ποια διεύθυνση IP στάλθηκε το μήνυμα ερωτήματος; Πρόκειται για τη διεύθυνση IP του τοπικού σας DNS server;

13.Εξετάστε το μήνυμα ερωτήματος. Ποιό το ‘είδος’ (‘Type’) του ερωτήματος; Περιέχονται ‘απαντήσεις’(‘answers’) στο μήνυμα ερώτησης;

14.Εξετάστε το μήνυμα απόκρισης. Πόσες ‘απαντήσεις’ περιέχονται στο μήνυμα αυτό; Τι περιέχει καθεμιά από τις απαντήσεις αυτές;

15.Παρέχετε ένα screenshot

Επαναλάβετε το προηγούμενο πείραμα για την εντολή Nslookup –type=NS mit.edu

Απαντήστε στις ακόλουθες ερωτήσεις (3)

16.Σε ποιά διεύθυνση IP στάλθηκε το μήνυμα ερωτήματος; Πρόκειται για τη διεύθυνση IP του τοπικού σας DNS server ;

17.Εξετάστε το μήνυμα ερωτήματος. Ποιό το 'είδος' ('type') του ερωτήματος; Περιέχονται 'απαντήσεις'('answers') στο μήνυμα ερωτήματος;

18.Εξετάστε το μήνυμα απόκρισης. Ποιά ονόματα nameservers του MIT παρέχονται στο μήνυμα απόκρισης; Στο μήνυμα απόκρισης παρέχονται και οι διευθύνσεις IP των nameservers του MIT;

19.Παρέχετε ένα screenshot

Επαναλάβετε το προηγούμενο πείραμα για την εντολή Nslookup www.aiit.or.kr διευθυνση_ip_του_bitsy.mit.edu
Απαντήστε στις ακόλουθες ερωτήσεις (4)

20.Σε ποια διεύθυνση IP στάλθηκε το μήνυμα ερωτήματος; Πρόκειται για τη διεύθυνση IP του τοπικού σας DNS server; Εάν όχι, σε τι αντιστοιχεί η συγκεκριμένη διεύθυνση IP;

21.Εξετάστε το μήνυμα ερωτήματος. Ποιο το 'είδος' ('Type') του ερωτήματος; Περιέχονται 'απαντήσεις' ('answers') στο μήνυμα ερωτήματος;

22.Εξετάστε το μήνυμα απόκρισης. Πόσες 'απαντήσεις' περιέχονται στο μήνυμα αυτό; Τι περιέχει καθεμία από τις απαντήσεις αυτές;

23.Παρέχετε ένα screenshot.

24.Εάν εκτελέσετε την εντολή nslookup www.aiit.or.kr αφού αδειάσετε την DNS cache , θα περάσετε από τους ίδιους nameservers για να λάβετε την απάντηση;

(1) φορώστε το αρχείο http://gaia.cs.umass.edu/wireshark-labs/wireshark-traces.zip και εξάγετε το αρχείο dns-ethereal-trace-1.τα traces που περιέχονται σε αυτό το αρχείο zip συλλέχθηκαν από το Wireshark ενώ εκτελούνταν τα βήματα που περιγράφονται στο εργαστήριο Wireshark για το DNS στον υπολογιστή του συγγραφέα.Αφου λάβετε το trace,μπορείτε να το φορτώσετε στο Wieshark και να το δείτε στο παράθυρο χρησιμοποιώντας το μενού File,επιλέγοντας Open και στη συνέχεια επιλέγοντας το αρχείο dns-ethereal-trace-1 του trace

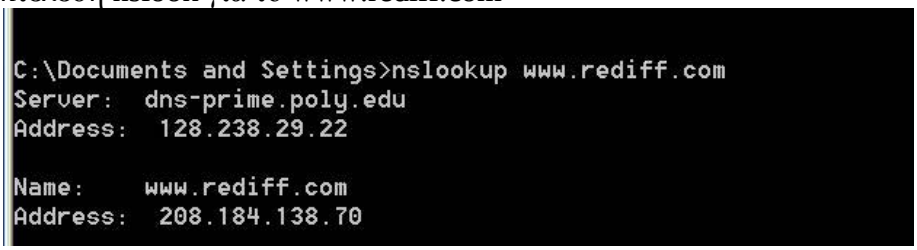
(2) εάν δεν είστε σε θέση να τρέξετε το wireshark μπορείτε να χρησιμοποιήσετε το trace πακέτων dns-ethereal-trace-2

(3) εάν δεν είστε σε θέση να τρέξετε το wireshark μπορείτε να χρησιμοποιήσετε το trace πακέτων dns-ethereal-trace-3

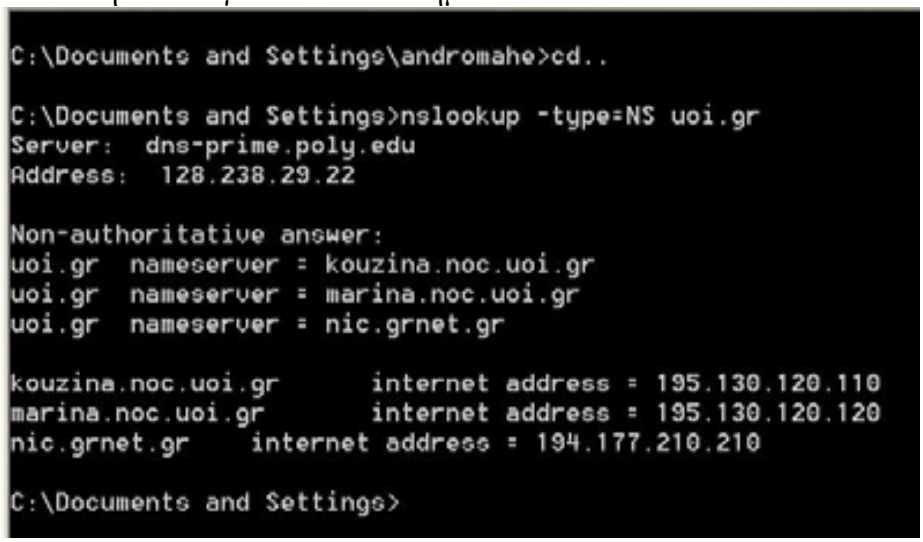
(4)εάν δεν είστε σε θέση να τρέξετε το wireshark μπορείτε να χρησιμοποιήσετε το trace πακέτων dns-ethereal-trace-4

ΑΠΑΝΤΗΣΕΙΣ

1.Η εκτέλεση nslook για το www.rediff.com

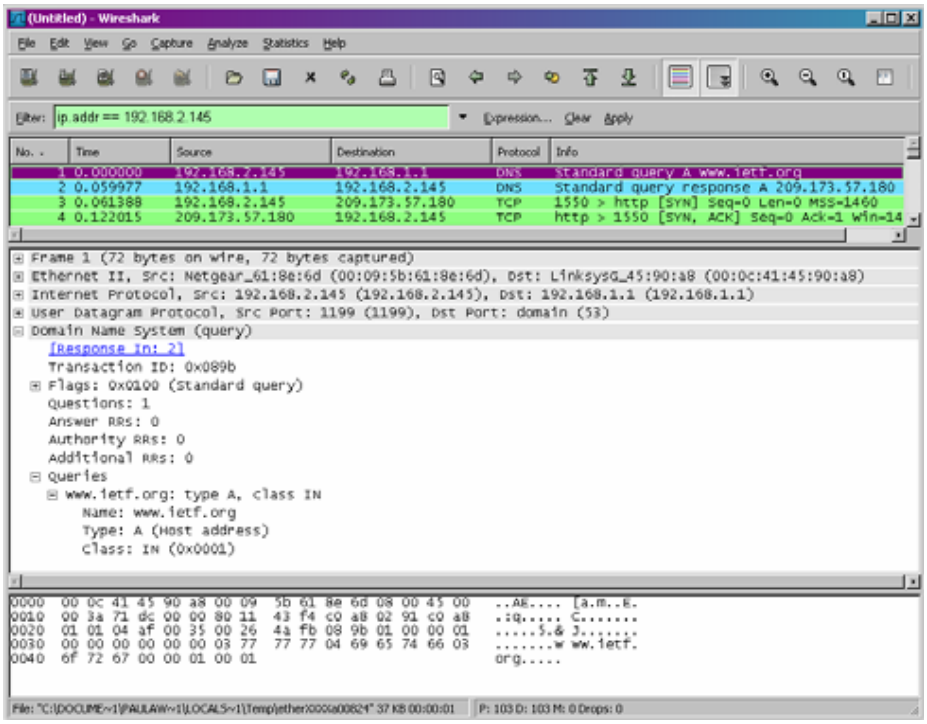


2. Η εκτέλεση nslook για το Πανεπιστήμιο Ιωαννίνων

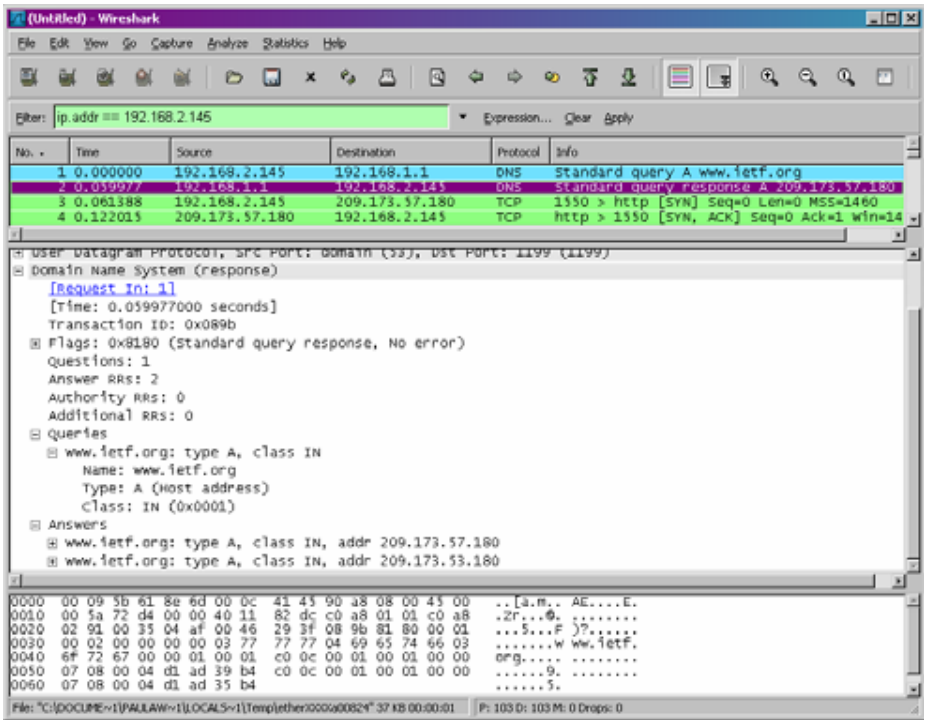


3.

4.



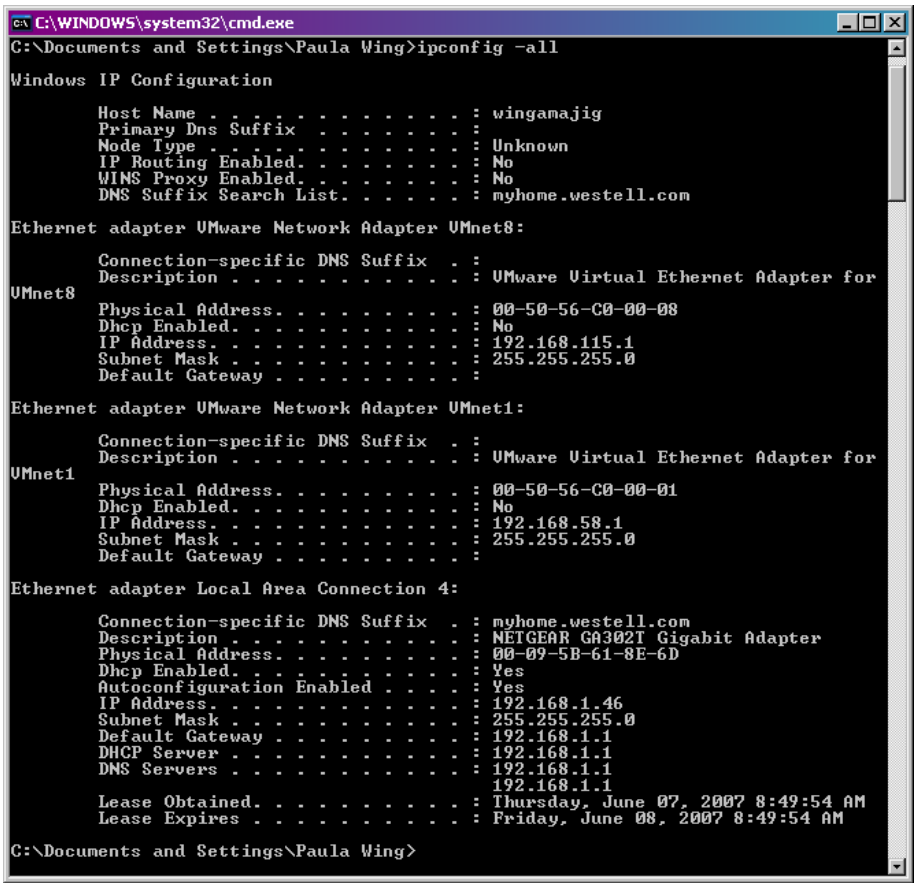
Screenshot για τα DNS μηνύματα ερωτημάτων



Screenshot για τα DNS μυνήματα αποκρίσεων

Για τη μεταφορά τους χρησιμοποιήθηκε το UDP

5.



Screenshot για το ipconfig

Η θύρα προορισμού του μηνύματος ερωτήσης είναι 53 και η θύρα πηγής του μηνύματος απόκρισης είναι 53

6.Στάλθηκε στην διεύθυνση 192.168.1.1 όπου είναι η IP διεύθυνση ενός από τους τοπικούς DNS servers

7. Είναι τύπου A Standart Query και δεν περιέχει απαντήσεις

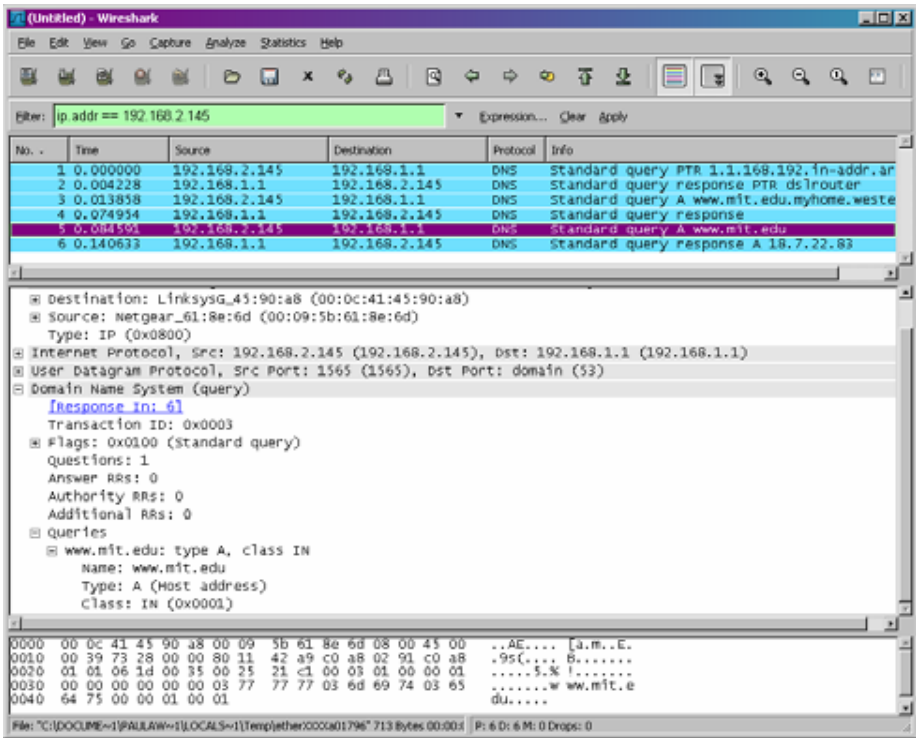
8.Υπάρχουν 2 απαντήσεις με πληροφορίες σχετικά με το όνομα του host, τον τύπο της διεύθυνσης, κατηγορία, TTL, μέγεθος και τη διεύθυνση
Answers
www.ietf.org: type A, class IN, addr 209.173.57.180
Name: www.ietf.org
Type: A (Host address)
Class: IN (0x0001)
Time to live: 30 minutes
Data length: 4
Addr: 209.173.57.180
www.ietf.org: type A, class IN, addr 209.173.53.180
Name: www.ietf.org
Type: A (Host address)
Class: IN (0x0001)
Time to live: 30 minutes

Data length: 4
Addr: 209.173.53.180

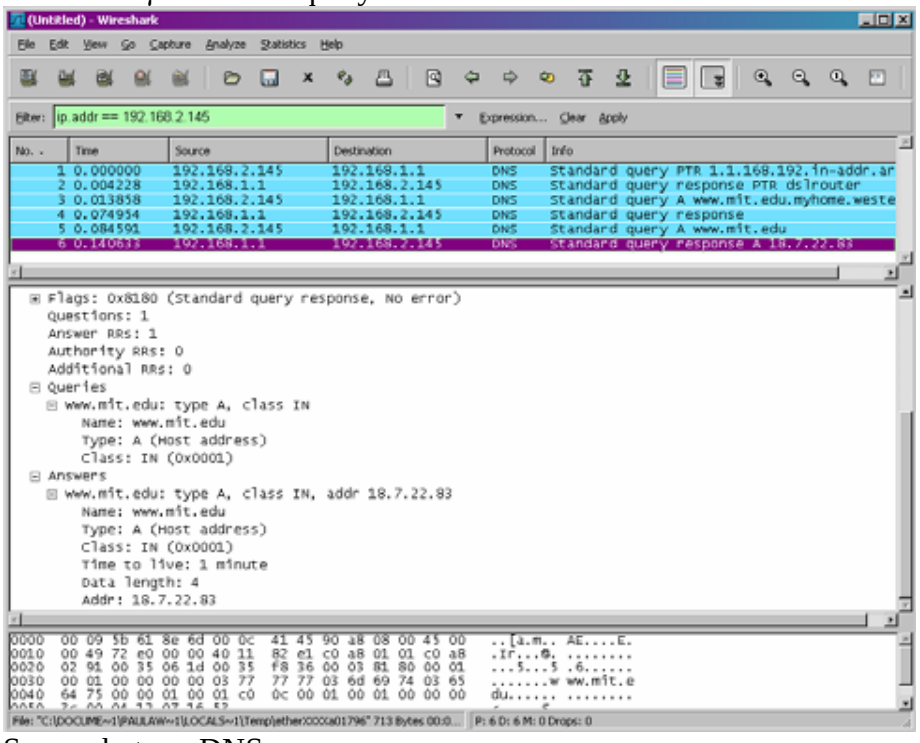
9.Το πρώτο πακέτο SYN στάλθηκε στο 209.173.57.180 και αντιστοιχεί στην πρώτη IP διεύθυνση που παρέχεται στο μήνυμα απόκρισης του DNS

10.Όχι

11.



Screenshot για το DNS query



Screenshot για DNS response

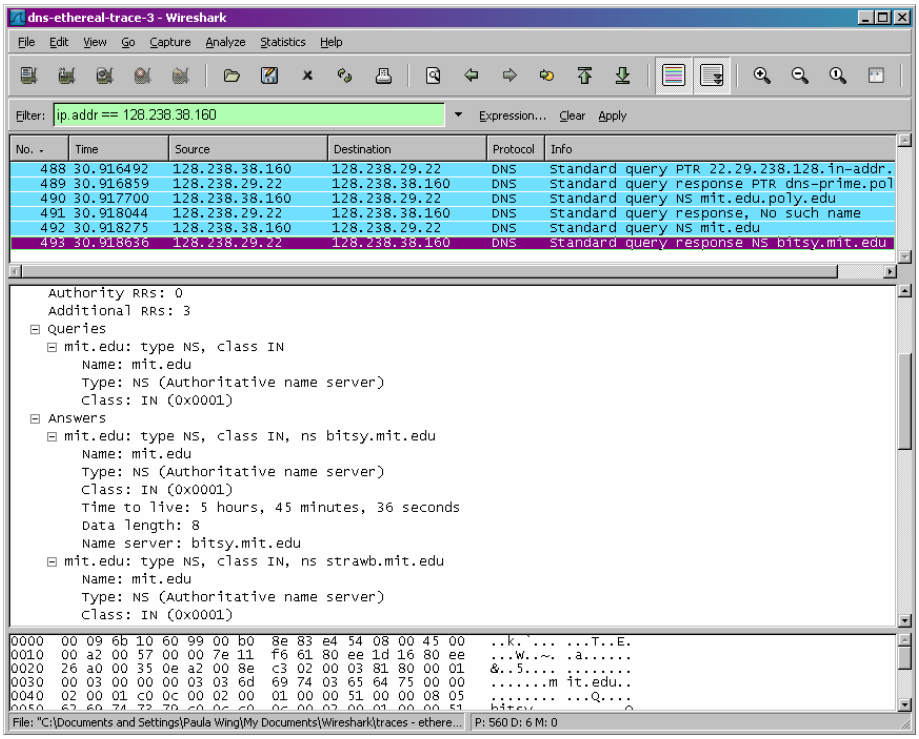
Η θύρα προορισμού του μηνύματος ερώτησης είναι 53 και η θύρα πηγής του μηνύματος απόκρισης είναι 53

12.Στάλθηκε στη διεύθυνση IP 192.168.1.1 και πρόκειται όπως μπορούμε να δούμε από το screenshot για τη διεύθυνση IP τοπικού DNS server

13.Είναι τύπου A και δεν περιέχει απαντήσεις

14.Περιέχει μια απάντηση η οποία περιέχει το όνομα του host, τον τύπο της διεύθυνσης, την κατηγορία και τη IP διεύθυνση
Answers
www.mit.edu: type A, class IN, addr 18.7.22.83
Name: www.mit.edu
Type: A (Host address)
Class: IN (0x0001)
Time to live: 1 minute
Data length: 4
Addr: 18.7.22.83

15.



16.Στάλθηκε στη 128.238.29.22 η οποία είναι διεύθυνση IP του τοπικού μας DNS server

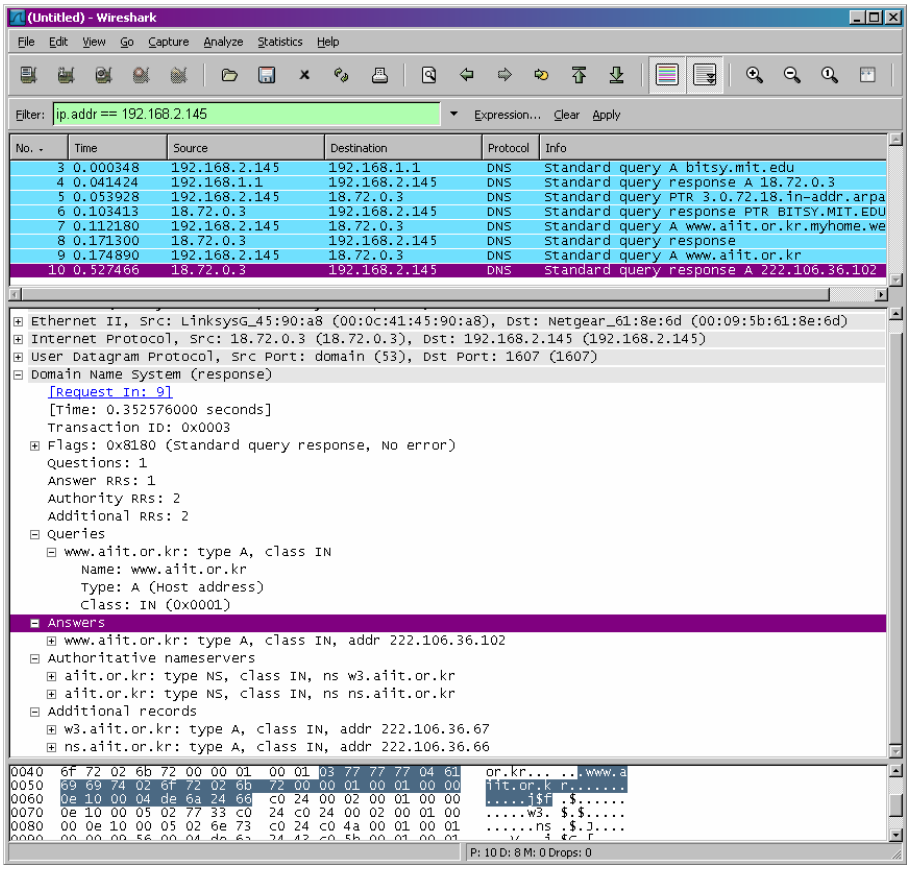
17.Είναι τύπου NS DNS query και δεν περιέχει καμία απάντηση

18.Μπορούμε να δούμε τις IP διευθύνσεις τους αν εκτείνουμε το πεδίο Additional records όπως φένεται παρακάτω
mit.edu: type NS, class inet, ns bitsy.mit.edu

mit.edu: type NS, class inet, ns strawb.mit.edu
mit.edu: type NS, class inet, ns w20ns.mit.edu

`Additional records`
bitsy.mit.edu: type A, class inet, addr 18.72.0.3
strawb.mit.edu: type A, class inet, addr 18.71.0.151
w20ns.mit.edu: type A, class inet, addr 18.70.0.160

19.



20.Στάλθηκε στη 18.72.0.3 η οποία αντιστοιχεί στο bitsy.mit.edu.

21.Είναι τύπου A και δεν περιέχει απαντήσεις

22.Οι απαντήσεις περιέχουν τα εξής
www.aiit.or.kr: type A, class inet, addr 222.106.36.102
Name: www.aiit.or.kr
Type: Host address
Class: inet
Time to live: 1 hour
Data length: 4
Addr: 222.106.36.102

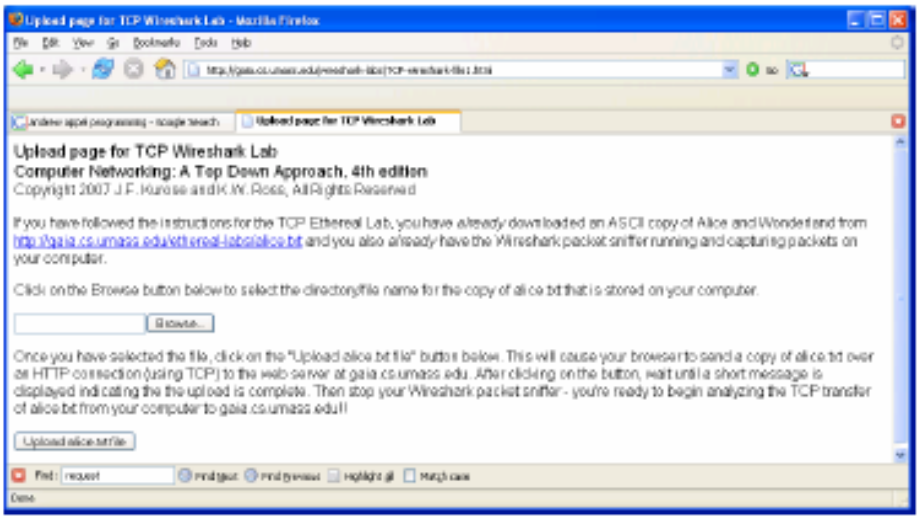
Κεφάλαιο 6
Ασκήσεις επίδειξης του πρωτοκόλλου TCP

Το TCP (Transmission Control Protocol - Πρωτόκολλο Ελέγχου Μεταφοράς) είναι ένα από τα κυριότερα πρωτόκολλα. Οι κύριοι στόχοι του πρωτοκόλλου TCP είναι να επιβεβαιώνεται η αξιόπιστη αποστολή και λήψη δεδομένων, επίσης να μεταφέρονται τα δεδομένα χωρίς λάθη μεταξύ του στρώματος δικτύου (network layer) και του στρώματος εφαρμογής (application layer) και, φτάνοντας στο πρόγραμμα του στρώματος εφαρμογής, να έχουν σωστή σειρά. Το TCP χρησιμοποιείται σχεδόν παντού, για αμφίδρομη επικοινωνία μέσω δικτύου.
Στο εργαστήριο αυτό θα εξετάσουμε λεπτομερώς τη συμπεριφορά του TCP. Θα το κάνουμε αυτό αναλύοντας ένα trace από TCP segments τα οποία στέλνονται και λαμβάνονται κατά τη μεταφορά ενός αρχείου 150 KB (που περιέχει το κείμενο του έργου του Lewis Carrol Alice's Adventure in Wonderland) από τον υπολογιστή μας σε έναν μακρινό server. Θα μελετήσουμε τον τρόπο που το TCP χρησιμοποιεί τους αριθμούς ακολουθίας και επιβεβαίωσης για να παρέχει αξιόπιστη μεταφορά δεδομένων, θα παρατηρήσουμε τον αλγόριθμο ελένχου συμφόρησης του TCP- αργή εκκίνηση και αποφυγή συμφόρησης- σε δράση και θα εξετάσουμε το μηχανισμό ελένχου ροής του TCP. Θα εξετάσουμε συνοπτικά την εγκαθίδρυση σύνδεσης TCP και θα διερευνήσουμε την απόδοση (throughput και round-trip time) της σύνδεσης TCP ανάμεσα στον υπολογιστή μας και τον server

6.1 Σύλληψη μαζικής μεταφοράς TCP από τον υπολογιστή σας σε έναν απομακρυσμένο server

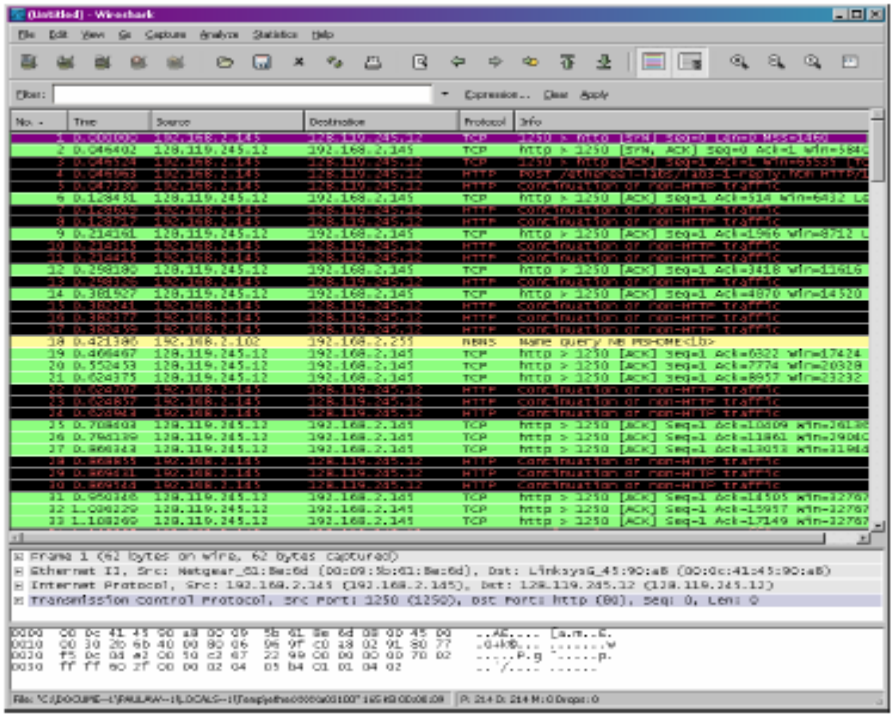
Πρίν ξεκινήσουμε την εξερεύνηση του TCP, θα χρειαστεί να χρησιμοποιήσουμε το Wireshark για να αποκτήσουμε το trace των πακέτων της μεταφοράς από το TCP ενός αρχείου από τον υπολογιστή μας σε έναν απομακρυσμένο server. Αυτό θα γίνει με την πρόσβαση σε μια ιστιοσελίδα η οποία θα μας επιτρέψει να εισάγουμε το όνομα ενός αρχείου αποθηκευμένου στον υπολογιστή μας (το οποίο περιέχει το κείμενο ASCII του Alice in Wonderland) και μετά μεταφέρουμε το αρχείο σε ένα Web server χρησιμοποιώντας τη μέθοδο HTTP POST. Χρησιμοποιούμε τη μέθοδο POST και όχι τη μέθοδο GET καθώς θέλουμε να μεταφέρουμε ένα μέγαλο όγκο δεδομένων από τον δικό μας υπολογιστή σε ένα άλλο υπολογιστή. Φυσικά, θα τρέχουμε το Wireshark κατά τη διάρκεια του χρόνου μεταφοράς ώστε να αποκτήσουμε το trace των TCP segments που στέλνονται και λαμβάνονται απο τον υπολογιστή μας.

Ακολουθούμε τα παρακάτω βήματα
>Ξεκινάμε τον browser μας. Πηγαίνουμε στο http://gaia.cs.umass.edu/wireshark-labs/alice.txt και ανακτούμε ένα αντίγραφο ASCII του Alice in Wonderland. Αποθηκεύουμε το αρχείο αυτό στον υπολογιστή μας.
>Στη συνεχειά πηγαίνουμε στο http://gaia.cs.umass.edu/wireshark-labs/TCP-wireshark-file1.html
>Ο browser μας θα πρέπει να εμφανίσει μια ιστιοσελίδα παρόμοια με την παρακάτω



Σχ.30 ιστιοσελίδα
http://gaia.cs.umass.edu/wireshark-labs/TCP-wireshark-file1.html

>Χρησιμοποιούμε το κουμπί Browse στη φόρμα αυτή για να εισάγουμε το όνομα του αρχείου (πλήρες path name) στον υπολογιστή μας που περιέχει το Alice in Wonderland. Δεν πατάμε ακόμη το κουμπί 'Upload alice.txt file'
>Ξεκινάμε τώρα το wireshark και τη σύλληψη πακέτων (Capture>Options) και στη συνέχεια πατάμε OK στο παράθυρο Επιλογές Σύλληψης Πακέτων (Packet Capture Options) του Wireshark (δεν θα χρειαστεί να διαλέξουμε κάποια από τις επιλογές εδώ)
>Επιστρέφοντας στον browser μας, πατάμε το κουμπί 'Upload alice.txt file' για να φορτώσουμε το αρχείο στον server gaia.cs.umass.edu. Αφού ολοκληρωθεί η μεταφορά του αρχείου, ένα μικρό συγχαρητήριο μήνυμα θα εμφανιστεί στον browser μας
>Σταματάμε τη σύλληψη πακέτων από το Wireshark. Το παράθυρο του Wireshark θα πρέπει να είναι παρόμοιο με το παράθυρο που φαίνεται παρακάτω



Σχ.31 Παράθυρο του Wireshark μετά την σύλληψη πακέτων

Εάν δεν είμαστε σε θέση να τρέξουμε το Wireshark σε μια ζωντανή σύνδεση δικτύου, μπορούμε να φορτώσουμε ένα αρχείο με το trace πακέτων που συνελήφθη κατά την εκτέλεση των βημάτων στον υπολογιστή του συγγραφέα (1) (Ενδεχομένως να διαπιστώσετε ότι αξίζει να φορτώσετε αυτό το trace ακόμη και αν έχετε συλλάβει το δικό σας και να το χρησιμοποιήστε παράλληλα με το δικό σας καθώς διερευνάτε τις ερωτήσεις που τίθενται παρακάτω)

6.2 Μια πρώτη ματιά στο trace

Πρίν αναλύσουμε λεπτομερώς τη συμπεριφορά της σύνδεσης TCP, ας κάνουμε μια γενική επισκόπηση του trace
>Πρώτα φιλτράρουμε τα πακέτα που παρουσιάζονται στο παράθυρο του Wireshark εισάγοντας 'tcp' (με μικρά γράμματα, χωρίς εισαγωγικά και χωρίς να ξεχάσουμε να πιέσουμε return μετά την εισαγωγή) στο παράθυρο των προδιαγραφών του φίλτρου παρουσίασης που βρίσκεται προς το επάνω μέρος του παραθύρου του Wireshark.

Στο παράθυρο κατάλογου πακέτων θα πρέπει να δούμε μια σειρά από μηνύματα TCP και HTTP να ανταλλάσσονται μεταξύ του υπολογιστή μας και του server gaia.cs.umass.edu. Θα πρέπει να δούμε την αρχική 'χειραψία' τρίων βημάτων που περιέχει ένα μήνυμα SYN. Θα πρέπει να δούμε ένα μήνυμα HTTP POST και μια σειρά απο μηνύματα 'HTTP Continuation' να στέλνονται από τον υπολογιστή μας στο gaia.cs.umass.edu. Υπενθυμίζεται, από το προηγούμενο κεφάλαιο για το HTTP, οτι δεν υπάρχουν μηνύματα Continuation στο HTTP- το Wireshark χρησιμοποιεί αυτόν τον τρόπο για να υποδείξει οτι χρησιμοποιούνται πολλαπλά TCP segments για τη μεταφορά ενός μηνύματος HTTP. Θα πρέπει επίσης να δούμε TCP segments με επιβεβαιώσεις (ACK) να επιστρέφουν από το gaia.cs.umass.edu στον υπολογιστή μας

Ανοίξτε το αρχείο tcp-ethereal-trace-1 των πακέτων που έχουν συλληφθεί από το Wireshark που περιέχεται στο http://gaia.cs.umass.edu/wireshark-labs/wireshark-traces.zip (δηλαδη, φορτώστε το trace και ανοίξτε το στο wireshark- βλ. Υποσημείωση 1) και στη συνέχεια απαντήστε στις ακόλουθες ερωτήσεις. Όπου είναι δυνατό, η απάντηση σας θα πρέπει να συνοδεύεται από μια εκτύπωση των πακέτων του trace που χρησιμοποιήσατε για να απαντήσετε σε κάθε ερώτηση. Σημειώστε επάνω στην εκτύπωση τα σημεία εκείνα που αιτιολογούν την απάντηση σας. Για να εκτυπώσετε ένα πακέτο, χρησιμοποιήστε File>Print, επιλέξτε Selected packet only, επιλέξτε Packet summary line και επιλέξτε το ελάχιστο ποσό λεπτομερειών πακέτου που χρειάζεστε για να απαντήσετε στην ερώτηση

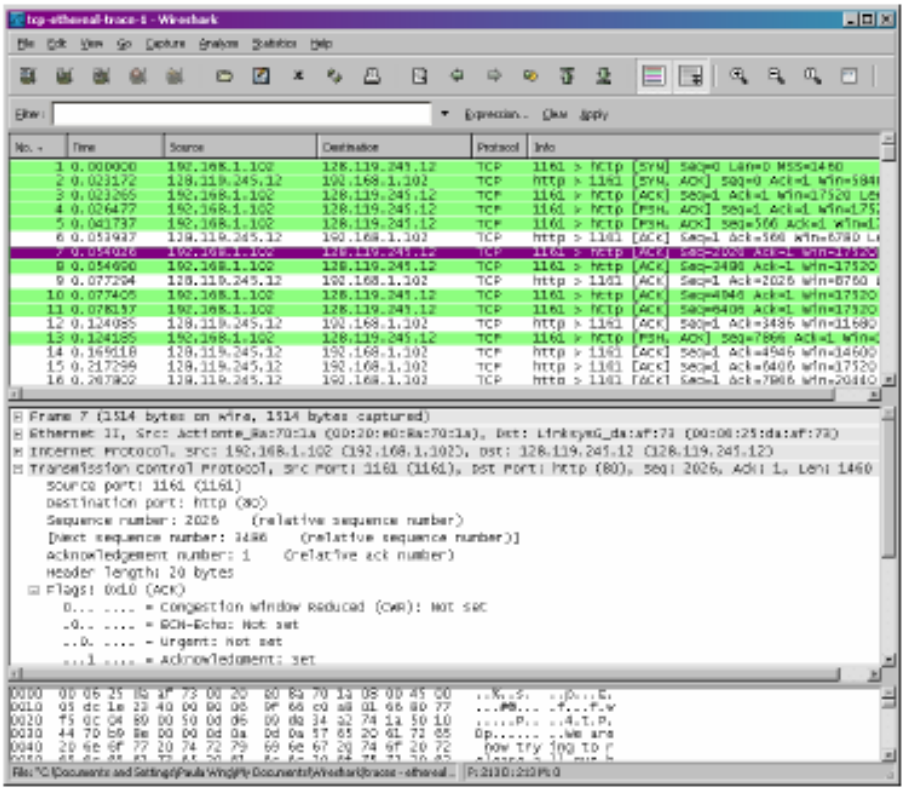
ΕΡΩΤΗΣΕΙΣ

1.Ποιά η διεύθυνση IP και ποιός ο αριθμός θύρας TCP που χρησιμοποιείται από τον client (πηγή) που μεταφέρει το αρχείο στο gaia.cs.umass.edu; Για να απαντήσετε στην ερώτηση αυτή είναι μάλλον ευκολότερο να επιλέξετε ένα μήνυμα HTTP και να εξετάσετε τις λεπτομέρειες του πακέτου TCP που χρησιμοποιήθηκε για να μεταφέρει αυτό το μήνυμα, χρησιμοποιώντας το παράθυρο με τις λεπτομέρειες επικεφαλίδας επιλεγμένου πακέτου (βλ. Σχήμα 23 στο εισαγωγικό εργαστήριο Wireshark για απορίες σχετικά με τα παράθυρα του Wireshark)

2.Ποιά η διεύθυνση IP του gaia.cs.umass.edu; Σε ποιό αριθμό θύρας στέλνει και λαμβάνει segments για αυτή τη σύνδεση TCP;
Εάν έχετε κατορθώσει να δημιουργήσετε το δικό σας trace, απαντήστε στην ακόλουθη ερώτηση

3.Ποιά η διεύθυνση IP και ποιός ο αριθμός θύρας TCP που χρησιμοποιείται από τον δικό σας client (πηγή) για τη μεταφορά του αρχείου στο gaia.cs.umass.edu;

Επειδή το εργαστήριο αυτό εστιάζει στο TCP και όχι στο HTTP, ας μεταβάλλουμε το παράθυρο καταλόγου πακέτων του Wireshark ώστε να παρουσιάζει πληροφορίες σχετικά με τα TCP segments που περιέχουν τα μηνύματα HTTP αντί για τα μηνύματα HTTP. Για να το κάνει αυτό το Wireshark, επιλέγουμε Analyse>Enabled Protocols. Στη σύνεχεια ξεμαρκάρουμε το κουτί HTTP και επιλέγουμε OK. Θα πρέπει τώρα να δούμε ένα παράθυρο Wireshark παρόμοιο με το ακόλουθο



Σχ.32 παράθυρο Wireshark με πληροφορίες σχετικά με τα TCP segments

Αυτός ήταν ο επιδιωκόμενος στόχος- μια σειρά από TCP segments που ανταλλάσσονται μεταξύ του υπολογιστή μας και του gaia.cs.umass.edu. Στο υπόλοιπο μέρος αυτού του εργαστηρίου, θα χρησιμοποιήσουμε το trace των πακέτων που έχουμε συλλάβει (και το trace πακέτων tcp-ethereal-trace-1 στο http://gaia.cs.umass.edu/wireshark-labs/wireshark-traces.zip -βλ.υποσημείωση 1) για να μελετήσουμε τη συμπεριφορά του TCP

6.3 Βασικά χαρακτηριστικά του TCP

Απαντήστε στις ακόλουθες ερωτήσεις για τα TCP segments

ΕΡΩΤΗΣΕΙΣ

4.Ποιός ο αριθμός ακολουθίας του TCP segments SYN που χρησιμοποιείται για την εκκίνηση της σύνδεσης TCP μεταξύ του client και του gaia.cs.umass.edu; Ποιό στοιχείο του segment προσδιορίζει οτι πρόκειται για ένα SYN segment;

5.Ποιός ο αριθμός ακολουθίας του segment SYNACK που στέλνεται από το gaia.cs.umass.edu στον client ως απόκριση στο segment SYN; Ποιά η τιμή του πεδίου ACK στο segment SYNACK; Με ποιό τρόπο καθορίστηκε η τιμή αυτή από το gaia.cs.umass.edu; Ποιο στοιχείο του segment προσδιορίζει οτι πρόκειται για ένα SYNACK segment;

6.Ποιος ο αριθμός ακολουθίας του TCP segment που περιέχει την εντολή HTTP POST; Σημειώνεται οτι για να εντοπίσετε την εντολή POST θα χρειαστεί να ψάξετε στο πεδίο περιεχομένων πακέτου που βρίσκεται στο κάτω μέρος του παραθύρου Wireshark αναζητώντας ένα segment που περιέχει τους χαρακτήρες 'POST' στο πεδίο των δεδομένων του.

7.Θεωρείστε το TCP segment που περιέχει την εντολή ως το πρώτο segment της σύνδεσης TCP. Ποιοί οι αριθμοί ακολουθιάς των πρώτων 6 segments της σύνδεσης TCP (συμπεριλαμβανόμενου και του segment που περιέχει την εντολή HTTP POST); Ποιός ο χρόνος αποστολής του κάθε segment; Ποιός ο χρόνος λήψης της επιβεβαίωσης ACK για κάθε segment; Δεδομένου της διαφοράς μεταξύ του χρόνου αποστολής ενός TCP segment και του χρόνου λήψης της επιβεβαίωσης του, ποιά η τιμή του RTT για καθένα από τα 6 segments; Ποιά η τιμή της μεταβλητής EstimatedRTT μετά τη λήψη της κάθε επιβεβαίωσης ACK; Υποθέστε οτι η τιμή του EstimatedRTT είναι ίση με τον μετρούμενο χρόνο RTT για το πρώτο segment
(Σημείωση-Το Wireshark διαθέτει ένα χαρακτηριστικό γνώρισμα που σας επιτρέπει να παραστήσετε γραφικά το χρόνο RTT για καθένα από τα σταλμένα TCP segments. Στο παράθυρο καταλόγου πακέτων επιλέξτε ένα TCP segment το οποίο στέλνεται από το client στον server gaia.cs.umass.edu. Στη συνέχεια επιλέξτε Statistics>TCP Stream Graph>Round Trip Time Graph)

8.Ποιό το μήκος καθενός από τα 6 πρώτα TCP segments (2);

9.Ποιός ο ελάχιστος διαθέσιμος χώρος αποθήκευσης (buffer space) που ανακοινώνεται από τον παραλήπτη σε ολόκληρο το trace; Συμβαίνει ποτέ η έλλειψη χώρου αποθήκευσης στον παραλήπτη να περιορίζει το ρυθμό του αποστολέα;

10.Υπάρχουν επαναμεταδιδόμενα segments στο αρχείο του trace; Σε τι είδους έλενχο του trace βασίσατε την απάντηση σας στην ερώτηση αυτή;

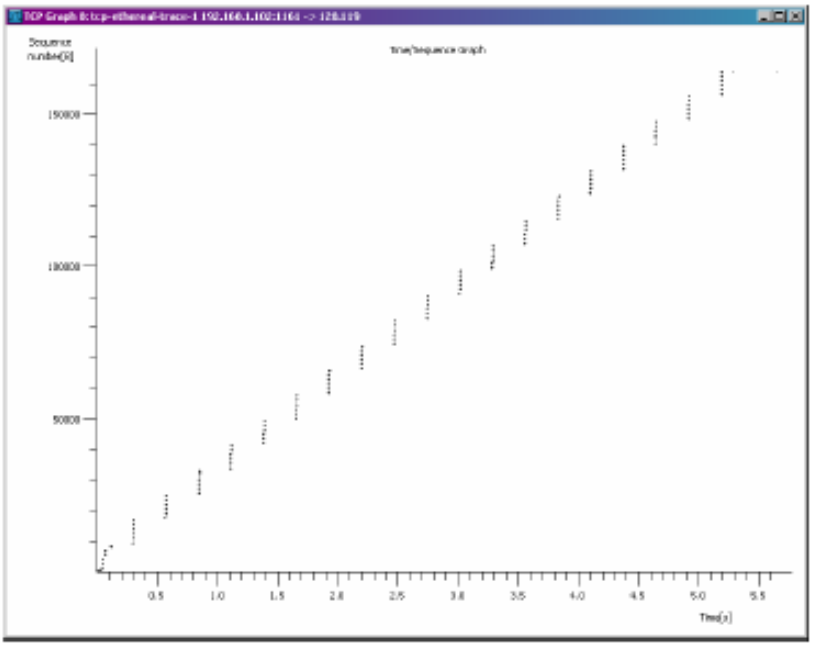
11.Πόσα bytes δεδομένων επιβεβαιώνει συνήθως ο παραλήπτης σε μια επιβεβαίωση; Μπορείτε να διακρίνετε περιπτώσεις όπου ο παραλήπτης επιβεβαιώνει κάθε δεύτερο λαμβανόμενο segment ;

12.Ποιό το throughput (αριθμός μεταφερόμενων bytes ανά μονάδα χρόμου) της σύνδεσης TCP; Εξηγείστε τον τρόπο με τον οποίο υπολογίσατε την τιμή αυτή.

6.4 Ο αλγόριθμος συμφόρησης του TCP σε δράση

Ας εξετάσουμε τώρα τον όγκο των δεδομένων που στέλνονται ανα μονάδα χρόνου από τον client στον server. Αντί να υπολογίσουμε το μέγεθος αυτό από τα ανεπεξέργαστα δεδομένα του παραθύρου του Wireshark, θα χρησιμοποιήσουμε ένα από τα βοηθητικά γραφικά εργαλεία του Wireshark για το TCP-Time-Sequence-Graph (Stevens) για να παραστήσουμε γραφικά τα δεδομένα

>Επιλέγουμε ένα TCP segment στο παράθυρο καταλόγου πακέτων του Wireshark. Μετά επιλέγουμε το μενού Statistics> TCP Stream Graph> Time-Sequence-Graph (Stevens). Θα πρέπει να δούμε μια γραφική παράσταση παρόμοια με την ακόλουθη η οποία δημιουργήθηκε για τα δεδομένα του trace πακέτων tcp-ethereal-trace-1 που περιέρχεται στο http://gaia.cs.umass.edu/wireshark-labs/wireshark-traces.zip (βλ.υποσημείωση 1)



Σχ.33 Γραφική παράσταση για τα δεδομένα του trace πακέτων tcp-ethereal-trace-1

Στην παραπάνω γραφική παράσταση, όπου κάθε κουκκίδα παριστάνει ένα απεσταλμένο TCP segment, δίνεται ο αριθμός ακολουθίας του segment και ο χρόνος αποστολής του. Παρατηρούμε ότι ένα σύνολο κουκκίδων, με τη μια κουκκίδα πάνω στην άλλη, αναπαριστά μια ακολουθία που στάλθηκαν το ένα αμέσως μετά το άλλο (back-to-back)

Απαντήστε στις ακόλουθες ερωτήσεις για τα TCP segments του trace πακέτων tcp-ethereal-trace-1 που περιέχεται στο http://gaia.cs.umass.edu/wireshark-labs/wireshark-traces.zip

ΕΡΩΤΗΣΕΙΣ

13.Χρησιμοποιείστε το γραφικό εργαλείο Time-Sequence-Graph(Stevens) για να λάβετε τη γραφική παράσταση του αριθμού ακολουθίας ως προς το χρόνο των segments που στέλνονται από τον client στον server gaia.cs.umass.edu. Μπορείτε να προσδιορίσετε πότε αρχίζει και πότε τέλειωνει η φάση αργής εκκίνησης (slow start) του TCP, και πότε γίνεται μετάβαση στη φάση αποφυγής συμφόρησης (congestion avoidance); Προσέξτε ότι στο 'πραγματικό' αυτό trace, η συμπεριφορά του TCP διαφέρει από την ιδανική.

14.Σχολιάστε τις διαφορές ανάμεσα στα δεδομένα των μετρήσεων και στην εξιδανικευμένη συμπεριφορά του TCP που μελετήσαμε στο βιβλίο

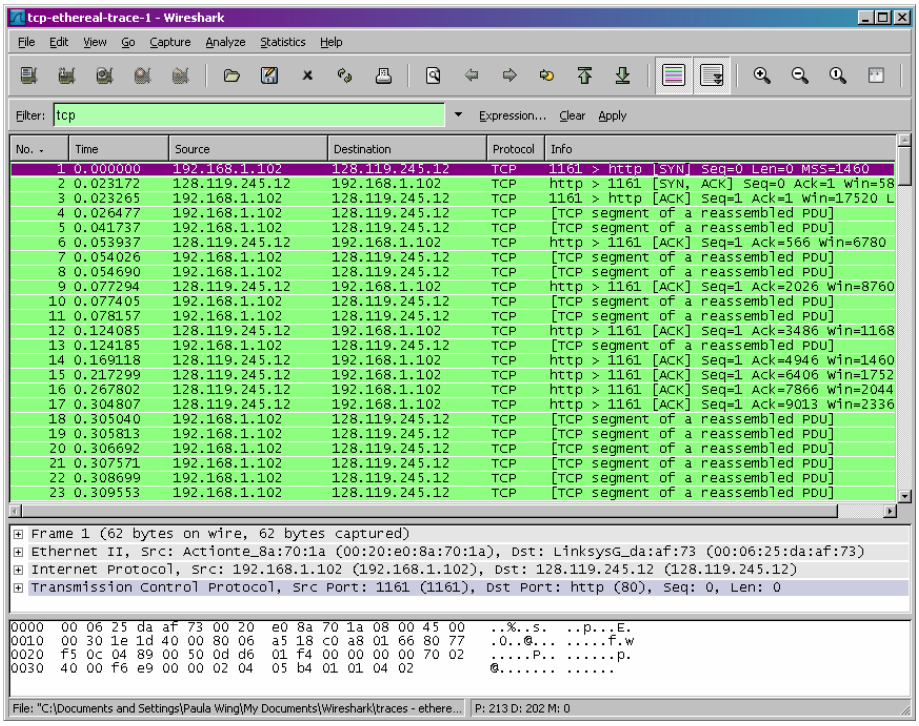
15.Απαντήστε σε κάθεμια από τις δυο παραπάνω ερωτήσεις για το trace που συλλέξατε εσείς κατά τη μεταφορά του αρχείου απο τον υπολογιστή σας στο gaia.cs.umass.edu

(1)Φορτώστε το αρχείο zip http://gaia.cs.umass.edu/wireshark-labs/wireshark-traces.zip και εξάγετε το αρχείο tcp-ethereal-trace-1.Τα traces που περιέχονται σε αυτό το αρχείο zip συλλέχθηκαν απο το Wireshark ενώ εκτελούνταν τα βήματα που περιέχονται στο εργαστήριο Wireshark στον υπολογιστή του συγγραφέα.Αφού λάβετε το trace,μπορείτε να το φορτώσετε στο Wireshark και να το δείτε στο παράθυρο χρησιμοποιώντας το μενού File, επιλέγοντας Open και στη συνέχεια επιλέγοντας το αρχείο tcp-ethereal-trace-1 του trace

(2)Τα TCP segments στο αρχείο tcp-ethereal-trace-1 του trace είναι όλα μικρότερα από 1460 bytes.Αυτό οφείλεται στο γεγονός οτι ο υπολογιστής που χρησιμοποιήθηκε για τη συλλογή του trace έχει μια κάρτα Ethernet η οποία περιορίζει το μέγιστο μήκος ενός IP datagram σε 1500 bytes (40 bytes για τις επικεφαλίδες TCP/IP και1460 bytes ως ωφέλιμου φορτίου TCP) .Αυτή η τιμή των 1500 bytes αποτελεί το καθιερωμένο μεγίστο επιτρεπτό μήκος στο Ethernet.Εάν το δικό σας trace εμφανίζει ένα TCP segment με μήκος μεγαλύτερο από 1400 bytes και ο υπολογιστής σας χρησιμοποιεί μια σύνδεση Ethernet,τότε το Wireshark αναφέρει λάθος μήκος TCP segment.Είναι πιθανό επίσης να δείχνει μόνο ένα μεγάλο TCP segment αντί για πολλαπλά μικρότερα segments.Στην πραγματικότητα,ο υπολογιστή σας μάλλον στέλνει πολλαπλά μικρότερα segments όπως υποδεικνύεται από τις πολλαπλές επιβεβαιώσεις που λαμβάνει.Αυτή η ασυνέπεια στα αναφερόμενα μήκη των segments οφείλεται στην αλληλεπίδραση μεταξύ του Ethernet driver και του λογισμικού Wireshark.Σε περίπτωση που αντιμετωπίσετε αυτό το πρόβλημα,συνιστούμε να χρησιμοποιήσετε το trace του αρχείουtcp-ethereal-trace-1 για το εργαστήριο αυτό.

ΑΠΑΝΤΗΣΕΙΣ

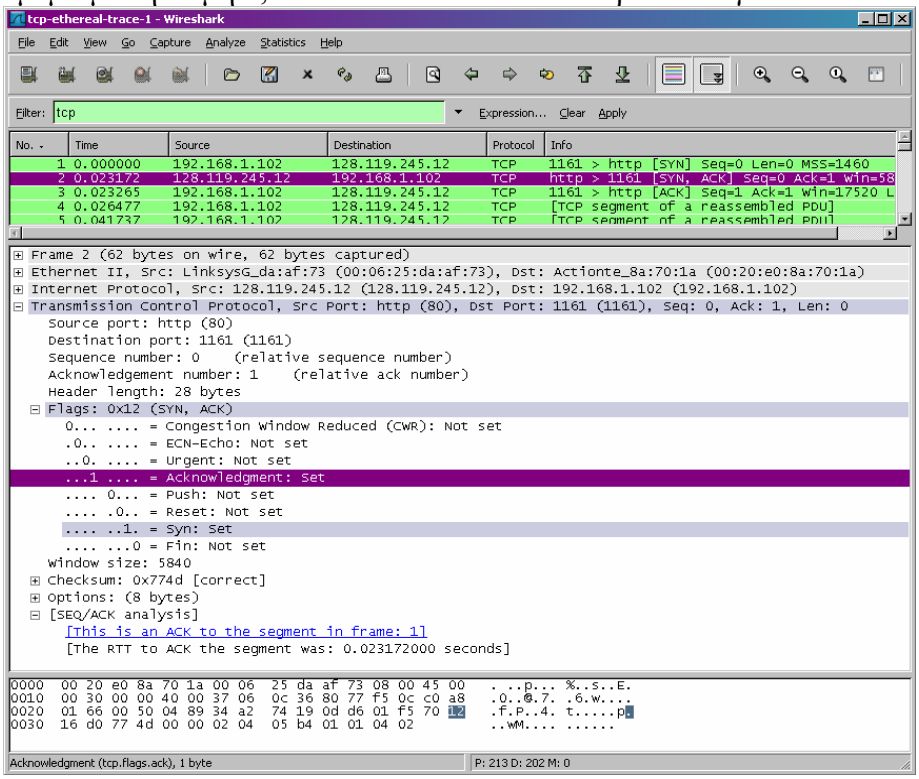
1. Η διεύθυνση IP είναι 192.168.1.102 και ο αριθμός θύρας TCP είναι 1161



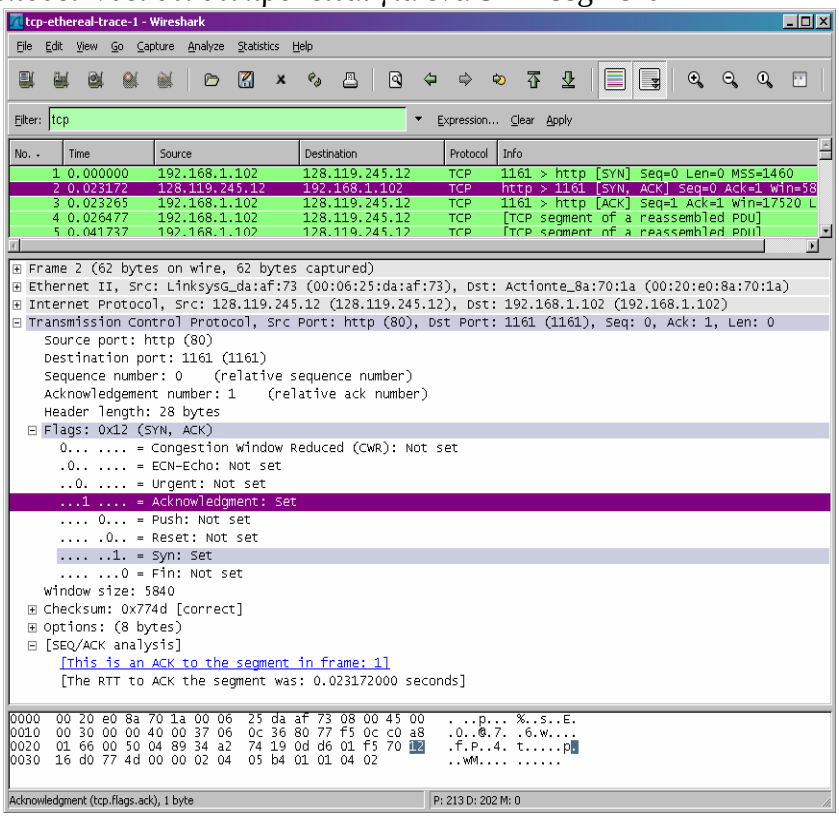
IP διευθύνσεις και TCP αριθμοί θύρας του client pc

2. Η διεύθυνση IP είναι 128.119.245.12 και ο αριθμός θύρας TCP είναι 80

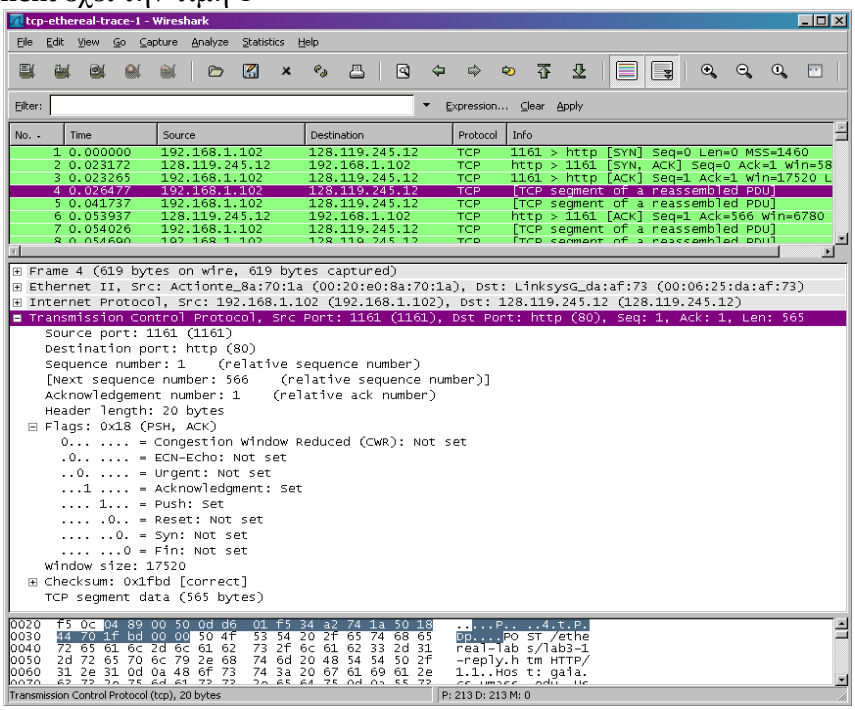
4.Ο αριθμός ακολουθίας έχει τιμή 0. Το SYN και το Acknowledgement είναι σεταρισμένα με την τιμή 1, το οποίο υποδεικνύει ότι πρόκειται για ένα SYN segment



5.Ο αριθμός ακολουθίας έχει τιμή 0. Η τιμή του πεδίου ACK στο segment SYNACK είναι 1. Η τιμή του καθορίστηκε από το gaia.cs.umass.edu προσθέτωντας 1 στον αρχικό αριθμό ακολουθίας του SYN segment του client υπολογιστή (η τιμή του αρχικά ήταν 0). Το SYN και το Acknowledgement είναι σεταρισμένα με την τιμή 1 , το οποίο υποδεικνύει ότι οτι πρόκειται για ένα SYN segment



6.Στο No. 4 segment περιέχεται η εντολή HTTP POST .Ο αριθμός ακολουθίας του TCP segment έχει την τιμή 1



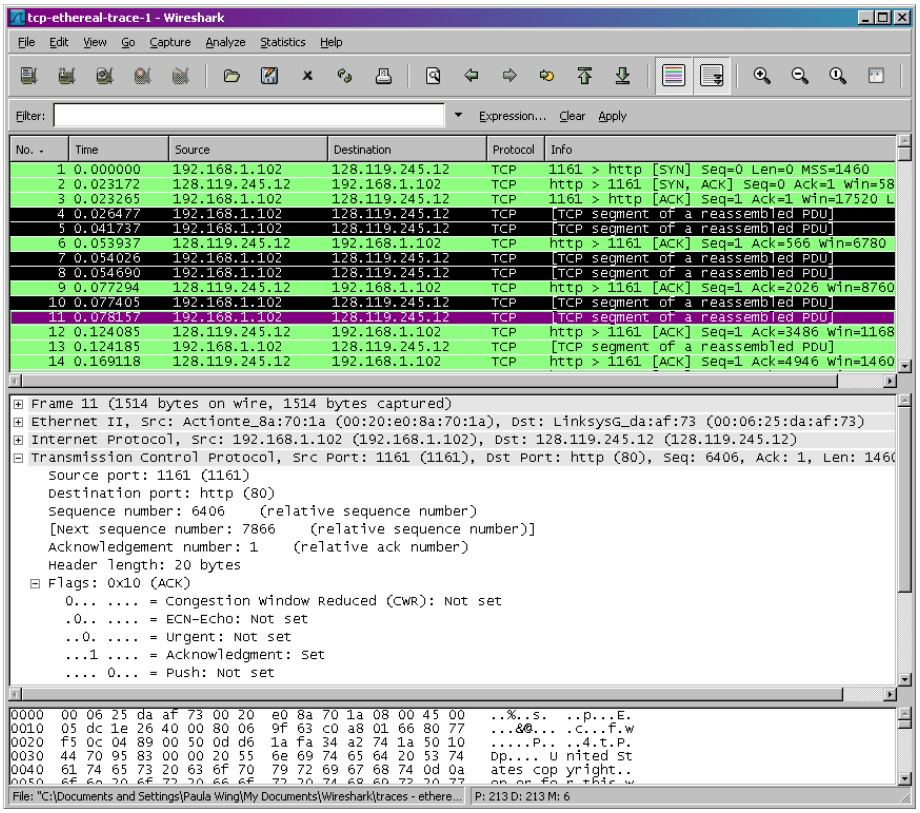
7.Segments 1-6 είναι No. 4, 5, 7, 8, 10 και 11
Τα ACKs των1-6 segments είναι No. 6, 9, 12, 14, 15 και 16

Αριθμός ακολουθίας Segment 1: 1
Αριθμός ακολουθίας Segment 2: 566
Αριθμός ακολουθίας Segment 3: 2026
Αριθμός ακολουθίας Segment 4: 3486
Αριθμός ακολουθίας Segment 5: 4946
Αριθμός ακολουθίας Segment 6: 6406

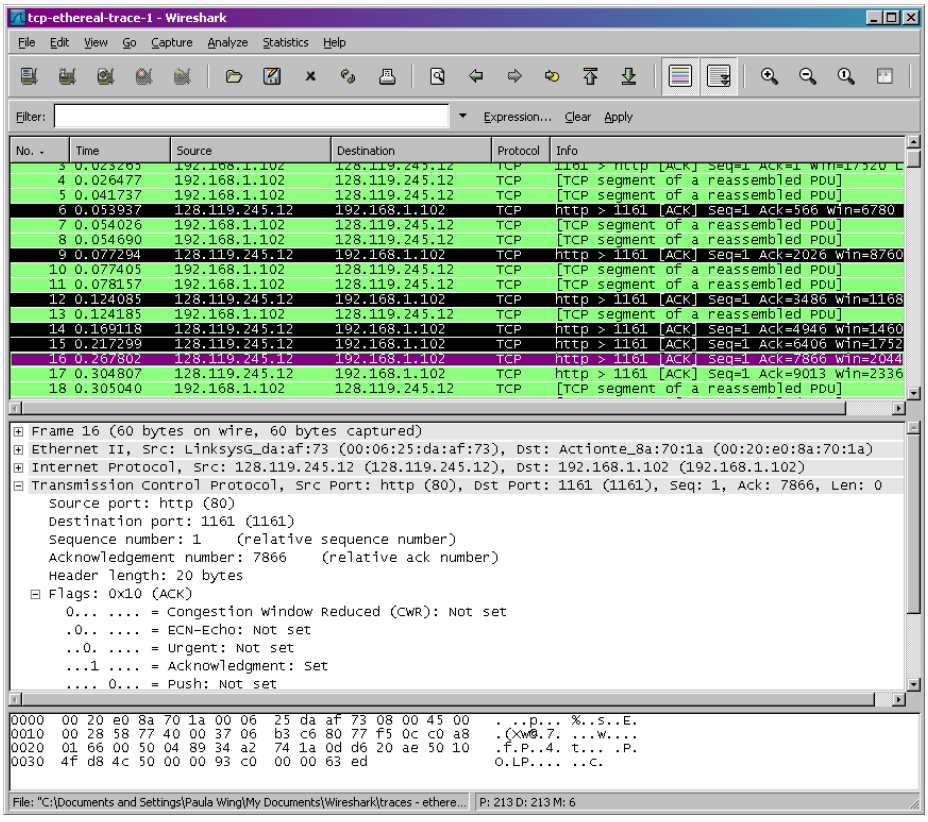
χρόνος αποστολής του κάθε segment /χρόνος λήψης της επιβεβαίωσης ACK

	Sent time	ACK received time	RTT (seconds)
Segment 1	0.026477	0.053937	0.02746
Segment 2	0.041737	0.077294	0.035557
Segment 3	0.054026	0.124085	0.070059
Segment 4	0.054690	0.169118	0.11443
Segment 5	0.077405	0.217299	0.13989
Segment 6	0.078157	0.267802	0.18964

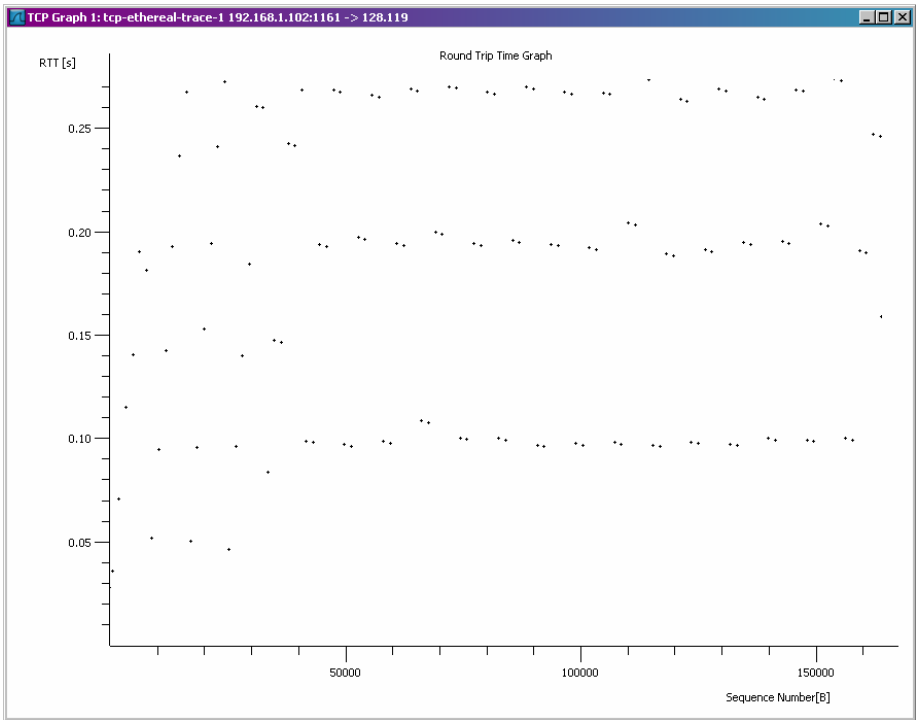
EstimatedRTT =0.875 * EstimatedRTT + 0.125 * SampleRTT
EstimatedRTT μετά τη λήψη της επιβεβαίωσης ACK για το segment 1:
EstimatedRTT = RTT for Segment 1 = 0.02746 second
EstimatedRTT τη λήψη της επιβεβαίωσης ACK για το segment 2:
EstimatedRTT = 0.875 * 0.02746 + 0.125 * 0.035557 = 0.0285
EstimatedRTT μετά τη λήψη της επιβεβαίωσης ACK για το segment 3:
EstimatedRTT = 0.875 * 0.0285 + 0.125 * 0.070059 = 0.0337
EstimatedRTT μετά τη λήψη της επιβεβαίωσης ACK για το segment 4:
EstimatedRTT = 0.875 * 0.0337+ 0.125 * 0.11443 = 0.0438
EstimatedRTT μετά τη λήψη της επιβεβαίωσης ACK για το segment 5:
EstimatedRTT = 0.875 * 0.0438 + 0.125 * 0.13989 = 0.0558
EstimatedRTT μετά τη λήψη της επιβεβαίωσης ACK για το segment 6:
EstimatedRTT = 0.875 * 0.0558 + 0.125 * 0.18964 = 0.0725
Second



Segments 1 – 6



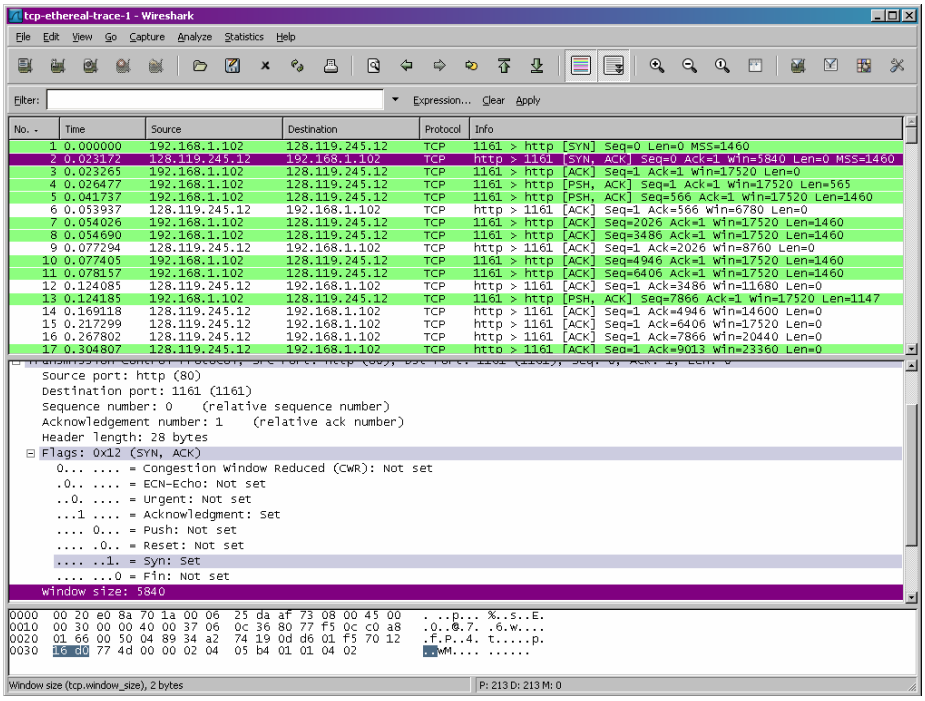
ACKs των segments 1 – 6



Round Trip Time Graph

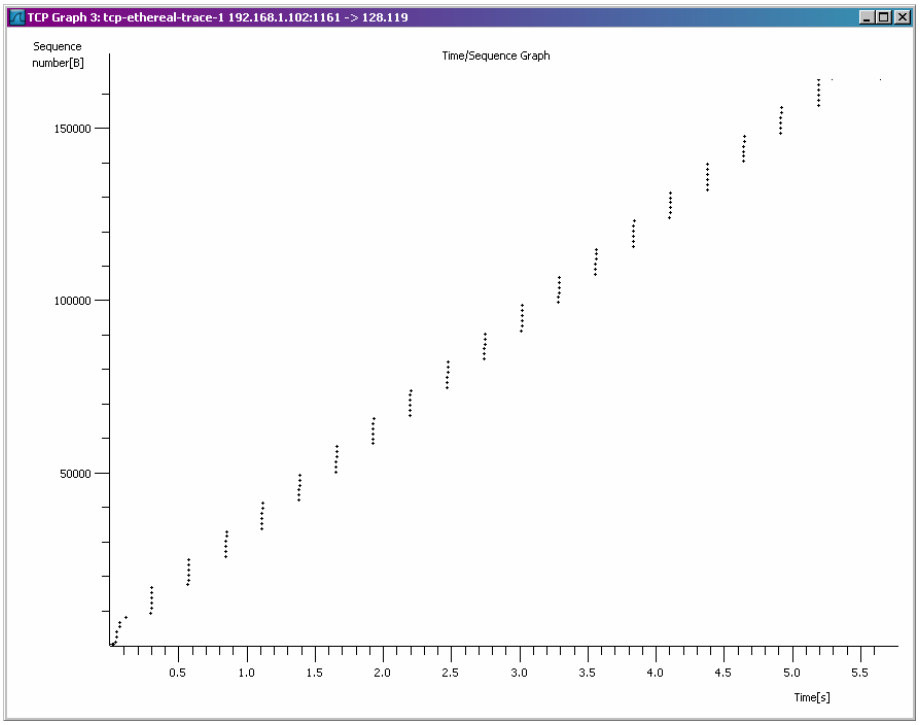
8.Το μέγεθος του 1ου TCP segment (περιέχει την εντολή HTTP POST): 565 bytes
Το μέγεθος των υπόλοιπων 5 TCP segments: 1460 bytes (MSS)

9.Ο ελάχιστος διαθέσιμος χώρος αποθήκευσης είναι 5840 bytes. Ο παραλήπτης έχει χώρο αποθήκευσης εώς και 62780 bytes άρα ο αποστολέας δεν περιορίζει ποτέ τον ρυθμό του.



10.Δεν υπάρχουν αναμεταδιδόμενα segments. Μπορούμε να το διαπιστώσουμε βλέποντας το χρονογράφημα όπου φαίνεται ότι όλοι οι αριθμοί ακολουθίας από την

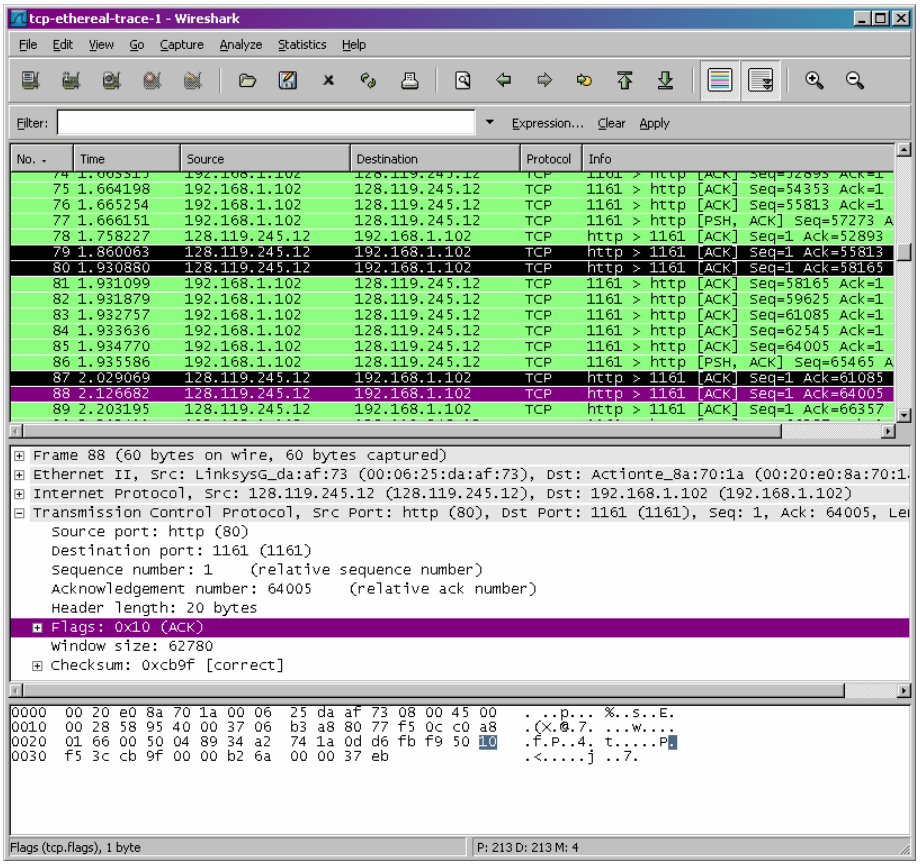
πηγή (192.168.1.102) μέχρι και τον προορισμό (128.119.245.12) αυξάνονται αναλογικά προς τον χρόνο .



11.Οι αριθμοί ακολουθίας και τα bytes δεδομένων

	acknowledged sequence number	acknowledged data
ACK 1	566	566
ACK 2	2026	1460
ACK 3	3486	1460
ACK 4	4946	1460
ACK 5	6406	1460
ACK 6	7866	1460
ACK 7	9013	1147
ACK 8	10473	1460
ACK 9	11933	1460
ACK 10	13393	1460
ACK 11	14853	1460
ACK 12	16313	1460

Η διαφορά μεταξύ των δυο αριθμών ακολουθίας δυο διαδοχικών ACKs υποδεικνύει το μέγεθος των δεδομένων που έλαβε ο server μεταξύ αυτών των 2 ACKs. Κοιτώντας τον πίνακα acknowledged data βλέπουμε ότι υπάρχουν περιπτώσεις όπου ο παραλήπτης επιβεβαιώνει κάθε δεύτερο λαμβανόμενο segment



Συσσωρευμένα ACKs (No. 80, 87, 88, κτλ)

12.Ο μέσος όρος υπολογίζεται ως ο λόγος μεταξύ του συνολικού όγκου πληροφορίας και του συνολικού χρόνου μεταφοράς. Το συνολικό μέγεθος μπορεί να υπολογιστεί από την διαφορά του 1ου (1 byte για το No. 4 segment) και τελευταίου (164091 bytes για το No. 202). Άρα 164091 - 1 = 164090 bytes. Ο συνολικός χρόνος μεταφοράς μπορεί να υπολογιστεί από την χρονική διαφορά μεταξύ του 1ου (0.026477 second για το No.4 segment) και του τελευταίου ACK (5.455830 second για το No. 202 segment). Άρα 5.455830 - 0.026477 = 5.4294 seconds. Το throughput υπολογίζεται ως 164090/5.4294 = 30.222 KByte/sec.

13.Στον ακόλουθο πίνακα δεν μπορούμε να δούμε το πώς ο όγκος σημαντικών δεδομένων αυξάνει γρήγορα στην εκκίνηση του TCP. Πάντως δεν υπερβαίνει τα 8192 Bytes και σίγουρα το μέγεθος του είναι μεγαλύτερο από 8192 Bytes. Παρόλαυτα δεν μπορούμε να καθορίσουμε το τέλος της αργής εκκίνησης και την έναρξη της φάσης αποφυγής συμφόρησης. Ο βασικός λόγος γι' αυτό είναι ότι ο αποστολέας δεν στέλνει πολλά δεδομένα ώστε να μπορεί να δημιουργηθεί συμφόρηση. Εξετάζοντας τον όγκο των σημαντικών δεδομένων μπορούμε να παρατηρήσουμε ότι η συσκευή στέλνει κυρίως δεδομένα κάτω των 8192 bytes. Πρίν λάβει την επιβεβαίωση για αυτά τα 8192 bytes, η συσκευή δεν στέλνει περισσότερα δεδομένα. Αυτό δείχνει ότι πρίν το τέλος της φάσης αργής εκκίνησης, η συσκευή έχει ήδη σταματήσει την μεταφορά.

Type	No.	Seq.	ACKed seq.	Outstanding data
Data	4	1		565
Data	5	566		2025
ACK	6		566	1460
Data	7	2026		2920
Data	8	3486		4380
ACK	9		2026	2920
Data	10	4946		4380
Data	11	6406		5840
ACK	12		3486	4380
Data	13	7866		5527
ACK	14		4096	4917
ACK	15		6006	3007
ACK	16		7866	1147
ACK	17		9013	0
Data	18	9013		1460
Data	19	10473		2920
Data	20	11933		4380
Data	21	13393		5840
Data	22	14853		7300
Data	23	16313		8192
ACK	24		10473	6732
ACK	25		11933	5272
ACK	26		13393	3812
ACK	27		14853	2352
ACK	28		16313	892
ACK	29		17205	0
Data	30	17205		1460
Data	31	18665		2920
Data	32	20125		4380
Data	33	21585		5840
Data	34	23045		7300
Data	35	24505		8192
ACK	36		18665	6732
ACK	37		20125	5272
ACK	38		21585	3812
ACK	39		23045	2352
ACK	40		24505	892
ACK	41		25397	0
Data	42	25397		1460
Data	43	26857		2920
Data	44	28317		4380
Data	45	29777		5840

Data	46	31237		7300
Data	47	32697		8192
ACK	48		26857	
ACK	49		28317	
ACK	50		29777	
ACK	51		31237	
ACK	52		33589	
Data	53	33589		6732
Data	54	35049		5272
Data	55	36509		3812
Data	56	37969		2352
Data	57	39429		892
Data	58	40889		0
ACK	59		35049	6732
ACK	60		37969	3812
ACK	61		40889	892
ACK	62		41781	0
Data	63	41781		1460
Data	64	43241		2920
Data	65	44701		4380
Data	66	46161		5840
Data	67	47621		7300
Data	68	49081		8192
ACK	69		44701	5272
ACK	70		47621	2352
ACK	71		49973	0
Data	72	49973		1460
Data	73	51433		2920
Data	74	52893		4380
Data	75	54353		5840
Data	76	55813		7300
Data	77	57273		8192
ACK	78		52893	5272
ACK	79		55813	2352
ACK	80		58165	0
Data	81	58165		

14.Η ιδανική συμπεριφορά του TCP υποθέτει ότι ο TCP αποστολέας στέλνει υπεραρκετά δεδομένα. Υπερβολική μεταφορά δεδομένων μπορεί να προκαλέσει συμφόρηση στο δίκτυο, γι' αυτό οι TCP αποστολείς πρέπει να ακολουθήσουν τον AIMD αλγόριθμο ώστε όταν σημειωθεί συμφόρηση στο δίκτυο, το μέγεθος του παραθύρου του αποστολέα να μειωθεί. Πρακτικά η συμπεριφορά του TCP εξαρτάται αρκετά από την συσκευή.

Κεφάλαιο 7
Ασκήσεις επίδειξης του πρωτοκόλλου UDP

Το πρωτόκολλο UDP (User Datagram Protocol) είναι ένα από τα βασικά πρωτόκολλα που χρησιμοποιούνται στο Διαδίκτυο. Διάφορα προγράμματα χρησιμοποιούν το πρωτόκολλο UDP για την αποστολή σύντομων μηνυμάτων (γνωστών και ως datagrams) από τον έναν υπολογιστή στον άλλον μέσα σε ένα δίκτυο υπολογιστών.
Σε αυτό το εργαστήριο, θα ρίξουμε μια γρήγορη ματιά στο UDP πρωτόκολλο μεταφοράς. Επειδή το UDP είναι απλό, θα το καλύψουμε σχετικά γρήγορα σε αυτό το εργαστήριο.

7.1 Η Εκχώρηση

Ξεκινάμε να συλλαμβάνουμε πακέτα στο wireshark και μετά δίνουμε εντολή στον host να στείλει και να λαμβάνει διάφορα UDP πακέτα. [Ένας τρόπος για να γίνει αυτό είναι να χρησιμοποιήσουμε την εντολή nslookup, όπως είδαμε στο εργαστήριο DNS. Αν δεν μπορούμε να τρέξουμε το wireshark σε μια ζωντανή σύνδεση, μπορουμε να κατεβάσουμε ένα trace αρχείο το οποίο συλλήφθηκε ακολουθόντας τα δυο πρώτα βήματα του nslookup του εργαστηρίου wireshark DNS σε κάποιον υπολογιστή του συγγραφέα (1)]
Αφού σταματήσει η σύλληψη πακέτων, ενεργοποιούμε το packet filter μιας και το wireshark αντιλαμβάνετε μόνο τα UDP πακέτα που στέλνονται και συλλέγονται στο δικό μας host. Επιλέγουμε ένα από τα UDP πακέτα και ανοίγουμε τα UDP πεδία στο παράθυρο λεπτομερειών.
Στις απαντήσεις των ερωτήσεων καλό θα ήταν να έχουμε μια εκτύπωση των πακέτων trace που χρησιμοποιήσαμε για να απαντήσουμε. Για να εκτυπώσουμε ένα πακέτο, File> Print, επιλέγουμε Seleced packet only, επιλέγουμε Packet summery line, και επιλέγουμε το μικρότερο δυνατό απόσμασμα του πακέτου που χρησιμοποιήσαμε για να απαντήσουμε στην ερώτηση.

ΕΡΩΤΗΣΕΙΣ

1.Επιλέξτε ένα πακέτο. Από το πακέτο αυτό, υπολογίστε πόσα πεδία υπάρχουν στο UDP header. (Μην κοιτάξετε το textbook! απαντήστε τις ερωτήσεις παρατηρώντας μόνο το trace πακέτο). Ονομάστε τα πεδία αυτά.

2.Από τα πεδίο περιεχομένων του πακέτου υπολογίστε το μέγεθος (σε bytes) του καθενός UDP header πεδίου.

3.Η τιμή στο Length field σε ποιά τιμή αναφέρεται; Επαληθεύεστε την απάντηση σας με το UDP πακέτο που συλλάβαται

4.Ποιός είναι ο μεγαλύτερος αριθμός bytes που μπορεί να συμπεριλιφθει σε ένα UDP payload;

5.Ποίος είναι ο μεγαλύτερος πιθανός αριθμός της θύρας πηγής;

6.Ποίος είναι ο αριθμός πρωτοκόλλου για το UDP? Δώστε την απάντηση σας σε 16δική και 10δική μορφή. (Για να απαντήσετε στην ερώτηση αυτή, πρέπει να κοιτάξετε στο IP header)

7.Ψάξτε "UDP" στο Google και προσδιορίστε τα πεδία στα οποία το UDP checksum υπολογίστηκε

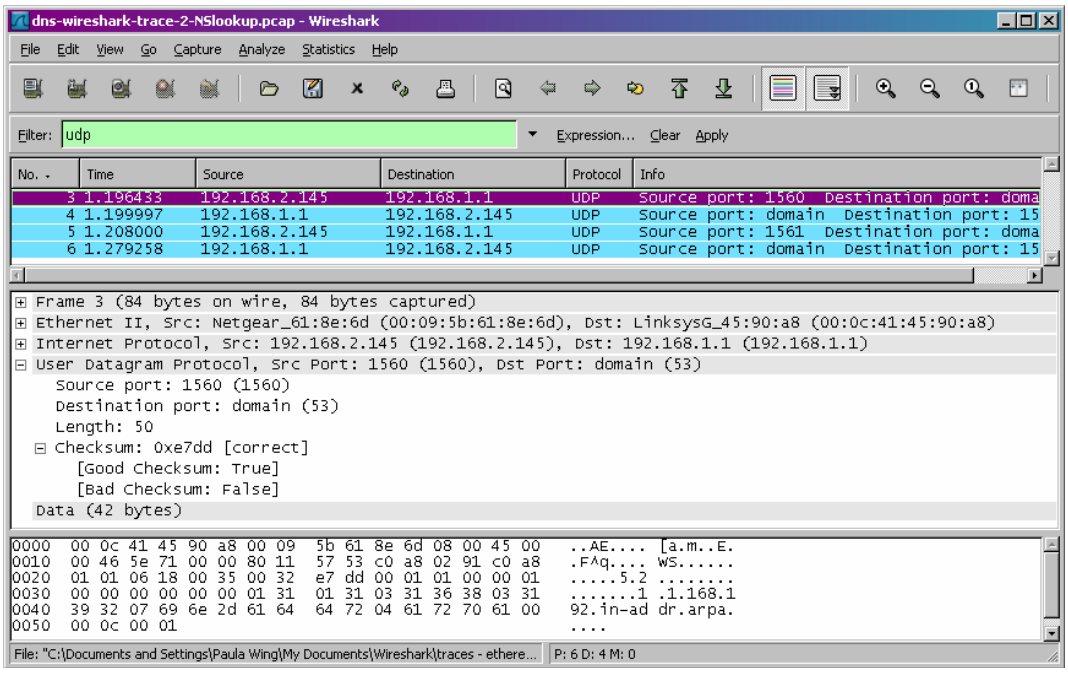
8.Εξετάστε ένα ζεύγος UDP πακέτων από τα οποία το πρώτο πακέτο στέλνεται από τον δικό σας host και το δεύτερο πακέτο είναι επανάληψη του πρώτου. Περιγράψτε τη σχέση μεταξύ των port numbers στα δυο πακέτα

Extra credit

1.Συλλάβετε ένα μικρό UDP πακέτο.Επαληθεύσετε χειρονακτικά το άθροισμα σε αυτό το πακέτο. Παρουσιάστε όλη την εργασία και εξηγήστε όλα τα βήματα

(1) Κατεβάστε το zip αρχείο http://gaia.cs.umass.edu/wireshark-labs/wireshark-traces.zip
και εξάγετε το αρχείο Tcp-ethereal-trace-1.Τα traces σε αυτό το zip συλλέχθηκαν τρέχοντας το wireshark σε ένα υπολογιστή του συγγραφέα ακολουθόντας τα βήματα που υποδεικνύει το εργαστήριο wireshark.Αφού κατεβάσετε το trace,μπορείτε να το φορτώσετε στο wireshark και να δείτε το trace χρησιμοποιώντας το μενού του file,επιλέγοντας open και μετά επιλέγουμε udp-wireshark-trace αρχείο trace.

ΑΠΑΝΤΗΣΕΙΣ



UDP Header Fields

1.Περιέχει 4 πεδία : θύρα πηγής, θύρα προορισμού, μέγεθος και άθροισμα

2.Και τα 2 έχουν μέγεθος 2 bytes

3.Είναι το άθροισμα των 8 bytes της header συν τα 42 data bytes που περικλείονται

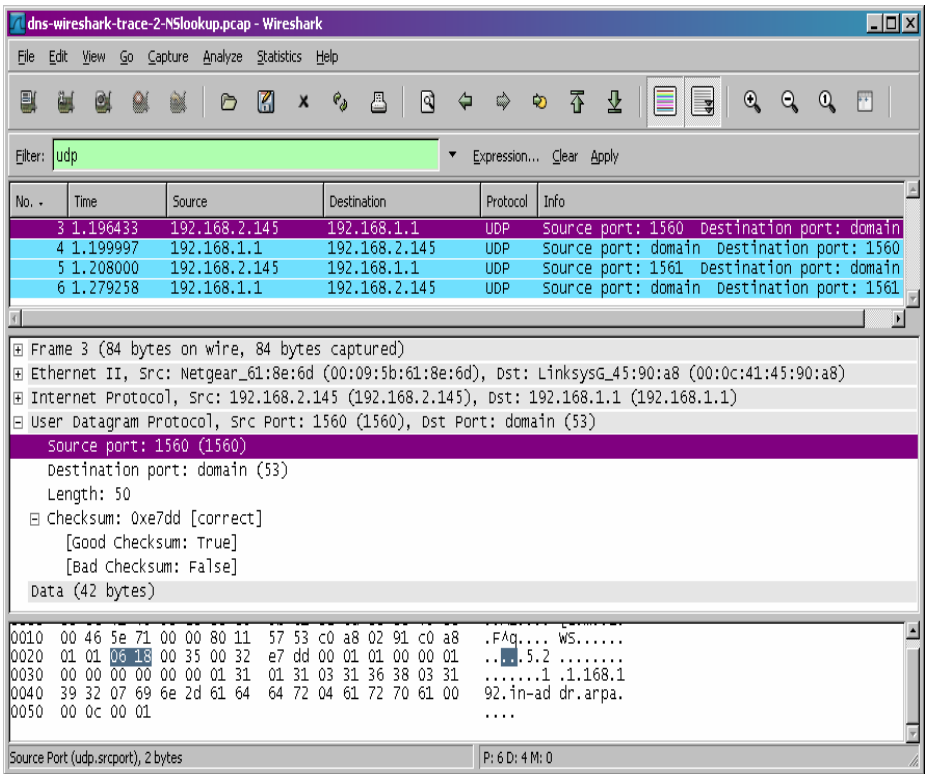
4.Ο μεγαλύτερος αριθμός bytes που μπορεί να συμπεριληφθεί σε ένα UDP payload είναι $2_{16} - 1 = 65535$ μείον τα bytes τις επικεφαλίδας. Αυτό σημαίνει 65535 – 8 = 65527 bytes

5.Ο μεγαλύτερος πιθανός αριθμός της θύρας πηγής είναι $2_{16} - 1 = 65535$= 65535.

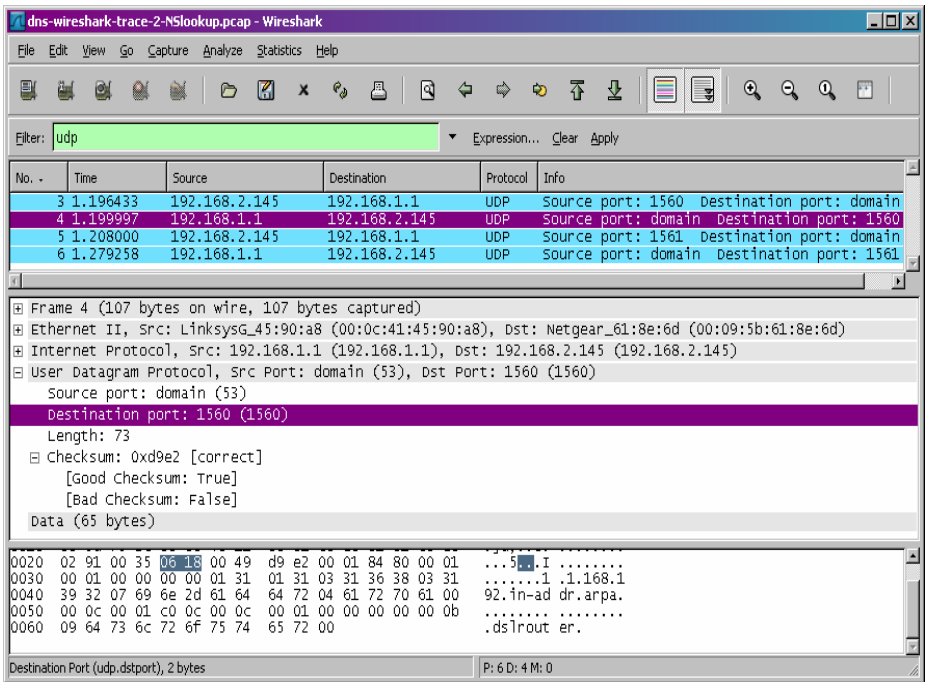
6.Ο αριθμός πρωτοκόλλου για το UDP σε 16δική μορφή είναι 0x11 και σε 10δική είναι 17.

7.Το UDP checksum έχει υπολογιστεί σαν 16-bit συμπλήρωμα ως προς το ένα του αθροίσματος της ψευδο επικεφαλίδας της πληροφορίας από την IP header, την UDP header και τα δεδομένα. Αυτό έχει στο τέλος μία ακολουθία μηδενικών ώστε να δημιουργηθεί μια πολλαπλότητα από δυο bytes. Αν το checksum είναι 0, τότε πρέπει να σεταριστεί στο 0Xffff

8.



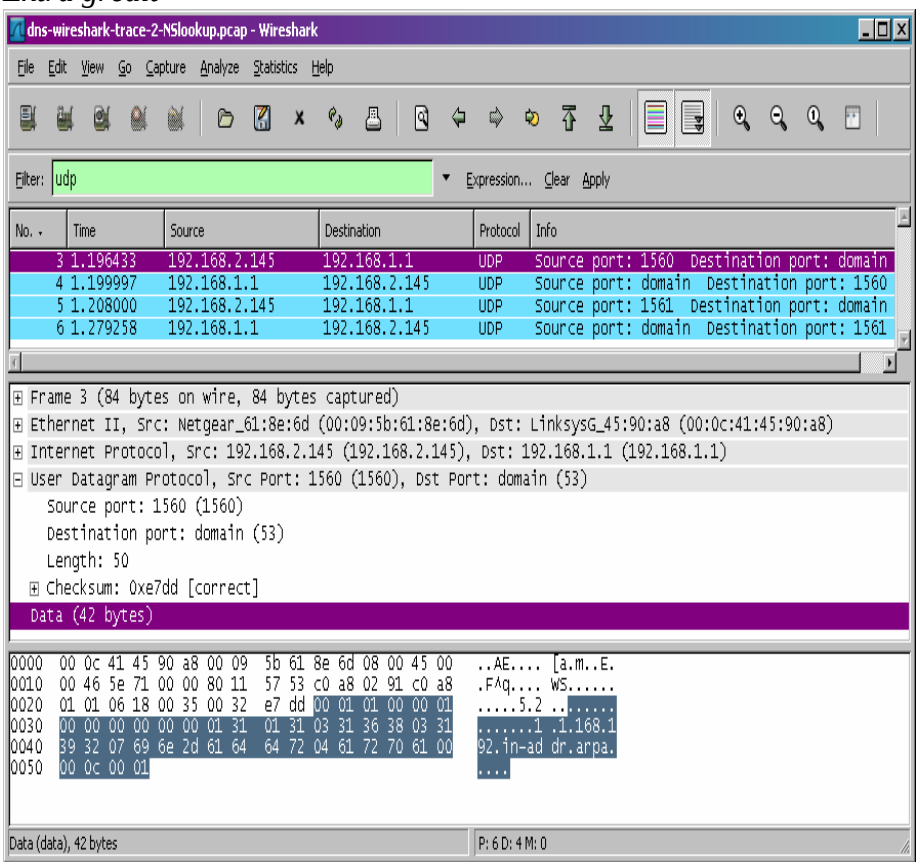
UDP σταλμένο από τον δικό μας host



UDP απόκριση προς τον δικό μας host

Η θύρα πηγής του UDP πακέτου που στέλνεται από τον host είναι ίδια με τη θύρα προορισμού του πακέτου απόκρισης, και αντιστρόφως η θύρα προορισμού του UDP πακέτου που στέλνεται από τον host είναι ίδια με την θύρα πηγής του πακέτου απόκρισης

Extra gredit



Κεφάλαιο 8
Ασκήσεις επίδειξης του πρωτοκόλλου IP

Το Πρωτόκολλο Διαδικτύου (IP, Internet Protocol), αποτελεί το κύριο πρωτόκολλο επικοινωνίας για τη μετάδοση αυτοδύναμων πακέτων (datagrams), δηλαδή, πακέτων δεδομένων, σε ένα διαδίκτυο. Το Πρωτόκολλο IP είναι υπεύθυνο για τη δρομολόγηση των πακέτων δεδομένων ανάμεσα στα διάφορα δίκτυα, ανεξάρτητα από την υποδομή τους, και αποτελεί το κύριο πρωτόκολλο πάνω στο οποίο είναι βασισμένο το Διαδίκτυο. Ανήκει στο Επίπεδο Δικτύου, στο Μοντέλο Διαστρωμάτωσης TCP/IP και καθορίζει τη μορφή των πακέτων που στέλνονται μέσω ενός διαδικτύου, καθώς και τους μηχανισμούς που χρησιμοποιούνται για την προώθηση των πακέτων από έναν υπολογιστή προς έναν τελικό προορισμό μέσω ενός ή περισσότερων δρομολογητών.

Σε αυτό το εργαστήριο θα εξετάσουμε το IP πρωτόκολλο, εστιάζοντας κυρίως στο IP datagram (πακέτα τα οποία δρομολογούνται ανεξάρτητα από τα προηγούμενά τους και περιέχουν τη διεύθυνση προορισμού). Αυτό θα γίνει αναλύοντας ενα trace από IP datagrams που στέλνονται και συλλαμβάνονται κατά την εκτέλεση του traceroute πρόγραμματος (το πρόγραμμα traceroute θα το διερευνήσουμε λεπτομερέστερα στο εργαστήριο Wireshark ICMP). Θα εξετάσουμε τα διάφορα πεδία ενός IP datagram και θα μελετήσουμε τον κατακερματισμό (fragmentation) στο IP.

8.1 Συλλαμβάνοντας Πακέτα από μια Εκτέλεση του traceroute

Προκειμένου να παράγουμε ένα trace από IP datagramς για αυτό το εργαστήριο, θα χρησιμοποιήσουμε το traceroute πρόγραμμα ώστε να στείλουμε datagrams διαφορετικών μεγεθών προς μια κατευθύνση X. Το traceroute λειτουργεί στέλνοντας αρχικά ένα ή περισσότερα datagrams με το πεδίο time to live (το πεδίο TTL περιέχει το χρονικό διάστημα σε δευτερόλεπτα μέσα στο οποίο θα πρέπει το πακέτο να έχει παραδοθεί) στην IP επικεφαλίδα (header) σεταρισμένο στο 1. Στην συνέχεια στέλνει ένα ή περισσότερα datagrams στον ίδιο προορισμό με TTL τιμή 2, στην συνέχεια στέλνει ένα ή περισσότερα datagrams στον ίδιο προορισμό με TTL τιμή 3, κ.ο.κ. Ο router πρέπει να μειώνει την TTL τιμή κατά 1 για κάθε datagram που λαμβάνει. Εάν το TTL μηδενιστεί ο router επιστρέφει ένα μήνυμα ICMP (type 11-TTL-exceeded) στον αποστολέα host. Άρα, ένα datagram με TTL=1 (που στέλνεται από τον host που εκτελεί το traceroute) αναγκάζει τον router που βρίσκεται σε απόσταση ενός άλματος από τον αποστολέα να στείλει πίσω στον αποστολέα ένα ICMP TTL-exceeded μήνυμα. Το datagram με TTL=2 θα έχει αποτέλεσμα ο router που βρίσκεται σε απόσταση 2 άλματών από τον αποστολέα να στείλει πίσω στον αποστολέα ένα ICMP μήνυμα . Το datagram με TTL=3 θα έχει αποτέλεσμα ο router που βρίσκεται σε απόσταση 3 άλματών από τον αποστολέα να στείλει πίσω στον αποστολέα ένα ICMP μήνυμα κ.ο.κ. Με αυτό τον τρόπο, ο host που εκτελεί το traceroute μπορεί να πληροφορηθεί την ταυτότητα των router μεταξύ του ίδιου και του προορισμού X εξετάζοντας τις IP διευθύνσεις πηγής των datagrams που περιέχουν τα μηνύματα ICMP 'TTL-exceeded'

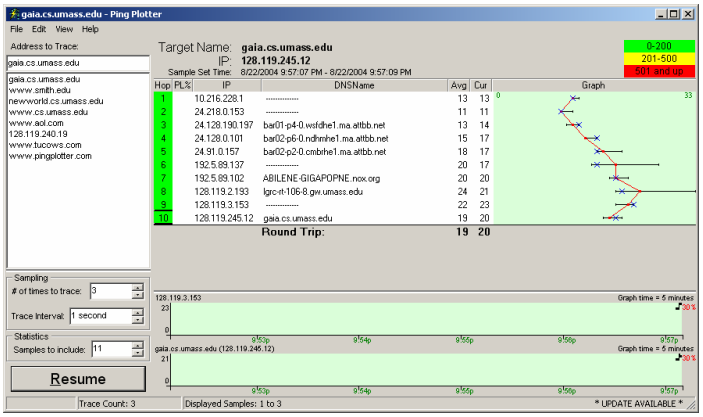
Στο εργαστήριο αυτό θα τρέξουμε το traceroute και θα το κάνουμε να στείλει daragrams με διαφορετικά μήκη.

WINDOWS-Το πρόγραμμα tracert (traceroute), που παρέχεται με τα windows ,και το οποίο χρησιμόποιουμε στο εργαστήριο ethereal για το ICMP, δεν επιτρέπει την αλλαγή

μεγέθους του μηνύματος ICMP ‘echo (ping) request’ που στέλνεται από το πρόγραμμα. Ένα καλύτερο πρόγραμμα traceroute για windows είναι το **pingplotter**, που είναι διαθέσιμο σε εκδόσεις freeware και shareware απο το site-http://www.pingplotter.com. Φορτώνουμε και εγκαταστούμε το pingplotter, και εκτελούμε μερικά traceroutes σε site της αρεσκείας μας. Το μέγεθος του μηνύνατος ICMP ‘echo request’ μπορεί να καθοριστεί ρητά, επιλέγοντας Edit >Options >Packet Options από το μενού και μετά συμπληρώνοντας το πεδίο packet size. Το προεπιλεγμένο μέγεθος πακέτου είναι 56 bytes. Αφού το Pingplotter στείλει μια σειρά πακέτων με αυξανόμενες τιμές TTL, ξαναρχίζει την διαδικασία αποστολής με TTL=1, αφού περιμένει για χρονικό διάστημα διάρκειας ίση με Trace Interval. Η τιμή του Trace Interval και ο αριθμός των χρονικών διαστημάτων μπορουν να καθοριστούν στο pingplotter

LINUX/UNIX-Στην εντολή traceroute του Unix, το μέγεθος των UDP datagrams που στέλνονται προς τον προορισμό μπορεί να καθορισθεί υποδεικνύοντας τον αριθμό των bytes στο datagram. Η τιμή αυτή εισάγεται στη γραμμή εντολής traceroute αμέσως μετά το όνομα ή την διεύθυνση προορισμού. Για παράδειγμα, η εντολή για να σταλούν datagrams των 2000 bytes προς τον προορισμό Gaia.cs.umass.edu είναι %traceroute gaia.cs.umass.edu 2000

Εκτελούμε τα παρακάτω βήματα
>Ξεκινάμε το Ethernal και τη σύλληψη πακέτων (capture>start) και πατάμε OK στο παράθυρο packet capture options του ethernal
>Εάν χρησιμοποιούμε windows, ξεκινάμε το pingplotter και εισάγουμε το όνομα προορισμού-στόχου στο πεδίο address to trace. Εισάγουμε τον αριθμό 3 στο πεδίο #of times to trace, ώστε να μην συλλέξουμε μεγάλο όγκο δεδομένων, επιλέγουμε Edit> Advanced Options>Packet Options από το μενού, εισάγουμε την τιμή 56 στο πεδίο packet size και πατάμε OK. Στη συνέχεια πατάμε Trace. Το παράθυρο του Pingplotter θα πρέπει να μοιάζει με το ακόλουθο



Σχ.34 Το παράθυρο του Pingplotter

Στη συνέχεια στέλνουμε ένα σύνολο από datagrams με μεγαλύτερο μήκος – επιλέγουμε Edit>Advanced Options>Packet Options, εισάγουμε την τιμή 2000 στο πεδίο Packet Size και κατόπιν πατάμε OK, έπειτα πατάμε το κουμπί Resume. Τέλος, στέλνουμε ένα σύνολο από datagrams με ακόμη μεγαλύτερο μήκος- επιλέγουμε Edit>Advanced Options>Packet Options, εισάγουμε την τιμή 3500 στο πεδίο Packet Size και κατόπιν πατάμε OK. Έπειτα πατάμε Resume. Σταματάμε τη σύλληψη πακέτων από το Ethereal

>Εαν χρησιμοποιούμε Unix, εισάγουμε 3 εντολές traceroute, μια με μήκος UDP datagram ίσο με 56 bytes, μια με 2000 bytes και μια με μήκος ίσο με 3500 bytes. Σταματάμε τη σύλληψη πακέτων από το Ethereal.

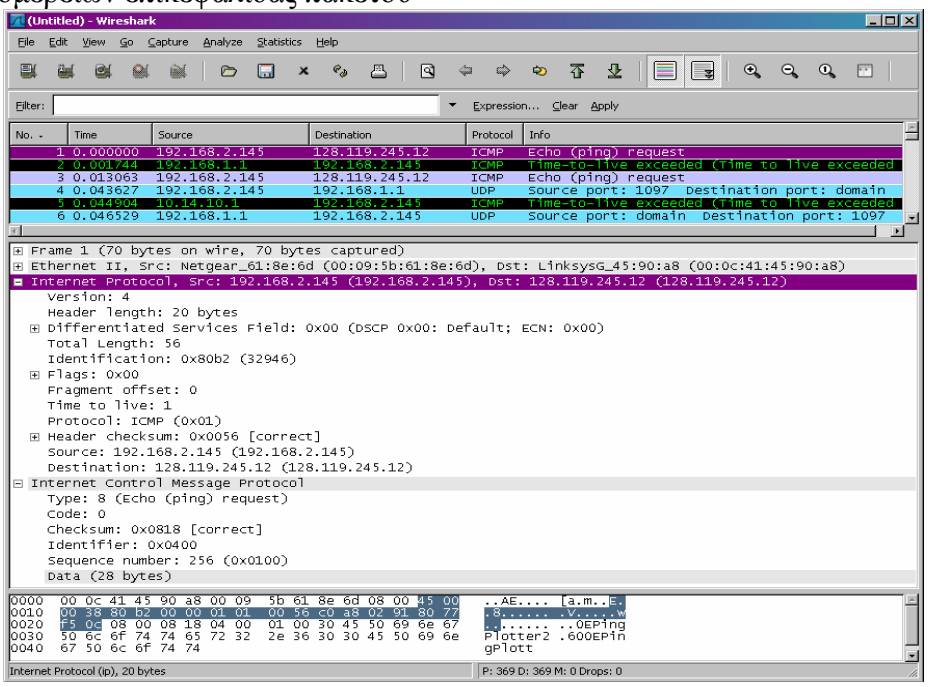
Εάν δεν είμαστε σε θέση να τρέξουμε το Ethernal σε μια ζωντανή σύνδεση δικτύου, μπορούμε να φορτώσουμε ένα αρχείο με το trace πακέτων που συλλέχθηκαν από το Ethernal ενώ εκτελούνταν τα βήματα που περιγράφονται στο εργαστήριο στον υπολογιστή του συγγραφέα (1). (Ενδεχομένως να διαπιστώσετε οτι αξίζει να φορτώσετε αυτό το trace ακόμη και αν έχετε συλλάβει το δικό σας, να το χρησιμοποιήσετε παράλληλα με το δικό σας καθώς διερευνάτε τις ερωτήσεις που τίθενται παρακάτω)

8.2 Μια Ματιά στο trace

Στο trace που συλλέξαμε θα πρέπει να' μαστε σε θέση να δούμε την σειρά των μηνυμάτων ICMP 'echo request' (για windows) ή των UDP Segments(για unix) που στάλθηκαν από τον υπολογιστή μας και τα μηνύματα ICMP 'TTL-exceeded' που επεστράφησαν στον υπολογιστή μας από τους ενδιάμεσους δρομολόγητες. Στις ερωτήσεις που ακολουθούν υποθέτουμε οτι χρησιμοποιείται windows. Όπου είναι δυνατό, η απάντηση σας θα πρέπει να συνοδεεύται από μια εκτύπωση των πακέτων του trace που χρησιμοποιείσατε για την απάντηση των ερωτήσεων. Σημειώστε πάνω στην εκτύπωση τα σημεία που αιτιολογούν την απάντηση σας. Για την εκτύπωση ενος πακέτου, χρησιμοποιείστε File>Print, επιλέξτε Selected Packet Only, επιλέξτε Packet summary και επιλέξτε το ελάχιστο ποσό λεπτομερειών πακέτου που χρειάζεται για να απαντήσετε στην ερώτηση.

ΕΡΩΤΗΣΕΙΣ

1.Επιλέξτε το πρώτο μήνυμα ICMP 'echo request' που στάλθηκε απο τον υπολογιστή σας και αναπτύξτε το τμήμα που αφορά το Internet Protocol στο παράθυρο λεπτομερειών επικεφαλίδας πακέτου



Σχ.35 παράθυρο λεπτομερειών επικεφαλίδας πακέτου

Ποιά η διεύθυνση IP του υπολογιστή σας;

2.Ποιά η τιμή του πεδίου upper layer protocol στην επικεφαλίδα IP του πακέτου;

3.Ποιός ο αριθμός των bytes στην επικεφαλίδα IP; Ποιός ο αριθμός των bytes στο ωφέλιμο φορτίο (payload) του IP datagram; Εξηγήστε τον τρόπο με τον οποίο υπολογίσατε τον αριθμό των bytes ωφέλιμου φορτίου

4.Έχει υποστεί κατακερματισμό το δεδομένο IP datagram; εξηγήστε. Στη συνέχεια ταξινομήστε τα πακέτα του trace σύμφωνα με τη διεύθυνση IP πηγής κάνοντας κλικ στην επικεφαλίδα της στήλης Source. Δίπλα στη λέξη Source θα πρέπει να εμφανισθεί ένα μικρό βέλος που δείχνει προς τα κάτω. Εάν το βέλος δείχνει προς τα πάνω, κάνετε ξανά κλικ στην επικεφαλίδα της στήλης Source. Επιλέξτε το πρώτο μήνυμα ICMP 'echo request' που στάλθηκε από τον υπολογιστή σας και αναπτύξτε το τμήμα που αφορά το Internet Protocol στο παράθυρο λεπτομερειών επικεφαλίδας του επιλεγμένου πακέτου. Στο παράθυρο καταλόγου συλληφθέντων πακέτων θα πρέπει να εμφανίζονται μετά από αυτό το πρώτο μήνυμα ICMP όλα τα επακόλουθα μηνύματα ICMP (μαζί ίσως με πρόσθετα πακέτα που στάλθηκαν από άλλα πρωτόκολλα που τρέχουν στον υπολογιστή σας). Χρησιμοποιήστε το πλήκτρο με το βέλος που δείχνει προς τα κάτω (down arrow)στο πληκτρολόγιο σας για να διατρέξετε τα μηνύματα ICMP που στάλθηκαν.

5.Στην ακολουθία μηνυμάτων ICMP που στάλθηκαν από τον υπολογιστή σας, ποιά από τα πεδία του IP datagram μεταβάλλονται πάντοτε από το ένα datagram στο επόμενο;

6.Ποιά πεδία παραμένουν αμετάβλητα; Ποιά απο τα πεδία πρέπει να παραμείνουν αμετάβλητα; Ποια πρέπει να αλλάξουν; Για ποιό λόγο;

7.Περιγράψτε τον τρόπο μεταβόλης της τιμής του πεδίου Identification της επικεφαλίδας από datagram σε datagram. Στη συνέχεια, και ενώ τα πακέτα παραμένουν ταξινομημένα σύμφωνα με τη διεύθυνση πηγής, εντοπίστε την ακολουθία αποκρίσεων ICMP 'TTL-exceeded' που στάλθηκαν στον υπολογιστή σας απο τον κοντινότερο (σε απόσταση ενός άλματος) δρομολόγητη.

8.Ποιές οι τιμές των πεδίων Identification και TTL;

9.Οι τιμές αυτές παραμένουν αμετάβλητες για όλες τις αποκρίσεις ICMP 'TTL-exceeded' που στάλθηκαν στον υπολογιστή σας απο τον κοντινότερο δρομολόγητη;
Για ποιό λόγο;

Κατακερματισμός (fragmentation)

Ταξινομήστε τον κατάλογο πακέτων και πάλι σύμφωνα με το χρόνο κάνοντας κλικ στη στήλη Time

ΕΡΩΤΗΣΕΙΣ

10.Βρείτε το πρώτο μήνυμα ICMP 'echo request' το οποίο στάλθηκε απο τον υπολογιστή σας αφού αλλάξατε την τιμή του Packet Size στο Pingplotter σε 2000. Έχει

κατακερματιστεί το μήνυμα αυτό σε περισσότερα από ένα IP datagrams; {Σημείωση>εάν το πακέτο δεν έχει κατακερματιστεί, φορτώστε το αρχείο zip http://gaia.cs.umass.edu/ethereal-labs/ethernal-traces.zip και εξάγετε το trace πακέτων ip-ethernal-trace-1. Έαν ο υπολογιστή σας χρησιμοποιεί μια διεπαφή Ethernet, τότε ένα μέγεθος πακέτου ισο με 2000 bytes πρέπει να προκαλέσει κατακερματισμό(2)}

11.Εκτυπώστε το πρώτο τεμάχιο (fragment) του κατακερματισμένου IP datagram. Ποία πληροφορία στην επικεφαλίδα IP υποδεικνύει οτι το datagram έχει κατακερματιστεί; Ποιά πληροφορία στην επικεφαλίδα IP υποδεικνύει οτι πρόκειται για το πρώτο και όχι κάποιο επακόλουθο τεμάχιο;Ποιό το μήκος αυτού του IP datagram;

12.Εκτυπώστε το δεύτερο τεμάχιο του κατακερματισμένου IP datagram. Ποία πληροφορία στην επικεφαλίδα IP υποδεικνύει οτι δεν πρόκειται για το πρώτο τεμάχιο του datagram; Υπάρχουν περισσότερα τεμάχια που έπονται; εξηγήστε

13.Σε ποιά πεδία της επικεφαλίδας IP διαφέρουν το πρώτο και δεύτερο τεμάχιο; Εντοπίστε τώρα το πρώτο μήνυμα ICMP 'echo request' που στάλθηκε από τον υπολογιστή σας αφου αλλαξατε την τιμή του Packet Size στο pingplotter σε 3500

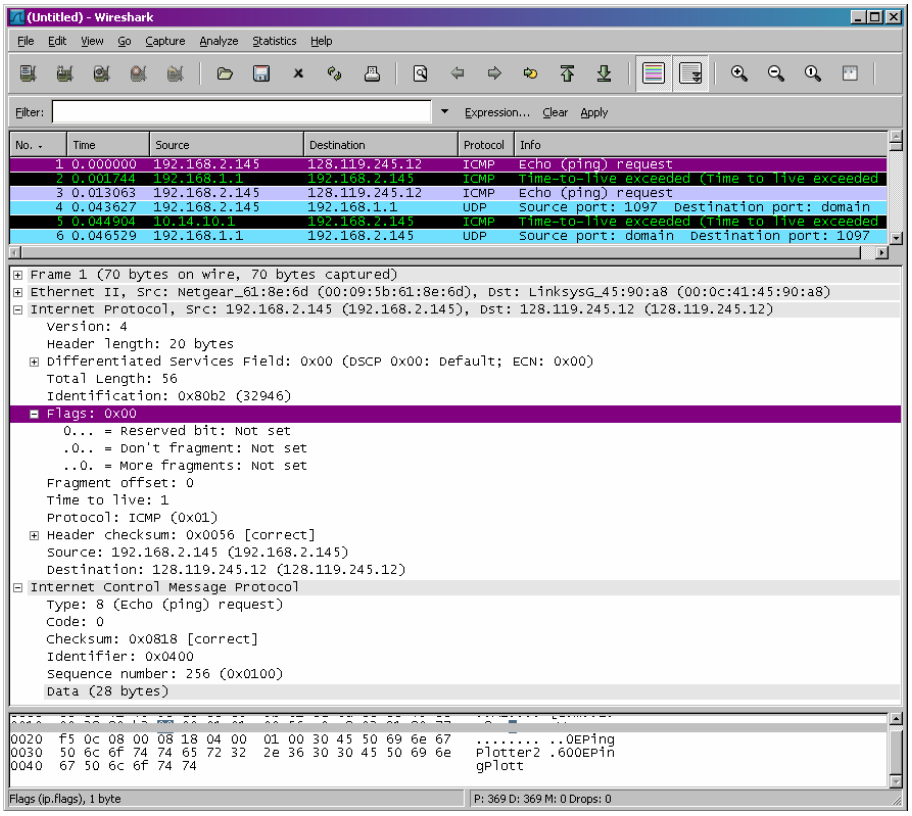
14.Πόσα τεμάχια δημιουργήθηκαν από το αρχικό datagram;

15.Ποιά πεδία της επικεφαλίδας IP μεταβάλλονται απο τεμάχιο σε τεμάχιο;

(1) φορτώστε το αρχείο zip http://gaia.cs.umass.edu/ethernal-labs/ethernal-traces.zip Και εξάγετε το αρχείο ip-ethernal-trace-1.αφου λάβετε το trace,Μπορείτε να το φορτώστε στο ethereal και να το δείτε στο Παράθυρο χρησιμοποιώντας το μενού File>>Open και Στη συνέχεια επιλέγοντας το αρχείο ip-ethernal-trace-1

(2) Τα πακέτα που περιέχονται στο αρχείο ip-ethernal-trace-1 του αρχείου http://gaia.cs.umass.edu/ethereal-labs/ethernal-traces.zipΈχουν όλα μήκος μικρότερο από 1500 bytes.Αυτό οφείλεται στο γεγονος οτι ο υπολογιστής ο οποίος χρησιμοποιήθηκε για τη συλλογή του trace έχει μια κάρτα Ethernet που περιορίζει το μέγιστο μήκος ενός πακέτου IP σε 1500 bytes.(40 bytes δεδομένα επικεφαλίδας TCP/IP και 1460 bytes ωφέλιμο φορτίο πρωτοκόλλων ανωτέρου επιπέδου).Αυτή η τιμή των 1500 bytes είναι το καθιερωμένο μέγιστο μήκος που επιτρέπει το Ethernet.Εαν το δικό σας trace εμφανίζει datagram με μήκος μεγαλύτερο από 1500 bytes και ο υπολογιστής σας χρησιμοποιεί μια σύνδεση Ethernet ,τότε το Ethernet αναφέρει λανθασμένη τιμή για το μήκος του IP datagram.ίναι πιθανό επίσης να δείχνει μόνο ένα μεγάλοIP datagram αντί για πολλαπλά datagrams.Αυτή η ασυνέπεια στα αναφερόμενα μήκη των segments οφείλεται στην αλληλεπίδραση μεταξυ του Ethernet driver και του λογισμικου ethereal.Σε περίπτωση που αντιμετωπίζετε αυτό το πρόβλημα,συνιστούμε να χρησιμοποιήσετε για το εργαστήριο αυτο το trace του αρχείου ip-ethernal-trace-1

ΑΠΑΝΤΗΣΕΙΣ



1.Η διεύθυνση του υπολογιστή μου είναι 192.168.1.46

2. ICMP (0x01)

3.Υπάρχουν 20 bytes στην IP επικεφαλίδα και 56 bytes είναι το συνολικό μέγεθος, το οποίο μας δίνει 36 bytes ωφέλιμο φορτίο του IP datagram

4.Τα περισσότερα fragments έχουν τιμή 0 άρα δεν έχει υποστεί κατακερματισμό

5.Time to live και Header checksum μεταβάλλονται συνεχώς

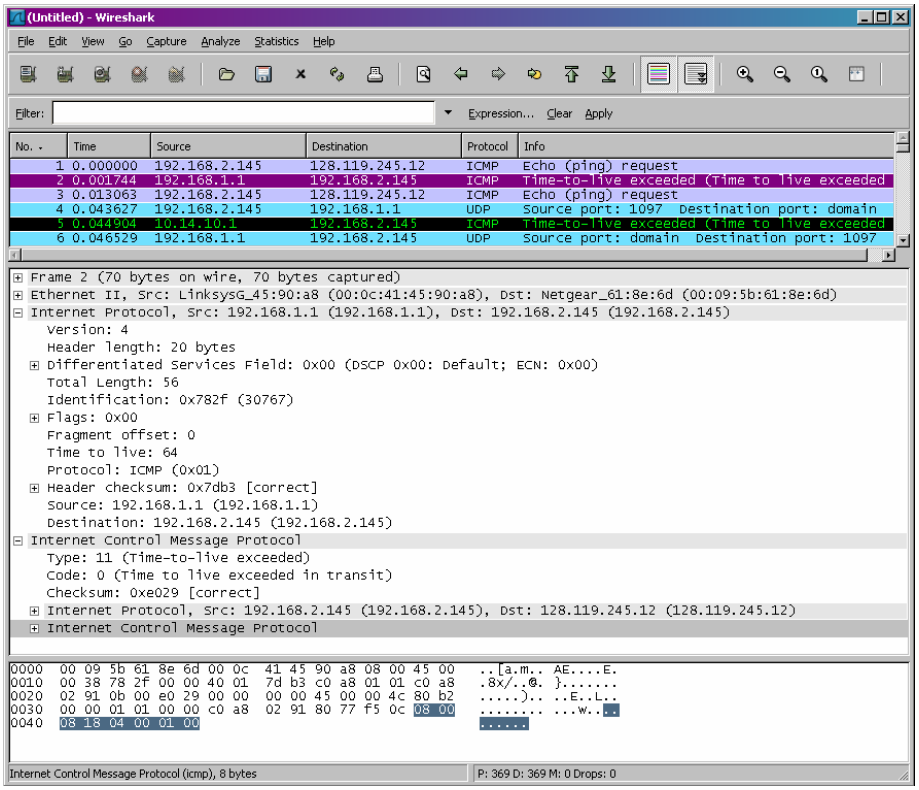
6.Τα πεδία που παραμένουν αμετάβλητα είναι
• Είδος (αν χρησιμοποιούμε IPv4 για όλα τα πακέτα)
• Μέγεθος επικεφαλίδας (αν υπάρχουν ICMP πακέτα)
• IP πηγής (αν στέλνουμε από την ίδια πηγή)
• IP προορισμού (αν στέλνουμε στον ίδιο προορισμό)
• Διακεκριμένες υπηρεσίες (αν όλα τα πακέτα είναι ICMP ίδιου τύπου)
• Ανώτερο επίπεδο πρωτοκόλλου (αν είναι όλα ICMP πακέτα)

Τα πεδία που πρέπει να παραμένουν αμετάβλητα είναι
• Είδος (αν χρησιμοποιούμε IPv4 για όλα τα πακέτα)
• Μέγεθος επικεφαλίδας (αν υπάρχουν ICMP πακέτα)
• IP πηγής (αν στέλνουμε από την ίδια πηγή)

• IP προρισμού (αν στέλνουμε στον ίδιο προορισμό)
• Διακεκριμένες υπηρεσίες (αν όλα τα πακέτα είναι ICMP ίδιου τύπου)
• Ανώτερο επίπεδο πρωτοκόλλου (αν είναι όλα ICMP πακέτα)

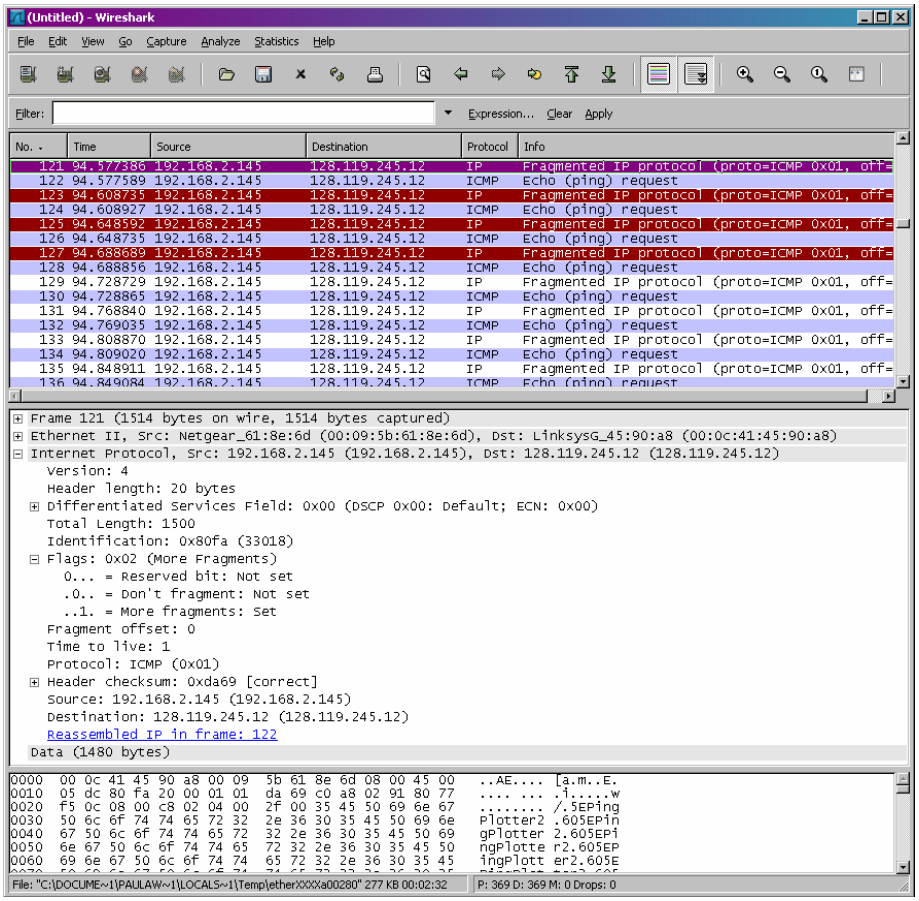
Τα πεδία που να πρέπει να αλλάξουν
• Ταυτότητα (τα IP πακέτα πρέπει να έχουν διαφορετικές ταυτότητες)
• Time to live (το traceroute προσαυξάνει με κάθε επακόλουθο πακέτο)
• Άθροισμα επικεφαλίδας (αφού αλλάζει η επικεφαλίδα, αλλάζει και το άθροισμα)

7.Το πεδίο IP header Identification fields προσαυξάνει με κάθε ICMP Echo (ping) αίτηση.



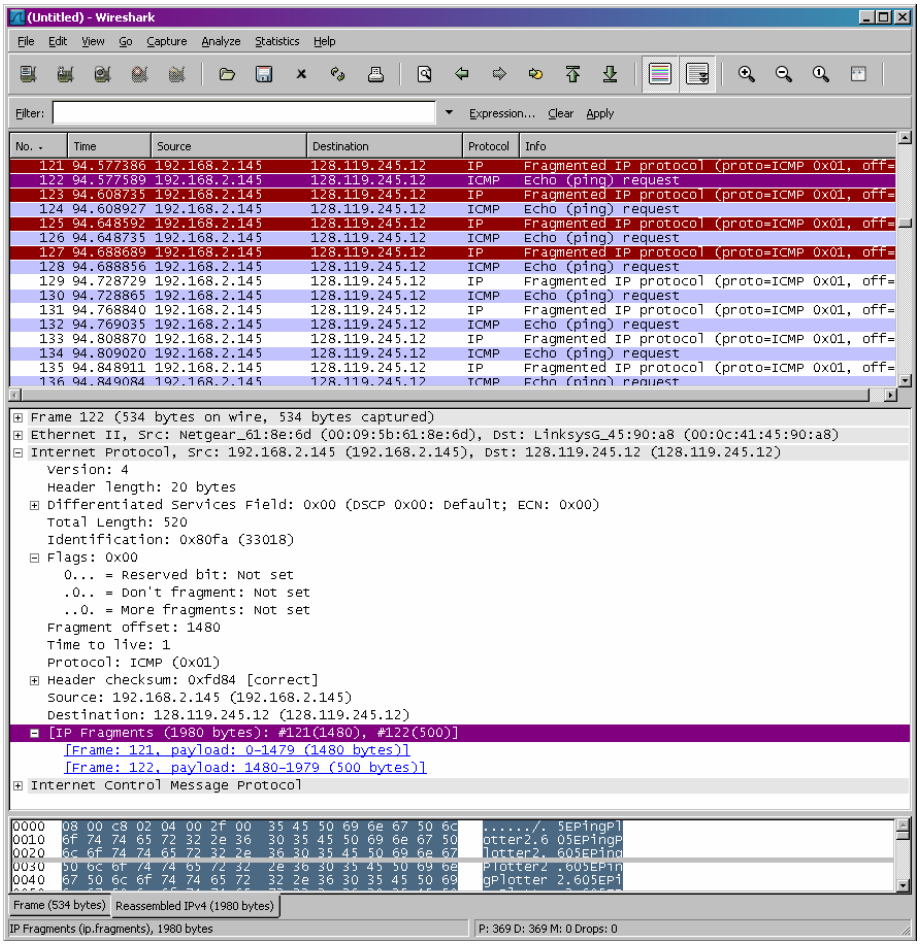
8. Identification: 30767
TTL: 64

9.Οι τιμές αλλάζουν για όλες τις αποκρίσεις ICMP 'TTL-exceeded' επειδή το πεδίο identification έχει μια μοναδική τιμή. Όταν δύο ή περισσότερα IP Datagrams έχουν την ίδια identification τιμή, σημαίνει πως αυτά τα IP datagrams είναι κομμάτια (fragments) ενός μοναδικού μεγάλου IP datagram. Το TTL πεδίο παραμένει αμετάβλητο επειδή το TTL για το πρώτο hop router είναι πάντα το ίδιο.



10. Ναι,έχει κατακερματιστεί το μήνυμα αυτό σε περισσότερα από ένα IP datagrams;

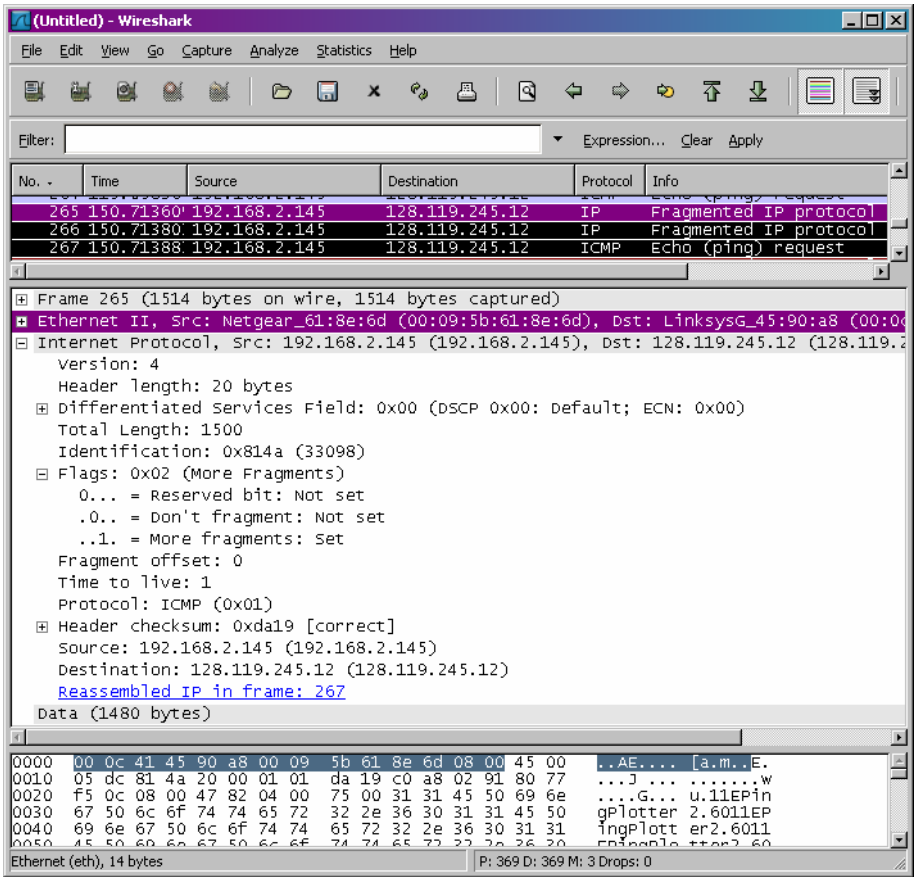
11.Το Flags bit για περισσότερα τεμάχια είναι σεταρισμένο,που σημαίνει ότι το datagram έχει κατακερματιστεί. Αν το fragment offset είχε την τιμή 0, θα ξέραμε ότι είναι το πρώτο τεμάχιο. Το πρώτο τεμάχιο έχει συνολικό μέγεθος 1500, συμπεριλαμβανομένου την επικεφαλίδα



12.Το fragment offset είναι 1480, άρα δεν είναι το πρώτο. Είναι το τελευταίο τεμάχιο μιας και τα περισσότερα δεν είναι σεταρισμένα

13.Συνολικό μέγεθος, flags, fragment offset και άθροισμα

14.3



15.Το fragment offset και το άθροισμα. Μεταξύ των δυο πρώτων και του τελευταίου πακέτου βλέπουμε αλλαγές στο συνολικό μέγεθος και στις flags. Τα δυο πρώτα έχουν συνολικό μέγεθος 1500 με τα περισσότερα σεταρισμένα fragments bit set στο 1, και το τελευταίο έχει συνολικό μέγεθος 540, με τα περισσότερα fragments σεταρισμένα στο 0.

Κεφάλαιο 9
Ασκήσεις επίδειξης του πρωτοκόλλου ICMP

Το πρωτόκολλο Internet Control Message Protocol (ICMP) είναι ένα από τα βασικά πρωτόκολλα του διαδικτύου. Χρησιμοποιείται κυρίως από τα λειτουργικά συστήματα των ηλεκτρονικών υπολογιστών ενός δικτύου για την ανταλλαγή μηνυμάτων λάθους, όπως για παράδειγμα την έλλειψη κάποιας υπηρεσίας από έναν server ή την απουσία ενός υπολογιστή από το δίκτυο.
Στο εργαστήριο αυτό θα εξετάσουμε μερικά χαρακτηριστικά του πρωτοκόλλου ICMP
>τα μηνύματα ICMP που προκύπτουν απο το πρόγραμμα Ping
>τα μηνύματα ICMP που προκύπτουν απο το πρόγραμμα Traceroute
>τη μορφή και το περιεχόμενο ενός μηνύματος ICMP

Το εργαστήριο αυτό περιγράφεται με βάση το λειτουργίκο σύστημα Windows. Είναι εύκολο να μεταφρασθεί σε περιβάλλον Unix ή Linux

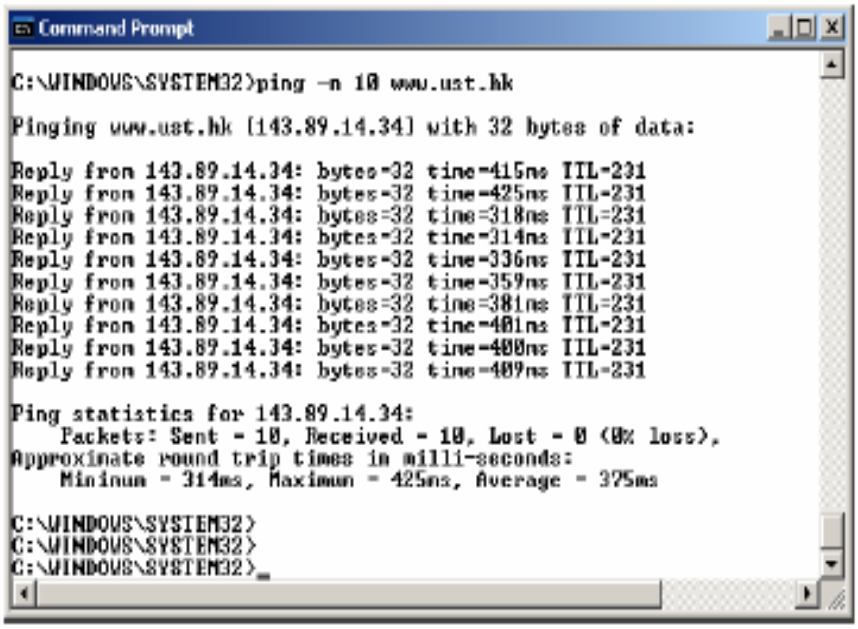
9.1 ICMP και PING

Θα ξεκινήσουμε να εξετάζουμε το ICMP με τη σύλληψη πακέτων που προκύπτουν από το πρόγραμμα Ping. Ενδεχομένως θα θυμάστε οτι το πρόγραμμα ping είναι ένα εργαλείο το οποίο επιτρέπει σε ένα χρήστη του διαδικτύου (σε κάποιον διαχειριστή δικτύου π.χ.) να επαληθεύσει εάν ένα τερματικό σύστημα (host) βρίσκεται σε λειτουργία ή όχι. Το πρόγραμμα Ping στον host πηγής στέλνει ένα πακέτο στη IP διεύθυνση του host στόχου. Έαν ο στόχος βρίσκεται σε λειτουργία, το πρόγραμμα ping στον host στόχο ανταποκρίνεται στέλνοντας ένα πακέτο πίσω στον host πηγής. Και τα 2 πακέτα ping, είναι πακέτα ICMP

Εκτελούμε τα ακόλουθα βήματα (1)
>Ανοίγουμε την εφαρμόγη Command Prompt των windows (η οποία βρίσκεται στο φάκελο Accessories)
>Ξεκινάμε τον packet sniffer και τη σύλληψη πακέτων
>Η εντολή ping βρίσκεται στο c;\windows\system32, οπότε πληκτρολογούμε είτε "*ping –n 10 hostname*" ή "*c:\windows\system32\ping –n 10 hostname*" στη γραμμή εντολής MS-DOS, όπου hostname το όνομα ενός host που βρίσκεται σε διαφορετική ήπειρο. Εάν βρίσκετε εκτός Ασίας, μπορείτε να χρησιμοποιήσετε το όνομα www.ust.hk του web server στο Πανεπιστήμιο επιστήμης και τεχνολογίας του Hong Kong (Hong Kong University of Science and Technology,HKUST). Το όρισμα '-n 10' υποδεικνύει οτι πρέπει να σταλούν 10 μηνύματα Ping. Στη συνέχεια τρέχουμε το πρόγραμμα Ping πληκτρολογώντας return.
>Όταν τερματιστεί το πρόγραμμα Ping, σταματάμε τη σύλληψη πακέτων από το Ethernal

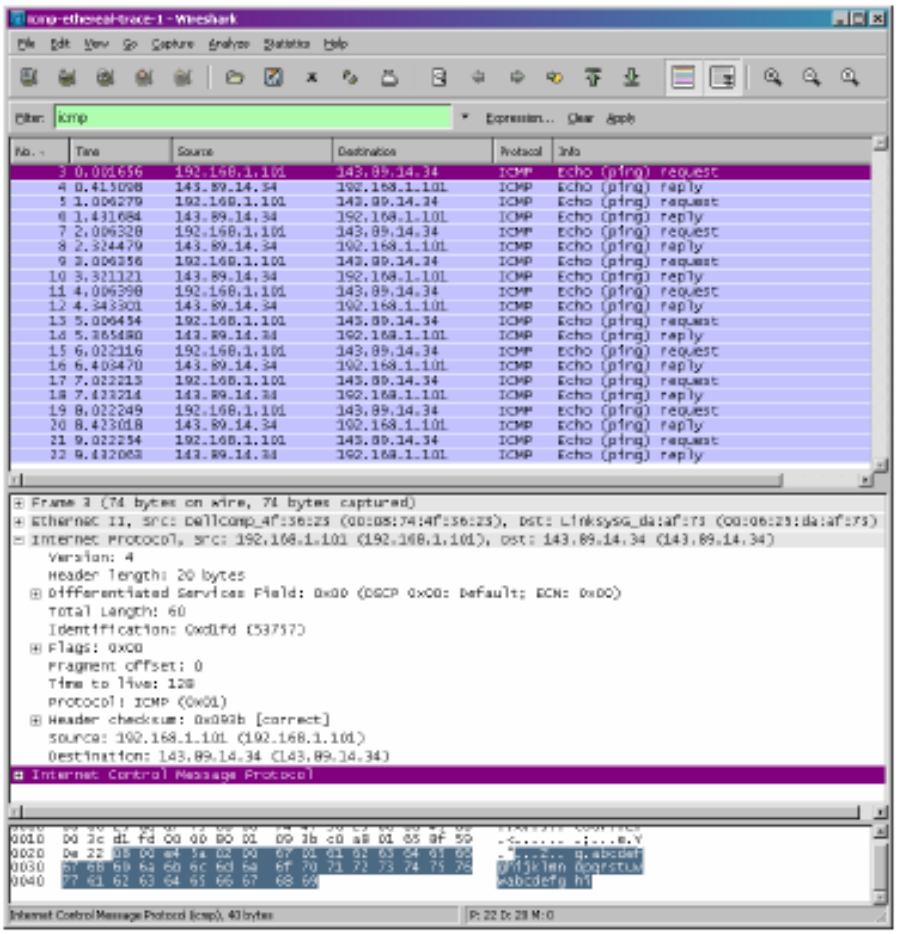
Στο τέλος του πειράματος το παράθυρο command prompt θα πρέπει να μοιάζει με αυτό του Σχ.36. Στο παράδειγμα αυτό, το πρόγραμμα Ping πηγής βρίσκεται στην πολιτεία Μασαχουσέτη των ΗΠΑ και το πρόγραμμα Ping προορισμού βρίσκεται στο Hong Kong. Από το παράθυρο αυτό διαπιστώνουμε οτι το πρόγραμμα Ping πηγής έστειλε 10 πακέτα ερωτημάτων (query packets) και έλαβε 10 αποκρίσεις. Παρατηρούμε επίσης οτι

η πηγή υπολογίζει το χρόνο διανομής μετ `επιστροφής (round-trip time, RTT) για κάθε απόκριση, ο οποίος για τα 10 πακέτα, κατά μέσο όρο, είναι ίσος με 375 msec.



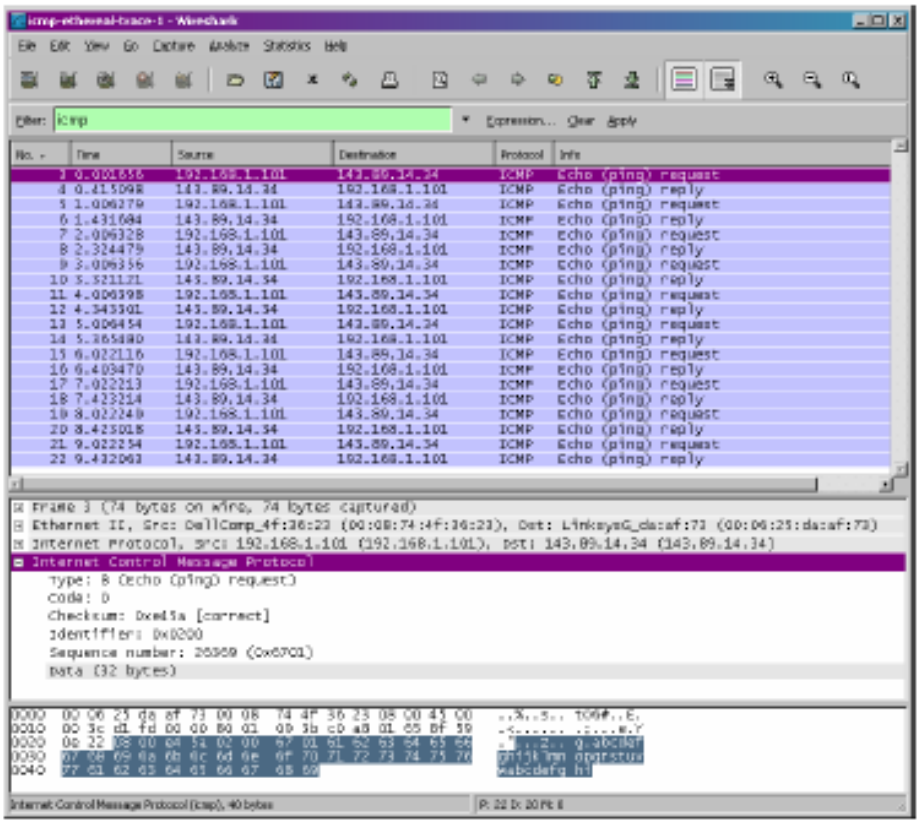
Σχ.36 το παράθυρο command prompt μετά την εισαγωγής της εντολής Ping

Στο σχήμα 37 απεικονίζεται η έξοδος του ethernal, αφού έχουμε εισάγει 'ICMP' στο παράθυρο του φίλτρου παρουσίασης. Παρατηρούμε οτι στον κατάλογο πακέτων εμφανίζονται 20 πακέτα - τα 10 ερωτήματα ping που στάλθηκαν από την πηγή και οι 10 αποκρίσεις ping που ελήφθησαν από την πηγή. Παρατηρούμε επίσης οτι η διεύθυνση IP της πηγής είναι μια ιδιωτική (private) διεύθυνση (που βρίσκεται πίσω από το NAT) της μορφής 192.168/12. Η διεθύνση IP είναι εκείνη του web server στο HKUST. Ας επικεντρώσουμε τώρα την προσοχή μας στο πρώτο πακέτο, το οποίο στάλθηκε από τον client. Στο παρακάτω σχήμα, το παράθυρο με τις λεπτομέρειες επικεφαλίδας πακέτων παρέχει πληροφοριές σχετίκα με το επιλεγμένο πακέτο. Βλέπουμε οτι το IP datagram που περιέχεται στο πακέτο έχει αριθμό πρωτοκόλλου που αντιστοιχεί στο ICMP. Αυτό σημαίνει οτι το ωφέλιμο φορτίο (payload) του IP datagram είναι ένα πακέτο ICMP.



Σχ.37 η έξοδος του ethernal για το πρόγραμμα ping με αναλυτική παρουσίαση της επικεφαλίδας του internet protocol (IP)

Το Σχ.38 εστιάζει στο ίδιο πακέτο ICMP αλλά στο παράθυρο με τις λεπτομέρειες επικεφαλίδας πακέτων παρουσιάζει αναλυτικά πληροφορίες σχετικά με το πρωτόκολλο ICMP. Παρατηρούμε οτι πρόκειται για ένα πακέτο ICMP με Type 8 και Code 0, δηλαδή για ένα ICMP πακέτο 'echo request'. Παρατηρούμε επίσης οτι το πακέτο ICMP αυτό περιέχει ένα άθροισμα ελέγχου (checksum), ένα πεδίο αναγνώρισης (identifier) και έναν αριθμό ακολουθίας (sequence number)



Σχ.38 η έξοδος του Ethernal για το πρόγραμμα Ping με αναλυτική παρουσίαση της επικεφαλίδας του Internet Control Message Protocol (ICMP)

Τι θα παραδώσετε

Θα πρέπει να παραδώσετε ένα screen shot του παραθύρου Command Prompt παρόμοιο με αυτό του Σχ.36. Όπου είναι δυνατό, η απάντηση σας σε κάθε μια από τις ακόλουθες ερωτήσεις θα πρέπει να συνοδεύται από μια εκτύπωση των πακέτων του trace που χρησιμοποιήσατε για να απαντήσετε στην ερώτηση. Σημειώστε πάνω στην εκτύπωση τα σημεία που αιτιολογούν την απάντηση σας. Για να εκτυπώσετε ένα πακέτο, χρησιμοποιήστε File>print, επιλέξτε selected packet only, επιλέξτε packet summary line και επιλέξτε το ελάχιστο ποσό λεπτομερείων που απαιτείται για να απαντησετε στην ερώτηση.

ΕΡΩΤΗΣΕΙΣ

1.Ποιά η διεύθυνση IP του δικού σας τερματικού συστήματος (host); Ποιά η διεύθυνση IP του host προορισμού

2.Γιατί ένα πακέτο ICMP δεν έχει αριθμούς θυρών πηγής και προορισμού;

3.Εξετάστε ένα απο τα πακέτα αίτησης Ping που στάλθηκαν από τον δικό σας host. Ποιές οι τιμές των πεδιών Type και Code; Τι άλλου είδους πεδία περιλαμβάνει αυτό το πακέτο ICMP; Πόσα bytes καταλαμβάνουν συνολικά τα πεδία checksum, identifier sequence field;

4.Εξετάστε το αντίστοιχο πακέτο απόκρισης Ping. Ποιές οι τιμές των πεδιών Type και Code; Τι άλλου είδους πεδία περιλαμβάνει αυτό το πακέτο ICMP; Πόσα bytes καταλαμβάνουν συνολικά τα πεδία checksum, identifier sequence field;

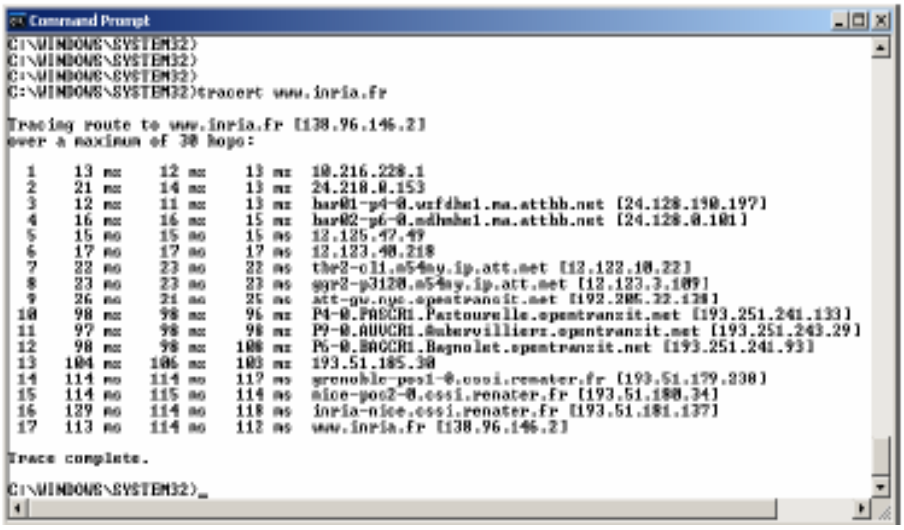
9.2 ICMP και Traceroute

Θα συνεχίσουμε να εξετάζουμε το ICMP με τη σύλληψη πακέτων που προκύπτουν από το πρόγραμμα Traceroute. Θυμόμαστε οτι το πρόγραμμα traceroute μπορεί να χρησιμοποιηθεί για να προσδιορίσουμε τη διαδρόμη που ακολουθεί ένα πακέτο από την πηγή στον προορισμό.

Το traceroute υλοποιείται με διαφορετικούς τρόπους στα λειτουργικά συστήματα unix/linux και στο λειτουργικό σύστημα windows. Στα unix/linux η πηγή στέλνει μια σειρά από πακέτα UDP στον προορισμό-στόχο χρησιμοποιώντας έναν απίθανο αριθμό θύρας προορισμού. Στα Windows, η πηγή στέλνει μια σειρά από πακέτα ICMP στον προορισμό-στόχο. Και στα δυο λειτουργικά συστήματα το πρόγραμμα στέλνει το πρώτο πακέτο με TTL=1, το δεύτερο πακέτο με TTL=2, κ.ο.κ. Υπενθυμίζεται οτι ο ένας δρομολογητής θα ελλαττώσει την τιμή TTL ενός πακέτου που διέρχεται από αυτόν. Όταν ένα πακέτο με TTL=1 φθάσει σε ένα δρομολογητή, ο δρομολογητής στέλνει πίσω στη πηγή ένα πακέτο ICMP σφάλματος. Στη συνέχεια θα χρησιμοποιήσουμε το πρόγραμμα tracert των windows. Μια πολύ καλύτερη έκδοση ενός προγράμματος traceroute για windows είναι το pingplotter (www.pingplotter.com). Θα χρησιμοποιήσουμε το pingplotter στο εργαστήριο Ethereal για το IP αφού παρέχει παρέχει πρόσθετη λειτουργικότητα την οποία θα χρειαστούμε σε εκείνο το εργαστήριο

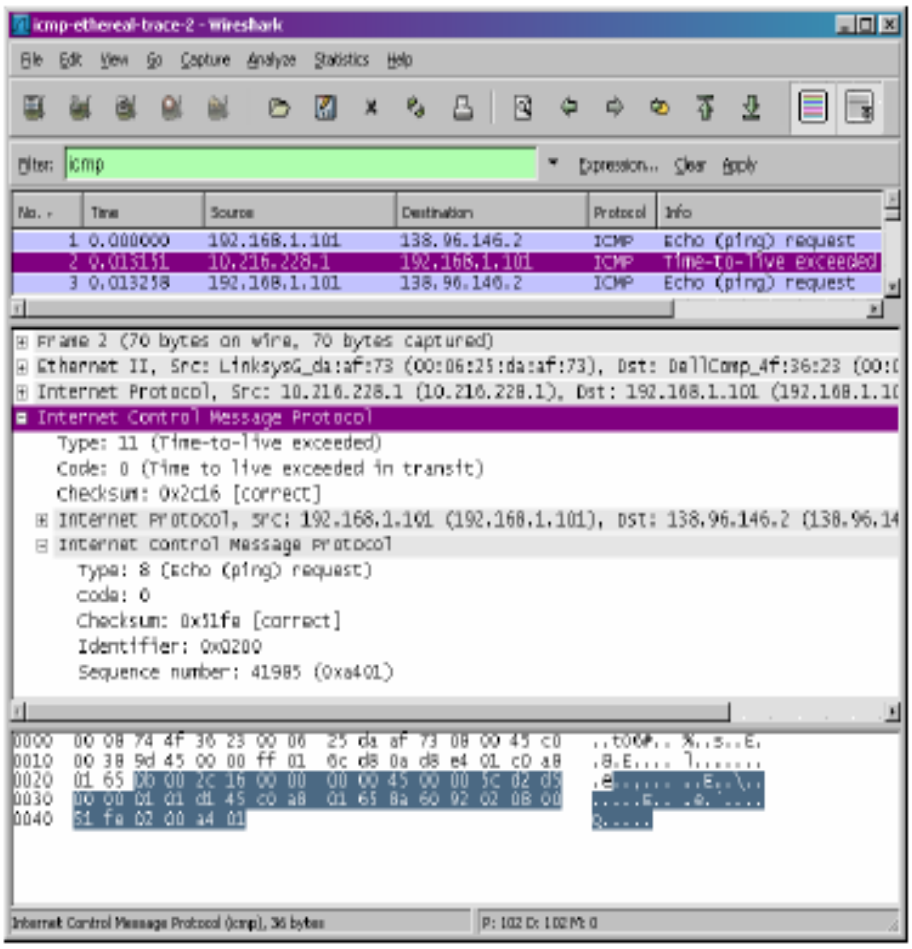
Εκτελούμε τα ακόλουθα βήματα (2)
>Ανοίγουμε την εφαρμόγη Command Prompt των windows (η οποία βρίσκεται στο φάκελο Accessories).
>Ξεκινάμε το Ethereal και τη σύλληψη πακέτων
>Η εντολή tracert βρίσκεται στο c;\windows\system32, άρα πληκτρολογούμε ‘tracert hostname’ ή c;\windows\system32\tracert hostname’ στη γραμμή εντολής MS-DOS, όπου hostname το όνομα ενός host που βρίσκεται σε μια διαφορετική ήπειρο. (Σημειώνεται οτι, σε ένα υπολογιστή με λειτουργικό σύστημα Windows, η εντολή είναι ‘tracert’ και όχι ‘traceroute’). Εάν βρίσκεστε εκτός Ευρώπης (3) μπορείτε να χρησιμοποιήσετε το όνομα www.inria.fr του Web server στο INRIA, ένα ερευνητικό ινστιτούτο επιστημής υπολογιστών στη Γαλλία. Στη συνέχεια τρέχουμε το πρόγραμμα Traceroute πληκτρολογώντας return
>Όταν το πρόγραμμα Traceroute τερματίσει, σταματάμε τη σύλληψη πακέτων από το Ethereal.

Στο τέλος του πειράματος το παράθυρο Command Prompt θα πρέπει να μοιάζει με αυτό του σχήματος 39. Στο σχήμα αυτό, το πρόγραμμα Traceroute πηγής βρίσκεται στη Μασαχουσέτη και ο προορισμός- στόχος στη Γαλλία. Από το σχήμα αυτό διαπιστώνουμε οτι για κάθε τιμή TTL το πρόγραμμα πηγής στέλνει τρία πακέτα ανίχνευσης (probe packets). Το traceroute δείχνει τους χρόνους RTT για καθένα από τα πακέτα ανίχνευσης καθώς επίσης και τη διεύθυνση IP (πιθανώς και το όνομα) του δρομολογητή που έστειλε στην πηγή το μήνυμα ICMP ‘TTL-exceeded’



Σχ.39 το παράθυρο Command Prompt με τα αποτελέσματα του προγράμματος Traceroute

Στο Σχ.40 φαίνεται το παράθυρο Ethereal για ένα πακέτο ICMP που επιστρέφεται από το router. Παρατηρούμε οτι το ICMP error πακέτο περιέχει πολλά περισσότερα πεδία από οτι τα μηνύματα ICMP Ping



Σχ.40 το παράθυρο Ethereal με αναλυτική παρουσίαση των λεπτομερειών του ICMP για ένα πακέτο ICMP σφάλματος

Τι θα παραδώσετε

Για αυτό το μέρος του εργαστηρίου θα πρέπει να παραδώσετε ένα screen shot του παραθύρου Command Prompt. Όπου είναι δυνατό, η απάντηση σε κάθε μια από τις ακόλουθες ερωτήσεις θα πρέπει να συνοδεύεται από μια εκτύπωση των πακέτων του trace που χρησιμοποιήσατε για να απαντήσετε στην ερώτηση. Σημειώστε πάνω στην εκτύπωση τα σημεία που αιτιολογούν την απάντηση σας. Για να εκτυπώσετε ένα πακέτο, χρησιμοποιήστε File>print, επιλέξτε selected packet only, επιλέξτε packet summary line και επιλέξτε το ελάχιστο ποσό λεπτομερείων που απαιτείται για να απαντησετε στην ερώτηση.

ΕΡΩΤΗΣΕΙΣ

5.Ποιά η διεύθυνση IP του δικού σας τερματικού συστήματος (host); Ποιά η διεύθυνση IP του host προορισμού;

6.Εάν το πρόγραμμα Tracert έστελνε πακέτα UDP αντί για πακέτα ICMP (όπως συμβαίνει σε Unix/Linux), θα παρέμενε 01 ο αριθμός πρωτοκόλλου της επικεφαλίδας IP για τα πακέτα ανίχνευσης; Εάν όχι, ποιά τιμή θα έπαιρνε το πεδίο αυτός;

7.Εξετάστε το πακέτο ICMP echo που εμφανίζεται στο παράθυρο Ethereal. Διαφέρει από τα πακέτα ICMP ερωτημάτων του Ping που εξετάσατε στο πρώτο μισό αυτό του εργαστηρίου; Εάν ναι, σε τι διαφέρει απο αυτά;

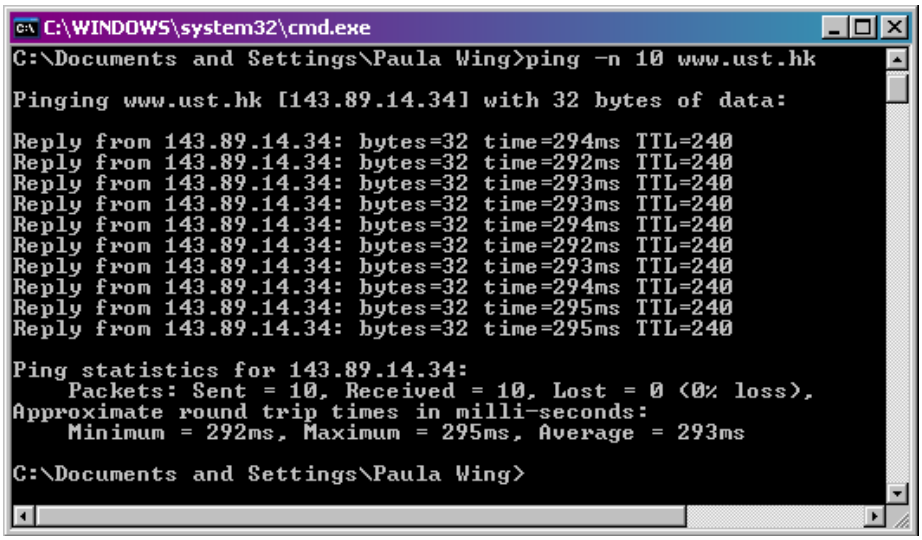
8.Εξετάστε το ICMP error πακέτο που εμφανίζεται στο παράθυρο Ethereal. Έχει περισσότερα απο τα πακέτα ICMP echo. Τι περιλαμβάνεται στα πεδία αυτά;
9.Εξετάστε τα τρία τελευταία πακέτα ICMP που λαμβάνει ο host πηγης. Σε τι διαφέρουν από τα πακέτα ICMP σφάλματος; Γιατί είναι διαφορετικά;

10.Εξετάζοντας τις μετρήσεις του tracert, εμφανίζενται κάποια ζεύξη της οποίας η καθυστέρηση είναι σημαντικά μεγαλύτερη από τις καθυστερήσης των άλλων ζευξεων;
Στο παράθυρο του Σχ.39, υπάρχει κάποια ζεύξη της οποίας η καθυστέρηση είναι σημαντικά μεγαλύτερη από τις καθυστερήσης των άλλων ζευξεων; Με βάση τα ονόματα των routers, μπορείτε να μαντέψετε τις τοποθεσίες των δυο routers στα άκρα αυτής της ζεύξης;

(1) Φορτώστε το αρχείο http://gaia.cs.umass.edu/wireshark-labs/ethernal-traces.zip και εξάγετε το αρχείο icmp-ethernal-trace-1.Τα traces που περιέχονται σε αυτό το αρχείο zip συλλέχθηκαν από το ethernal ενώ εκτελούνταν τα βήματα που περιγραφονται στο εργαστήριο ethernal στον υπολόγιστη του συγγραφέα.Αφού λάβουμε το trace,το φορτώνουμε στο ethernal ,και απο το μενού file επιλέγουμε open και στη συνέχεια επιλέγουμε το αρχείο icmp-ethernal-trace-1.
στη συνέχεια μπορίτε να χρησιμοποιήσετε αυτό το αρχείο του trace για να απαντήσετε
στις ερωτησεις που θέτονται παρακάτω
(2) Φορτώστε το αρχείο http://gaia.cs.umass.edu/wireshark-labs/ethernal-traces.zip και εξάγετε το αρχείο icmp-ethernal-trace-1.Τα traces που περιέχονται σε αυτό το αρχείο zip συλλέχθηκαν από το ethernal ενώ εκτελούνταν τα βήματα που περιγράφονται στο εργαστήριο ethernal στον υπολόγιστη του συγγραφέα.Αφού λάβουμε το trace,το φορτώνουμε στο ethernal ,και απο το μενού file επιλέγουμε open και στη συνέχεια επιλέγουμε το αρχείο icmp-ethernal-trace-2. στη συνέχεια μπορείτε να χρησιμοποιήσετε αυτό το αρχείο του trace για να απαντήσετε στις ερωτησεις που θέτονται παρακάτω

(3)Εάν βρίσκεστε στην Ευρώπη μπορείτε να χρησιμοποιήσετε το όνομα Gaia.cs.umass.edu του web server στο Τμήμα Επιστήμης Υπολογιστών του Πανεπιστήμιου της Μασαχουσέτης (University of Massachustts)

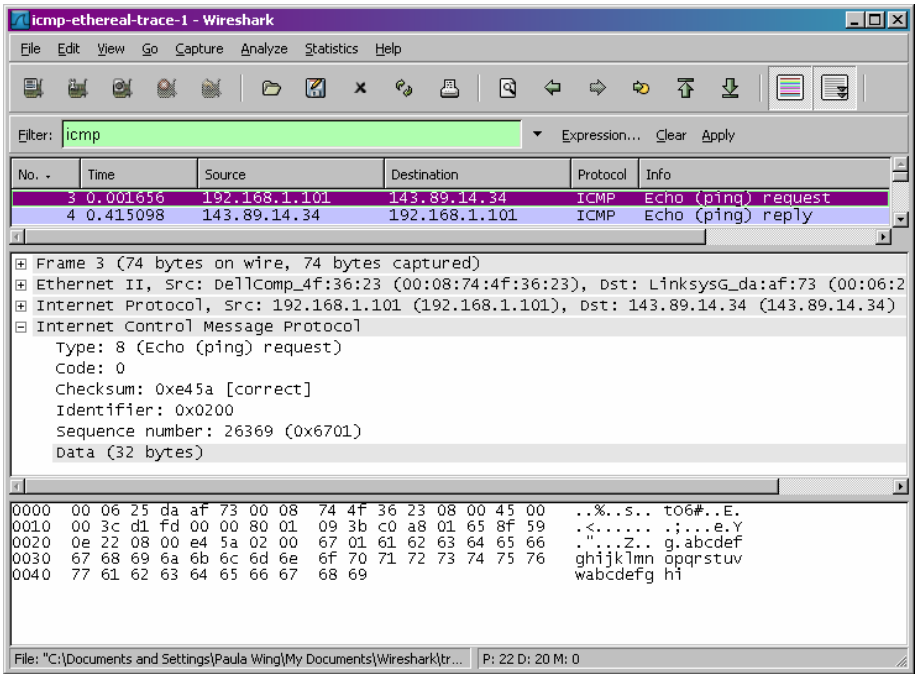
ΑΠΑΝΤΗΣΕΙΣ



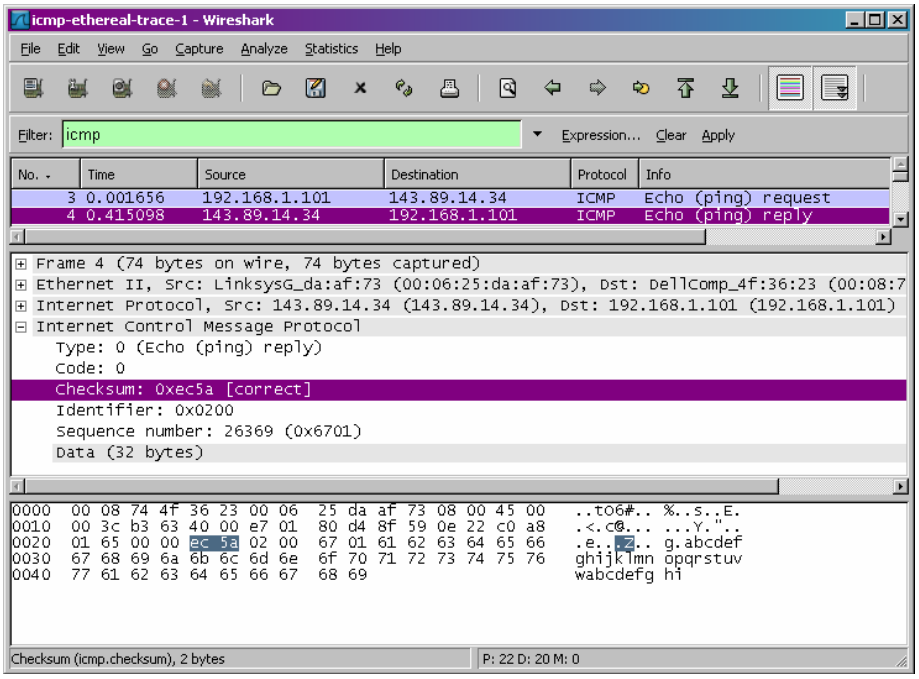
Το παράθυρο command prompt μετά την εισαγωγής της εντολής Ping

1.Η διεύθυνση IP του δικού μου τερματικού είναι 192.168.1.101
Η διεύθυνση IP προορισμού είναι 143.89.14.34.

2.Επειδή έχει σχεδιαστεί για την επικοινωνία μεταξύ hosts και routers,και όχι μεταξύ των στρωμάτων εφαρμογής. Κάθε ICMP πακέτο έχει ένα "Type" και ένα "Code". Ο συνδιασμός Type/Code προσδιορίζει το μήνυμα που συλλέχθηκε. Από τη στιγμή που το σύστημα από μόνο του ερμηνεύει όλα τα ICMP μυνήματα, δεν χρειάζονται αριθμοί θύρας και προορισμού για να κατευθύνουν το ICMP μήνυμα



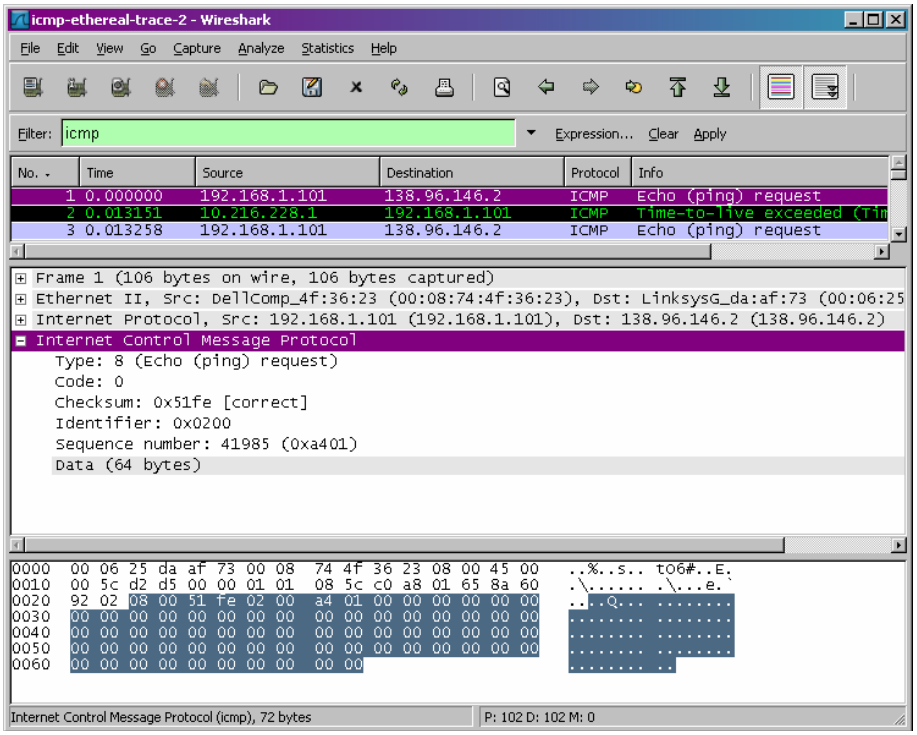
3. Το ICMP type έχει τιμή 8 και ο code 0. Περιλαμβάνει επίσης τα πεδία checksum, identifier, sequence number, και data. Τα πεδία checksum, sequence number και identifier καταλαμβάνουν 2 bytes εξίσου.



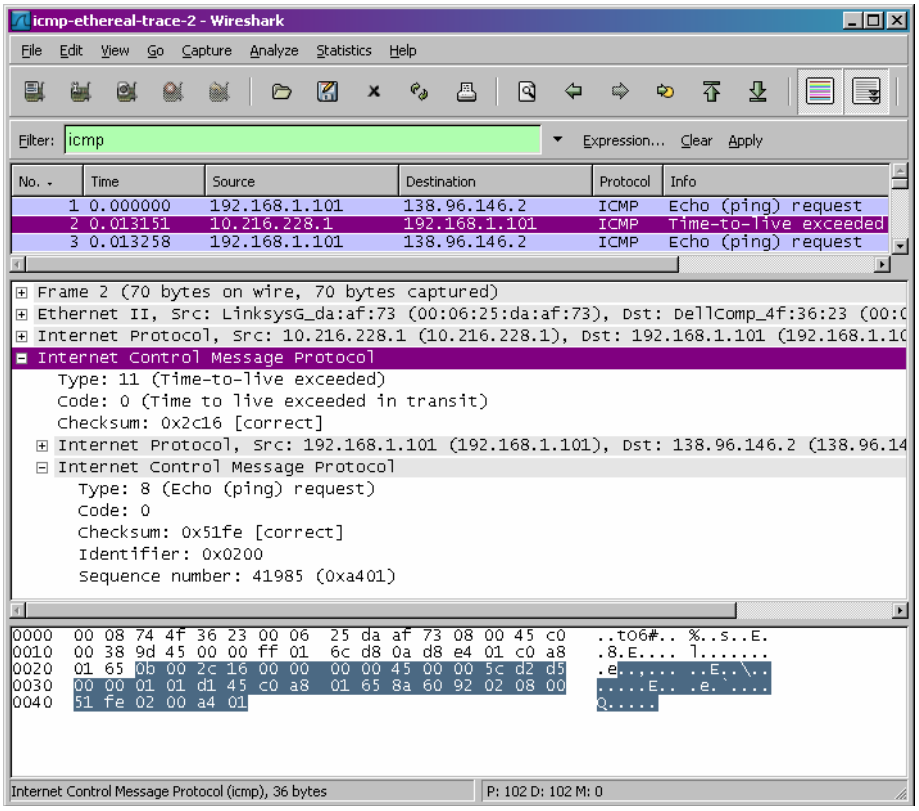
4. Το ICMP type έχει τιμή 0 και ο code 0. Περιλαμβάνει επίσης τα πεδία checksum, identifier, sequence number, και data .Τα πεδία checksum, sequence number και identifier καταλαμβάνουν 2 bytes εξίσου.

5. Η διεύθυνση IP του δικού μας τερματικού συστήματος είναι 192.168.1.101 και η IP διεύθυνση προορισμού είναι 138.96.146.2.

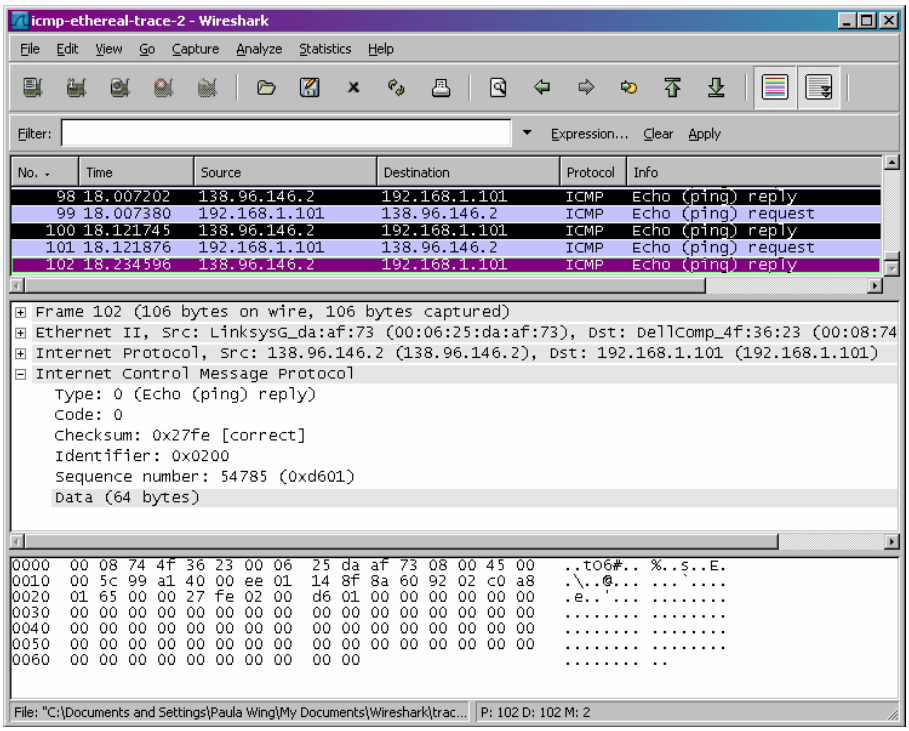
6. Όχι. Αν έστελνε UDP πακέτα ,ο IP αριθμός πρωτοκόλλου θα ήταν 0x11



7. Δεν διαφέρει. Έχουν τα ίδια πεδία

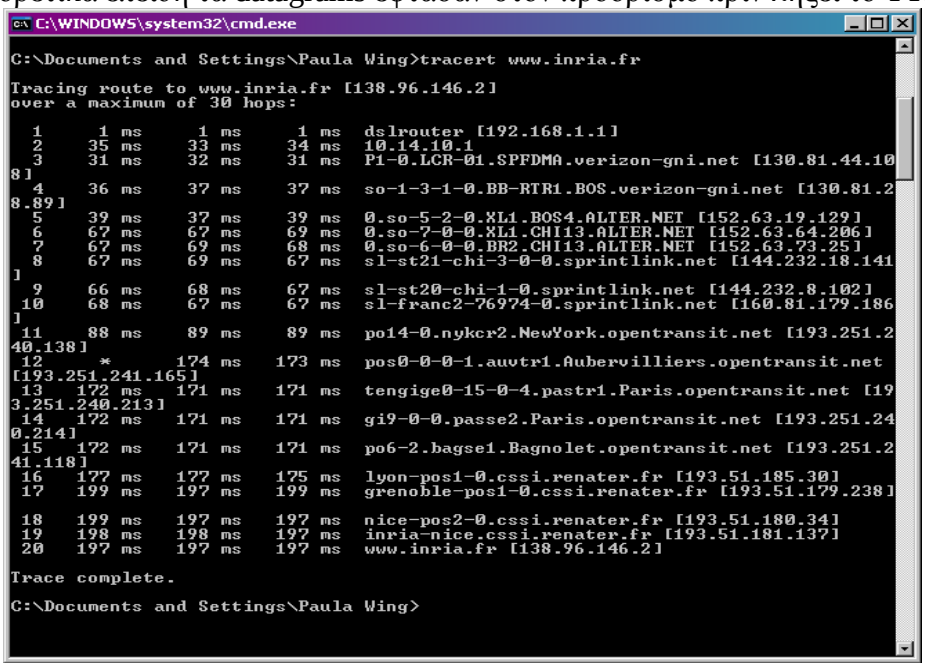


8.Το ICMP error πακέτο δεν είναι το ίδιο με τα πακέτα ερωτημάτων του Ping. Περιλαμβάνει την IP επικεφαλίδας και τα πρώτα 8 bytes του αρχικού εσφαλμένου ICMP πακέτου



9. Τα τρία τελευταία πακέτα ICMP που λαμβάνει ο host πηγης.Σε τι διαφέρουν από τα πακέτα ICMP σφάλματος;Γιατί είναι διαφορετικά;

Τα τρία τελευταία ICMP πακέτα είναι μυνήματα με type 0 αντι για 11.Είναι διαφορετικά επειδή τα datagrams έφτασαν στον προορισμό πρίν λήξει το TTL



10. Υπάρχει μια ζεύξη μεταξύ των βημάτων 11 και 12 με σημαντικά μεγαλύτερη καθυστέρηση. Αυτή είναι μια υπερατλαντική ζεύξη από τη Νέα Υόρκη στο Aubervilliers της Γαλλίας. Στο σχήμα η ζεύξη είναι μεταξύ Νέας Υόρκης και το Pastourelle της Γαλλίας.

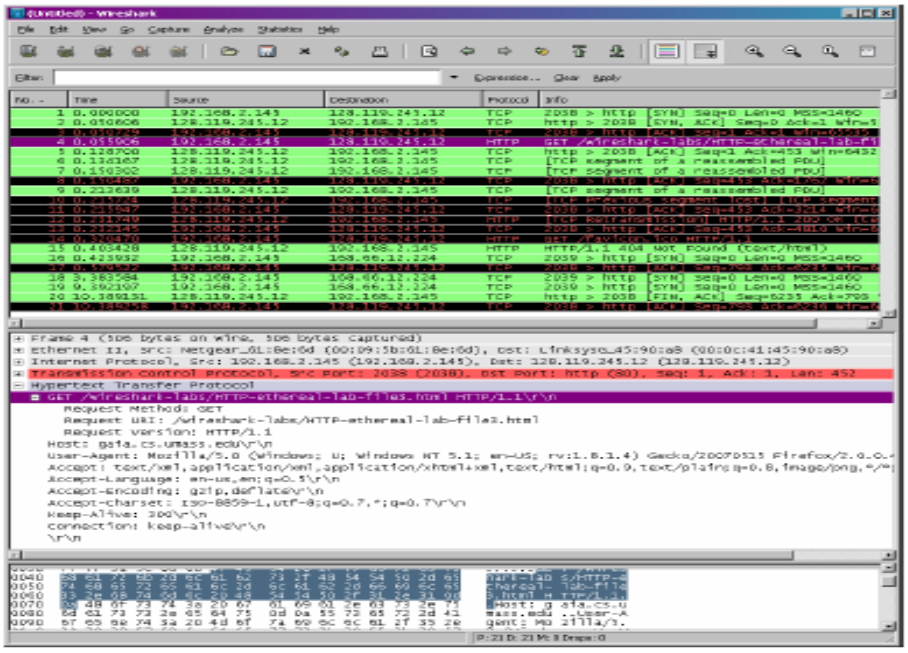
Κεφάλαιο 10
Ασκήσεις επίδειξης του πρωτοκόλλου Ethernet και ARP

Το Address Resolution Protocol (ARP) (Πρωτόκολλο Επίλυσης Διευθύνσεων) χρησιμοποείται για να βρεθεί μια διεύθυνση του επιπέδου ζεύξης δεδομένων (link layer) ή διεύθυνση υλικού (hardware address) ενός host με βάση μια διεύθυνση του επιπέδου επικοινωνίας (network layer).
Σε αυτό το εργαστήριο θα ερευνήσουμε το πρωτόκολλο Ethernet και το πρωτόκολλο ARP.

10.1 Συλλαμβάνοντας και αναλύοντας Ethernet frames

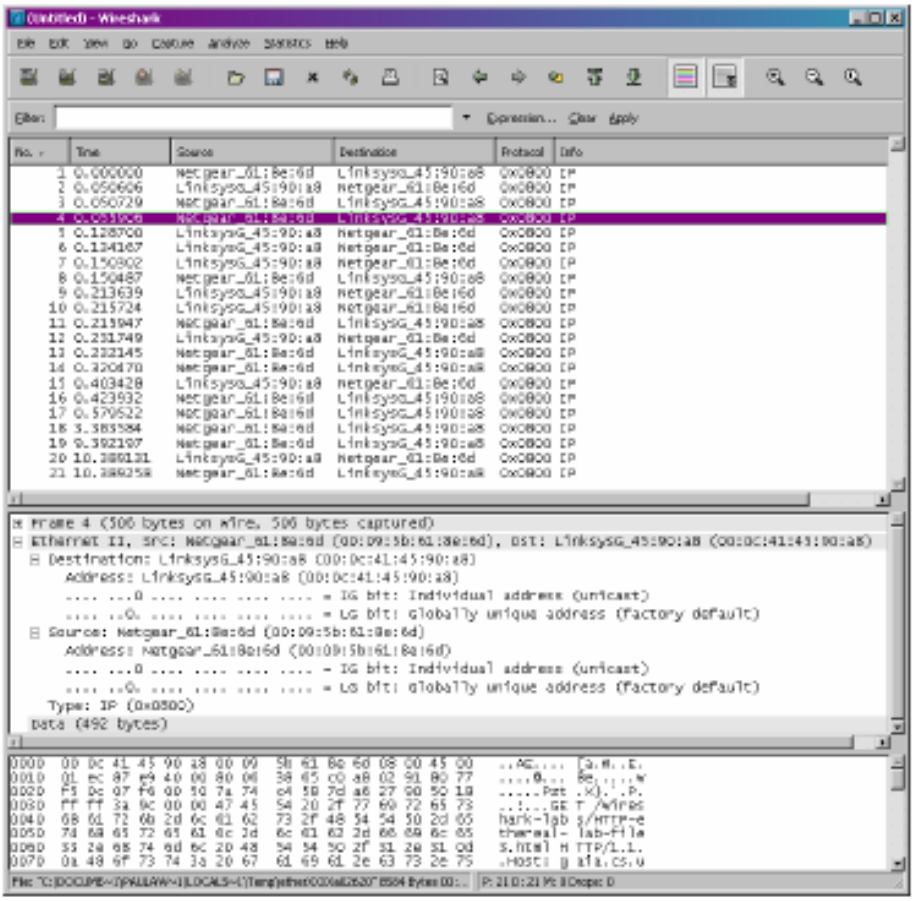
Ας ξεκινήσουμε συλλαμβάνοντας ένα πακέτο από Ethernet frames ακολουθώντας τα παρακάτω βήματα (1)

>Αρχικά, σιγουρευόμαστε οτι το cache του browser μας είναι άδειο. (Για να γίνει αυτό με Netscape 7.0 επιλέγουμε Edit> Preferences> Advanced> Cache και καθαρίζουμε την μνήμη και το disk cache. Για Internet Explorer, επιλέγουμε Tools> Internet Options> Delete Files. Για Firefox επιλέγουμε Tools> Clear Private Data.
>Ενεργοποιούμε τον Wireshark packet sniffer
>Πληκτρολογούμε το ακόλουθο URL στο browser μας http://gaia.cs.umass.edu/wireshark-labs/HTTP-ethereal-lab-file3.html. Ο browser μας θα πρέπει τώρα να παρουσιάζει την αμερικάνικη διακύρηξη δικαιωμάτων.
>Σταματάμε την σύλληψη πακέτων. Αρχικά, βρίσκουμε τον αριθμό πακέτων (στην δεξιότερη στηλη του παραθύρου του Wireshark) από τα HTTP GET μηνύνατα που στάλθηκαν από τον υπολογιστή μας στο gaia.cs.umass.edu, όπως το μύνημα στο ξεκίνημα της HTTP απόκρισης που έστειλε το gaia.cs.umass.edu στον υπολογιστή μας. Θα δούμε στην οθόνη την παρακάτω εικόνα



Σχ.41 Screenshot μετά τη σύλληψη πακέτων

>Από τη στιγμή που το εργαστήριο αυτό αφορά το Ethernet και το ARP, δεν μας αποσχολούν το IP και άλλα ανώτερα πρωτόκολλα. Ας αλλάξουμε λοιπόν το παράθυρο του Wireshark "listing of captured packets" που μας δείχνει πληροφορίες σχετικά με ανώτερα πρωτόκολλα. Για να το κάνει το Wireshark αυτό, επιλέγουμε Analyze> Enabled Protocols.
Μετά ξε-τσεκάρουμε το IP κουτάκι και επιλέγουμε OK. Τώρα πρέπει να βλέπουμε ένα παράθυρο του Wireshark όμοιο με το παρακάτω.



Σχ.42 πληροφορίες σχετικά με ανώτερα πρωτόκολλα μετά την επιλογή Analyze> Enabled Protocols.

Για να απαντήσετε στις παρακάτω ερωτήσεις, θα πρέπει να κοιτάξετε στα παράθυρα packet details και packet contents (το μεσαίο και το κατώτερο παράθυρο στο Wireshark).
Επιλέγουμε το Ethernet frame που περιέχει το HTTP GET μήνυμα (υπένθυμίζουμε οτι το HTTP GET μύνημα μεταφέρεται μέσω ενός TCP segment, το οποίο μεταφέρεται μέσω ενός IP datagram, το οποίο μεταφέρεται μέσω ενός Ethernet frame). Επεκτήνουμε την Ethernet II πληροφορία στο παράθυρο πληροφοριών παρακάτω. Σημειώστε οτι τα περιεχόμενα του Ethernet frame εμφανίζονται στο παράθυρο περιεχομένων

Απαντήστε τις παρακάτω ερωτήσεις, με βάση τα περιεχόμενα του Ethernet frame που περιλαμβάνουν το HTTP GET μήνυμα. Αν υπάρχει δυνατότητα, όταν απαντάτε μια ερώτηση, καλό θα ήταν να έχετε και μια εκτύπωση των πακέτων του trace που χρησιμοποιήσατε για να απαντήσετε στις ερωτήσεις. Σημειώστε στην εκτύπωση για να

εξηγήστε τις απαντήσεις σας. Για να εκτυπώστε, χρησιμοποιήστε File> Print, επιλέξτε Selected packet only, επιλέξτε Packet summary line, και επιλέξτε την μικρότερη τιμή πακέτου λεπτομερειών που χρειαζεστε για να απαντήσετε στις ερωτήσεις.

ΕΡΩΤΗΣΕΙΣ

1.Ποιά είναι η 48-bit Ethernet διεύθυνση του υπολογιστή σας;

2.Ποιά είναι η 48-bit διεύθυνση προορισμού του Ethernet frame; Είναι η Ethernet διεύθυνση του gaia.cs.umass.edu; (σημείωση-η απάντηση είναι οχι). Σε τι είδους συσκευή ανήκει αυτή η Ethernet address;

3.Δώστε την 16δικη τιμή για το Frame 2-byte type field.

4.Πόσα bytes από την έναρξη του Ethernet frame είναι το ASCII "G" στο 'GET' που εμφανίζονται στο Ethernet frame;

5.Ποιά η 16δική τιμή του CRC πεδίου σε αυτό το Ethernet frame;

Αργότερα, απαντήστε στις παρακάτω ερωτήσεις, με βάση τα περιεχόμενα του Ethernet frame που περιέχουν το πρώτο byte του HTTP μηνύματος απόκρισης
6.Ποιά είναι η τιμή της διεύθυνσης της Ethernet πηγής; Είναι αυτή η διεύθυνση του υπολογιστή σας ή του gaia.cs.umass.edu (σημείωση-η απάντηση είναι όχι); Σε τι είδους συσκευή ανήκει αυτή η Ethernet address;

7.Ποιά είναι η διεύθυνση προορισμού του Ethernet frame; Είναι αυτή η Ethernet διεύθυνση του υπολογιστή σας;

8.Δώστε την 16δικη τιμή για το Frame 2-byte type field.

9.Πόσα bytes από την έναρξη του Ethernet frame είναι το ASCII "O' στο 'OK' (Σημείωση –ο κωδικός HTTP απόκρισης) που εμφανίζονται στο Ethernet frame;

10. Ποιά η 16δική τιμή του CRC πεδίου σε αυτό το Ethernet frame;

10.2 The Address Resolution Protocol

Σε αυτό το σημείο,θα παρακολουθήσουμε το πρωτόκολλο ARP σε δράση.

ARP Caching.
Υπενθυμίζουμε ότι το πρωτόκολλο ARP τυπικά διατηρεί ένα cache με ένα ζεύγος IP διευθύνσεων μεταφρασμένο σε Ethernet διευθύνσεις στον υπολογιστή μας. Η εντολη ARP χρησιμοποιήται (και για MSDOS και για Linux/Unix) ώστε να ταξινομηθούν τα περιεχόμενα από αυτό το cache. Από τη στιγμή που η ARP εντολή και το πρωτόκολλο ARP έχουν το ίδιο όνομα, είναι εύκολο να μας μπερδέψει. Πρέπει να ξέρουμε όμως οτι είναι διαφορετικά πράγματα- η εντολή ARP χρησιμοποιείται για να κοιτάξουμε και να ταξινομίσουμε τα περιεχόμενα του ARP cache, ενώ το πρωτόκολλο ARP ορίζει τη

μορφή και το νόημα των μηνυμάτων που στέλνονται ή συλλέγονται και ορίζει τις διαδικασίες που χρειάζονται για την μεταφορά και παραλαβή των μηνυμάτων.

Ας ρίξουμε μια ματιά στα περιεχόμενα του APR cache του υπολογιστή μας

>**MS-DOS.** Η εντολή arp βρίσκεται στο c:\windows\system32, άρα πληκτρολογούμε 'arp' ή 'c:\windows\system32' στην γραμμή εντολών της MS-DOS
>**Linux/Unix.** Το εκτελέσιμο για την arp εντολή μπορεί να βρίσκεται σε πολλά μέρη. Δημοφιλείς τοποθεσείες είναι /sbin/arp (για Linux) και /usr/etc/arp (για κάποιες Unix παραλλαγές)

Η εντολή arp αναμφισβήτητα θα παρουσιάσει τα περιεχόμενα του ARP cache στον υπολογιστή μας. Τρέχουμε την εντολή arp

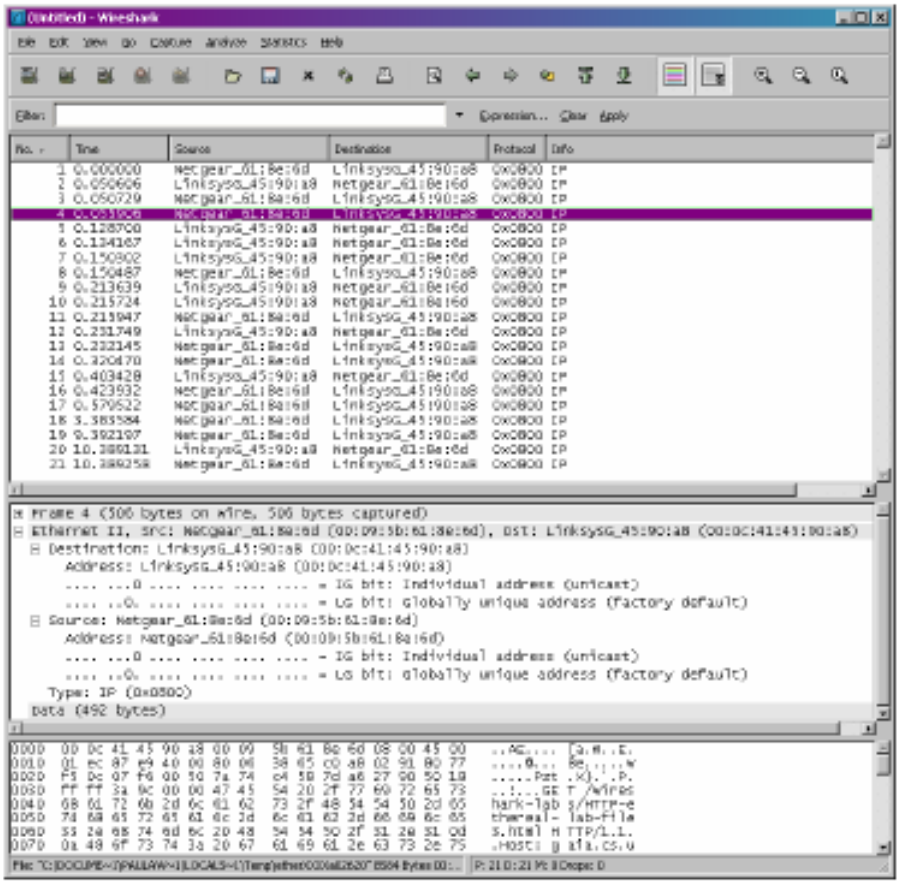
11.Καταγράψτε τα περιεχόμενα του ARP cache του υπολογιστή μας. Ποιά η σημασία της τιμής κάθε στήλης;

Για να παρατηρήσουμε στον υπολογιστή μας τα ARP μηνύματα που στέλνονται και λαμβάνονται, θα χρειαστεί να αδείασουμε το ARP cache, εκτός εάν βρει ο υπολογιστή μας το απαιτούμενο IP- Ethernet μεταφρασμένο ζεύγος και κατα συνέπεια δεν χρειαστεί να στείλουμε ένα ARP μήνυμα

>**MS-DOS.** Η εντολή arp-d* θα καθαρίσει το ARP cache μας. Το σύμβολο –d υποδεικνύει την λειτουργία διαγραφής, και το * επιπρόσθετα υποδεικνύει οτι πρέπει να διαγραφούν όλες οι εισόδοι.
>**Linux/Unix**. Η εντολή arp-d* θα καθαρίσει το ARP cache μας. Για να τρέξουμε την εντολή αυτή θα χρειαστούμε δικαιώματα πρόσβασης. Αν δεν έχουμε δικαιώματα πρόσβασης και δεν μπορούμε να τρέξουμε το Wireshark στα windows, μπορούμε να παρακάμψουμε το κομμάτι της συλλογής trace για αυτό το εργαστήριο και να χρησιμοποιήσουμε το trace που αναφέρθηκε στην υποσημείωση 1

10.3 Παρατηρώντας το ARP σε Δράση

Πράτουμε τα ακόλουθα (2)
>Καθαρίζουμε το ARP cache μας, όπως περιγράφηκε παραπάνω
>Σιγουρευόμαστε οτι το cache του browser μας είναι άδειο (για να το κάνουμε αυτό για Netscape 7.0, επιλέγουμε Edit> Preferences> Advanced> Cache και καθαρίζουμε τη μνήμη του cache. Για Internet Explorer, επιλέγουμε Tools> Internet Options> Delete Files)
>Ενεργοποιούμε το Wireshark packet sniffer
>Δίνουμε το ακόλουθο URL στον browser μας http://gaia.cs.umass.edu/wireshark-labs/HTTP-wireshark-lab-file3.html. Ο browser μας πρέπει να παρουσιάζει την US Bill of Rights.
>Σταματάμε την σύλληψη πακέτων του wireshark. Ξανά, δεν ενδιαφερόμαστε για IP ή ανώτερα πρωτόκολλα, γι'αυτό αλλάζουμε το παράθυρο του Wireshark "listing of captured packets" ώστε να μας δείχνει πληροφορίες μόνο για πρωτόκολλα κατώτερα από το IP. Για να γίνει αυτό, επιλέγουμε Analyze> Enabled Protocols. Μετά ξεμαρκάρουμε το IP κουτάκι και επιλέγουμε OK. Τώρα βλέπουμε ένα παράθυρο του Wireshark που μοιάζει με το παρακάτω



Σχ.43 πληροφορίες σχετικά με κατώτερα πρωτόκολλα μετά την επιλογή Analyze> Enabled Protocols.

.

Στο παραπάνω παράδειγμα, τα δυο πρώτα frames στο trace περιέχουν ARP μηνύματα (όπως και το 6^{ο} μήνυμα). Η screen shot παραπάνω αντιστοιχεί στο trace που προαναφέρθηκε στο trace της υποσημείωσης 1.
Απαντήστε στις ακόλουθες ερωτήσεις

12.Ποιά είναι η 16δική τιμή της διεύθυνσης πηγής και προορισμού στο Ethernet frame που περιέχει ένα ARP μήνυμα αίτησης;

13.Δώστε την 16δική τιμή για την 2-byte Ethernet Frame type field.

14.Κατεβάστε το ARP παράρτημα από ftp://ftp.rfc-editor.org/innotes/std/std37.txt. Μια άλλη λεπτομερής αναφορά για το ARP βρίσκεται στο http://www.erg.abdn.ac.uk/users/gorry/course/inet-pages/arp.html.
Α)Πόσα bytes είναι το ARP opcode πεδίο από την έναρξη του Ethernet frame ;
Β)Ποιά η τιμή του opcode πεδίου του τμήματος ARP -payload του Ethernet frame στο οποίο έγινε η ARP αίτηση;
Γ)Περιέχει το ARP μήνυμα την IP διεύθυνση του αποστολέα;
Δ)Που εμφανίζεται η ‘ερώτηση’ στην ARP αίτηση

15.Βρείτε την ARP απάντηση που στάλθηκε στην ARP αίτηση
Α)Πόσα bytes είναι το ARP opcode πεδίο από την έναρξη του Ethernet frame ;
Β) Ποιά η τιμή του opcode πεδίου του τμήματος ARP -payload του Ethernet frame στο οποίο έγινε η ARP απόκριση δόθηκε ;

Γ)Που βρίσκεται στο ARP μήνυμα η ‘απάντηση’ στην προηγούμενη ARP αίτηση

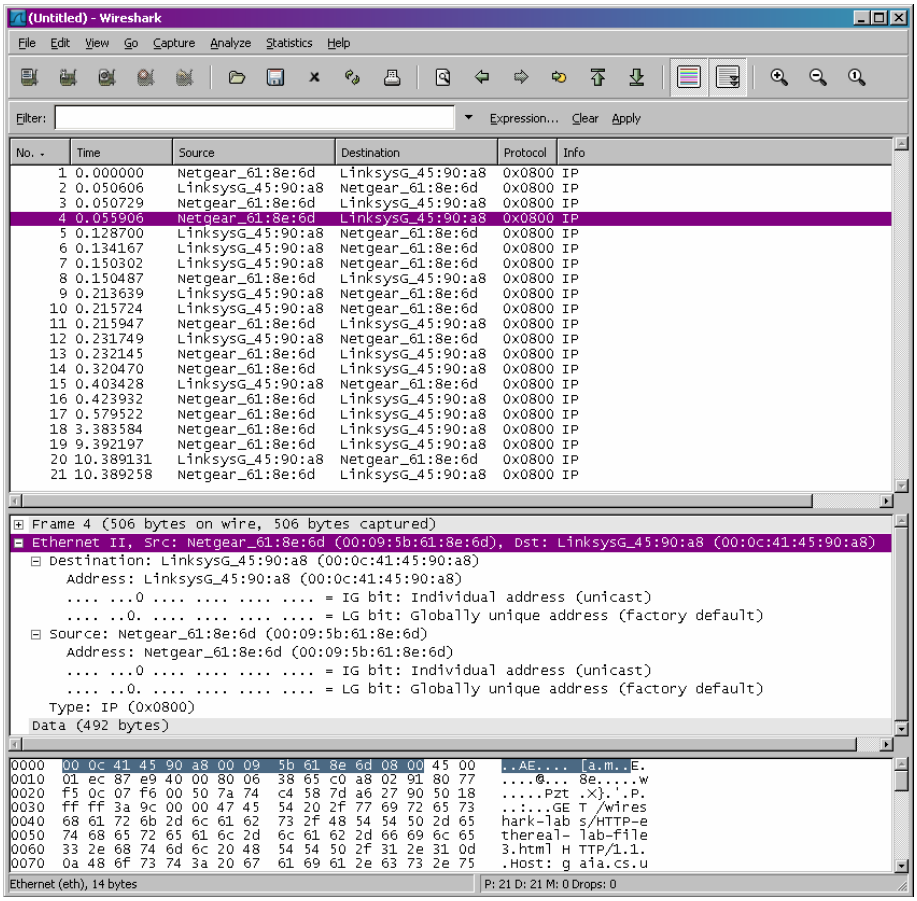
16. Ποιά είναι η 16δική τιμή της διεύθυνσης πηγής και προορισμού στο Ethernet frame που περιέχει ένα ARP μήνυμα απάντησης;

17.Ανοίξτε το ethernet-ethereal-trace-1 trace από http://gaia.cs.umass.edu/wireshark-labs/wireshark-traces.zip. Το πρώτο και το δεύτερο ARP πακέτο στο trace αντιστοιχεί σε μια ARP αίτηση που στέλνει ο υπολογιστής ενώ τρέχει το Wireshark, και η ARP απάντηση στέλνεται στον υπολογιστή που τρέχει το Wireshark από τον υπολογιστή με την ARP που ζητά την Ethernet διεύθυνση. Αλλά υπάρχει ένας ακόμα υπολογιστής σε αυτή τη διαδικασία όπως υποδεικνύεται στο πακέτο 6 –άλλη μια ARP αίτηση. Γιατί δεν υπάρχει ARP απάντηση (στέλνεται ως απάντηση στην ARP αίτηση στο πακέτο 6) στο πακέτο trace

(1) Αν δεν μπορείτε να τρέξετε το wireshark σε μια ‘ζωντανή’ σύνδεση,μπορείτε νακατεβάσετε το zip αρχείο http://gaia.cs.umass.edu/wireshark-labs/wireshark-traces.zipκαι να εξάγετε το αρχείο *ethernet--ethereal-trace-1*.Τα traces από αυτό το zip συλλέχθηκαν από το Wireshark σε υπολογιστή του συγγραφέα.Αφου καταβάσετε το trace,μπορείτε να το φορτώσετε στο Wireshark Και να δείτε το trace χρησιμοποιόντας το file μενού,επιλέγοντας Open, και μετά επιλέγουμε το trace ethernet-ethereal-trace-1.Μπορείτε να χρησιμοποιήσετε το trace για να απαντήσετε στις παρακάτω ερωτήσεις.

(2)Το *ethernet-ethereal-trace-1* trace στη σελιδα http://gaia.cs.umass.edu/wireshark-labs/wireshark-traces.zip

ΑΠΑΝΤΗΣΕΙΣ



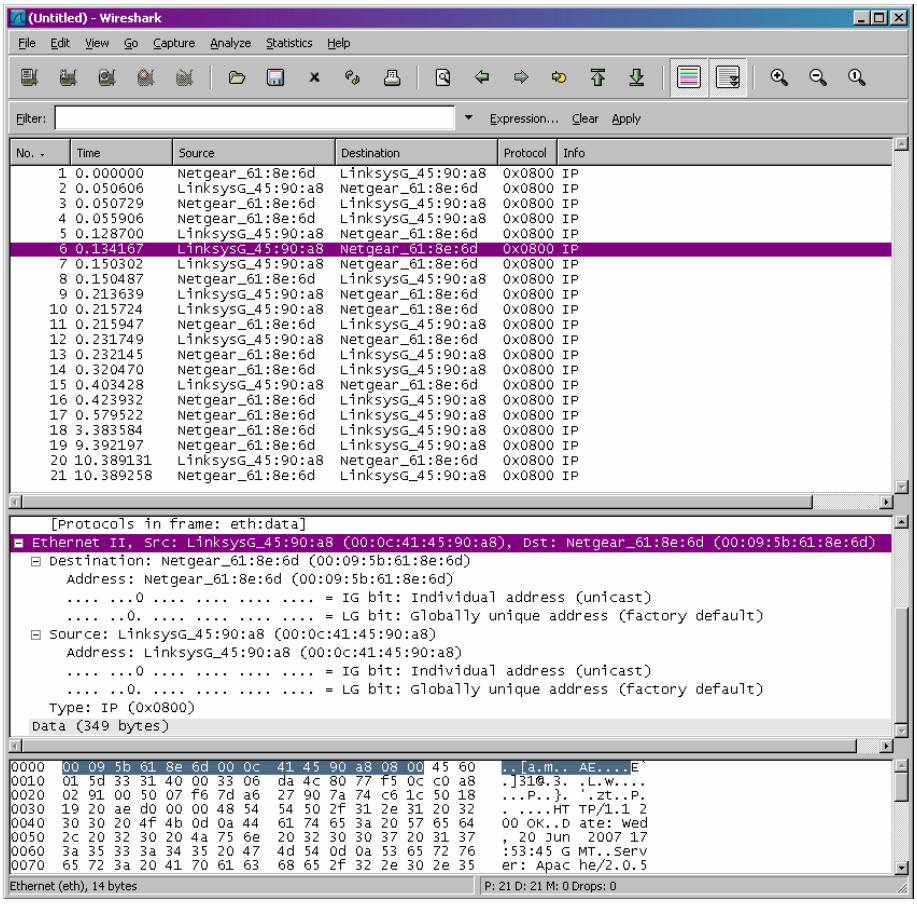
1.Η Ethernet διεύθυνση του υπολογιστή μου είναι 00:09:5b:61:8e:6d

2.Η διεύθυνση προορισμού είναι 00:0c:41:45:90:a8, δεν είναι η διεύθυνση του gaia.cs.umass.edu. Είναι η διεύθυνση του δικού μου Linksys router

3.Η 16δικη τιμή του frame είναι *0x0800*

4.Το ASCII "G" είναι 52 bytes από την έναρξη του Ethernet frame . Υπάρχουν 14B Ethernet frame και 20 bytes της IP επικεφαλίδας ακολουθούμενα από 20 bytes της TCP επικεφαλίδας

5.Η 16δική τιμή του CRC πεδίου είναι *0x 0d0a 0d0a.*



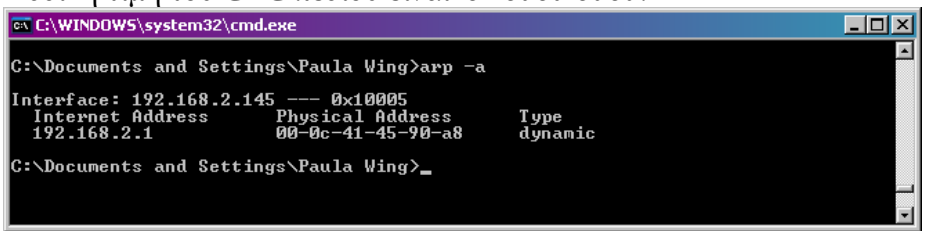
6. Η τιμή της διεύθυνσης της Ethernet πηγής *00:0c:41:45:90:a8* .Δεν είναι διεύθυνση του υπολογιστή μας ούτε της gaia.cs.umass.edu. Είναι η διεύθυνση του δικού μου Linksys router

7.Η διεύθυνση προορισμού είναι *00:09:5b:61:8e:*6d και είναι η διεύθυνση προορισμού του pc μας.

8. Η 16δικη τιμή είναι *0x0800.*

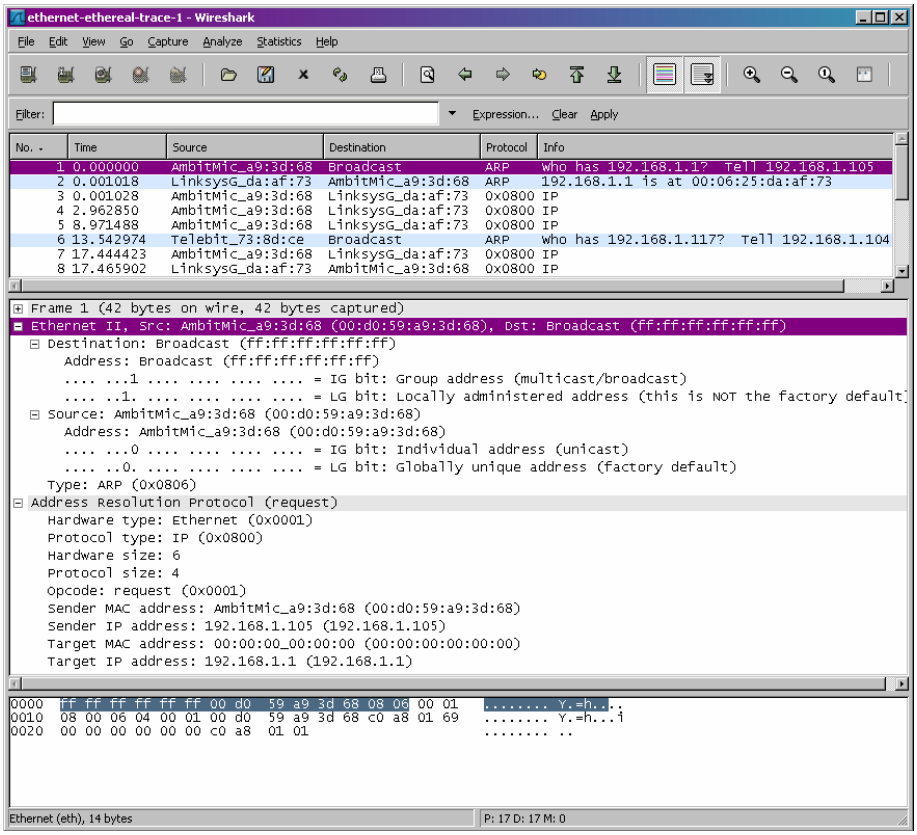
9. Το ASCII "G" είναι 52 bytes από την έναρξη του Ethernet frame. Υπάρχουν 14B Ethernet frame και 20 bytes της IP επικεφαλίδας ακολουθούμενα από 20 bytes της TCP επικεφαλίδας

10.Η 16δική τιμή του CRC πεδίου είναι *0x 0d0a 0d0a.*



Command prompt after executing arp

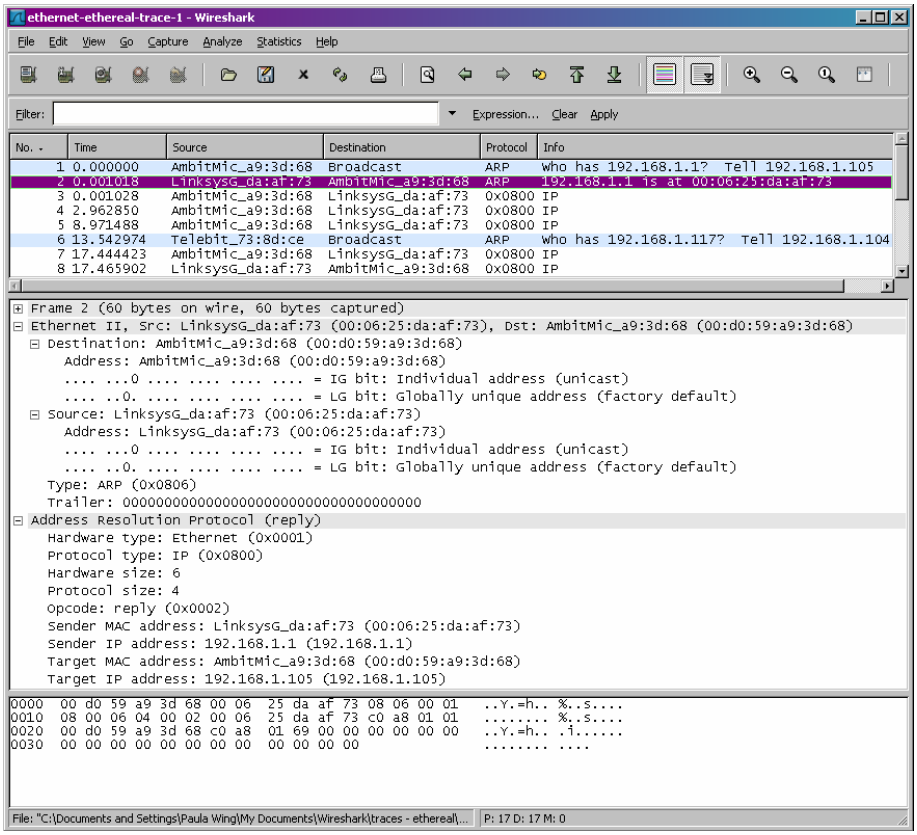
11. Η Internet Address περιέχει την IP διεύθυνση, η Φυσική διεύθυνση περιέχει την MAC διεύθυνση. Υποδεικνύουν τον τύπο του πρωτοκόλου.



12.Η 16δική τιμή της διεύθυνσης πηγής είναι 00:d0:59:a9:3d:68 και η 16δική τιμή της διεύθυνσης προορισμού είναι is ff:ff:ff:ff:ff:ff

13. Η 16δική τιμή για το Ethernet Fram type πεδίο είναι 0x0806

14. Α.20 bytes από την έναρξη του Ethernet frame
Β. 0x0001
Γ. Ναι, περιέχει την διεύθυνση αποστολέα 192.168.1.105
Δ. Το πεδίο “Target MAC address” είναι σεταρισμένο στο 00:00:00:00:00:00 για να αντιστοιχεί με την IP address (192.168.1.1) is being queried.



15. A. 20 bytes από την έναρξη του Ethernet frame

B. *0x0002*

Γ. εμφανίζεται στο πεδίο ”Sender MAC address” , περιέχει και την the Ethernet διεύθυνση 00:06:25:da:af:73 για τον αποστολέα με IP διεύθυνση 192.168.1.1.

16. Η 16δική τιμή της διεύθυνσης πηγής 00:06:25:da:af:73 και προορισμού είναι 00:d0:59:a9:3d:68 .

17. Δεν υπάρχει απάντηση γιατί δεν βρισκόμαστε στο pc που στέλνει την αίτηση.Η ARPαπάντηση στέλνεται πίσω στην Ethernet διεύθυνση του αποστολέα

Κεφάλαιο 11
Ασκήσεις επίδειξης του πρωτοκόλλου DHCP

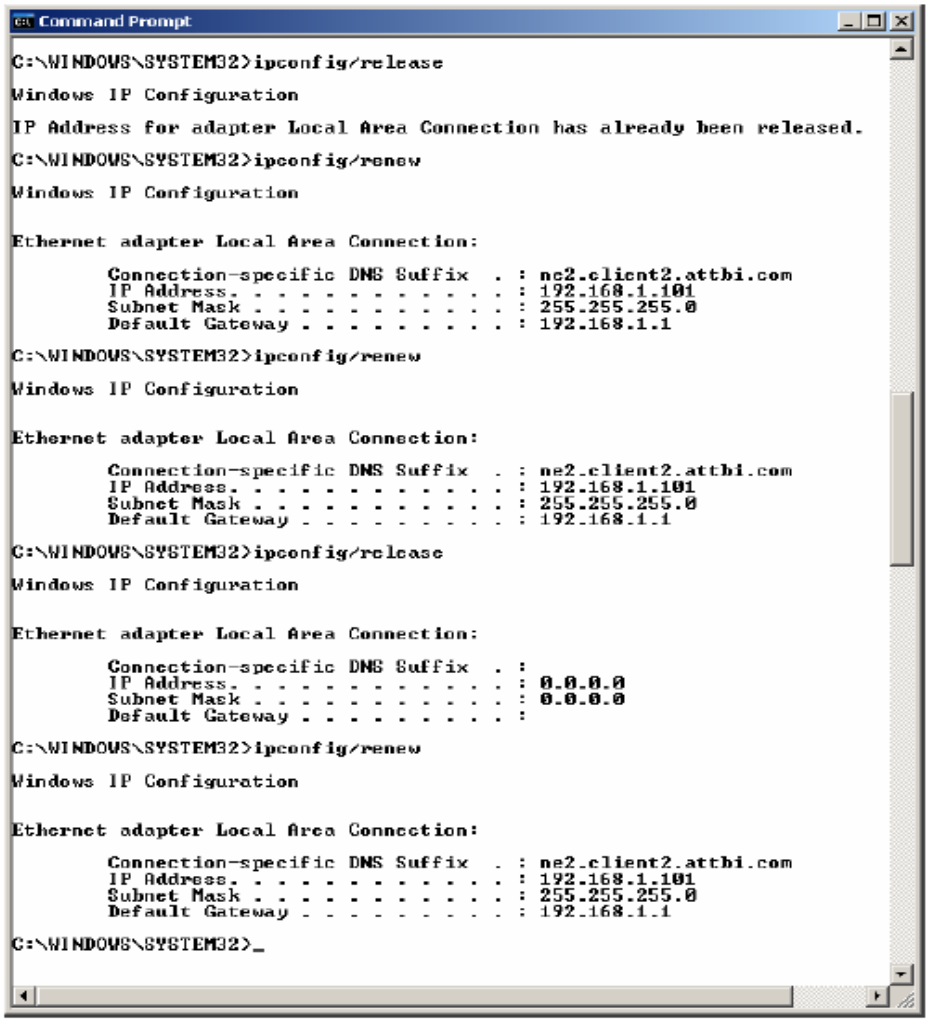
Με τον όρο DHCP (Dynamic Host Configuration Protocol- Πρωτόκολλο Δυναμικής Διαμόρφωσης Κεντρικού Υπολογιστή) αναφερόμαστε σε ένα μηχανισμό διαχείρισης πρωτοκόλλων TCP/IP. Το TCP/IP πρωτόκολλο είναι ουσιαστικά ένα λογισμικό που τρέχει σε έναν router και σε υπολογιστή και διευθετεί όλα τα θέματα επικοινωνίας με αυτόν τον υπολογιστή και άλλους που χρησιμοποιούν αυτό το πρωτόκολλο ως γλώσσα.

Στο εργαστήριο αυτό θα ρίξουμε μια σύντομη ματιά στο DHCP. Υπενθυμίζεται οτι το DHCP χρησιμοποιείται ευρέως σε εταιρικά, πανεπιστημιακά και οικιακά ενσύρματα και ασύρματα τοπικά δίκτυα για τη παροχή διευθύνσεων IP στους hosts [καθώς επίσης για τη διευθέτηση (configuration) άλλου είδους πληροφορίας δικτύου]. Το εργαστήριο αυτό είναι σύντομο καθώς θα εξετάσουμε μόνο τα πακέτα DHCP που συλλαμβάνονται από ένα host. Έαν έχετε εξουσιοδοτημένη πρόσβαση σε έναν DHCP server, ενδεχομένως να θελήσετε να επαναλάβετε αυτό το εργαστήριο αφού κάνετε κάποιες αλλαγές στη διευθέτηση, π.χ. στο χρόνο μίσθωσης (lease time) της διευθυνσης IP. Έαν έχετε router στο σπίτι, πιθανώς να μπορείτε να ρυθμίσετε τον DCHP server σας. Έπειδη πολλά από τα συστήματα που χρησιμοποιούν Linux/Unix (ειδικά αυτά που εξυπηρετούν πολλούς χρήστες) έχουν στατική διεύθυνση IP και επειδή ο χειρισμός του DCHP σε τέτοια συστήματα απαιτεί συνήθως δικαιώματα super-user, θα παρουσιάσουμε μια μόνο έκδοση αυτού του εργαστηρίου για Windows

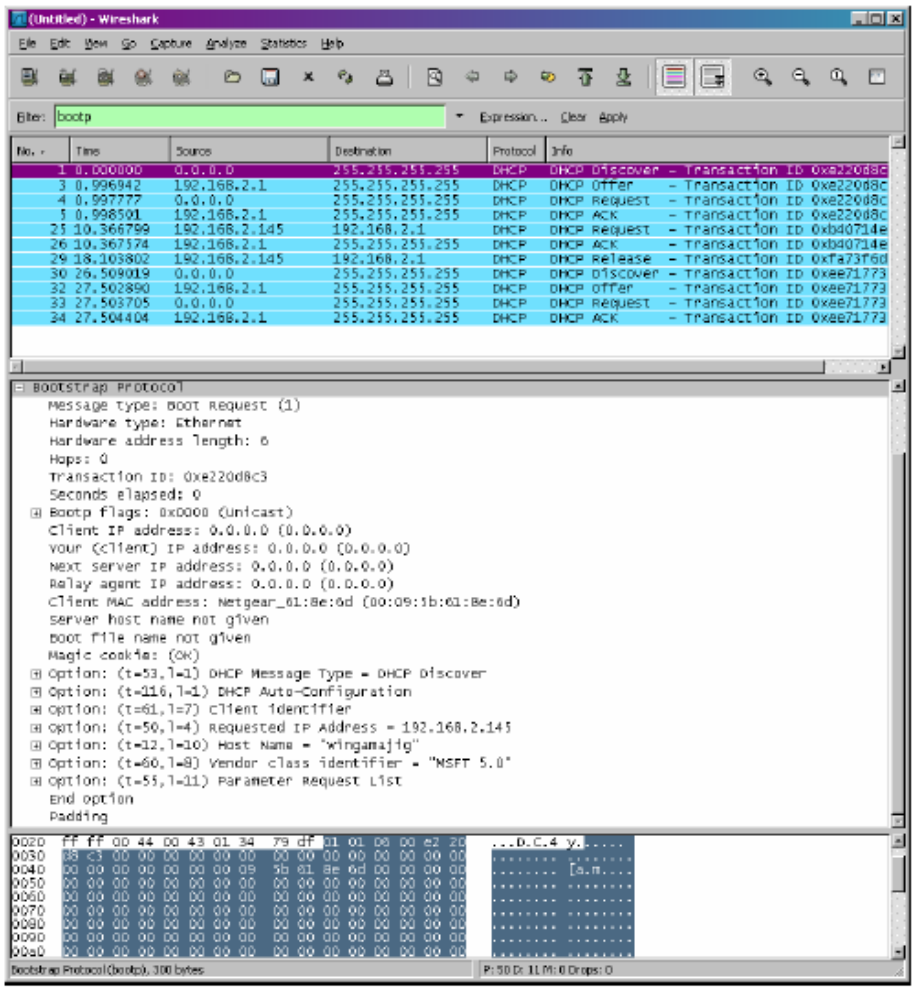
11.1 Πείραμα DHCP

Για να παρατηρήσουμε το DHCP σε δράση, θα εκτελέσουμε μερικές εντολές που σχετίζονται με το DHCP και θα συλλάβουμε τα μηνύματα DHCP που ανταλλάσσονται ως αποτέλεσμα της εκτέλεσης αυτών των εντολών. Εκτελούμε τα ακόλουθα βήματα (1)

1.Ξεκινάμε ανοίγοντας την εφαρμογη Command Prompt των Windows (η οποία βρίσκεται στον φάκελο Accessories). Εισάγουμε την εντολή 'ipconfig/release' όπως φαίνεται στο Σχ.43. Το εκτελέσιμο πρόγραμμα για την εντολή ipconfig βρίσκεται στο C;\\windows\sysstem32. Η εντολή αυτή αποδεσμεύει την τρέχουσα διεύθυνση IP του host μας, οπότε η διεύθυνση IP του host γίνεται 0.0.0.0.
2.Ξεκινάμε το Ethernal και τη σύλληψη πακέτων
3.Επιστρέφουμε τώρα στο παράθυρο Command Prompt των Windows και εισάγουμε την εντολή 'ipconfig/renew'. Η εντολή αυτή οδηγεί τον host μας να αποκτήσει μια διεύθυνση δικτύου (network configuration), συμπεριλαμβανόμενης μιας νέας διεύθυνσης IP. Στο Σχ.44, ο host αποκτά τη διεύθυνση IP 192.168.1.101.
4.Περιμένουμε μέχρι να ολοκληρωθεί η εντολή 'ipconfig/renew'. Κατόπιν εισάγουμε ξανά την ίδια εντολή 'ipconfig/renew'
5.Όταν ολοκληρωθεί η δεύτερη εντολή 'ipconfig/renew', εισάγουμε την εντολή 'ipconfig/release', ώστε να αποδεσμεύσουμε την προηγούμενη διεύθυνση IP που είχε εκχώρηθεί στον υπολογιστή μας.
6.Τέλος, εισάγουμε ξανά 'ipconfig/renew' ώστε να εκχωρηθεί ξανά μια διεύθυνση IP στον υπολογιστή μας.
7.Σταματάμε τη σύλληψη πακέτων από το Ethereal.



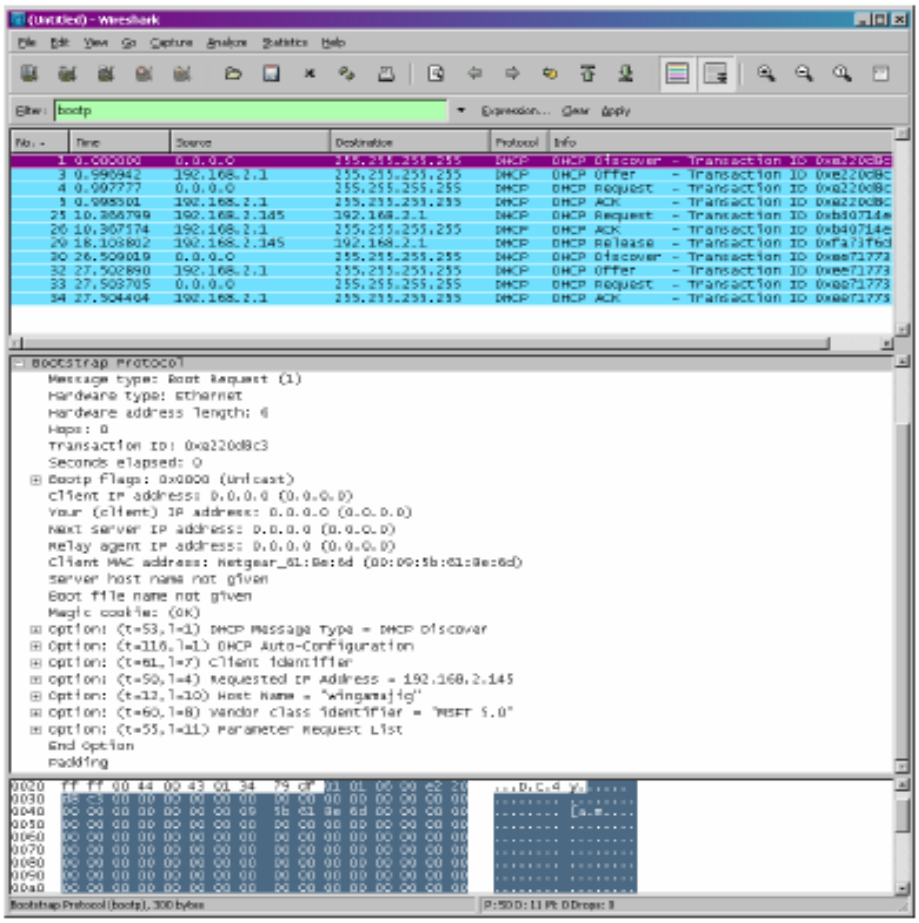
Σχ.43 το παράθυρο Command Prompt που υποδεικνύει την ακολουθία εντολών ipconfig που πρέπει να εισάγουμε



Σχ.44 παράθυρο του Ethereal με ανεπτυγμένο το πρώτο πακέτο DHCP Discover

Ας ρίξουμε μια ματιά στο παράθυρο Ethereal που προέκυψε. Για να δούμε μόνο τα πακέτα DHCP, εισάγουμε ‘bootp’ στο πεδίο του φίλτρου. Το DHCP προέρχεται από ένα παλαιότερο πρωτόκολλο ονομαζόμενο BOOTP. Και τα δύο πρωτόκολλα, BOOTP και DHCP, χρησιμοποιούν τους ίδιους αριθμούς θύρας, 67 και 68. Για να δούμε τα πακέτα DCHP στην παρούσα έκδοση του Ethereal χρειάζεται να εισάγουμε στο φίλτρο ‘bootp’ και όχι ‘dhcp’

Από το Σχ.45 φαίνεται οτι η πρώτη εντολή ‘ipconfig/renew’ είχε ως αποτέλεσμα να προκύψουν τέσσερα πακέτα DHCP Discover, ένα πακέτο DHCP Offer, ένα πακέτο DHCP Request και ένα πακέτο DHCP ACK



Σχ.45 το παράθυρο του wireshark

Τι θα παραδώσετε

Εάν ήσασταν σε θέση να εκτελέσετε το πείραμα, θα πρέπει να παραδώσετε ένα screen shot του παραθύρου Command Prompt παρόμοιο με αυτό του σχήματος 43. Σε αντίθετη περίπτωση, χρησιμοποιήστε το trace πακέτων dhcp-ethereal-trace-1 για να απαντήσετε στις παρακάτω ερωτήσεις. Όπου είναι δυνατό, η απάντηση σας θα πρέπει να συνοδεύεται από μια εκτύπωση των πακέτων του trace που χρησιμοποιήσατε για να απαντήσετε στην ερώτηση. Σημειώσετε επάνω στην εκτύπωση τα σημεία εκείνα που αιτιολογούν την απάντηση σας. Για να εκτυπώσετε ένα πακέτο, χρησιμοποιήστε File> Print, επιλέξτε Selected Packet Only, επιλέξτε Packet summary line και το ελάχιστο ποσό λεπτομερειων που απαιτείται για να απαντήσετε στην ερώτηση.

ΕΡΩΤΗΣΕΙΣ

1.Τα μηνύματα DHCP στέλνονται μέσω UDP ή TCP;

2.Σχεδιάστε ένα χρονικό διάγραμμα που να δείχνει την ακολούθια ανταλλαγής των τεσσάρων πρώτων πακέτων DCHP Discover/Offer/Request/ACK μεταξύ client και server. Υποδείξτε τους αριθμούς θύρας πηγής και προορισμού για κάθε πακέτο.

3.Ποια η διεύθυνση επιπέδου ζεύξης (π.χ. διεύθυνση Ethernet) του host σας;

4.Ποιές οι τιμές του μηνύματος DCHP Discover που διαφοροποιούν αυτό το μήνυμα από το μήνυμα DHCP Request;

5.Ποιά η τιμή του πεδίου Transaction-ID σε καθένα από τα τέσσερα πρώτα μηνύματα DCHP (Discover/Offer/Request/ACK); Ποιά η τιμή του πεδίου Transaction-ID στο δεύτερο σύνολο μηνυμάτων DCHP (Request/ACK); Ποιός ο ρόλος του πεδίου Transaction-ID;

6.Ένας host χρησιμοποίει το DCHP για να αποκτήσει, εκτός των άλλων, μια διεύθυνση IP. Όμως η διεύθυνση IP ενός host δεν επιβεβαιώνεται παρά στο τέλος αυτής της ανταλλαγής των τεσσάρων μηνυμάτων; Υποδείξτε τις διευθύνσης IP πηγής και προορισμού των IP datagrams μέσα στα οποία είναι ενθυλακωμένα καθένα από τα τέσσερα μηνύματα μηνύματα DCHP (Discover/Offer/Request/ACK)

7.Ποιά η διεύθυνση IP του DHCP server σας;

8. Ποιά η διεύθυνση IP που προσφέρει ο DHCP server στον host σας στο μήνυμα DHCP offer; Υποδείξτε το μήνυμα DHCP που περιέχει την προσφερόμενη διεύθυνση IP

9.Στο screen shot που χρησιμοποιήθηκε ως παράδειγμα σε αυτό το εργαστήριο, δεν υπάρχει relay agent μεταξύ του host και του DHCP server. Ποιές τιμές του trace υποδεικνύουν την απούσια ενός relay agent; Υπάρχει relay agent στο δικό σας πείραμα; Σε αυτή την περίπτωση, ποιά η διεύθυνση IP του relay agent;

10.Εξηγείστε το ρόλο των γραμμών Router και Subnet Mask στο μήνυμα DHCP Offer

11.Στο παράδειγμα που χρησιμοποιήθηκε σε αυτό το εργαστήριο, ο host ζητά την προσφερόμενη διεύθυνση IP στο μήνυμα DHCP Request. Τι συμβαίνει στο δικό σας πείραμα;

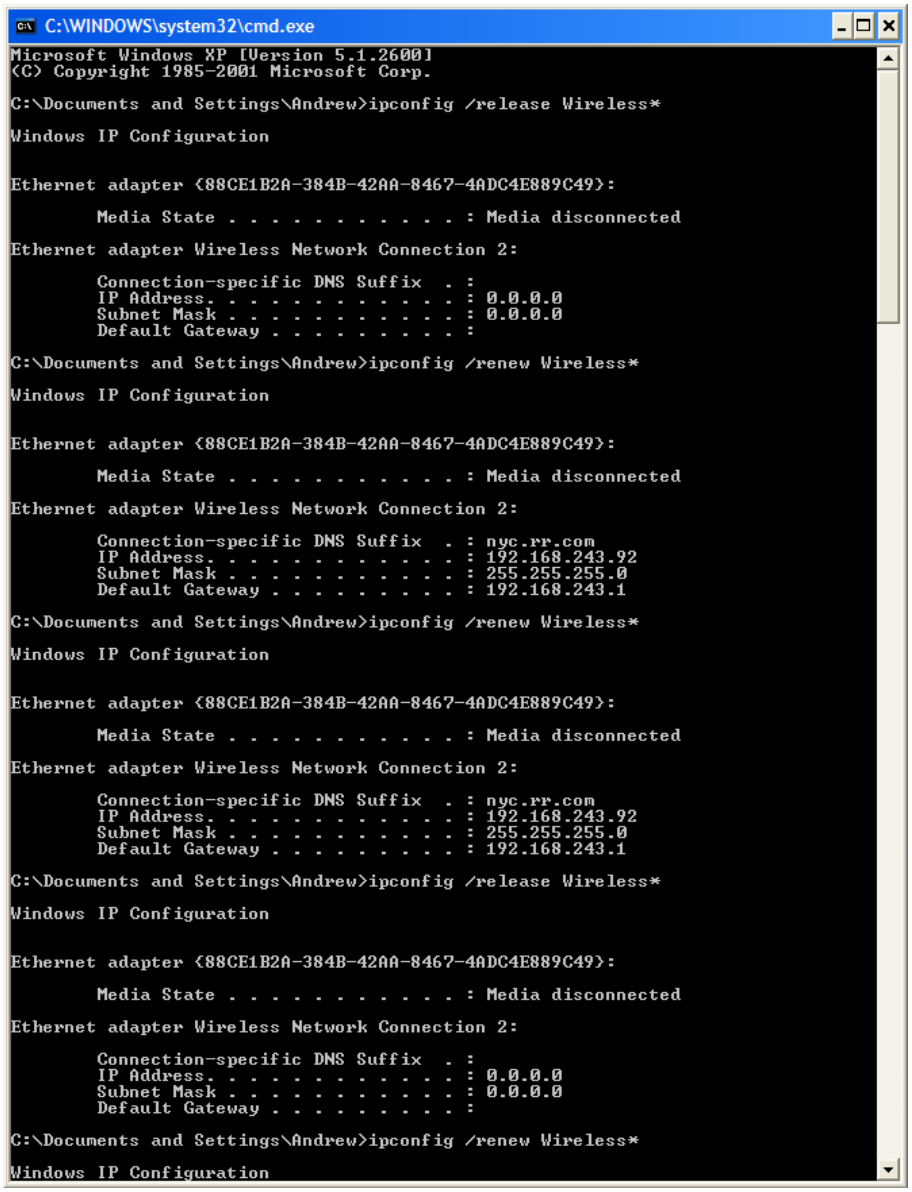
12.Εξηγείστε το ρόλο του χρόνου μίσθωσης (lease time). Ποιά η διάρκεια του χρόνου μίσθωσης στο πείραμα σας;

13.Ποιός ο ρόλος του μηνύματος DHCP Release; Στέλνει ο DHCP server επιβεβαίωση της λήψης του μηνύματος DHCP Request από τον client; Τι θα συνέβαινε εάν χάνοταν το μήνυμα DHCP Release του client;

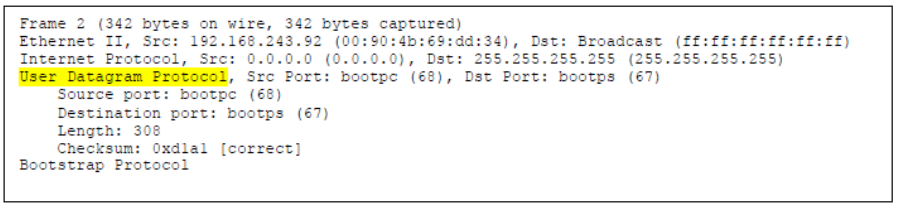
14.Καθαρίστε το bootp filter από το παράθυρο του Wireshark.Υπάρχουν άλλα ARP πακέτα που στάλθηκαν ή συλλήφθηκαν κατά την περίοδο ανταλλαγής πακέτων DCHP; Αν ναι, εξηγήστε το σκοπό αυτών των ARP πακέτων

(1) εάν δεν είστε σε θέση να τρέξετε το Ethereal σε μια σύνδεση δικτύου,ή έαν η σύνδεση δικτύου σας δεν σας επιτρέπει να εκτελέσετε το πείραμα DHCP φορώστε το αρχείο http://gaia.cs.umass.edu/ethereal-labs/etherreal-traces.zip και εξάγετε το αρχείο dhcp-ethereal-trace-1.Τα traces που περιέχονται σε αυτό το αρχείο zip συλλέχθηκαν από το Wireshark ενώ εκτελούνταν τα βήματα που περιγράφονται στο εργαστήριο Wireshark για το DNS στον υπολογιστή του συγγραφέα.Αφου λάβετε το trace,μπορείτε να το φορτώσετε στο Wieshark και να το δείτε στο παράθυρο χρησιμοποιώντας το μενού File,επιλέγοντας Open και στη συνέχεια επιλέγοντας το αρχείο dhcp-ethereal-trace-1 .Μετά μπορείτε να χρησιμοποιήσετε το trace που περιέχεται στο αρχείο αυτό για να απαντήσετε στις ερωτήσεις που θέτονται πατακάτω

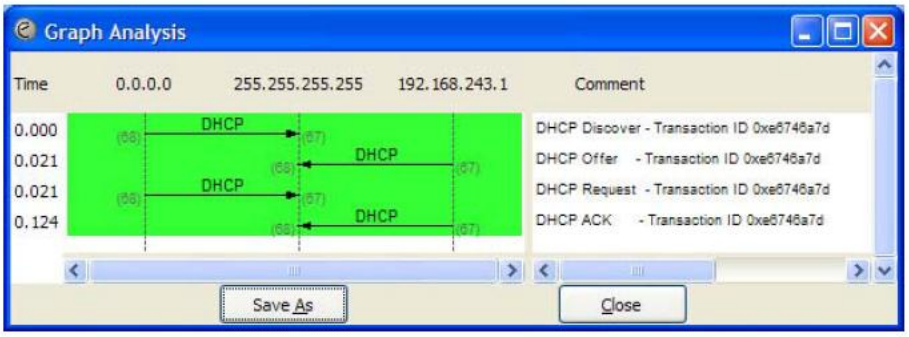
ΑΠΑΝΤΗΣΕΙΣ



1. Στέλνονται μέσω UDP



2.Είναι οι ίδιες με το παράδειγμα του εργαστηρίου.



2. Η διεύθυνση επιπέδου ζεύξης είναι 00:90:4b:69:dd:34

```
Frame 1 (342 bytes on wire, 342 bytes captured)
Ethernet II, Src: 192.168.243.92 (00:90:4b:69:dd:34), Dst: Broadcast
(ff:ff:ff:ff:ff:ff)
    Destination: Broadcast (ff:ff:ff:ff:ff:ff)
    Source: 192.168.243.92 (00:90:4b:69:dd:34)
    Type: IP (0x0800)
Internet Protocol, Src: 0.0.0.0 (0.0.0.0), Dst: 255.255.255.255 (255.255.255.255)
User Datagram Protocol, Src Port: bootpc (68), Dst Port: bootps (67)
Bootstrap Protocol
```

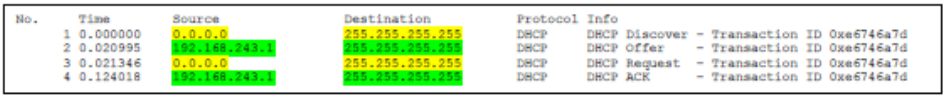
```
Frame 1 (342 bytes on wire, 342 bytes captured)
Ethernet II, Src: 192.168.243.92 (00:90:4b:69:dd:34), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
Internet Protocol, Src: 0.0.0.0 (0.0.0.0), Dst: 255.255.255.255 (255.255.255.255)
User Datagram Protocol, Src Port: bootpc (68), Dst Port: bootps (67)
Bootstrap Protocol
    Message type: Boot Request (1)
    Hardware type: Ethernet
    Hardware address length: 6
    Hops: 0
    Transaction ID: 0xe6746a7d
    Seconds elapsed: 1280
    Bootp flags: 0x0000 (Unicast)
    Client IP address: 0.0.0.0 (0.0.0.0)
    Your (client) IP address: 0.0.0.0 (0.0.0.0)
    Next server IP address: 0.0.0.0 (0.0.0.0)
    Relay agent IP address: 0.0.0.0 (0.0.0.0)
    Client MAC address: 192.168.243.92 (00:90:4b:69:dd:34)
    Server host name not given
    Boot file name not given
    Magic cookie: (OK)
    Option 53: DHCP Message Type = DHCP Discover
    Option 116: DHCP Auto-Configuration (1 bytes)
    Option 61: Client identifier
    Option 50: Requested IP Address = 192.168.243.92
    Option 12: Host Name = "homelt"
    Option 60: Vendor class identifier = "MSFT 5.0"
    Option 55: Parameter Request List
    End Option
    Padding

Frame 3 (350 bytes on wire, 350 bytes captured)
Ethernet II, Src: 192.168.243.92 (00:90:4b:69:dd:34), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
Internet Protocol, Src: 0.0.0.0 (0.0.0.0), Dst: 255.255.255.255 (255.255.255.255)
User Datagram Protocol, Src Port: bootpc (68), Dst Port: bootps (67)
Bootstrap Protocol
    Message type: Boot Request (1)
    Hardware type: Ethernet
    Hardware address length: 6
    Hops: 0
    Transaction ID: 0xe6746a7d
    Seconds elapsed: 1280
    Bootp flags: 0x0000 (Unicast)
    Client IP address: 0.0.0.0 (0.0.0.0)
    Your (client) IP address: 0.0.0.0 (0.0.0.0)
    Next server IP address: 0.0.0.0 (0.0.0.0)
    Relay agent IP address: 0.0.0.0 (0.0.0.0)
    Client MAC address: 192.168.243.92 (00:90:4b:69:dd:34)
    Server host name not given
    Boot file name not given
    Magic cookie: (OK)
    Option 53: DHCP Message Type = DHCP Request
    Option 61: Client identifier
    Option 50: Requested IP Address = 192.168.243.92
    Option 54: Server Identifier = 192.168.243.1
    Option 12: Host Name = "homelt"
    Option 81: FQDN
    Option 60: Vendor class identifier = "MSFT 5.0"
    Option 55: Parameter Request List
    End Option
```

4.Οι τιμές του μηνύματος DCHP Discover βρίσκονται στο “Option 53: DHCP Message Type”.

5.Η τιμή του πεδίου είναι 0xe6746a7d. Η τιμή του πεδίου στο δεύτερο σύνολο μηνυμάτων DCHP είναι 0xe4eff25f. Ο Transaction-ID χρησιμοποιείται ώστε να μπορεί ο DHCP server να διαφοροποιεί μεταξύ τους τις αιτήσεις πελατών κατά τη διάρκεια της διαδικασίας αιτήσεων.

```
No.     Time        Source                Destination           Protocol Info
      3 5.000175    0.0.0.0               255.255.255.255       DHCP     DHCP Discover - Transaction ID 0xe6746a7d
     27 12.075229   192.168.243.92        192.168.243.1         DHCP     DHCP Request  - Transaction ID 0xe4eff25f
```

6.Η διευθύνση προορισμού είναι 255.255.255.255 και για τον DHCP server και για τον client. Ο πελάτης και ο server χρησιμοποιούν τη διεύθυνση πηγής IP 0.0.0.0

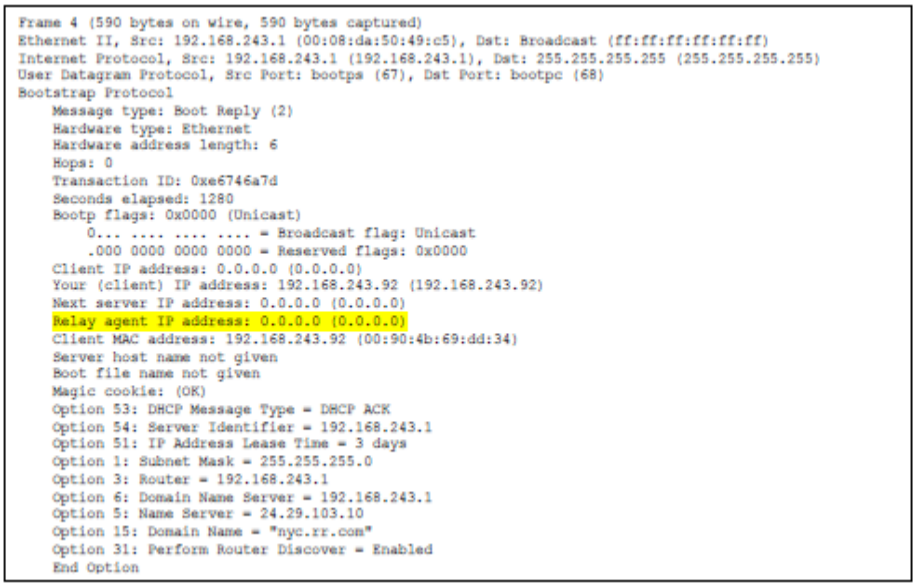


No.	Time	Source	Destination	Protocol	Info
1	0.000000	0.0.0.0	255.255.255.255	DHCP	DHCP Discover - Transaction ID 0xe6746a7d
2	0.020995	192.168.243.1	255.255.255.255	DHCP	DHCP Offer - Transaction ID 0xe6746a7d
3	0.021346	0.0.0.0	255.255.255.255	DHCP	DHCP Request - Transaction ID 0xe6746a7d
4	0.124018	192.168.243.1	255.255.255.255	DHCP	DHCP ACK - Transaction ID 0xe6746a7d

7.Η διεύθυνση IP του DHCP server είναι 192.168.243.1

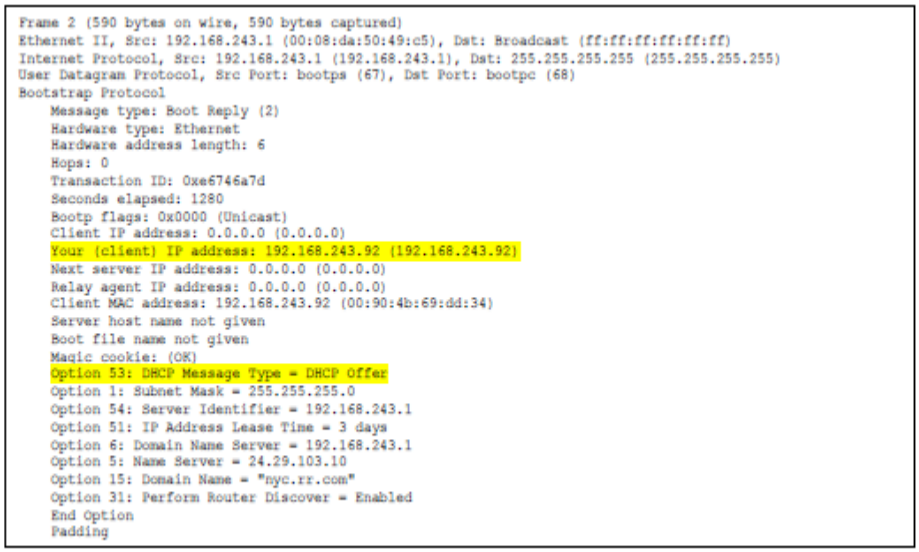
No.	Time	Source	Destination	Protocol	Info
4	0.124018	192.168.243.1	255.255.255.255	DHCP	DHCP ACK - Transaction ID 0xe6746a7d

8. Η διεύθυνση IP που προσφέρει ο DHCP server είναι 192.168.243.92. Το μήνυμα DHCP με "DHCP Message Type = DHCP Offer"περιέχει την προσφερόμενη διεύθυνση IP



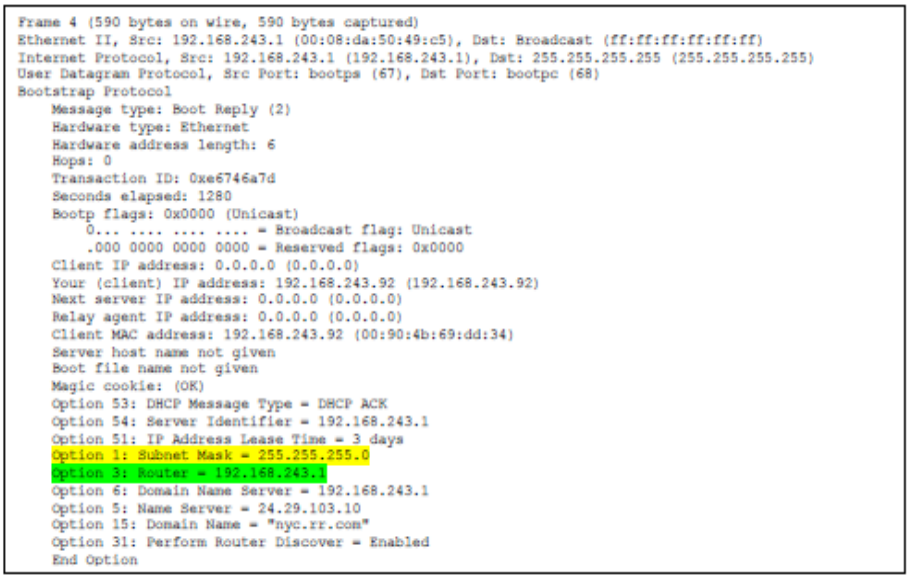
```
Frame 4 (590 bytes on wire, 590 bytes captured)
Ethernet II, Src: 192.168.243.1 (00:08:da:50:49:c5), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
Internet Protocol, Src: 192.168.243.1 (192.168.243.1), Dst: 255.255.255.255 (255.255.255.255)
User Datagram Protocol, Src Port: bootps (67), Dst Port: bootpc (68)
Bootstrap Protocol
    Message type: Boot Reply (2)
    Hardware type: Ethernet
    Hardware address length: 6
    Hops: 0
    Transaction ID: 0xe6746a7d
    Seconds elapsed: 1280
    Bootp flags: 0x0000 (Unicast)
        0... .... .... .... = Broadcast flag: Unicast
        .000 0000 0000 0000 = Reserved flags: 0x0000
    Client IP address: 0.0.0.0 (0.0.0.0)
    Your (client) IP address: 192.168.243.92 (192.168.243.92)
    Next server IP address: 0.0.0.0 (0.0.0.0)
    Relay agent IP address: 0.0.0.0 (0.0.0.0)
    Client MAC address: 192.168.243.92 (00:90:4b:69:dd:34)
    Server host name not given
    Boot file name not given
    Magic cookie: (OK)
    Option 53: DHCP Message Type = DHCP ACK
    Option 54: Server Identifier = 192.168.243.1
    Option 51: IP Address Lease Time = 3 days
    Option 1: Subnet Mask = 255.255.255.0
    Option 3: Router = 192.168.243.1
    Option 6: Domain Name Server = 192.168.243.1
    Option 5: Name Server = 24.29.103.10
    Option 15: Domain Name = "nyc.rr.com"
    Option 31: Perform Router Discover = Enabled
    End Option
```

9. Η διεύθυνση IP του relay agent είναι 0.0.0.0 ,το οποίο υποδεικνύει την απούσια ενός relay agent? Δεν υπάρχει relay agent στο πείραμα μου



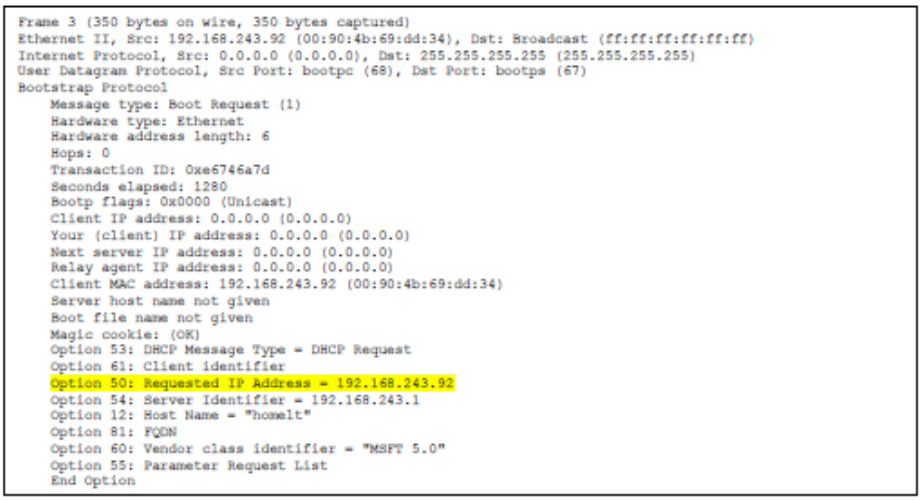
```
Frame 2 (590 bytes on wire, 590 bytes captured)
Ethernet II, Src: 192.168.243.1 (00:08:da:50:49:c5), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
Internet Protocol, Src: 192.168.243.1 (192.168.243.1), Dst: 255.255.255.255 (255.255.255.255)
User Datagram Protocol, Src Port: bootps (67), Dst Port: bootpc (68)
Bootstrap Protocol
    Message type: Boot Reply (2)
    Hardware type: Ethernet
    Hardware address length: 6
    Hops: 0
    Transaction ID: 0xe6746a7d
    Seconds elapsed: 1280
    Bootp flags: 0x0000 (Unicast)
    Client IP address: 0.0.0.0 (0.0.0.0)
    Your (client) IP address: 192.168.243.92 (192.168.243.92)
    Next server IP address: 0.0.0.0 (0.0.0.0)
    Relay agent IP address: 0.0.0.0 (0.0.0.0)
    Client MAC address: 192.168.243.92 (00:90:4b:69:dd:34)
    Server host name not given
    Boot file name not given
    Magic cookie: (OK)
    Option 53: DHCP Message Type = DHCP Offer
    Option 1: Subnet Mask = 255.255.255.0
    Option 54: Server Identifier = 192.168.243.1
    Option 51: IP Address Lease Time = 3 days
    Option 6: Domain Name Server = 192.168.243.1
    Option 5: Name Server = 24.29.103.10
    Option 15: Domain Name = "nyc.rr.com"
    Option 31: Perform Router Discover = Enabled
    End Option
    Padding
```

10. Η γραμμή router υποδεικνύει στον client ποιά είναι η προεπιλεγμένη έξοδος. Η γραμμή subnet mask υποδεικνύει στον client ποιά subnet mask να χρησιμοποιήσει.



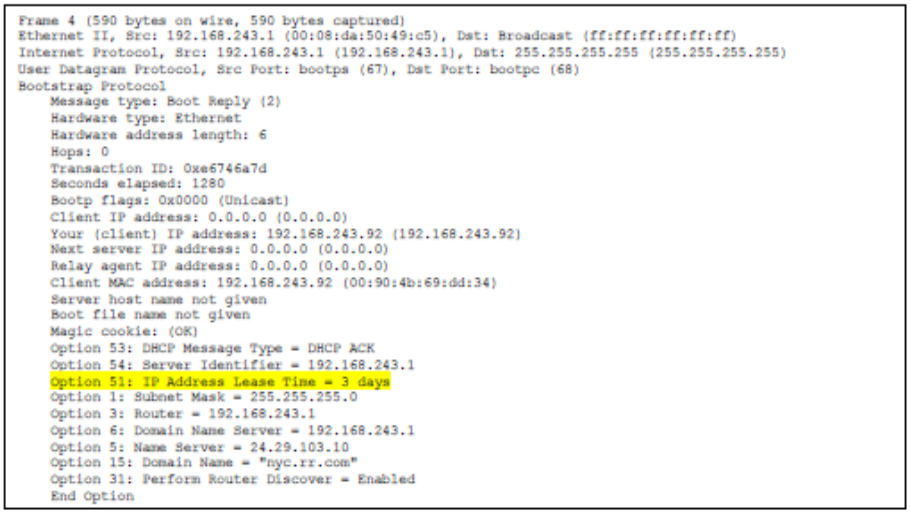
```
Frame 4 (590 bytes on wire, 590 bytes captured)
Ethernet II, Src: 192.168.243.1 (00:08:da:50:49:c5), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
Internet Protocol, Src: 192.168.243.1 (192.168.243.1), Dst: 255.255.255.255 (255.255.255.255)
User Datagram Protocol, Src Port: bootps (67), Dst Port: bootpc (68)
Bootstrap Protocol
    Message type: Boot Reply (2)
    Hardware type: Ethernet
    Hardware address length: 6
    Hops: 0
    Transaction ID: 0xe6746a7d
    Seconds elapsed: 1280
    Bootp flags: 0x0000 (Unicast)
        0... .... .... .... = Broadcast flag: Unicast
        .000 0000 0000 0000 = Reserved flags: 0x0000
    Client IP address: 0.0.0.0 (0.0.0.0)
    Your (client) IP address: 192.168.243.92 (192.168.243.92)
    Next server IP address: 0.0.0.0 (0.0.0.0)
    Relay agent IP address: 0.0.0.0 (0.0.0.0)
    Client MAC address: 192.168.243.92 (00:90:4b:69:dd:34)
    Server host name not given
    Boot file name not given
    Magic cookie: (OK)
    Option 53: DHCP Message Type = DHCP ACK
    Option 54: Server Identifier = 192.168.243.1
    Option 51: IP Address Lease Time = 3 days
    Option 1: Subnet Mask = 255.255.255.0
    Option 3: Router = 192.168.243.1
    Option 6: Domain Name Server = 192.168.243.1
    Option 5: Name Server = 24.29.103.10
    Option 15: Domain Name = "nyc.rr.com"
    Option 31: Perform Router Discover = Enabled
    End Option
```

11. Στο δικό μας πείραμα ο host ζητά την προσφερόμενη IP διεύθυνση στο DHCP Request μύνημα



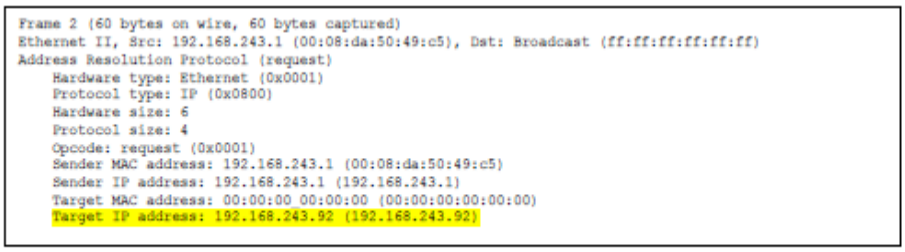
```
Frame 3 (350 bytes on wire, 350 bytes captured)
Ethernet II, Src: 192.168.243.92 (00:90:4b:69:dd:34), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
Internet Protocol, Src: 0.0.0.0 (0.0.0.0), Dst: 255.255.255.255 (255.255.255.255)
User Datagram Protocol, Src Port: bootpc (68), Dst Port: bootps (67)
Bootstrap Protocol
    Message type: Boot Request (1)
    Hardware type: Ethernet
    Hardware address length: 6
    Hops: 0
    Transaction ID: 0xe6746a7d
    Seconds elapsed: 1280
    Bootp flags: 0x0000 (Unicast)
    Client IP address: 0.0.0.0 (0.0.0.0)
    Your (client) IP address: 0.0.0.0 (0.0.0.0)
    Next server IP address: 0.0.0.0 (0.0.0.0)
    Relay agent IP address: 0.0.0.0 (0.0.0.0)
    Client MAC address: 192.168.243.92 (00:90:4b:69:dd:34)
    Server host name not given
    Boot file name not given
    Magic cookie: (OK)
    Option 53: DHCP Message Type = DHCP Request
    Option 61: Client identifier
    Option 50: Requested IP Address = 192.168.243.92
    Option 54: Server Identifier = 192.168.243.1
    Option 12: Host Name = "homelt"
    Option 81: FQDN
    Option 60: Vendor class identifier = "MSFT 5.0"
    Option 55: Parameter Request List
    End Option
```

12. Ο χρόνος μίσθωσης είναι το σύνολο του χρόνου που ο DHCP server αναθέτει μια IP διεύθυνση στον client. Στην διάρκεια αυτή ο DHCP server δεν αναθέτει την IP σε άλλον client. Όταν ο χρόνος λήξει η IP διεύθυνση μπορει να ξαναχρησιμοποιηθεί από τον DHCP server και να δωθεί σε άλλον client. Στο δικό μας πείραμα, ο χρόνος μίσθωσης είναι 3 μέρες.



13. Ο client στέλνει ένα μήνυμα DHCP Release για να ακυρώσει την μίσθωση της IP διεύθυνσης που του δώθηκε από τον DHCP server. Ο DHCP server δεν στέλνει μύνημα επιβεβαίωσης. Αν το μήνυμα DHCP Release του client χαθεί, ο DHCP server πρέπει να περιμένει να λήξει ο χρόνος μίσθωσης

14. Ναι,υπάρχουν. Πρίν προσφέρει μια IP διεύθυνση στον πελάτη, ο DHCP server στέλνει μια ARP αίτηση για την προσφερόμενη IP ώστε να είναι σίγουρο ότι δεν χρησιμοποιείται σε κάποια άλλη εργασία.



Κεφάλαιο 12
Ασκήσεις επίδειξης του πρωτοκόλλου 802.11

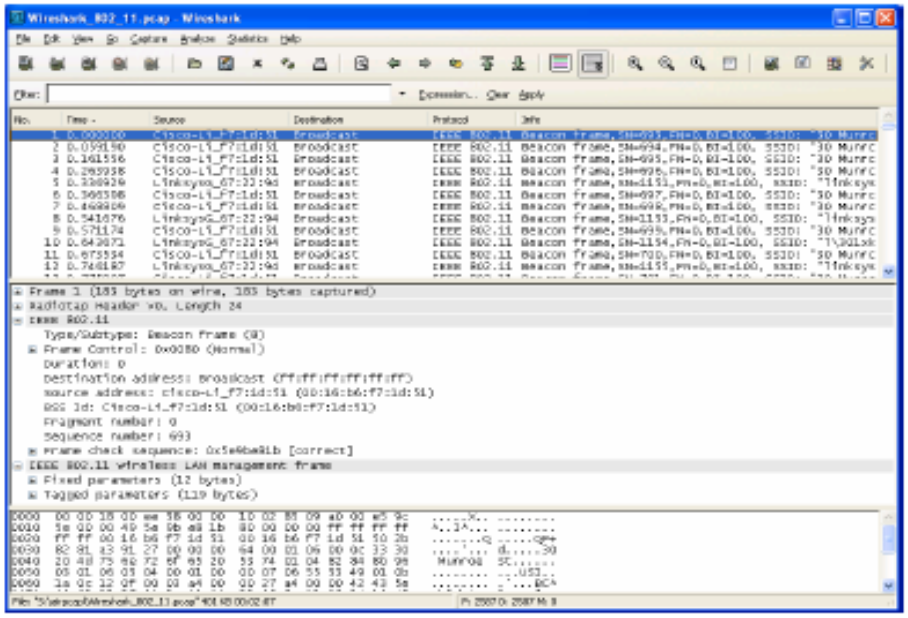
Ως ασύρματο δίκτυο χαρακτηρίζεται το τηλεπικοινωνιακό δίκτυο, συνήθως τηλεφωνικό ή δίκτυο υπολογιστών, το οποίο χρησιμοποιεί, ραδιοκύματα ως φορείς πληροφορίας. Το IEEE 802.11 είναι μια οικογένεια πρωτοκόλλων που αποτελεί το καθιερωμένο πρότυπο της βιομηχανίας στο χώρο των ασύρματων τοπικών δικτύων. Τα πρότυπα 802.11 είναι ευρύτερα γνωστά και ως WiFi.
Σ'αυτο το εργαστήριο θα εξετάσουμε το πρωτόκολλο για ασύρματο Δίκτυο 802.11. Σε όλα τα εργαστήρια wireshark μέχρι τώρα, συλλαμβάναμε μυνήματα σε καλωδιομένη σύνδεση Ethernet. Εδώ, από τη στιγμή που το 802.11 είναι πρωτόκολλο ασύρματης σύνδεσης, θα συλλαμβάνουμαι μηνύματα 'στον αέρα'. Δυστυχώς, οι περισσότερες συσκευές εγκατάστασης του ασύρματου 802.11 NIC (ειδικά τα windows) δεν υποστηρίζουν τη σύλληψη μηνυμάτων του 802.11 πρωτοκόλλου για να τα χρήσιμοποιήσουμε στο wireshark. Γι' αυτό, σε αυτό το εργαστήριο θα χρησιμοποιήσουμε ενα trace με συλληφθέντα μηνύματα του 802.110

12.1 Αρχικά

Κατεβάστε το αρχείο http://gaia.cs.umass.edu/wireshark-labs/wireshark-traces.zip και εξάγετε (extract) το αρχειο wireshark_802_11.pcap. Αυτό το trace συλλέχθηκε χρησιμοποιόντας AirPcap και το wireshark σε pc του συγγραφέα. Σε αυτό το trace, θα δούμε μυνηματα που συλλήφθηκαν στο κανάλι 6. Επειδή υπάρχουν και άλλοι χρήστες που χρησιμοποίουν το κανάλι 6, θα δούμε μηνύματα τα οποία δεν αφορούν αυτό το εργαστήριο.

Οι δραστηριότητες του ασύρματου δεκτη σε αυτό το trace είναι
>Ο δέκτης είναι ήδη συνδεμένος με την 30 Munroe St AP όταν ξεκινά το trace
>Όταν t=24.82, ο δέκτης κάνει μια HTTP αίτηση στη http://gaia.cs.umass.edu/wireshark-labs/alice.txt. Η IP διεύθυνση της gaia.cs.umass.edu είναι 128.119.245.12
>Όταν t=32.82, ο δέκτης κάνει μια HTTP αίτηση στη http://www.cs.umass.edu .Η IP διεύθυνση της είναι 128.119.240.19.
>Όταν t=49.58, ο δέκτης αποσυνδεέται απο την 30 Munroe St AP και προσπαθεί να συνδεθεί με τη linksys_ses_24086 AP. Δεν είναι σημείο ελεύθερης πρόσβασης με αποτέλεσμα να μην τα καταφέρνει
>Όταν t=63.0 ο δέκτης σταματά τις προσπάθειες να συνδεθεί με linksys_ses_24086 AP και συνδεέται ξανά με την 30 Munroe St

Αφού κατεβάσαμε το trace, μπορούμε να το φορτώσουμε στο wireshark και να το μελετήσουμε. Από το file επιλέγουμε open και επιλέγουμε Wireshark_802_11.pcap trace file και εμφανίζεται το Σχ.46



Σχ.46 το παράθυρο του wireshark αφού ανοίξουμε το αρχείο Wireshark_802_11.pcap

12.2 Συλλαμβανόμενα Μυνήματα

Επαναφέρουμε τα συλλαμβανόμενα μυνήματα που χρησιμοποιήθηκαν από το 802.11AP. Για να απαντήσουμε σε κάποιες ερωτήσεις παρακάτω, θα πρέπει να δούμε τις λεπτομέρειες στη λίστα "IEEE 802.11"frame και στις λεπτόμερειες στο μεσαίο παράθυρο του wireshark

ΕΡΩΤΗΣΕΙΣ

1.Ποιά τα SSIDs των 2 σημείων πρόσβασης που εμφανίζονται περισσότερο στα συλλαμβανόμενα μυνήματα σε αυτο το trace;

2.Ποιά τα χρονικά διαστήματα μεταξύ της μετάδοσης των συλλαμβανόμενων μηνύματων του σημείου πρόσβασης της linksys _ses_24086; Από το σημείο πρόσβασης της *30 Munroe St.* ;

3.Ποιά είναι (σε δεκαεξαδική μορφή) η διεύθυνση της πηγής MAC του συλλαμβανόμενου μηνύματος απο την 30 Munroe St;

4.Ποιά είναι (σε δεκαεξαδική μορφή) η διεύθυνση του προορισμου MAC του συλλαμβανόμενου μηνύματος απο την 30 Munroe St;

5.Ποιά ειναι (σε δεκαεξαδική μορφή) η MAC BSS id του συλλαμβανόμενου μηνύματος απο την 30 Munroe St;

6.Τα συλλαμβανόμενα μυνήματα του σημείου πρόσβασης 30 Munroe St υποστήριζουν 4 μεγέθη πληροφορίας και 8 επιπλέον επεκτηνώμενα (extended supported rates). Ποιά είναι τα μεγέθη αυτά;

12.3 Μεταφορά Δεδομένων

Το trace ξεκινά έχοντας ήδη συνδεθεί ο δέκτης με την AP, ας ρίξουμε μια ματιά στη μεταφορά δεδομένων της 802.11 σύνδεσης πριν ασχοληθουμε με την AP συνδεση/αποσύνδεση. Θυμόμαστε οτι σε αυτό το trace όταν t=24.82, ο δέκτης κάνει μια HTTP αίτηση στη http://gaia.cs.umass.edu/wireshark-labs/alice.txt Η IP διεύθυνση της gaia.cs.umass.edu είναι 128.119.245.12. Όταν t=32.82 ο δέκτης κάνει μια HTTP αίτηση στη http://www.cs.umass.edu

ΕΡΩΤΗΣΕΙΣ

7.Βρείτε το μύνημα της 802.11 σύνδεσης που περιέχει το SYN TCP πακέτο για την πρώτη TCP διαδικασία (αυτή που κατέβασε το alice.txt). Ποιά χρονική στιγμή στάλθηκε το TCP SYN; Ποιές είναι οι 3 mac διευθύνσεις σε αυτο το μήνυμα; Ποιά MAC διευθυνση αντιστοιχεί στον ασύρματο δέκτη (σε 16δικη μορφή); Ποιά στο σημείο πρόσβασης; Ποιά στον δρομολογητή πρώτου άλματος (first-hop router); Ποιά η IP διευθυνση του ασύρματου δέκτη που στέλνει αυτό το TCP πακέτο; Ποιά η IP διεύθυνση προορισμού; Αντιστοιχεί αυτή η IP διεύθυνση στο δέκτη, στο σημείο πρόσβασης, στο first-hop router, ή άλλες network-attached συσκευές; Εξηγήστε

8.Βρείτε το μύνημα της 802.11 σύνδεσης που περιέχει το SYNACK πάκετο για την πρώτη TCP διαδικασία. Ποιά στιγμή συλληφθηκε το TCP SYNACK; Ποιές είναι οι 3 mac διευθύνσεις σε αυτό το μυνημα που περιέχουν το SYNACK; Ποιά MAC διεύθυνση αντιστοιχεί στον ασύρματο δέκτη (σε 16δικη μορφή) στο σημείο πρόσβασης, στο first-hop router; Ποιά η IP διεύθυνση του ασύρματου δέκτη που στέλνει αυτο το TCP τμήμα; Ποιά η διεύθυνση προορισμού; Αντιστοιχεί η MAC διεύθυνση του αποστολέα στην IP διεύθυνση της συσκεής που στέλνει το TCP πακέτο;

12.4 Σύνδεση /Αποσύνδεση

Θυμόμαστε οτι ο δέκτης πρέπει πρώτα να συνδεθεί με ένα σημείο πρόσβασης πρίν αρχίσει να στέλνει πληροφορίες. Η σύνδεση 802.11 εκτελείται με την αίτηση σύνδεσης (που στέλνεται απο τον δέκτη στην AP) και την απάντηση σύνδεσης (που στέλνεται απο την AP στον δέκτη). Για περισσότερες λεπτόμεριες κοιτάξτε την σελ.34 (section 7) http://gaia.cs.umass.edu/wireshark-labs/802.11-1999.pdf.

ΕΡΩΤΗΣΕΙΣ

9.Ποιές είναι οι 2 δραστηριότητες (τα μυνήματα έχουν σταλεί) του δέκτη σε αυτό το trace από τη στιγμή t=49, μέχρι το τέλος της σύνδεσης με τη *30 Munroe St* AP και ποιά στιγμή στάλθηκαν τα μηνύματα; (Σημείωση>η μια δραστηριότητα είναι της IP και

η άλλη της 802.11). Κοιτώντας τις λεπτομέρειες της 802.11 υπάρχει κάποιο μήνυμα που περιμένατε να δείτε αλλά δεν είναι εδώ;

10.Εξετάστε το trace αρχείο και βρείτε AUTHENICATION μηνύματα που σταλθηκαν από το δέκτη στην AP και αντιστρόφως. Πότε στάλθηκε το πρώτο AUTHENICATION μήνυμα από τον ασύρματο δέκτη στο linksys_ses_24086 AP (με MAC διεύθυνση Cisco_Li_f5:ba:bb) μετά τη στιγμή t=49;

11.Χρειάζεται ο host την authentication ώστε να απαιτεί κλειδί ή να΄ναι ανοιχτός;

12.Βλέπετε απάντηση στο AUTHENICATION της linksys_ses_24086 AP σε αυτό το trace;

13.Τώρα ας δούμε τη συμβαίνει όταν ο δέκτης παρατά τις προσπάθειες (κάπου μετά από t=63) να συνδεθεί με το linksys_ses_24086 AP και μετά προσπαθεί να συνδεθεί με 30 Munroe St AP. Εξετάστε και βρείτε AUTHENICATION μηνύματα που στάλθηκαν από το δέκτη στην AP και αντιστρόφως. Ποιές στιγμές υπάρχουν AUTHENTICATION μυνήματα από τον δέκτη στην 30 Munroe St. AP, και πότε υπάρχει AUTHENTICATION απάντηση από την AP στον δέκτη; (σημειώστε οτι μπορείτε να χρησιμοποιήσετε τα φίλτρα filter "wlan.fc.subtype==11 και wlan.fc.type ==0 και wlan.addr==IntelCor_d1:b6:4f" ώστε να βλέπετε μόνο τα AUTHENTICATION frames σε αυτό το trace)

14.Ας συνεχίσουμε με την σύνδεση μεταξύ του ασύρματου δέκτη και 30 Munroe St AP που συμβαίνει μετά την στιγμή t=63.0. Μια αίτηση σύνδεσης απο τον δέκτη στην AP και μια απάντηση από το AP αντίστοιχα στο δέκτη χρησιμοποιούνται ώστε να συνδεθεί ο δέκτης με την AP. Ποιά στιγμή στέλνεται η αίτηση απο το δέκτη στη AP; Πότε στέλνεται η αντίστοιχη απάντηση;
(Σημειώστε ότι μπορείτε να χρησιμοποιήσετε τα φίλτρα "wlan.fc.subtype<2 και wlan.fc.type==0 και wlan.addr==IntelCor_d1:b6:4f" ώστε να βλέπετε μόνο τα ASSOCIATE REQUEST και ASSOCIATE RESPONSE σε αυτό το trace)

15.Ποιές τιμές μεταβίβασεις επιθυμεί ο δέκτης να χρησιμοποιήσει; Ποιές ο AP; Για να απαντήσετε σε αυτή την ερώτηση, θα χρειαστεί να κοιτάξετε στις παραμέτρους του 802.11 wireless LAN management frame.

12.5 Άλλοι Τύποι Μηνυμάτων

Στο trace περιέχονται PROBE REQUEST και PROBE RESPONSE frames.

16.Εξετάστε το πρώτο PROBE REQUEST και την ακόλουθη PROBE RESPONSE PAIR που εμφανίζεται μετά από t=2.0 sec στο trace. Πότε στάλθηκαν τα μηνύματα αυτά και ποιές είναι οι διευθύνσεις του αποστολέα, δέκτη, BSS ID MAC; Ποιός ο σκοπός αυτών των δυο τύπων μηνυμάτων;

ΑΠΑΝΤΗΣΕΙΣ

1.30 Munroe St και linsys_SES_24086

2.Και για τις δυο το χρονικό διάστημα είναι 0.1024 seconds.

3.Η διεύθυνση της πηγής είναι 00:16:b6:f7:1d:51

4.Η διεύθυνση του προορισμού είναι ff:ff:ff:ff:ff:ff

5.Η MAC BSS id του συλλαμβανόμενου μηνύματος είναι 00:16:b6:f7:1d:51 (είναι η ίδια με την διεύθυνση πηγής)

6.Τα μεγέθη πληροφορίας που υποστηρίζει είναι 1.0, 2.0, 5.5, 11.0 Mbps. The επεκτενόμενα μεγέθη είναι 6.0, 9.0, 12.0, 18.0, 24.0, 36.0, 48.0 και 54.0 Mbps

7.Το TCP SYN στάλθηκε τη χρονική στιγμή t = 24.811093 seconds. Η MAC διεύθυνση του host που στέλνει το TCP SYN είναι 00:13:02:d1:b6:4f.Η MAC διεύθυνση προορισμού η οποία αντιστοιχεί και στον πρώτο hop router είναι 00:16:b6:f4:eb:a8. Η MAC διεύθυνση για το BSS είναι 00:16:b6:f7:1d:51. Η IP διεύθνση του host που στέλνει το TCP SYN είναι 192.168.1.109. Η διεύθυνση προορισμού είναι 128.199.245.12. και αντιστοιχεί στον server του gaia.cs.umass.edu.

8.Το TCP SYNACK συλλέχθηκε όταν t = 24.827751 seconds . Η MAC διεύθυνση του αποστολέα είναι 00:16:b6:f4:eb:a8, η οποία είναι και η διεύθυνση του πρώτουι hop router. Η MAC διεύθυνση προορισμού είναι 91:2a:b0:49:b6:4f. Η MAC διεύθυνση του BSS is 00:16:b6:f7:1d:51. Η IP διεύθυνση του server που έστειλε το TCP SYNACK είναι 128.199.245.12 (gaia.cs.umass.edu). Η διεύθυνση προορισμού είναι 192.168.1.109 (το δικό μας ασύρματο PC).

9.Την χρονική στιγμή t = 49.583615 ένα DHCP release στέλνεται από τον host (με IP διεύθυνση 192.168.1.1) στο δίκτυο όπου φιλοξενείται ο. Την στιγμή t=49.609617, ο host στέλνει ένα DEAUTHENTICATION frame (Frametype = 00[Management], subframe type = 12[Deauthentication]). Το μήνημα που περιμέναμ ε να δούμε είναι μια DISASSOCIATION request

10.Στάλθηκε τη χρονική στιγμή t = 49.638857.

11.Την χρειάζεται ώστε ο σύνδεσμος (association) να είναι ανοιχτός (be open (by specifying Authentication Algorithm: Open System).)

12.Όχι

13.Την στιγμή t = 63.168087 υπάρχει ένα AUTHENTICATION frame σταλμένο από την 00:13:02:d1:b6:4f (ο ασύρματος host) στο 00:16:b7:f7:1d:51 (BSS). Την στιγμή t =63.169071 υπάρχει ένα AUTHENTICAN με την αντίθετη κατεύθυνση

14.Την στιγμή t= 63.169910 υπάρχει ένα ASSOCIATE REQUEST frame σταλμένο από την 00:13:02:d1:b6:4f (ο ασύρματος host) στο 00:16:b7:f7:1d:51 (BSS). Την

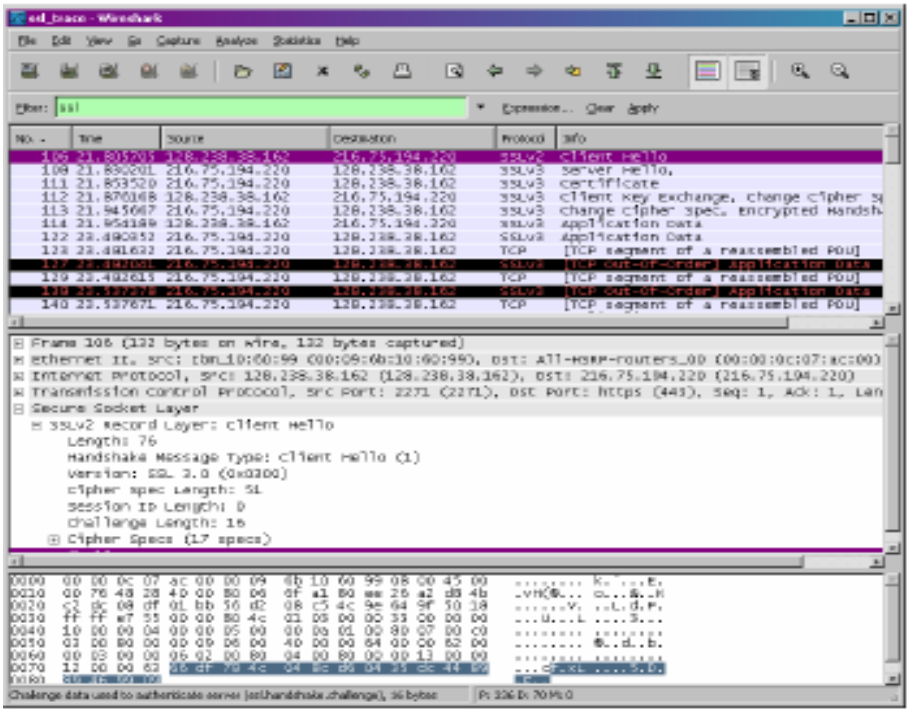
στιγμή t=63.192101 υπάρχει ένα ASSOCIATE RESPONSE με την αντίθετη κατεύθυνση

15.Για το ASSOCIATION REQUEST οι τιμές είναι 1, 2, 5.5, 11, 6, 9, 12, 18, 24, 32, 48, και 54 Mbps. Ομοίως για ASSOCIATION RESPONSE.

16.Την στιγμή t=2.297613 υπάρχει ένα PROBE REQUEST σταλμένο από 00:12:f0:1f:57:13 με προορισμό: ff:ff:ff:ff:ff:ff, και BSSID of ff:ff:ff:ff:ff:ff. Την στιγμή t=2.300697 υπάρχει ένα PROBE RESPONSE σταλμένο από 00:16:b6:f7:1d:51, με προορισμό και BSSID 00:16:b6:f7:1d:51. Το PROBE REQUEST χρησιμοποιείται από τον host για να βρεί ένα σημείο πρόσβασης. Ένα PROBE RESPONSE στέλνεται από το σημείο πρόσβασης στον host που έστειλε την αίτηση.

Κεφάλαιο 13
Ασκήσεις επίδειξης του πρωτοκόλλου SSL

Το πρωτόκολλο SSL (Secure Sockets Layer) σχεδιάστηκε για να παρέχει ασφάλεια κατά την μετάδοση ευαίσθητων δεδομένων στο διαδίκτυο. Το SSL χρησιμοποιεί μεθόδους κρυπτογράφησης των δεδομένων που ανταλλάσσονται μεταξύ δύο Ηλεκτρονικών Υπολογιστών εγκαθιδρύοντας μία ασφαλή σύνδεση μεταξύ τους μέσω του διαδικτύου.
Σε αυτό το εργαστήριο, θα εξετάσουμε το πρωτόκολλο ασφάλειας SSL, επικεντρώνοντας στα SSL δεδομενα που στελνονται μέσω μιας TCP σύνδεσης. Θα αναλήσουμε ένα trace όπου SSL δεδομένα στέλνονται μεταξύ του δικού μας host και ενός server ηλεκτρονικού εμπορίου. Θα εξετάσουμε τους διάφορους τύπους SSL δεδομένων καθώς και τα πεδία των SSL μηνυμάτων



Σχ.47 SSL δεδομένα

13.1 Συλλαμβάνοντας Πακέτα σε μια SSL session

Το πρώτο βήμα είναι η σύλληψη πακέτων σε μια SSL session. Για να γίνει αυτό, θα πρέπει να επισκευθούμε το αγαπημένο μας site ηλεκτρονικού εμπορίου και να ξεκινήσουμε την διαδικασία αγοράς ενός αντικειμένου (αλλά θα διακόψουμε την διαδικασία πρίν γίνει πραγματική αγορά). Αφού συλλέξουμε τα πακέτα με το Wireshark, θα πρέπει να ρυθμίσουμε το φιλτρο ώστε να εμφανίζει εκείνα τα ethernet μηνύματα τα οποία περιέχουν SSL δεδομένα σταλμένα και ληφθέντα από τον δικό μας host (ένα SSL δεδομένα είναι το ίδιο με ένα SSL μήνυμα)

Αν είναι δύσκολο να δημιουργήσετε ένα trace, μπορείτε να φορτώσετε το αρχείο http://gaia.cs.umass.edu/wireshark-labs/wireshark-traces.zip και να εξάγετε το trace ssl-ethereal-trace-1

13.2 Μια Ματιά στο trace

Το Wireshark GUI θα πρέπει να εμφανίζει μόνο τα Ethernet frames που έχουν SSL δεδομένα. Είναι σημαντκό να έχουμε στο νου μας οτι ένα Ethernet frame μπορεί να περιέχει ένα ή περισσότερα SSL δεδομένα. (Αυτό είναι πολύ διαφορετικό από το HTTP, επειδή κάθε frame περιέχει ένα ολοκληρομένο HTTP μήνυμα ή ένα μέρος του HTTP μηνύματος). Επίσης, ένα SSL δεδομένο μπορεί να μην ταιριάζει απόλυτα στο Ethernet frame, οπότε σε αυτή την περίπτωση πολλαπλά frames θα χρειαστούν να φέρουν το δεδομένο.

ΕΡΩΤΗΣΕΙΣ

1. Για κάθε ένα από τα πρώτα 8 Ethernet frames, προσδιορίστε την πηγή του frame (πελάτης ή server), καθορίστε τον αριθμό SSL δεδομένων που περιέχονται στο frame, και ταξινομήστε τους τύπους των SSL δεδομένων που περιέχονται στο frame. Σχεδίαστε ένα χρονικό διάγραμμα μεταξύ πελάτη και server, με ένα βέλος για κάθε SSL δεδομένο.

2. Καθένα από τα SSL δεδομένα ξεκινά με τα ίδια τρία πεδία (πιθανόν με διαφορετικές τιμές). Ένα από τα πεδία είναι ‘τύπος περιεχομένου’ (‘content type’) και έχει μέγεθος ένα byte. Ταξινομήστε και τα τρια πεδία και τα τρια μεγέθη τους.

ClientHello

3.Εκτείνετε το ClientHello Record (αν στο δικό σας trace περιέχονται πολλαπλά ClientHello αρχεία, εκτείνετε το frame που περιέχει το πρώτο). Ποιά είναι η τιμή του content type;

4.Περιέχει το ClientHello αρχείο ένα nonce (επίσης γνωστό σαν ‘πρόκληση’); Αν ναι, ποιά η τιμή του nonce σε δεκαεξαδική μορφή;

5.Το ClientHello προβάλλει τα ‘cyber suites’ που υποστηρίζει; Αν ναι, στο πρώτο suite της λίστας ποίος είναι ο αλγόριθμος δημοσιοποιημένου κλειδιού, ο αλγόριθμος συμμετρικού κλειδιού, και ο αλγόριθμος hash;

ServerHello Record

6.Εντοπίστε το ServerHello SSL δεδομένο. Προσδιορίζει αυτό το αρχείο ένα επιλεγμένο cipher suite (πακέτο των αλγορίθμων κρυπτογράφησης); Ποιός είναι ο αλγόριθμος του επιλεγμένου cipher suite;

7.Περιέχει αυτό το δεδομένο ένα nonce; Αν ναι, πόσο μεγάλο είναι; Ποιός είναι ο σκοπός των nonce του πελάτη και του server στο SSL;

8.Περιέχει αυτό το δεδομένο ένα session ID; Ποιός είναι ο σκοπος του session ID;

9.Περιέχει αυτό το δεδομένο ένα πιστοποιητικό (certificate), ή το πιστοποιητικό περιέχεται σε ξεχωριστό δεδομένο; Εντάσσεται το πιστοποιητικό σε ένα ξεχωριστό Ethernet frame;

Client Key Exchange Record

10. Εντοπίστε το αρχείο κωδικού συναλλαγής πελάτη. Περιέχει αυτό το αρχείο μια μυστική ερώτηση; Που χρησιμοποιείται αυτή η ερώτηση; Είναι η ερώτηση κρυπτογραφημένη; Αν ναι, πώς; Πόσο μεγάλη είναι η κρυπτογραφημενη ερώτηση;

Change Cipher Spec Record (στέλνεται από τον client) και Encrypted Handshake Record:

11.Ποιός είναι ο σκοπός της αλλαγής κρυπτογραφημένου δεδομένου; Πόσα bytes είναι το δεδομένο στο trace σας;

12.Στο κρυπτογραφημένο handshake record, τι έχει αποκρυπτογραφηθεί; Πώς;

13.Έστειλε και ο server ένα change cipher record και ένα κρυπτογραφημένο handshake record στον client;

Αρχεία εφαρμογής

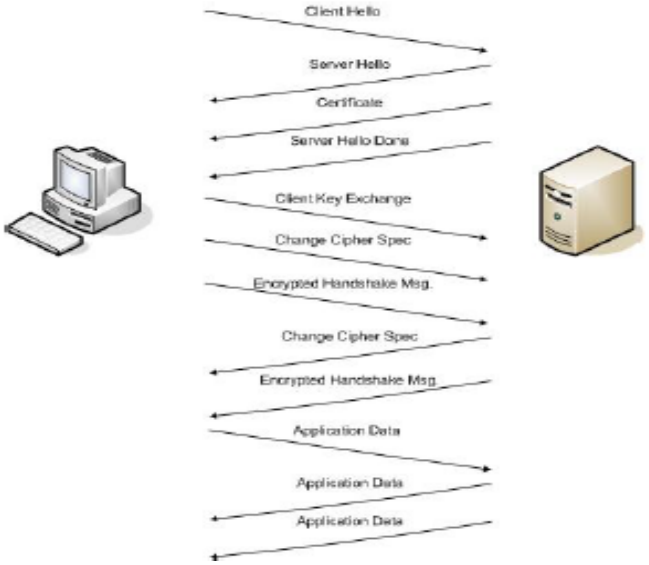
14.Πως κρυπτογραφούνται τα αρχεία εφαρμογης; Τα μυνήματα που περιέχουν τα αρχεία εφαρμογής περιέχουν επίσης ένα MAC; Ξεχωρίζει το Wireshark τα κρυπτογραφηνένα αρχεία εφαρμογής από το MAC;

15. Σχολιάστε και εξηγείστε ότιδήποτε άλλο που βρήκατε ενδιαφέρον στο trace.

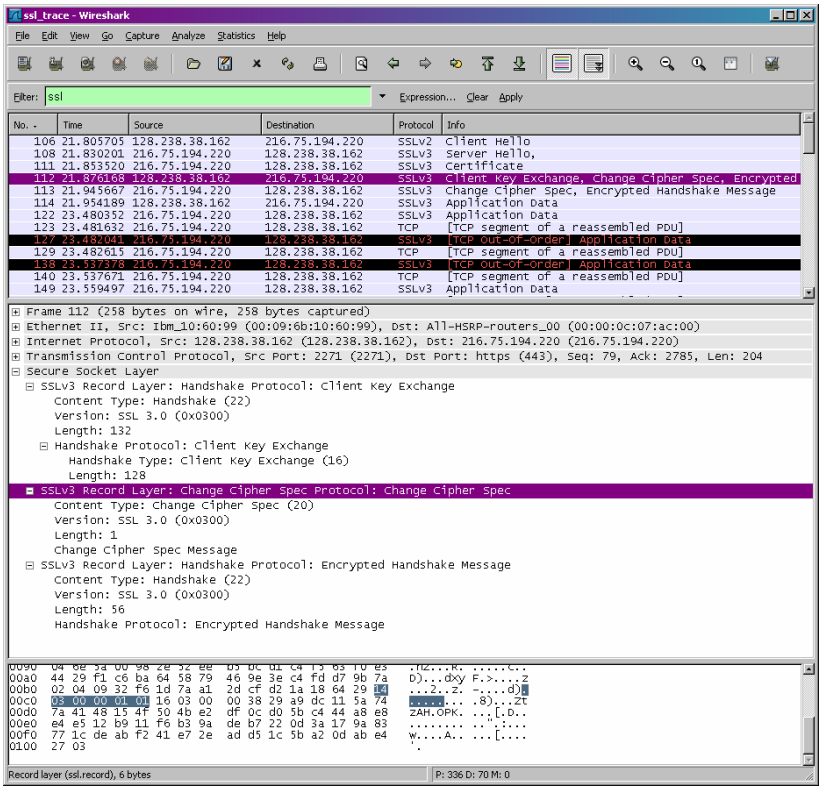
ΑΠΑΝΤΗΣΕΙΣ

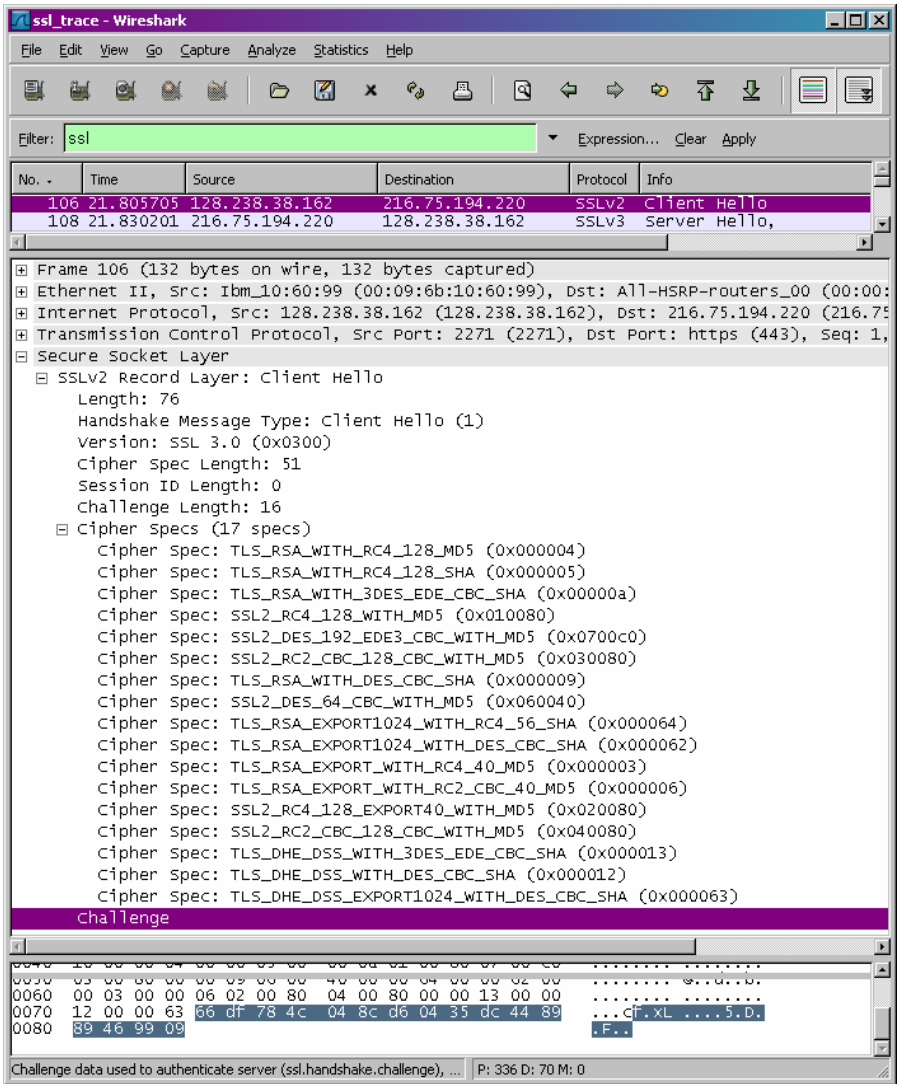
1.

Frame	Source	SSL Count	SSL Type
106	Client	1	Client Hello
108	Server	1	Server Hello
111	Server	2	Certificate Server Hello Done
112	Client	3	Client Key Exchange Change Cipher Spec Encrypted Handshake Message
113	Server	2	Change Cipher Spec Encrypted Handshake Message
114	Client	1	Application Data
122	Server	1	Application Data
127	Server	1	Application Data



2. Τα τρία πρώτα πεδία είναι
Τύπος Περιεχομένου - 1 byte
Είδος (Version) - 2 bytes
Μέγεθος - 2 bytes.


ssl_trace - Wireshark
File Edit View Go Capture Analyze Statistics Help
Filter: ssl
Expression... Clear Apply
No. - Time Source Destination Protocol Info
106 21.805705 128.238.38.162 216.75.194.220 SSLv2 Client Hello
108 21.830201 216.75.194.220 128.238.38.162 SSLv3 Server Hello,
111 21.853520 216.75.194.220 128.238.38.162 SSLv3 Certificate
112 21.876168 128.238.38.162 216.75.194.220 SSLv3 Client Key Exchange, Change Cipher Spec, Encrypted
113 21.945667 216.75.194.220 128.238.38.162 SSLv3 Change Cipher Spec, Encrypted Handshake Message
114 21.954189 128.238.38.162 216.75.194.220 SSLv3 Application Data
122 23.480352 216.75.194.220 128.238.38.162 SSLv3 Application Data
123 23.481632 216.75.194.220 128.238.38.162 TCP [TCP segment of a reassembled PDU]
127 23.482041 216.75.194.220 128.238.38.162 SSLv3 [TCP Out-Of-Order] Application Data
129 23.482615 216.75.194.220 128.238.38.162 TCP [TCP segment of a reassembled PDU]
138 23.537378 216.75.194.220 128.238.38.162 SSLv3 [TCP Out-Of-Order] Application Data
140 23.537671 216.75.194.220 128.238.38.162 TCP [TCP segment of a reassembled PDU]
149 23.559497 216.75.194.220 128.238.38.162 SSLv3 Application Data
Frame 112 (258 bytes on wire, 258 bytes captured)
Ethernet II, Src: Ibm_10:60:99 (00:09:6b:10:60:99), Dst: All-HSRP-routers_00 (00:00:0c:07:ac:00)
Internet Protocol, Src: 128.238.38.162 (128.238.38.162), Dst: 216.75.194.220 (216.75.194.220)
Transmission Control Protocol, Src Port: 2271 (2271), Dst Port: https (443), Seq: 79, Ack: 2785, Len: 204
Secure Socket Layer
SSLv3 Record Layer: Handshake Protocol: Client Key Exchange
Content Type: Handshake (22)
Version: SSL 3.0 (0x0300)
Length: 132
Handshake Protocol: Client Key Exchange
Handshake Type: Client Key Exchange (16)
Length: 128
SSLv3 Record Layer: Change Cipher Spec Protocol: Change Cipher Spec
Content Type: Change Cipher Spec (20)
Version: SSL 3.0 (0x0300)
Length: 1
Change Cipher Spec Message
SSLv3 Record Layer: Handshake Protocol: Encrypted Handshake Message
Content Type: Handshake (22)
Version: SSL 3.0 (0x0300)
Length: 56
Handshake Protocol: Encrypted Handshake Message
00a0 44 29 f1 c6 ba 64 58 79 46 9e 3e c4 fd d7 9b 7a D)...dXy F.>....z
00b0 02 04 09 32 f6 1d 7a a1 2d cf d2 1a 18 64 29 14 ...2..z. -....d)
00c0 03 00 00 01 01 16 03 00 00 38 29 a9 dc 11 5a 74 .....8)...Zt
00d0 7a 41 48 15 4f 50 4b e2 df 0c d0 5b c4 44 a8 e8 zAH.OPK. ...[.D..
00e0 e4 e5 12 b9 11 f6 b3 9a de b7 22 0d 3a 17 9a 83 ....".:...
00f0 77 1c de ab f2 41 e7 2e ad d5 1c 5b a2 0d ab e4 w....A.. ...[....
0100 27 03 '.
Record layer (ssl.record), 6 bytes
P: 336 D: 70 M: 0
SSL Frames

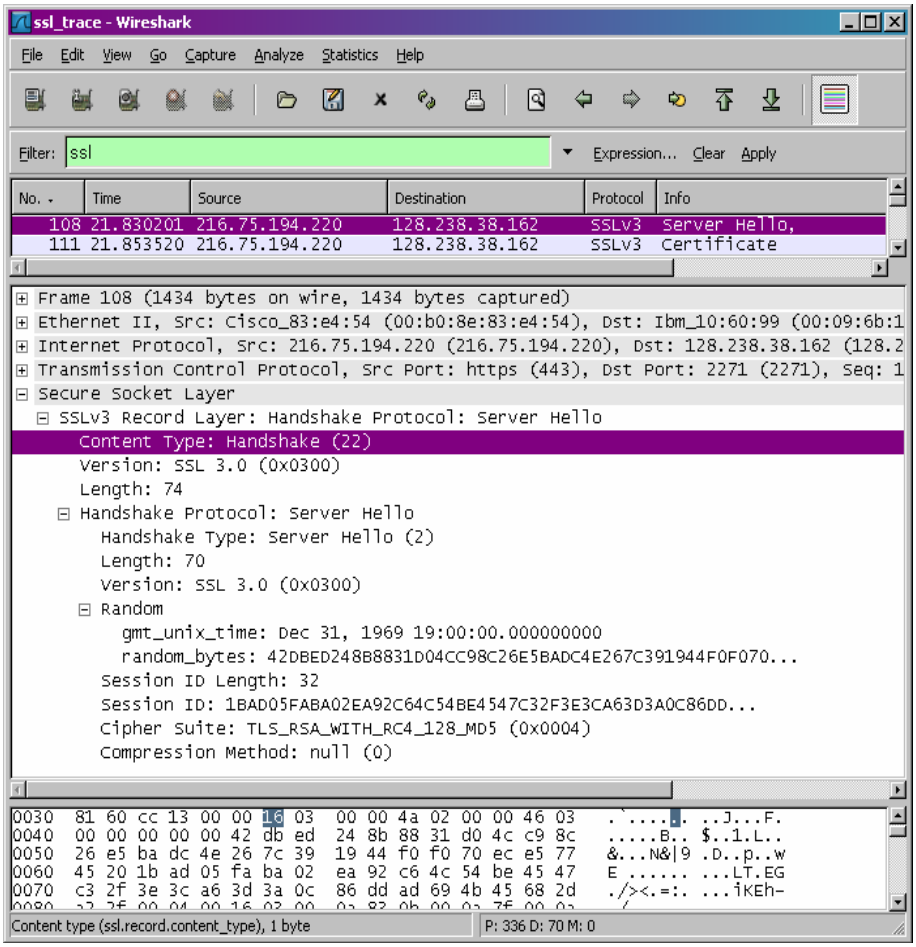


Client Hello record

3. Η τιμή του content type είναι 22

4. Η τιμή του nonce είναι 66df 784c 048c d604 35dc 4489 8946 9909

5. Το πρώτο suite της λίστας χρησιμοποιεί RSA για κοινό-κλειδί αλγόριθμο, RC4 για συμμετρικό-κλειδί και χρησιμοποιεί *τον* MD5 hash αλγόριθμο



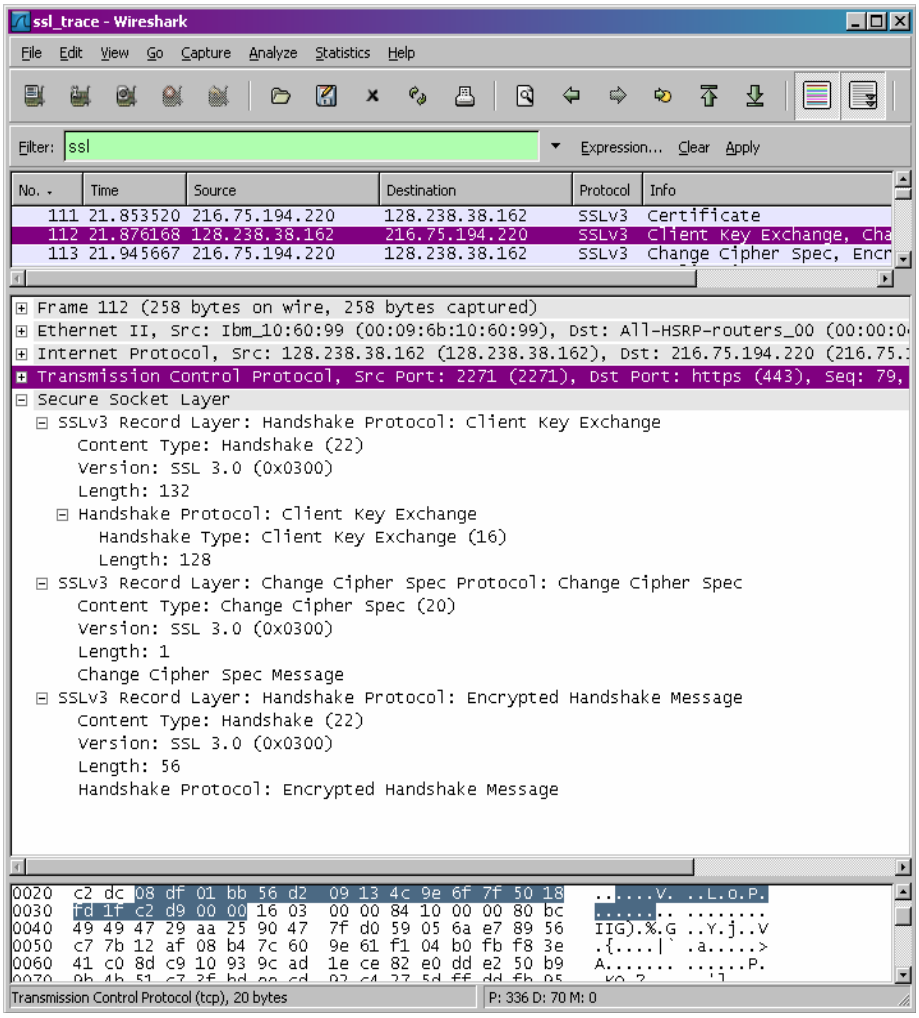
Server Hello record

6. Το cipher suite χρησιμοποιεί RSA για τον αλγόριθμο κοινού κλειδιού, RC4 για συμμετρικό κλειδί και χρησιμοποιεί τον MD5 αλγόριθμο

7. Ναί, στη λίστα Random.Το μέγεθος του είναι 32 bits, 28 για την πληροφορία (data) και 4 για το χρόνο. Ο σκοπός των nonce είναι να προλαμβάνει συνεχόμενες επιθέσεις

8.Ναί, περιέχει. Σκοπός του είναι να παρέχει μια μοναδική αναγνωριστική ταυτότητα για το SSL session. Ο πελάτης μπορεί να επαναλάβει το ίδιο session αργότερα χρησιμοποιώντας το serverprovided session όταν αυτό στέλνει ClientHello.

9.Όχι δεν περιέχει. Το πιστοποιητικό περιέχεται σε ξεχωριστό δεδομένο. Το πιστοποιητικό εντάσσεται σε ένα ξεχωριστό Ethernet frame



10. Ναι, περιέχει. Η ερώτηση αυτή χρησιμοποιείται και από τον client και τον server ώστε να δημιουργηθεί ένας ενιαίος μυστικός κώδικας, που θα χρησιμοποιηθεί ώστε να δημιουργηθεί ένα σετ απαντήσεων για MAC και για κρυπτογράφηση. Η μυστική ερώτηση κρυπτογραφήθηκε χρησιμοποιώντας το δημοσιοποιημένο κλειδί του server, το οποίο ο client εξήγαγε από το πιστοποιητικό που έστειλε ο server. Το μέγεθος της κρυπτογραφημένης μυστικής ερώτησης είναι 128 bytes

11. Ο σκοπός της αλλαγής κρυπτογραφημένου δεδομένου είναι να υποδεικνύει ότι τα περιεχόμενα των επακόλουθων SSL δεδομένων που στάλθηκαν από τον client θα αποκρυπτογραφηθούν. Αυτό το δεδομένο έχει μέγεθος 6 bytes, 5 bytes για την επικεφαλίδα και 1 byte για το μήνυμα

12.Παράγεται ένα MAC από την συνένωση όλων των προηγούμενων handshake μυνημάτων που στάλθηκαν από τον client και στέλνεται στον server

13. Ναι, έστειλε. Το κρυπτογραφημένο handshake record του server είναι διαφορετικό από αυτό που στάλθηκε από τον client, επειδή περιέχει την συνένωση όλων των handshake μυνημάτων που στάλθηκαν από τον server, αντί του client.Κατά τα άλλα, τα μυνήματα είναι τα ίδια με αυτά που στάλθηκαν από τον client.

14. Τα αρχεία εφαρμογής κρυπτογραφήθηκαν χρησιμοποιώντας αλγόριθμο κρυπτογράφησης συμμετρικού κλειδιού στην handshake φάση χρησιμοποιώντας τα

συμμετρικά κλειδιά κρυπτογράφησης που παράχθηκαν από το pre-master key και τα nonces (από τον client και τον server). Το κλειδί κρυπτογράφησης του client χρησιμοποιείται για να κρυπτογραφήσει τα αρχεία που στέλνονται από τον client στον server και το κλειδί κρυπτογράφησης του server χρησιμοποιείται για να κρυπτογραφήσει τα αρχεία που στέλνονται από server στον client. Το μήνυμα περιέχει ένα MAC, παρολαυτά το Wireshark δεν ξεχωρίζει τα κρυπτογραφηνένα αρχεία εφαρμογής από το MAC

15. Το αρχικό ClientHello μήνυμα χρησιμοποιεί SSLv2 (version 2). Παρόλαυτα, όταν ο server απαντά με ένα frame χρησιμοποιώντας SSLv3 (version 3), το επακόλουθο SSL μήνυμα ανταλλαγής είναι σε μορφή version 3.
Υπάρχουν αρκετές φορές όπου η session συνεχίζεται. Η handshake διαδικασία εδώ διαφέρει από την αρχική handshake που φένεται στο σχ.1. Για να επαναληφθεί η session, ο client στέλνει ένα ClientHello μύνημα που περιλαμβάνει το SessionID που πρώτα στάλθηκε από τον server στον client κατά τη διάρκεια της αρχικής SSL handshake. Ο Server δεν χτειάζεται να απαντήσει με ένα certificate,από τη στιγμή που ο client το έχει ήδη. Αντί αυτού, ένα ServerHello μήνυμα που περιέχει ένα νέο nonce στέλνεται,ακολουθόυμενο από Change Cipher Spec και Encrypted Handshake records, από τον server στον client. Ο client τότε απαντά με Change Cipher Spec και Encrypted Handshake records,το αρχείο εφαρμογής στέλνεται.

Κεφάλαιο 14
Συμπεράσματα

Μέσα από την παρούσα πτυχιακή εργασία αναλύσαμε τα βασικά πρωτόκολλα που αφορούν τα TCP/IP δίκτυα, τα παρατηρήσαμε σε δράση και είδαμε πως αλληλοεπιδρούν μεταξύ τους. Αν και το TCP/IP είναι πολύ μεγαλύτερο και θα χρειαζόταν χιλιάδες σελίδες για να περιγράψουν εκτενώς τις λεπτομέρειες και τις παραμέτρους που το απαρτίζουν, κάποιος που έχει κατανοήσει το εισαγωγικό κομμάτι και τις ασκήσεις επίδειξης βρίσκεται σε ένα πολύ καλό επίπεδο γνώσης των TCP/IP δικτύων και με αρκετό διάβασμα και πρακτική εξάσκηση μπορεί να μάθει πολύ περισσότερα.
Το Wireshark είναι πλήρως εναρμονισμένο με το TCP/IP και αποτελεί ιδανικό εργαλείο για αυτή την εξάσκηση. Το εύχρηστο γραφικό περιβάλλον του το καθιστά εύκολο να αναλύσει όλη την κυκλοφορία ενός δικτύου που χρησιμοποιεί ποικίλα πρωτόκολλα.

ΒΙΒΛΙΟΓΡΑΦΙΑ

http://el.wikipedia.org/wiki/Πρωτόκολλο_Μεταφοράς_Υπερκειμένου
(τελευταία επίσκεψη στις 1.3.2013)
http://el.wikipedia.org/wiki/Domain_Name_System
(τελευταία επίσκεψη στις 1.3.2013)
http://el.wikipedia.org/wiki/Transmission_Control_Protocol
(τελευταία επίσκεψη στις 1.3.2013)
http://el.wikipedia.org/wiki/UDP (τελευταία επίσκεψη στις 1.3.2013)
http://el.wikipedia.org/wiki/Internet_Protocol (τελευταία επίσκεψη στις 1.3.2013)
http://el.wikipedia.org/wiki/ICMP (τελευταία επίσκεψη στις 1.3.2013)
http://el.wikipedia.org/wiki/Ethernet (τελευταία επίσκεψη στις 1.3.2013)
http://el.wikipedia.org/wiki/Address_Resolution_Protocol
(τελευταία επίσκεψη στις 1.3.2013)
http://el.wikipedia.org/wiki/DHCP (τελευταία επίσκεψη στις 1.3.2013)
http://el.wikipedia.org/wiki/IEEE_802.11. (τελευταία επίσκεψη στις 1.3.2013)
http://el.wikipedia.org/wiki/SSL (τελευταία επίσκεψη στις 1.3.2013)
http://el.wikipedia.org/wiki/TCP/IP (τελευταία επίσκεψη στις 1.3.2013)
http://el.wikipedia.org/wiki/Διεύθυνση_MAC (τελευταία επίσκεψη στις 1.3.2013)
http://el.wikipedia.org/wiki/Κατάλογος_των_TCP_και_UDP_ports
(τελευταία επίσκεψη στις 1.3.2013)
https://foss.ntua.gr/wiki/ (τελευταία επίσκεψη στις 1.3.2013)
http://www.neural.uom.gr/Documents/Networks/chapter5.pdf
(τελευταία επίσκεψη στις 1.3.2013)
http://conta.uom.gr/conta/ekpaideysh/seminaria/common/networks/osi.htm
(τελευταία επίσκεψη στις 1.3.2013)
http://el.wikipedia.org/wiki/Δίκτυο_υπολογιστών (τελευταία επίσκεψη στις 1.3.2013)
http://el.wikipedia.org/wiki/Μοντέλο_αναφοράς_OSI
(τελευταία επίσκεψη στις 1.3.2013)
http://networking-basics.wikispaces.com/ (τελευταία επίσκεψη στις 1.3.2013)
http://www.diktyas.gr/ (τελευταία επίσκεψη στις 1.3.2013)
http://el.wikipedia.org/wiki/Τοπικό_δίκτυο_υπολογιστών
(τελευταία επίσκεψη στις 1.3.2013)
http://diktia.weebly.com/ (τελευταία επίσκεψη στις 1.3.2013)
http://www.ece.ntua.gr/ (τελευταία επίσκεψη στις 1.3.2013)
http://openmaniak.com/gr/wireshark_stat.php (τελευταία επίσκεψη στις 1.3.2013)
http://www.wireshark.org/ (τελευταία επίσκεψη στις 1.3.2013)
http://el.wikipedia.org/wiki/Wireshark (τελευταία επίσκεψη στις 1.3.2013)
http://www.techrepublic.com/blog/products/review-wireshark-network-analyzer/642
(τελευταία επίσκεψη στις 1.3.2013)

J.F. Kurose, K.W. Ross.Δικτύωση Υπολογιστών - Προσέγγιση απο Πάνω προς τα Κάτω .εκδοσεις Γκιούρδας
Κ.Ξαρχακος,Δ.Καρολιδης.Βασικά Πακέτα Πληροφορικής.Εκδόσεις Άβακας
Φουληράς Παναγιώτης.Δίκτα Υπολογιστών.Μια Πρακτική Προσέγγιση.Εκδόσεις Ζυγός
Τανεμπάουμ Άντριου.Δίκτυα Υπολογιστών .Εκδόσεις Κλειδάριθμος
Χαιντέν Μάτ.Δίκτυα.Θεωρία και Πράξη.Εκδόσεις Γκιούρδας
Χάλμπεργκ Μπρους.Οδηγός Για Τα Δίκτυα.Εκδόσεις Γκιούρδας