Τεχνολογικό Ίδρυμα Κρήτης

Σχολή Τεχνολογικών Εφαρμογών
Τμήμα Εφαρμοσμένης Πληροφορικής και Πολυμέσων

Πτυχιακή Εργασία:

Θέμα:
"Σχεδίαση και ανάπτυξη ενιαίου δικτύου διασύνδεσης εργαστηρίων με υποστήριξη δικτυακών υπηρεσιών "

- **Εισηγητής:** Γαλάτης Παναγιώτης
- **Υλοποίηση:** Πουλτσάκης Μιχαήλ

"*The effort of using machines to mimic the human mind has always struck me as rather silly.*
I would rather use them to mimic something better."

- *Edsger Dijkstra*

Περιεχόμενα

Κεφάλαιο 1ο: Γενικά…

1.1 Εισαγωγή

Σκοπός του εγγράφου αυτού είναι να γίνει αρχικά μία μικρού βάθους εισαγωγή στην ιστορία των δικτύων, στις τεχνολογίες τους καθώς και στο πώς βοηθούν το σημερινό άνθρωπο, έμμεσα ή άμεσα, στις καθημερινές του ανάγκες και δραστηριότητες. Επίσης θα γίνει γενική αναφορά στον τρόπο και φιλοσοφία λειτουργίας τους καθώς επίσης και μία σύντομη περιγραφή στα πρότυπα που έχουν επικρατήσει και στις μελλοντικές εξελίξεις των τελευταίων. Διάφορα θεωρητικά και πρακτικά θέματα θα καλυφθούν με σκοπό την καλύτερη και γρηγορότερη κατανόηση του κυρίως μέρους από τον αναγνώστη.

Ως κύριο θέμα, θα εστιαστούμε κυρίως στην εκτενή περιγραφή και παρουσίαση δικτυακών εφαρμογών και υπηρεσιών σε θεωρητικό αλλά και σε πρακτικό επίπεδο. Θα περιγραφεί το υλοποιημένο τμήμα της εργασίας καθώς η όλη φιλοσοφία του.

Ωστόσο, πρέπει να αναφέρουμε ότι απαιτείται από τον αναγνώστη να γνωρίζει κάποιες βασικές έννοιες, όπως:

- Τί είναι ο υπολογιστής
- Τί είναι το δίκτυο υπολογιστών

Στο πρακτικό κομμάτι της εργασίας έχει υλοποιηθεί ένα υποδίκτυο με:

- ένα σύνολο απλών υπολογιστών
- ενός δρομολογητή (**router**)
- δύο switches
- ενός εξυπηρετητή (**server**) με ιδιότητες δρομολογητή

Το βασικό αντικείμενο μελέτης, σχεδιασμού και υλοποίησης είναι ο ίδιος ο εξυπηρετητής αφού αυτός εκτελεί όλες τις απαραίτητες διεργασίες και υπηρεσίες για τη σωστή και ασφαλή λειτουργία του δικτύου. Πρόκειται για ένα σύστημα με τα χαρακτηριστικά:

CPU	Intel Xeon 2.4 GHz
Motherboard	Intel Xeon Ultra Server
RAM	1GB DDR
SCSI Controller	Adaptec-Intel RAID SRCZCR
Hard Disks	2 IDE 2 SCSI RAID (Mirroring)

Υλοποίηση Server

Η υλοποίηση του server έχει γίνει σε λειτουργικό σύστημα **Linux** και πιο συγκεκριμένα, στη διανομή **Slackware 11**. Η επιλογή του λειτουργικού Linux έγινε επειδή μέσω αυτού μπορούν να ρυθμιστούν όλες οι παράμετροι για την ομαλή λειτουργία του καθώς κι επειδή είναι διαθέσιμος όλος ο πηγαίος κώδικας κάθε διεργασίας. Με τον τρόπο αυτό είναι εμφανής στο χρήστη (και ακόμα περισσότερο στο διαχειριστή) του συστήματος ο τρόπος με τον οποίο εκτελούνται και πραγματοποιούνται οι διάφορες λειτουργίες του συστήματος. Επίσης, είναι δυνατόν μέσω του λειτουργικού αυτού, ο διαχειριστής του συστήματος να το τροποποιήσει σύμφωνα με τις δικές του και μόνο ανάγκες με σκοπό τη βελτιστοποίηση της λειτουργίας του.

Η επιλογή της διανομής Slackware έγινε κατόπιν έντονης μελέτης και έρευνας και σύγκρισης μεταξύ των openSUSE 10, Ubuntu, Fedora Core 5 και Debian. Οι λόγοι που το αποτέλεσμα της σύγκρισης υπέδειξε τη συγκεκριμένη διανομή είναι πολλοί και διάφοροι οι οποίοι είναι δύσκολο να αναφερθούν καθώς δεν είναι αντικείμενο του παρόντος εγγράφου. Ένας σημαντικός λόγος είναι ωστόσο, το ότι η συγκεκριμένη διανομή είναι η πιο λιτή από όλες όσων αφορά τον τομέα των υλοποιημένων υπηρεσιών. Ουσιαστικά είναι το λειτουργικό σύστημα και μόνο χωρίς να διαθέτει οποιαδήποτε εφαρμογή ή εργαλείο. Απαιτεί καλές γνώσεις λειτουργικών συστημάτων σχετικά με τον τρόπο λειτουργίας τους καθώς και αρκετό “ιδρώτα” για να προσαρμοστεί στις ανάγκες ή στις απαιτήσεις του διαχειριστή τους (customize). Ένας ακόμα λόγος ήταν η επιθυμία του υπογράφοντα να διεισδύσει στον κόσμο των λειτουργικών συστημάτων Linux με σκοπό την εκμάθησή κι εξοικείωση του με αυτά.

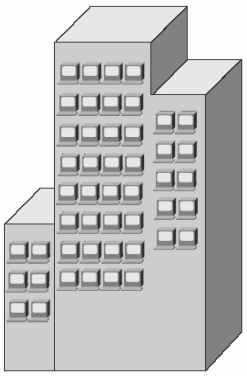
Υλοποίηση Clients

Από την πλευρά των clients το λειτουργικό σύστημα που επιλέχθηκε είναι το **Microsoft Windows XP Professional Edition**. Όλες οι διεργασίες από τη μεριά των clients εκτελούνται και τρέχουν μέσα από το λειτουργικό σύστημα αυτό. Η επιλογή αυτή έγινε γιατί, σύμφωνα πάνα με τα ελληνικά δεδομένα, το πιο διαδεδομένο λειτουργικό σύστημα στην Ελλάδα είναι τα Windows.

Έχει επικρατήσει η άποψη οι περισσότεροι χρήστες να χρησιμοποιούν τη συγκεκριμένη σειρά λειτουργικών και ότι τα Linux απευθύνονται σε ιδιαίτερα έμπειρους χρήστες (power users). Κάτι τέτοιο βέβαια είναι αναληθές. Δε θα υπάρξει περαιτέρω ανάλυση του συγκεκριμένου ζητήματος παρ’ όλο που πρέπει να εξεταστεί και να διερευνηθεί και που άλλωστε, απασχολεί πολύ κόσμο. Ένας επιπλέον λόγος που επιλέχθηκε το συγκεκριμένο λειτουργικό σε συνδυασμό με εκείνο του server είναι ότι θέλουμε να δείξουμε πώς αυτά μπορούν να συνυπάρξουν αρμονικά μέσα σε ένα δίκτυο και να συνεργαστούν (π.χ Samba Server).

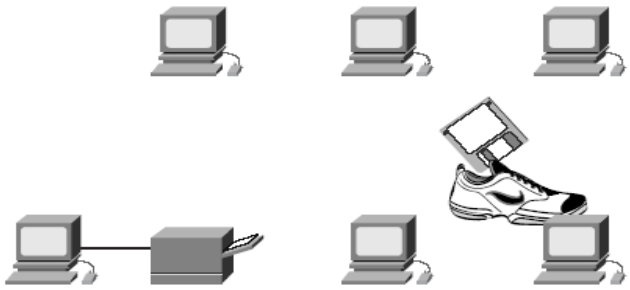
1.2 Ιστορική Αναδρομή

Τα δίκτυα δεδομένων (Data Networks) αναπτύχθηκαν ως αποτέλεσμα των αναγκών επιχειρήσεων καθώς και κυβερνήσεων να ανταλλάσσουν ηλεκτρονικής μορφής πληροφορίες μεταξύ μεγάλων αποστάσεων. Εκείνη την εποχή, οι μικροϋπολογιστές δεν ήταν συνδεδεμένοι μεταξύ τους, σε αντίθεση με τους mainframe τερματικούς υπολογιστές και γι' αυτό το λόγο δεν υπήρχε κανένας τρόπος να μοιράζονται δεδομένα (shared data) μεταξύ πολλών μικροϋπολογιστών. Το Σχήμα 1.1 (CCNA 1 and 2, Third Edition, Cisco Press) απεικονίζει μία εταιρία με μικροϋπολογιστές χωρίς δικτυακή διασύνδεση.



Σχήμα 1.1 Εταιρία με πολλούς μη διασυνδεδεμένους (standalone) υπολογιστές

Τις πρώτες μέρες, μία εταιρεία επένδυε σε υπολογιστές που δεν ήταν συνδεδεμένοι μεταξύ τους οι οποίοι, πολλές φορές είχαν κάποιον εκτυπωτή συνδεδεμένο με αυτούς. Όταν οι υπάλληλοι που δεν είχαν εκτυπωτή χρειαζόντουσαν να εκτυπώσουν αρχεία και έγγραφα, ήταν αναγκασμένοι να αντιγράφουν τα δεδομένα αυτά σε δισκέτες και να τα φορτώσουν κατόπιν, στον υπολογιστή κάποιου συναδέλφου τους που τύχαινε να διαθέτει εκτυπωτή και να πραγματοποιήσουν την εκτύπωση από εκεί. Αυτή η κάπως "σκληρή" έκδοση δικτύου έγινε αργότερα γνωστή ως ***sneakernet****, αφού απαιτούσε τη διαρκή μετακίνηση υπαλλήλων μέσα στους χώρους της επιχείρησης. Το Σχήμα 1.2 (CCNA 1 and 2, Third Edition, Cisco Press) αναπαριστά με έναν εύθυμο τρόπο τη μορφή ενός sneakernet.*



Σχήμα 1.2 SneakerNet

Έγινε έτσι σαφές, ότι η μετακίνηση και το μοίρασμα δεδομένων μέσω δισκετών δεν ήταν αποτελεσματικός τρόπος εργασίας και δημιουργούσε έλλειψη ευλυγισίας στην εταιρεία. Κάθε φορά που ένα αρχείο τροποποιούταν και άλλαζε, έπρεπε να μοιραστεί ξανά σε κάθε υπάλληλο που το χρειαζόταν και το χρησιμοποιούσε. Κάτι τέτοιο δημιουργούσε προβλήματα στη λειτουργία των επιχειρήσεων κι έτσι, κατά συνέπεια κρίθηκε αναγκαία η δημιουργία δικτύων υπολογιστών αφού με τη βοήθεια αυτών η παραγωγικότητα θα αυξανόταν παράλληλα με την εξοικονόμηση χρημάτων.

Έτσι λοιπόν, όλες οι επιχειρήσεις αποφάσισαν να συμπεριλάβουν τα δίκτυα δεδομένων στο δυναμικό τους, με αποτέλεσμα αυτά να εξαπλωθούν και να αναπτυχθούν ραγδαία αφού διαρκώς, νέες τεχνολογίες αναπτύσσονταν πάνω σε αυτά.

Στα μισά της δεκαετίας του ’80, κάθε εταιρεία που παρήγαγε τεχνολογίες και υλικό δικτύων, χρησιμοποιούσε τα δικά της πρότυπα και standards, για λόγους ανταγωνισμού. Έτσι πολλές τεχνολογίες δικτύων ήταν ασύμβατες μεταξύ τους και, κατά συνέπεια, έγινε πολύ δύσκολο δίκτυα διαφορετικών τεχνολογιών να επικοινωνήσουν μεταξύ τους.

Λαμβάνοντας όλα αυτά υπόψη, κρίθηκε απαραίτητο να δημιουργηθούν και να οριστούν πρότυπα στον τομέα των δικτύων με σκοπό την αντιμετώπιση των παραπάνω προβλημάτων. Διάφορες τεχνολογίες συστήθηκαν στο κοινό με σκοπό την ευρεία χρήση και υιοθέτησή τους. Με τον τρόπο αυτό οι εταιρείες παραγωγής δικτυακού υλικού αναγκάστηκαν να ακολουθήσουν και να συμμορφωθούν με τα νέα αυτά πρότυπα (FDDI, Token Ring, Ethernet κ.τ.λ)

Ένας τρόπος περιγραφής και προτυποποίησης της λειτουργίας και της τεχνολογίας των δικτύων, ήταν η περιγραφή τους με τη μορφή επιπέδων. Είναι ένας τρόπος καλύτερης και ευκολότερης κατανόησης για τον τρόπο που λειτουργούν τα δίκτυα υπολογιστών. Έτσι, συστήθηκε στο κοινό το μοντέλο αναφοράς OSI ή αλλιώς, το Μοντέλο των Επτά Επιπέδων:

1.3 OSI

Το **μοντέλο OSI** διαιρεί τις λειτουργίες ενός πρωτοκόλλου σε μια σειρά από επίπεδα. Κάθε επίπεδο χρησιμοποιεί μόνο τις λειτουργίες του κάτω επιπέδου και προσφέρει λειτουργικότητα στο πάνω επίπεδο. Ένα σύστημα που παρουσιάζει συμπεριφορά πρωτοκόλλου και που είναι διαστρωματωμένο σε επίπεδα ονομάζεται **στοίβα πρωτοκόλλων** ή απλά στοίβα. Οι στοίβες κατασκευάζονται με υλικό είτε με λογισμικό. Τυπικά, τα κατώτερα επίπεδα κατασκευάζονται με υλικό, ενώ τα ανώτερα επίπεδα είναι εφαρμογές λογισμικού.

Το μοντέλο OSI είναι βασικά συνδεδεμένο με τον κλάδο των υπολογιστών και την δικτύωσή τους. Το κύριο χαρακτηριστικό του είναι η διεπαφή μεταξύ των επιπέδων, η οποία υπαγορεύει τις προδιαγραφές της αλληλεπίδρασης αυτών των επιπέδων. Αυτό σημαίνει ότι ένα επίπεδο δημιουργημένο από έναν κατασκευαστή μπορεί να συνεργαστεί με το διπλανό επίπεδο που έχει κατασκευάσει άλλος (με την προϋπόθεση ότι έχει γίνει αντιληπτή η προδιαγραφή σωστά).

Συνήθως, η κατασκευή ενός πρωτοκόλλου έχει διαστρωμάτωση σε επίπεδα, όπως και η σχεδίασή του. Μπορεί όμως να κατασκευαστεί έτσι η πιο συχνή

συναλλαγή (ή οι περισσότερο συχνές συναλλαγές) του συστήματος ώστε να γίνεται άμεσα από μια συσκευή που αποτελείται από πολλά συγχωνευμένα επίπεδα.

Αυτός ο λογικός διαχωρισμός των επιπέδων διευκολύνει πολύ την μελέτη της συμπεριφοράς των πρωτοκόλλων, και επιτρέπει να σχεδιάζουμε πολύπλοκες αλλά και πολύ αξιόπιστες στοίβες πρωτοκόλλων. Κάθε επίπεδο προσφέρει υπηρεσίες στο ανώτερό του και ζητά στοιχεία από το κατώτερό του.

Το μοντέλο OSI είναι μια ιεραρχική δομή επτά επιπέδων που καθορίζει τις απαιτήσεις για επικοινωνία δύο υπολογιστών μεταξύ τους και καθορίστηκε ως πρότυπο ISO 7498-1. Θεωρήθηκε ότι θα επέτρεπε την διαλειτουργικότητα μεταξύ διαφόρων συσκευών που προσέφεραν στην αγορά οι διάφοροι κατασκευαστές. Το μοντέλο επιτρέπει σε όλα τα στοιχεία ενός δικτύου να συλλειτουργούν ανεξάρτητα από το ποιος είναι ο κατασκευαστής τους. Περί τα τέλη της δεκαετίας 1980 ο ISO συνιστούσε την εφαρμογή του μοντέλου OSI ως δικτυακού προτύπου.

Τα επίπεδα του μοντέλου OSI φαίνονται παρακάτω:

- Application layer (Επίπεδο Εφαρμογής)
- Presentation layer (Επίπεδο Παρουσίασης)
- Session layer (Επίπεδο Συνόδου)
- Transport layer (Επίπεδο Μεταφοράς)
- Network layer (Επίπεδο Δικτύου)
- Data link layer (Επίπεδο Ζεύξης Δεδομένων)
- Physical layer (Φυσικό Επίπεδο)

Μόνο ένα υποσύνολο του μοντέλου OSI χρησιμοποιείται σήμερα. Η γενική αντίληψη είναι ότι οι περισσότερες προδιαγραφές του είναι περίπλοκες και η πλήρης λειτουργικότητά του θα χρειαζόταν μεγάλο χρόνο κατασκευής, αν και υπάρχουν πολλοί άνθρωποι που υποστηρίζουν σθεναρά το μοντέλο OSI.

Στην παρούσα εργασία και υλοποίηση θα ασχοληθούμε με την τεχνολογία Ethernet 100BASE-T και με τη σουίτα πρωτοκόλλων επικοινωνίας TCP/IP. Ακολουθεί μία σύντομη περιγραφή των τεχνολογιών αυτών. Δε θα ασχοληθούμε περισσότερα με το συγκεκριμένο αντικείμενο αφού υπάρχουν πιο συγκεκριμένα θέματα να καλύψουμε.

1.4 Ethernet

Το πρωτόκολλο **Ethernet** παρουσιάστηκε πρώτη φορά το 1973 στα εργαστήρια Xerox PARC, από τους Robert Metcalfe και David Boggs. Αποτελεί την πλέον διαδεδομένη μέθοδο υλοποίησης τοπικών δικτύων (Local Area Network, LAN) με τοπολογία αστέρα (star) ή αρτηρίας - διαύλου (bus). Οι πρώτες προδιαγραφές του Ethernet υποστήριζαν ταχύτητα μεταφοράς δεδομένων 2,94Mbps. Σήμερα, υποστηρίζονται οι ταχύτητες 10Mbps (10Base-T), 100Mbps (100Base-T ή Fast Ethernet) και 1.000Mbps (1Gbps, Gigabit Ethernet). Εξάλλου, πολλές κάρτες δικτύου Ethernet ονομάζονται και «10/100», διότι υποστηρίζουν ταχύτητες τόσο 10Mbps όσο και 100Mpbs. Έτσι, μπορούν να χρησιμοποιούνται εδώ και τώρα σε ένα LAN των 10Mbps, το οποίο αύριο θα αναβαθμιστεί σε LAN των 100Mbps. Το Ethernet επιτρέπει τη μετάδοση πακέτων δεδομένων (packets) μεταβλητού μεγέθους από 72 έως και 1.518Byte με χρήση της τεχνολογίας CSMA/CD (Carrier Sense Multiple Access with Collision Detection). Η τεχνολογία CSMA/CD δίνει τη δυνατότητα να προβλέπονται, να εντοπίζονται και να αποφεύγονται τυχών συγκρούσεις πακέτων, με αποτέλεσμα να βελτιώνεται κατά πολύ η απόδοση της λειτουργίας του δικτύου. Στο πρωτόκολλο Ethernet κάθε πακέτο περιέχει μια κεφαλίδα (header) στην οποία περιλαμβάνονται πληροφορίες όπως η διεύθυνση του μηχανήματος-αποστολέα, καθώς και αυτή του παραλήπτη.

Οι κάρτες δικτύου Ethernet αποτελούν ένα από τα πλέον οικονομικά μέσα δικτύωσης. Ταυτόχρονα προσφέρουν ικανοποιητικότατες ταχύτητες, από 10 έως και 1.000Mbps (υψηλότερο κόστος). Για το δίκτυο στο σπίτι ή στο γραφείο αρκούν οι κάρτες των 10Mbps, εκτός και αν προβλέπεται η συχνή μεταφορά μεγάλων αρχείων ή στο δίκτυο συμμετέχουν αρκετοί υπολογιστές, οπότε σε περιπτώσεις όπως αυτές η χρήση καρτών 10/100Mbps ή 100Mbps έχει περισσότερο νόημα. Οι δε κάρτες των 1.000Mbps βρίσκουν το ρόλο τους σε δίκτυα με υψηλές απαιτήσεις από πλευράς διακινούμενου όγκου δεδομένων, κάτι που πιθανότατα δεν ισχύει σε ένα γραφείο, πόσο μάλλον στο σπίτι. Μοναδικό μειονέκτημα των δικτύων Ethernet αποτελεί η καλωδίωση. Εάν, π.χ., στο σπίτι οι υπολογιστές βρίσκονται σε διαφορετικά δωμάτια (ή ορόφους), τότε για να συνδεθούν θα πρέπει να «τρέχουν» καλώδια από τον έναν στον άλλο. Το ίδιο πρόβλημα μπορεί να συναντάται και στο περιβάλλον ενός γραφείου, εκτός και αν προσφέρεται δομημένη καλωδίωση.

Παραπάνω είπαμε ότι το Ethernet είναι το βασικό πρωτόκολλο που χρησιμοποιείται σήμερα. Εκτός από αυτό όμως υπάρχουν και άλλα, όπως το Token Ring, το πιο παλιό ARCnet, το FDDI (Fiber Distributed Data Interface) κ.τ.λ.

Η δημοτικότητα του Ethernet ξεκίνησε όταν συστήθηκε στο κοινό το 10BASE5 ομοαξονικό καλώδιο (Thicknet). Αργότερα εμφανίστηκε το 10BASE2 ή αλλιώς γνωστό ως Thinnet. Το δεύτερο είχε το πλεονέκτημα να είναι λεπτότερο με αποτέλεσμα η εγκατάσταση του να είναι ευκολότερη αλλά το μέγιστο μήκος του έφανε τα μόλις 180 μέτρα σε αντίθεση με το Thicknet που έφτανε ως τα 800. Το κριτήριο που επικράτησε ωστόσο ήταν η ευκολία στην εγκατάσταση, και κατά συνέπεια μικρότερο κόστος, με αποτέλεσμα να μην αργήσει να εμφανιστεί το 10BASE-T με τύπο καλωδίου UTP (Unshielded Twisted Pair). Ωστόσο, το γεγονός ότι το μήκος του δεν μπορεί να ξεπεράσει τα 100 μέτρα έκρινε απαραίτητη η προσθήκη και ενσωμάτωση αναμεταδοτών (repeaters) και, αργότερα, πολλαπλών αναμεταδοτών (Hubs). Η ευελιξία, η ευκολία εγκατάστασης και το χαμηλό κόστος των δικτύων που βασίζονταν στο 10BASE-T είχαν ως αποτέλεσμα τη ραγδαία αύξηση των χρηστών τους, των χρηστών του Internet καθώς και την αύξηση της

πολυπλοκότητας των εφαρμογών τους. Η ανάγκη για υψηλότερο εύρος ζώνης (Bandwidth = πληροφορία / χρόνος) οδήγησε στο λεγόμενο Fast Ethernet ή αλλιώς 100BASE-T που βασιζόταν στο πρότυπο IEEE 802,3 και αναπτύχθηκαν τεχνικές για τη συμβατότητα αυτού με το 10BASE-T. Έτσι, με το ρυθμό αυτό, νέες τεχνολογίες εισήχθησαν όπως το Gigabit Ethernet (1000BASE-T), που υποστηρίζει εύρος ζώνης της τάξης το ενός Gigabit ανά δευτερόλεπτο (1Gbps), καθώς και τα δίκτυα οπτικών ινών και πολλές ακόμα άλλες.

Όλες οι παραπάνω ονομασίες των τύπων συνδέσεων, όπως για παράδειγμα ο τύπος 10BASE-T, αποτελούνται από τρία τμήματα που το κάθε ένα έχει την ερμηνεία του. Ο πρώτος αριθμός (στην προκειμένη περίπτωση το 10) δηλώνει εύρος ζώνης, δηλαδή τι ταχύτητα μετάδοσης μπορεί να επιτευχθεί με τη χρήση της συγκεκριμένης σύνδεσης. Η λέξη που ακολουθεί δηλώνει το είδος της συχνότητας μετάδοσης που στην προκειμένη περίπτωση, η λέξη BASE υποδηλώνει baseband, δηλαδή βασική ζώνη. Το γράμμα που ακολουθεί δηλώνει τον τύπο φυσικής σύνδεσης και καλωδίωσης που στην προκειμένη περίπτωση είναι το UTP. Στην περίπτωση του 100BASE-FX τα αρχικά FX υποδηλώνουν σύνδεση μέσω πολύτροπων οπτικών ινών.

1.5 TCP/IP

Το **TCP/IP** ή και **Σουίτα Πρωτοκόλλων Διαδικτύου** (Internet protocol suite) είναι μια συλλογή πρωτοκόλλων επικοινωνίας στα οποία βασίζεται το Internet αλλά και το μεγαλύτερο 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, όπως και πολλές άλλες άλλωστε, είναι οργανωμένη σε στρώματα ή επίπεδα (layers). Το καθένα τους απαντά σε συγκεκριμένα προβλήματα μεταφοράς δεδομένων και παρέχει μια καθορισμένη υπηρεσία στα υψηλότερα επίπεδα. Τα ανώτερα επίπεδα είναι πιο κοντά στη λογική του χρήστη και εξετάζουν πιο αφηρημένα δεδομένα, στηριζόμενα σε πρωτόκολλα χαμηλότερων επιπέδων για να μεταφράσουν δεδομένα σε μορφές που μπορούν να διαβιβαστούν με φυσικά μέσα, όπως ακριβώς περιγράφηκε το μοντέλο OSI.

Τα πρωτόκολλα Διαδικτύου κάνουν χρήση της ενθυλάκωσης (encapsulation) για να παρέχουν αφηρημένα πρωτόκολλα και υπηρεσίες. Ένα πρωτόκολλο υψηλών επιπέδων χρησιμοποιεί τα πρωτόκολλα των κατωτέρων για να ολοκληρώσει τους στόχους του.

Έτσι λοιπόν, κατά αντιστοιχία με το Μοντέλο των Επτά Επιπέδων, το TCP/IP αποτελείται από και αυτό από επίπεδα. Αυτά είναι τα:

- Application Layer (Επίπεδο Εφαρμογής)
- Transport Layer (Επίπεδο Μεταφοράς)
- Internet Layer (Επίπεδο Διαδικτύου)
- Network Access (Επίπεδο Πρόσβασης Δικτύου)

Επίπεδο Εφαρμογής

Το επίπεδο εφαρμογής χρησιμοποιείται από την πλειοψηφία των δικτυακών και δικτυωμένων προγραμμάτων. Το πρόγραμμα παραδίδει τα δεδομένα σε μια μορφή που ορίζει τα ίδιο. Εφ' όσον το TCP/IP δεν παρέχει επίπεδα μεταξύ των στρωμάτων εφαρμογής και μεταφοράς, όλες οι λειτουργίες παρουσίασης και συνόδου πρέπει να υλοποιηθούν σ' αυτό το επίπεδο. Αυτή η διαδικασία διευκολύνεται με την χρήση βιβλιοθηκών.

Επίπεδο Μεταφοράς

Το επίπεδο μεταφοράς είναι υπεύθυνο για την μεταφορά μηνυμάτων, ανεξαρτήτως του υποκείμενου δικτύου, με έλεγχο σφαλμάτων (error control), κατάτμηση (fragmentation) και ρύθμιση ροής (flow control). Η μετάδοση μηνυμάτων μεταξύ δυο οντοτήτων μπορεί να κατηγοριοποιηθεί ως εξής:

- Connection Oriented π.χ (TCP)
- Connectionless π.χ (UDP)

Η λειτουργία του επιπέδου αυτού μπορεί να συγκριθεί με αυτή οποιουδήποτε μηχανισμού / μέσου μεταφοράς, όπως για παράδειγμα την περίπτωση ενός οχήματος που πρέπει να εξασφαλίζει την πλήρη και ασφαλή διακίνηση του φορτίου του. Το επίπεδο μεταφοράς παρέχει αυτή την υπηρεσία σύνδεσης εφαρμογών μεταξύ τους, κάνοντας χρήση θυρών (ports). Καθώς το IP προσφέρει μόνο παράδοση όσο το δυνατόν καλύτερα (best effort delivery), το επίπεδο μεταφοράς είναι το πρώτο επίπεδο όπου λαμβάνεται υπόψη το θέμα της αξιοπιστίας.

Παραδείγματος χάρη, σε μια προσπάθεια αξιόπιστης μετακίνησης δεδομένων, το TCP που είναι ένα Connection oriented πρωτόκολλο, έχει τα ακόλουθα χαρακτηριστικά:

- τα δεδομένα έρχονται στην ίδια σειρά με την οποία στάλθηκαν
- ελάχιστος έλεγχος σφαλμάτων
- ανεπιθύμητα αντίγραφα απορρίπτονται
- χαμένα πακέτα ξαναστέλνονται
- έλεγχος κυκλοφοριακής συμφόρησης (congestion control)

Τα πρωτόκολλα δυναμικής δρομολόγησης (dynamic routing), που κανονικά θα έπρεπε να βρίσκονται σε αυτό το επίπεδο του TCP/IP (αφού λειτουργούν πάνω από το IP) αντιμετωπίζονται συχνά ως τμήματα του επίπεδου δικτύου (π.χ. το OSPF).

Το νέο SCTP είναι επίσης ένας "αξιόπιστος", Connection oriented μηχανισμός μεταφοράς. Είναι stream oriented, όχι byte oriented όπως το TCP, και προσφέρει την δυνατότητα multiplexing πολλών ρευμάτων (stream) σε μια μόνο σύνδεση. Προτείνει υποστήριξη multi-homing, την δυνατότητα δηλαδή για μια οντότητα να μπορέσει, στα πλαίσια μιας συγκεκριμένης σύνδεσης, να κάνει χρήση πολλαπλών (εφ' όσον υπάρχουν) διευθύνσεων IP, που αντιπροσωπεύουν

πολλαπλές interfaces (διασυνδετικές διατάξεις), έτσι ώστε αν κάποια παρουσιάσει βλάβη, να μη χαθεί η σύνδεση.

Το UDP είναι ένα connectionless πρωτόκολλο διαγραμμάτων δεδομένων (datagrams). Όπως και το IP, είναι ένα best effort αλλά και "αναξιόπιστο" πρωτόκολλο: ο έλεγχος σφαλμάτων είναι αδύναμος (απλό checksum). Χρησιμοποιείται κυρίως σε εφαρμογές streaming μέσων (ήχος, βίντεο, κλπ.) όπου η έγκαιρη άφιξη των δεδομένων είναι πιο σημαντική από την ακεραιότητα τους. Ο χρόνος που κερδίζεται σε σχέση με τα Connection oriented πρωτόκολλα, που πρέπει να καθιερώσουν μια αξιόπιστη σύνδεση, το καθιστά ιδανικό για απλές ερώτηση / απάντηση εφαρμογές (π.χ. DNS).

Το TCP και το UDP εκμεταλλεύονται από εφαρμογές που διακρίνονται (στο επίπεδο του δικτύου) από την θύρα TCP ή UDP τους. Ορισμένοι αριθμοί θυρών είναι κλειστοί και αναφέρονται σε πολύ συγκεκριμένες εφαρμογές

Το RTP είναι ένα πρωτόκολλο διαγραμμάτων δεδομένων σχεδιασμένο για στοιχεία πραγματικού χρόνου (real-time) όπως τα streaming audio και video. Αν και παρουσιάζεται στο επίπεδο μεταφοράς (αντί για το επίπεδο συνόδου), βασίζεται στο UDP για την λειτουργία του.

Επίπεδο Διαδικτύου

Ο σκοπός του επιπέδου διαδικτύου είχε αρχικά καθοριστεί ως η μεταφορά πακέτων μέσω ενός ενιαίου δικτύου.

Με την εμφάνιση πιο σύνθετων μορφών δικτύων, προστέθηκαν επιπλέον χαρακτηριστικά στο επίπεδο αυτό, έτσι ώστε ο ρόλος του να είναι πια η διακίνηση δεδομένων από το δίκτυο πηγή στο δίκτυο προορισμού. Αυτό προϋποθέτει συνήθως την δρομολόγηση πακέτων διαμέσου ενός δικτύου δικτύων (internetwork) ή διαδικτύου (με μικρά γράμματα).

Στην σουίτα πρωτοκόλλων Διαδικτύου, το IP μεταφέρει τα πακέτα δεδομένων από την πηγή, στον προορισμό. Το IP μπορεί να εξυπηρετήσει διάφορα πρωτόκολλα ανωτέρων επιπέδων (upper layer protocols) το καθένα εκ των οποίων προσδιορίζεται με έναν αποκλειστικό αριθμό πρωτοκόλλου: π.χ. το ICMP και το IGMP έχουν τους αριθμούς 1 και 2 αντίστοιχα.

Μερικά πρωτόκολλα που στηρίζονται στο IP, π.χ. το ICMP (χρησιμοποιείται για την διάδοση διαγνωστικών πληροφοριών σχετικά με την μεταφορά πακέτων μέσω IP) παρουσιάζονται πάνω από το IP αλλά παρέχουν υπηρεσίες επιπέδου διαδικτύου, απεικονίζοντας έτσι την ασυμβατότητα μεταξύ του Διαδικτύου, των πρωτοκόλλων Διαδικτύου και του μοντέλου OSI. Όλα τα πρωτόκολλα δρομολόγησης ανήκουν επίσης στο επίπεδο διαδικτύου, αν και θα μπορούσαν να τοποθετηθούν σε ανώτερα επίπεδα.

Επίπεδο Πρόσβασης Δικτύου

Το επίπεδο αυτό, ρόλος του οποίου είναι η διακίνηση πακέτων του επιπέδου δικτύου μεταξύ δυο οντοτήτων, δεν είναι στην ακρίβεια μέρος της σουίτας πρωτοκόλλων Διαδικτύου, διότι το IP λειτουργεί με διάφορα επίπεδα συνδέσμου. Η διαδικασία διαβίβασης πακέτων σε ένα συγκεκριμένο επίπεδο συνδέσμου μπορεί

να ελέγχεται είτε από τον οδηγό της διασυνδετικής διάταξης (interface), είτε το firmare ή σύνολο εξειδικευμένων κυκλωμάτων (chipsets), είτε τέλος από ένα συνδυασμό τον προ-αναφερθέντων. Αυτά θα εκτελέσουν τις λειτουργίες ζεύξης δεδομένων (data link), όπως π.χ. την πρόσθεση επικεφαλίδας (packet header) πριν την αποστολή, την ίδια τη διαβίβαση του πλαισίου (frame) με τη χρήση ενός φυσικού μέσου.

Το Επίπεδο Πρόσβασης Δικτύου είναι επίσης το επίπεδο όπου τα πακέτα μπορούν να αναχαιτιστούν για να σταλθούν σ' ένα ιδεατό ιδιωτικό δίκτυο (Virtual Private Network, VPN). Σ' αυτήν την περίπτωση, τα δεδομένα του επιπέδου αυτού αντιμετωπίζονται ως δεδομένα εφαρμογής, και "ξανακατεβαίνουν" την στοίβα πρωτοκόλλων Διαδικτύου για να σταλούν. Στη λαμβάνουσα πλευρά, τα δεδομένα ανεβαίνουν δυο φορές την στοίβα (μια για το VPN και μια δεύτερη για τη δρομολόγηση).

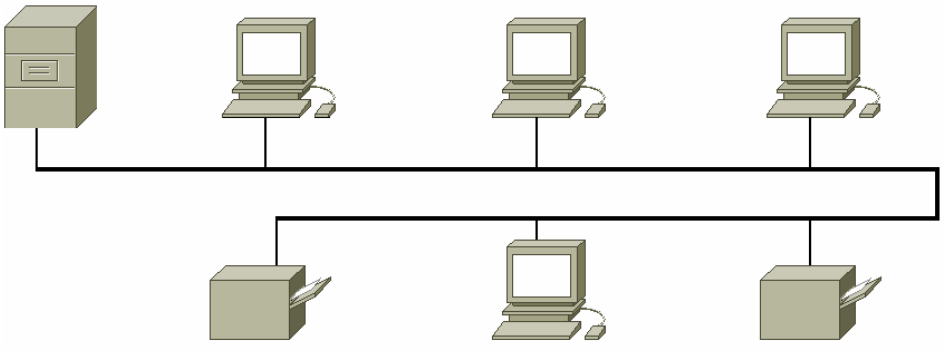
Το φυσικό επίπεδο του μοντέλου OSI, που αποτελείται από τα φυσικά στοιχεία του δικτύου (π.χ. hubs, repeaters, καλώδια δικτύου, οπτικές ίνες, ομοαξονικά καλώδια, κάρτες δικτύων) και τις προδιαγραφές χαμηλού επιπέδου των σημάτων (τάση, συχνότητα, κλπ.), θεωρείται συχνά ως μέρος του Επιπέδου Πρόσβασης Δικτύου.

1.6 Τοπολογίες Δικτύων

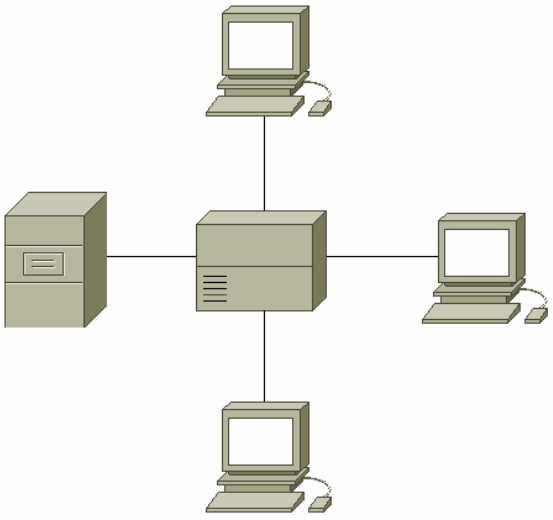
Μία τοπολογία δικτύου ορίζει τον τρόπο με τον οποίο υπολογιστές, εκτυπωτές, οδηγοί δικτύου, δικτυακές συσκευές και άλλα στοιχεία συνδέονται μεταξύ τους. Με άλλα λόγια, μία τοπολογία δικτύου περιγράφει τον τρόπο κατανομής των στοιχείων που τα απαρτίζουν. Η τοπολογία επηρεάζει και καθορίζει την ποιότητα λειτουργίας του δικτύου. Συγκεκριμένα, οι κατηγορίες τοπολογιών είναι οι εξής:

- Τοπολογία Αστέρα (Star)
- Τοπολογία Διαύλου (Bus)
- Τοπολογία Δακτυλίου (Ring)
- Τοπολογία Ιεραρχίας (Hierarchical)
- Τοπολογία Εκτεταμένου Αστέρα (Extended Star)
- Τοπολογία Διχτυού (Mesh)

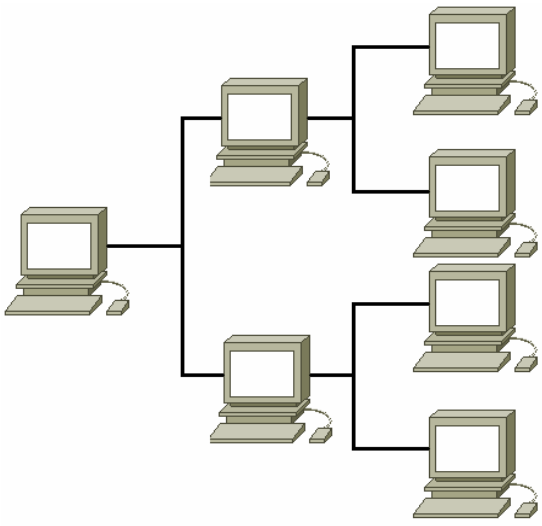
Στην επόμενη σελίδα απεικονίζονται σχηματικά μερικές από τις παραπάνω τοπολογίες



Σχήμα 1.3 Τοπολογία Διαύλου



Σχήμα 1.4 Τοπολογία Αστέρα



Σχήμα 1.5 Τοπολογία Ιεραρχίας

Κεφάλαιο 2ο: Υλικό

2.1 Κάρτα Δικτύου (Network Interface Card - NIC)

Οι κάρτες δικτύου γνωστές και με την συντομογραφία **NIC** (Network Interface Cards) είναι από τα πιο βασικά και συνηθισμένα εξαρτήματα ενός δικτύου και όλοι σχεδόν έχουμε δει και μπορούμε να τις αναγνωρίσουμε. Συνήθως πρόκειται για εξαρτήματα που τοποθετούνται με μορφή κάρτας στο εσωτερικό ενός υπολογιστή και επιτρέπουν τη σύνδεση του στο καλωδιακό μέσο του δικτύου.

Για να εξασφαλίσουμε ότι κάθε κάρτα δικτύου μπορεί να αναγνωριστεί σαν μοναδική από το δίκτυο, έχει από κατασκευή της ένα εσωτερικό αριθμό διεύθυνσης, το λεγόμενο MAC Address. Πρόκειται για ένα αριθμό διαφορετικό για κάθε κάρτα δικτύου, ο οποίος παρέχεται από τον κατασκευαστή. Αποτελείται από 6 αριθμούς του δεκαεξαδικού συστήματος και είναι της μορφής xx-xx-xx-xx-xx-xx. Η κατανομή των διευθύνσεων αυτών μεταξύ των κατασκευαστών – που αναφέρονται στο φυσικό επίπεδο της κάρτας – γίνεται από την επιτροπή IEEE (Institute of Electrical and Electronics Engineers, Inc.).

Οι κάρτες δικτύου εκτελούν πολλές από τις λειτουργίες της δικτυακής επικοινωνίας που περιλαμβάνονται στη συμφωνία των σταθμών που επικοινωνούν. Μεταξύ άλλων περιλαμβάνουν, ρύθμιση παραμέτρων επικοινωνίας όπως ρυθμός μετάδοσης, μέγεθος πακέτου, λήξη χρόνου κ.τ.λ.

Οι κάρτες δικτύου έχουν συγκεκριμένα ηλεκτρικά και μηχανικά χαρακτηριστικά:

- **Ηλεκτρικά Χαρακτηριστικά** είναι αυτά που καθορίζουν τον τρόπο λειτουργίας της κάρτας ως ηλεκτρονικό κύκλωμα και ρυθμίζουν τους τρόπους μετάδοσης των δυαδικών ψηφίων, το χρονισμό, τα σήματα ελέγχου κ.τ.λ.

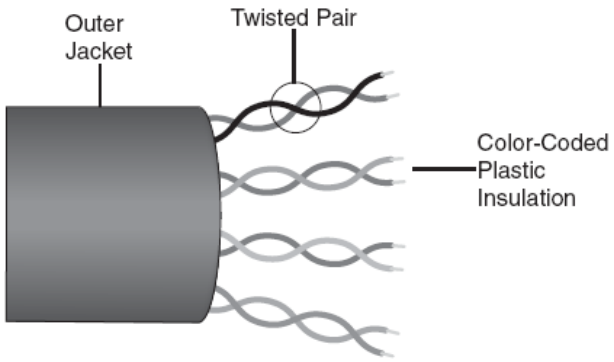
- **Μηχανικά Χαρακτηριστικά** είναι αυτά που καθορίζουν τους τρόπους της φυσικής σύνδεσης της κάρτας με το μέσο μετάδοσης. Για παράδειγμα, το είδος της υποδοχής (connector) που διαθέτει η κάρτα και το οποίο καθορίζει και το είδος του καλωδίου που μπορούμε να συνδέσουμε (π.χ. ομοαξονικό, συνεστραμμένων ζευγών κ.τ.λ).

Οι κάρτες δικτύου περιέχουν κυκλώματα τα οποία μπορούν να ανιχνεύσουν (και σε κάποιες περιπτώσεις να διορθώσουν) λάθη κατά τη μετάδοση ή λήψη δεδομένων. Επίσης ανάλογα με την κάρτα και το είδος του δικτύου, μπορεί να γίνεται και ανίχνευση συγκρούσεων στην ίδια την κάρτα.

Τέλος οι κάρτες δικτύου καθορίζουν και τις μεθόδους πρόσβασης στο μέσο του τοπικού δικτύου, σύμφωνα με τα πρότυπα IEEE-802.X ή άλλα.

2.2 Καλώδιο UTP (Unshielded Twisted Pair)

Το καλώδιο αθωράκιστων συνεστραμμένων ζευγών **UTP** είναι το πιο συνηθισμένο όπως αναφέρθηκε, μέσο δικτύωσης. Αποτελείται από τέσσερα ζεύγη λεπτών, χάλκινων καλωδίων εκ των οποίων το κάθε ένα είναι καλυμμένο με χρωματιστή πλαστική επένδυση όπως φαίνεται και στο Σχήμα 2.1 (CCNA 1 and 2, Third Edition, Cisco Press). Όλα μαζί είναι καλυμμένα με μία πλαστική εξωτερική επένδυση. Ονομάζεται αθωράκιστο γιατί απλούστατα δεν περιέχει καμία θωράκιση από εξωτερικές παρεμβολές. Ο τύπος βύσματος που χρησιμοποιείται για ο UTP είναι το RJ-45.



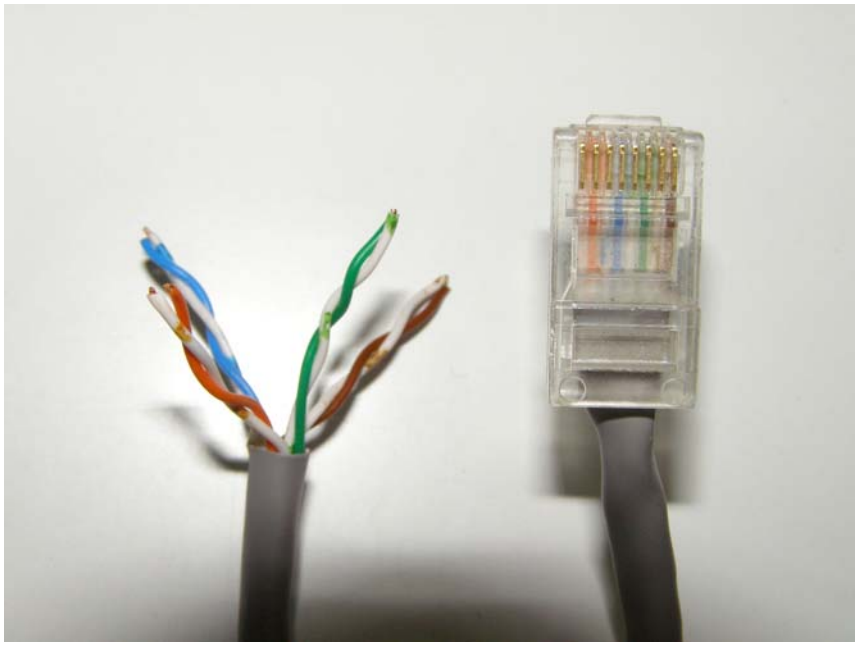
Σχήμα 2.2.1 Αναπαράσταση καλωδίου UTP

Το καλώδιο UTP έχει πολλά πλεονεκτήματα. Έχει πολύ μικρό μέγεθος διατομής ή πάχους και δε χρειάζεται γείωση, πράγμα που το κάνει το πιο φιλικό στον τομέα της εγκατάστασης καλώδιο. Είναι το φθηνότερο και η κατασκευή του είναι πολύ εύκολη. Υποστηρίζει τις ίδιες ταχύτητες μεταγωγής δεδομένων με τα άλλα χάλκινα μέσα.

Παρ' όλα αυτά, παρουσιάζει και αρκετά μειονεκτήματα. Το κυριότερο είναι ότι είναι πολύ επιρρεπές σε παρεμβολές και θόρυβο από κάθε άλλο μέσο αφού δεν διαθέτει καμία απολύτως θωράκιση. Βασίζεται μόνο στην ακύρωση σημάτων μέσω των συνεστραμμένων ζευγών για να μειώσει τα αποτελέσματα του θορύβου. Ένα ακόμα μειονέκτημά του είναι ότι έχει το μικρότερο μέγιστο μήκος μετάδοσης από τα υπόλοιπα χάλκινα και οπτικών ινών καλώδια.

Παλιότερα θεωρούταν το αργότερο μέσο για μετάδοση δεδομένων αλλά πλέον κάτι τέτοιο δεν είναι αληθές. Στην πραγματικότητα, στις μέρες μας είναι το γρηγορότερο χάλκινο μέσο μετάδοσης. Τα χαρακτηριστικά του είναι τα παρακάτω:

- Εύρος Ζώνης (Bandwidth) – 10 έως 1000 Mbps
- Κόστος αγοράς και εγκατάστασης – Το φθηνότερο της αγοράς
- Μέγεθος καλωδίου και βύσματος – Μικρό
- Μέγιστο μήκος – 100 μέτρα



Σχήμα 2.2.2 Καλώδιο UTP

Κάθε καλώδιο έχει διαφορετικό χρώμα και απαιτείται συγκεκριμένη τοποθέτησή τους για την ομαλή λειτουργία του καλωδίου και, κατά συνέπεια του δικτύου. Δεν χρησιμοποιούνται όλα τα καλώδια σε κάθε τύπο δικτύου αλλά, ειδικά στο Gigabit Ethernet είναι απαραίτητο όλα τα καλώδια να είναι συνδεδεμένα σωστά. Υπάρχουν δύο διαφορετικοί τρόποι σχεδιασμού του καλωδίου, οι T568A και T568B. Ο πρώτος χρησιμοποιείται για σύνδεση υπολογιστών με hubs και hubs με δρομολογητές και ο δεύτερος από υπολογιστή σε υπολογιστή και από hub σε hub. Στον πίνακα που ακολουθεί φαίνεται η σειρά των καλωδίων για κάθε ένα από τους δύο τύπους.

PIN – ακροδέκτες	T568A	T568B
1	Άσπρο - Πράσινο	Άσπρο - Πορτοκαλί
2	Πράσινο	Πορτοκαλί
3	Άσπρο - Πορτοκαλί	Άσπρο – Πράσινο
4	Μπλε	Μπλε
5	Άσπρο – Μπλε	Άσπρο – Μπλε
6	Πορτοκαλί	Πράσινο
7	Άσπρο – Καφέ	Άσπρο – Καφέ
8	Καφέ	Καφέ

Πίνακας 2.1 Σειρά καλωδίωσης για καλώδια UTP

Οι πιο κοινοί τύποι καλωδίων συνεστραμμένων ζευγών είναι οι ακόλουθοι:

- **Category 1 (CAT 1)** – Χρησιμοποιείται αποκλειστικά σε τηλεφω-νικές συνδέσεις.

- **Category 2 (CAT 2)** – Ικανό να στείλει δεδομένα σε ταχύτητες μέχρι 4 Mbps.

- **Category 3 (CAT 3)** – Χρησιμοποιείται σε 10BASE-T Ethernet δίκτυα. Μπορεί να στείλει δεδομένα σε ταχύτητες έως και 10Mbps.

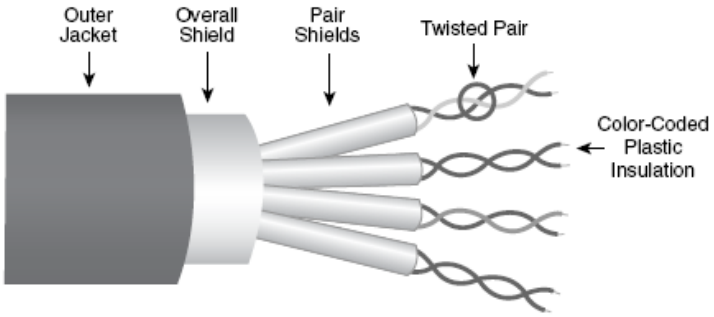
- **Category 4 (CAT 4)** – Χρησιμοποιείται σε δίκτυα σκυτάλης και δακτυλίου Token Ring. Μπορεί να μεταδώσει πληροφορία με ταχύτητες έως και 16Mbps.

- **Category 5 (CAT 5)** – Χρησιμοποιείται κυρίως στα δίκτυα Fast Ethernet. Μέγιστη ταχύτητα μετάδοσης: 100Mbps.

- **Category 5e (CAT 5e)** – Χρησιμοποιείται κυρίως σε δίκτυα που τρέχουν σε ταχύτητες της τάξης των 1000Mbps (1Gbps). Κατάλληλο για Gigabit Ethernet (GigE).

- **Category 6 (CAT 6)** – Καινούργιος τύπος καλωδίου που διατέθηκε στο κοινό στις 3 Φεβρουαρίου του έτους 2003. Χρησιμοποιείται σε δίκτυα Gigabit Ethernet.

Παράλληλα με το καλώδιο αθωράκιστων συνεστραμμένων ζευγών UTP, υπάρχει και το καλώδιο **STP** (Shielded Twisted Pair) που είναι ίδιο με το καλώδιο UTP με τη διαφορά ότι περιλαμβάνει ξεχωριστή μεταλλική θωράκιση για κάθε ζεύγος καλωδίων και στο σύνολό τους μία ενιαία κάτω ακριβώς από την πλαστική επένδυση (Σχήμα 2.2.2 CCNA 1 and 2, Third Edition, Cisco Press).



Σχήμα 2.2.3 Αναπαράσταση Καλωδίου STP

Τυπικά, τα καλώδια από την κατηγορία CAT 5 και πάνω αποτελούνται από τέσσερα ζεύγη 24 AWG χάλκινων καλωδίων. Παλιότερες καλωδιακές εγκαταστάσεις χρησιμοποιούσαν CAT 3 για μετάδοση φωνής και CAT 5 για

δεδομένα. Όταν συγκρίνουμε τα καλώδια UTP και STP πρέπει να λαμβάνουμε τα ακόλουθα υπόψη:

- Η ταχύτητα μετάδοσης και των δύο τύπων καλωδίων είναι υπερ-ικανοποιητικός για τοπικά δίκτυα (LANs).
- Είναι τα λιγότερο ακριβά μέσα για μεταφορά δεδομένων. Το STP είναι λίγο ακριβότερο.
- Πρέπει να λαμβάνεται πάντοτε υπόψη η δομημένη καλωδίωση των κτηρίων. Για παράδειγμα αν η καλωδίωση ενός κτηρίου είναι από καλώδια UTP τύπου CAT 3 δε θα μπορέσει να υποστηρίξει δίκτυο Fast Ethernet αφού το τελευταίο απαιτεί καλωδίωση τύπου τουλάχιστον CAT 5.

2.3 Αναμεταδότες (Repeaters)

Οι **αναμεταδότες** (repeaters) είναι δικτυακές συσκευές που λειτουργούν στο φυσικό επίπεδο του μοντέλου αναφοράς OSI. Για την κατανόηση του τρόπου λειτουργίας ενός αναμεταδότη είναι σημαντικό να καταλάβουμε ότι όταν τα δεδομένα φεύγουν από μία πηγή και πρέπει να κινηθούν στο δίκτυο, πρέπει να μετατραπούν είτε σε ηλεκτρικούς είτε σε οπτικούς παλμούς (παλμούς φωτός) για να περάσουν μέσα από το μέσο σύνδεσης (καλώδια χαλκού, οπτικών ινών κ.τ.λ). Αυτοί οι παλμοί ονομάζονται σήματα. Όταν τα σήματα φεύγουν από έναν πομπό είναι καθαρά και ευδιάκριτα. Ωστόσο, όσο αυξάνεται η απόσταση την οποία διανύει το σήμα επάνω στο μέσο σύνδεσης τόσο αυτό γίνεται πιο αδύναμο και δυσδιάκριτο. Ο σκοπός λοιπόν των αναμεταδοτών είναι λοιπόν να ενισχύσουν και να “αναβιώσουν” το σήμα, με σκοπό αυτό να μπορέσει να ταξιδέψει μακρύτερα χωρίς απώλειες πληροφορίας. Τοποθετούνται κοντά στο τέλος της απόστασης που μπορεί να διανύσει το σήμα και το αναμεταδίδουν.

Παλιότερα, οι αναμεταδότες θεωρούνταν σαν συσκευές με μία υποδοχή και μία έξοδο για την εξυπηρέτηση μίας και μόνο γραμμής. Σήμερα, υπάρχουν και αναμεταδότες πολλαπλών θυρών (multiport). Οι αναμεταδότες λειτουργούν στο πρώτο επίπεδο του μοντέλου OSI γιατί δρουν μόνο σε bit επίπεδο και δεν ασχολούνται με καμία επιπλέον πληροφορία και δεν παίρνουν αποφάσεις σε αντίθεση με άλλες, πιο πολύπλοκες συσκευές ανωτέρων επιπέδων.

2.4 Hubs

Σκοπό της λειτουργίας των **Hubs** είναι να ενισχύουν και να επαναχρονίζουν τα σήματα στα δίκτυα ‘όπως και οι αναμεταδότες. Τα χαρακτηριστικά τους είναι ίδια με αυτά των αναμεταδοτών. Τα Hubs είναι κοινά σημεία σύνδεσης συσκευών σε ένα δίκτυο. Επάνω τους συνδέονται διάφορες από αυτές αφού, σε αντίθεση με τους αναμεταδότες, περιέχουν περισσότερες από μία υποδοχές ή αλλιώς θύρες (ports). Όταν ένα πακέτο πληροφορίας καταφθάνει σε ένα port, αυτό αντιγράφεται και αναμεταδίδεται σε όλα τα υπόλοιπα ports, με αποτέλεσμα κάθε τμήμα του δικτύου να μπορεί να δει ο,τι κινείται μέσα στο δίκτυο.

Επειδή τα hubs και οι αναμεταδότες έχουν τα ίδια χαρακτηριστικά, τα hubs λέγονται αλλιώς και αναμεταδότες πολλαπλών θυρών (multiport repeaters). Η

διαφορά τους είναι ότι ενώ οι αναμεταδότες έχουν τυπικά μόνο δύο θύρες, τα hubs μπορούν να έχουν έως και 24 θύρες.

Τα ακόλουθα είναι τα σημαντικότερα χαρακτηριστικά των Hubs:

- Τα Hubs ενισχύουν σήματα
- Τα Hubs διαδίδουν σήματα σε όλο το δίκτυο
- Τα Hubs δε χρειάζονται φιλτράρισμα (filtering)
- Τα Hubs δε χρειάζονται ορισμό δρομολογίων και γενικότερα δρομολόγηση (routing)
- Τα Hubs χρησιμοποιούνται ως κεντρικά σημεία δικτύων

Τα Hubs χρησιμοποιούνται κοινώς σε Ethernet 10BASE-T ή 100BASE-T δίκτυα. Συνθέτουν τοπολογίες αστέρα (star) και προσφέρουν ανθεκτικότητα στο δίκτυο αφού αν ένας υπολογιστής ή γενικώς μία σύνδεση χαλάσει δεν καταρρέει ολόκληρο το δίκτυο, όπως θα συνέβαινε στην τοπολογία διαύλου (bus), παρά μόνο το σημείο που εμφάνισε το πρόβλημα.

2.5 Κατανεμητές (Switches)

Οι κατανεμητές ή switches είναι συσκευές δικτύου ακριβώς όπως τα Hubs με τη διαφορά ότι λειτουργούν στο δεύτερο επίπεδο του μοντέλου OSI (data-link layer). Μοιάζουν πολύ με τα hubs αλλά έχουν μία πολύ βασική διαφορά. Σε αντίθεση με τα hubs, τα οποία όποιο πακέτο λάβουν το αναμεταδίδουν σε όλες τους τις εξόδους, οι κατανεμητές από πριν γνωρίζουν σε ποια έξοδο είναι ο προορισμός του πακέτου με αποτέλεσμα να το αναμεταδίδουν μόνο σε αυτή. Με τον τρόπο αυτό μειώνεται κατά πολύ μεγάλο βαθμό η κίνηση και, κατά συνέπεια η συμφόρηση, του δικτύου αφού κάθε πακέτο που μεταδίδεται δε μεταφέρεται σε κάθε μονάδα του δικτύου παρά μόνο στον προορισμό.

Ουσιαστικά, κατά την εκκίνησή τους “μαθαίνουν” τη φυσική διεύθυνση (MAC Address) της συσκευής που είναι συνδεδεμένη σε κάθε θύρα τους με αποτέλεσμα να διατηρούν ένα πίνακα καταχωρήσεων. Όταν για παράδειγμα, ένα switch λάβει από τη θύρα 4 ένα πακέτο το οποίο έχει στην κεφαλίδα του τη φυσική διεύθυνση του προορισμού που στην προκειμένη περίπτωσή μας είναι ο υπολογιστής που είναι συνδεδεμένος στη θύρα 11, ελέγχοντας τον πίνακα καταχωρήσεων που διαθέτει, εντοπίζει τη θύρα που αντιστοιχεί στον προορισμό και αναμεταδίδει το πακέτο μόνο σε εκείνη.

Έτσι, επειδή εκτός του να αναμεταδίδουν «ασχολούνται» και με φυσικές διευθύνσεις, οι κατανεμητές λειτουργούν στο δεύτερο επίπεδο του μοντέλου OSI.

Ωστόσο, υπάρχουν και switches που λειτουργούν στο τρίτο επίπεδο, το επίπεδο δικτύου. Μπορούν να ρυθμιστούν (manageable) για να λειτουργούν σύμφωνα με τις δικές μας απαιτήσεις και ανάγκες.

Στο Σχήμα 2.5.1 φαίνεται ένα switch στο οποίο διακρίνονται καθαρά τα 24 ports του.

Σχήμα 2.5.1 Ένα 24-port Switch

2.6 Δρομολογητές (Routers)

Οι δρομολογητές ή routers είναι συσκευές των οποίων σκοπός είναι να περνάει πακέτα μεταξύ δικτύων χρησιμοποιώντας διευθύνσεις τρίτου επιπέδου. Ένας δρομολογητής μπορεί να παίρνει αποφάσεις σχετικά με το μονοπάτι παράδοσης των δεδομένων στο δίκτυο αφού μπορεί να προωθεί πακέτα μέσω των διευθύνσεων δικτύου (network addresses). Με άλλα λόγια σε αντίθεση με τα switches οι δρομολογητές ξέρουν ακριβώς πού να στέλνουν τα δεδομένα. Στο Σχήμα 2.6.1 μπορούμε να δούμε την εικόνα ενός δρομολογητή της κατασκευάστριας δικτυακού εξοπλισμού Cisco.

Σχήμα 2.6.1 Cisco Router 800

Λειτουργώντας στο τρίτο επίπεδο του μοντέλου OSI οι δρομολογητές, παίρνουν αποφάσεις σύμφωνα με διευθύνσεις δικτύου αντί για μοναδικές MAC addresses. Επίσης μπορούν να συνδέσουν διαφορετικές τεχνολογίες δευτέρου επιπέδου όπως Ethernet, Token Ring, FDDI. Επίσης συνδέουν Asynchronous Transfer Mode (ATM) και σειριακές συνδέσεις. Ωστόσο, εξαιτίας της ικανότητάς τους να δρομολογούν πακέτα βασισμένα σε πληροφορία τρίτου επιπέδου (Layer 3) οι δρομολογητές έχουν γίνει η ραχοκοκαλιά του Internet.

Σκοπός ενός δρομολογητή είναι να εξετάσει και να διερευνήσει κάθε εισερχόμενο πακέτο, να επιλέξει την καλύτερη δυνατή διαδρομή γι' αυτά και να τα δρομολογήσει στην αντίστοιχη έξοδό του. Οι δρομολογητές είναι οι σημαντικότερες συσκευές ρύθμισης κυκλοφορίας σε μεγάλα δίκτυα. Ουσιαστικά δίνουν τη δυνατότητα, σε κάθε υπολογιστή να επικοινωνεί εικονικά με οποιονδήποτε άλλο στον κόσμο.

Κεφάλαιο 3ο: Ρυθμίσεις δικτύου

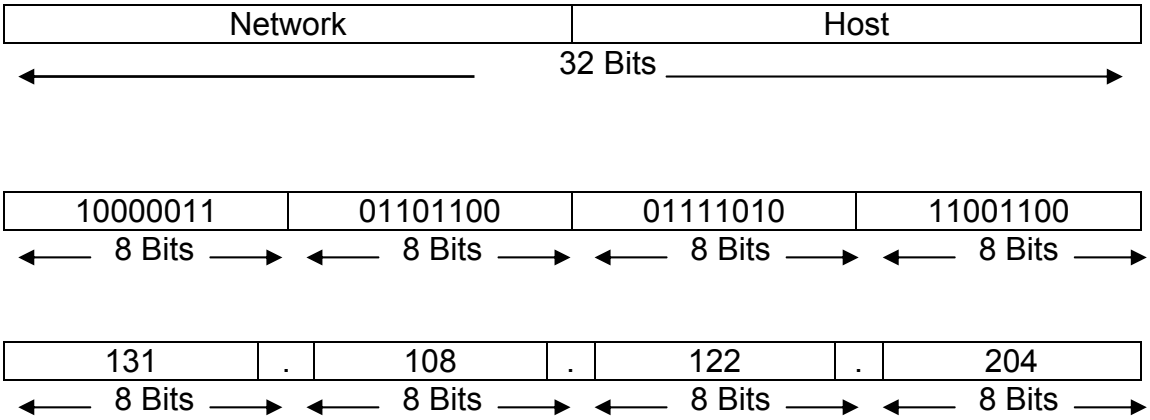
3.1 Διεύθυνση δικτύου (IP address)

Μία διεύθυνση IP (Internet protocol address), είναι ένας μοναδικός αριθμός που χρησιμοποιείται από συσκευές για τη μεταξύ τους αναγνώριση και συνεννόηση σε ένα δίκτυο υπολογιστών που χρησιμοποιεί το Internet Protocol Standard. Κάθε συσκευή που ανήκει στο δίκτυο - όπως επίσης δρομολογητές (routers), υπολογιστές, time servers, εκτυπωτές, μηχανές για fax μέσω Internet, και ορισμένα τηλέφωνα - πρέπει να έχει τη δική της μοναδική διεύθυνση. Μία διεύθυνση IP μπορεί να θεωρηθεί το αντίστοιχο μιας διεύθυνσης κατοικίας ή ενός αριθμού τηλεφώνου για έναν υπολογιστή ή άλλη συσκευή μέσα στα δίκτυα ή στο Internet. Όπως κάθε διεύθυνση κατοικίας και αριθμός τηλεφώνου αντιστοιχούν σε ένα και μοναδικό κτίριο ή τηλέφωνο, μια IP address χρησιμοποιείται για τη μοναδική αναγνώριση ενός υπολογιστή ή άλλης συσκευής που συνδέεται στο δίκτυο.

Ένας υπολογιστής μπορεί να είναι συνδεδεμένος με περισσότερα από ένα δίκτυα, για παράδειγμα με δύο. Αυτό γίνεται με το να διαθέτει δύο αντί για μία κάρτες δικτύου. Αυτός ο υπολογιστής ονομάζεται dual-home. Αυτό που πρέπει να τονιστεί είναι ότι οι δύο κάρτες του υπολογιστή ανήκουν σε δύο τελείως διαφορετικά δίκτυα συνεπώς πρέπει να έχουν διαφορετικά χαρακτηριστικά ώστε να προσδιορίζεται σε ποιο δίκτυο ανήκει η κάθε μία. Ένα ακόμη χαρακτηριστικό του υπολογιστή της περίπτωσης αυτής είναι ότι δεν περνάει πακέτα από το ένα δίκτυο στο άλλο εκτός αν οριστεί κάτι τέτοιο. Έτσι, πρέπει να του δοθούν δύο διαφορετικές διευθύνσεις, κάθε μία για να αναγνωρίζει τη σύνδεσή του σε κάθε δίκτυο. Αυστηρά και με κάθε επιφύλαξη πρέπει να τονιστεί ότι δεν μπορεί μία συσκευή όπως ο υπολογιστής να έχει διεύθυνση δικτύου αλλά κάθε σημείο σύνδεσης αυτής, όπως στο παράδειγμά μας οι κάρτες δικτύου.

3.2 Δομή IP Διευθύνσεων (IP address format)

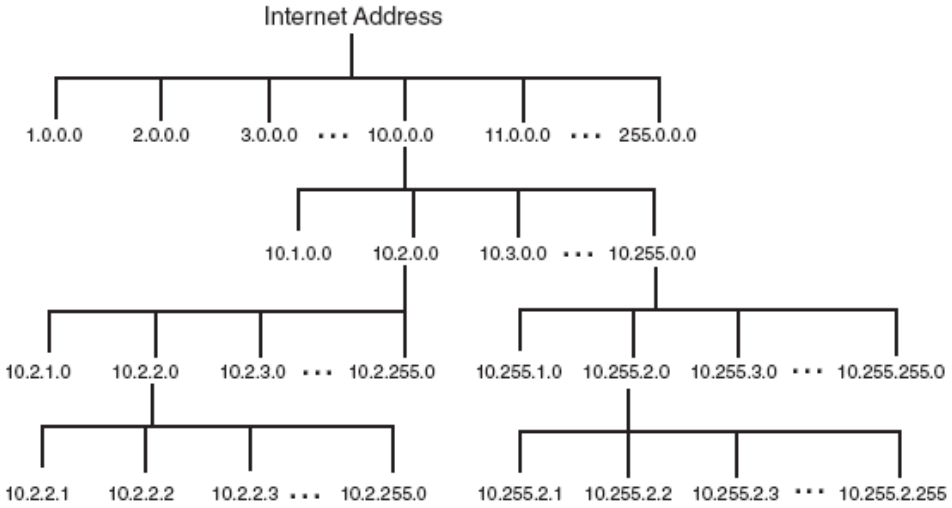
Μέσα σε έναν υπολογιστή μία διεύθυνση IP είναι αποθηκευμένη σαν μία ακολουθία 32-bit από 1 και 0 όπως φαίνεται στο Σχήμα 3.1.1. Για να καταστεί πιο εύκολη η χρήση της διεύθυνσης IP, είναι γραμμένη σαν ένα σύνολο τεσσάρων αριθμών χωρισμένοι με τελείες (για παράδειγμα 192.168.0.1). Με τον τρόπο αυτό κάθε διεύθυνση IP αποτελείται από τέσσερις αριθμούς. Κάθε αριθμός ονομάζεται octet επειδή είναι από οχτώ (8) δυαδικά ψηφία. Για παράδειγμα η διεύθυνση 192.168.1.8 είναι 11000000.10101000. 00000001.00001000 σε δυαδική μορφή. Η δεκαδική μορφή είναι όμως ευκολότερο κατανοητή για τους ανθρώπους και μειώνει τις πιθανότητες λάθους αν κάποιο ψηφίο εισαχθεί λάθος. Στο Σχήμα 3.2.1 φαίνεται η δομή μίας διεύθυνσης IP:



Σχήμα 3.2.1 Δομή διευθύνσεων IP με παράδειγμα την 131.108.122.204

Ουσιαστικά, κάθε IP διεύθυνση αποτελείται από δύο μέρη. Το πρώτο μέρος χαρακτηρίζει το δίκτυο στο οποίο είναι συνδεδεμένος ο υπολογιστής και το δεύτερο το ίδιο σύστημα μέσα στο δίκτυο. Δηλαδή το πρώτο μέρος μας λέει σε ποιο δίκτυο ανήκει το σύστημά μας και το δεύτερο ποια διεύθυνση έχει μέσα στο δίκτυο αυτό. Αυτή η πολιτική ονομάζεται ιεραρχική διευθυνσιοδότηση (hierarchical addressing) επειδή εισάγει επίπεδα ιεραρχίας.

Δημιουργεί δηλαδή ένα δέντρο ιεραρχίας όπως φαίνεται και στο Σχήμα 3.2



Σχήμα 3.2.2 Ιεραρχική ταξινόμηση IP διευθύνσεων

3.3 Κατηγορίες (Classes)

Εδώ όμως προκύπτει το ερώτημα: πώς αναγνωρίζουμε ποιο τμήμα της διεύθυνσής μας είναι η διεύθυνση του δικτύου και ποιο του συστήματός μας;

Η απάντηση αρχίζει από τους δημιουργούς του Internet, οι οποίοι σκέφτηκαν πως τα δίκτυα ποικίλουν ως προς το μέγεθός τους, ανάλογα με τον αριθμό των υπολογιστών (hosts) που περιέχουν. Έτσι δημιουργήθηκαν κάποιες κατηγορίες (classes) που ορίζουν τα μεγέθη των δικτύων. Οι κατηγορίες είναι τέσσερις, οι A, B, C και D. Στον Πίνακα 3.1 φαίνεται η κατηγοριοποίηση των δικτύων σε classes, και τα χαρακτηριστικά της κάθε μίας.

Address Class	Bits Υψηλής Τάξης	Εύρος τιμών πρώτης οκτάδας (octet)	Αριθμός Bits στη Διεύθυνση Δικτύου
A	0	0 έως 126*	8
B	10	128 έως 191	16
C	110	192 έως 223	24
D (multicast)	1110	224 έως 239	28

*Η διεύθυνση 127.x.x.x είναι δεσμευμένη ως **loopback** address*

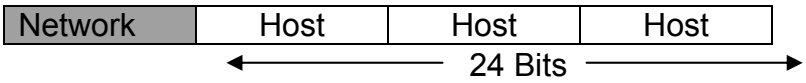
Πίνακας 3.3.1 IP Address Classes

Πρέπει να επισημάνουμε ωστόσο ότι υπάρχει μία ακόμη κατηγορία, η E. Όμως η Internet Engineering Task Force (IETF) έχει δεσμεύσει τις διευθύνσεις της κατηγορίας αυτής για δικούς της σκοπούς ερευνητικής φύσης. Γι' αυτό καμία διεύθυνση E Class δεν έχει δοθεί στο Internet.

Class A Addresses

Η class A σχεδιάστηκε για να υποστηρίζει δίκτυα πολύ μεγάλων διαστάσεων. Μία διεύθυνση της κατηγορίας αυτής χρησιμοποιεί μόνο την πρώτη οκτάδα για ορισμό του δικτύου. Οι υπόλοιπες χρησιμοποιούνται για ορισμό υπολογιστών (hosts)

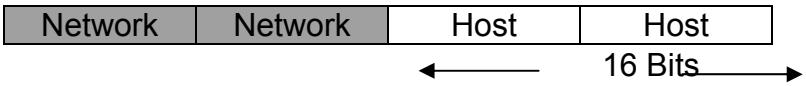
.



Το πρώτο bit μίας A Class IP Address είναι πάντα μηδέν (0). Έτσι, ο μικρότερος αριθμός που μπορεί να αναπαρασταθεί είναι ο 00000000 (στο δεκαδικό 0) και ο μεγαλύτερος ο 01111111 (στο δεκαδικό 127). Ωστόσο, οι αριθμοί 0 και 127 είναι δεσμευμένοι και δεν μπορούν να χρησιμοποιηθούν για διευθύνσεις IP. Έτσι, οποιαδήποτε διεύθυνση IP που έχει τιμή 1 έως 126 στην πρώτη οκτάδα είναι διεύθυνση A Class.

Class B Addresses

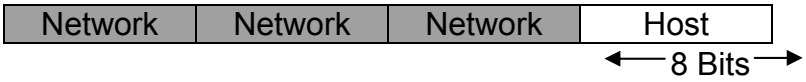
Η κατηγορία αυτή δημιουργήθηκε για να εξυπηρετεί τις ανάγκες δικτύων αρκετά μεγάλων διαστάσεων. Οι διευθύνσεις B Class χρησιμοποιούν τις δύο πρώτες οκτάδες για ορισμό της διεύθυνσης δικτύου. Οι άλλες δύο χρησιμοποιούνται για τους hosts – μέλη που εμπεριέχονται σε αυτό.



Τα πρώτα δύο bits της πρώτης οκτάδας των διευθύνσεων B Class είναι πάντα 10. Τα υπόλοιπα 6 μπορούν να είναι είτε 1 είτε 0. Έτσι ο μικρότερος αριθμός που μπορί να αναπαρασταθεί είναι ο 10000000 (στο δεκαδικό 128) και ο μεγαλύτερος ο 10111111 (στο δεκαδικό 191). Όποια διεύθυνση ξεκινά με τιμή από 128 έως 191 είναι B Class.

Class C Addresses

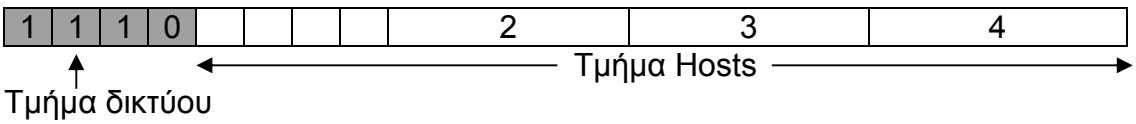
Οι C Class διευθύνσεις IP είναι οι πιο διαδεδομένες και αφορούν δίκτυα μικρότερου μεγέθους.



Στις C Class διευθύνσεις η πρώτη οκτάδα ξεκινά πάντοτε με 110. Έτσι η μικρότερη τιμή που μπορεί να πάρει είναι 11000000 (στο δεκαδικό 192) και η μεγαλύτερη 11011111 (στο δεκαδικό 223). Όποια IP διεύθυνση ξεκινάει με 192 έως 223 είναι C Class.

Class D Addresses

Η κατηγορία D Class δημιουργήθηκε για να δώσει τη δυνατότητα πολλαπλής μετάδοσης (multicasting) σε μία IP διεύθυνση. Μία multicast address είναι μία μοναδική διεύθυνση δικτύου η οποί χρησιμοποιείται για την αποστολή πακέτων σε προδιαγεγραμμένες ομάδες διευθύνσεων IP. Με τον τρόπο αυτό ένας σταθμός έχει τη δυνατότητα να αποστείλει μία σειρά δεδομένων σε περισσότερους από έναν προορισμούς.



Όπως και οι άλλες κατηγορίες, η κατηγορία D είναι μαθηματικά ορισμένη. Τα πρώτα 4 bits της πρώτης οκτάδας πρέπει να είναι 1110. Έτσι, από εκεί και πέρα, η μικρότερη τιμή για την πρώτη οκτάδα είναι η 11100000 (στο δεκαδικό 224) και η μεγαλύτερη 11101111 (στο δεκαδικό 239).

3.4 Δεσμευμένες Διευθύνσεις (reserved IP addresses)

Υπάρχουν κάποιες διευθύνσεις οι οποίες είναι δεσμευμένες και των οποίων η χρήση είναι συγκεκριμένη.

Network Addresses

Οι διευθύνσεις αυτές χρησιμοποιούνται για το προσδιορισμό του ίδιου του δικτύου ως οντότητα. Δεδομένα που είναι να σταλούν από κάποιο σταθμό έξω από το δίκτυο προς έναν υπολογιστή που ανήκει σε αυτό στέλνονται ουσιαστικά στο ίδιο το δίκτυο αρχικά και μετά στον host που είναι και ο προορισμός. Στο παράδειγμα αυτό αν η διεύθυνση του host είναι η 198.150.11.254 η διεύθυνση δικτύου (network address) είναι η 198.150.11.0. Γενικότερα όταν το τμήμα που αναφέρεται στους hosts είναι ίσο με 0 πρόκειται για διεύθυνση δικτύου.

Broadcast Addresses

Οι διευθύνσεις broadcast (στα ελληνικά εκπομπής) χρησιμοποιούνται για την εκπομπή δεδομένων προς όλους τους σταθμούς του δικτύου. Όταν ένας υπολογιστής με διεύθυνση 198.150.11.5 θέλει να αποστείλει δεδομένα προς όλα τα μέλη του δικτύου, αντί να στείλει ένα πακέτο με τη διεύθυνση το κάθε ένα, στέλνει ένα μόνο στη διεύθυνση 198.150.11.255 με αποτέλεσμα το πακέτο να φτάνει σε κάθε μία διεύθυνση, από 198.150.11.1 έως 198.150.11.254. Γενικότερα για να γίνει broadcast δεδομένων σε ένα δίκτυο χρησιμοποιείται ο αριθμός 255 στο τμήμα της διεύθυνσης των hosts.

Loopback Address

Η διεύθυνση loopback χρησιμοποιείται για την αναφορά του ίδιου του host στον ίδιο του τον εαυτό. Η δεσμευμένη αυτή διεύθυνση είναι η 127.0.0.1 και είναι για έναν υπολογιστή η διεύθυνση με την οποία μπορεί να δει τον εαυτό του. Κάθε κίνηση που δημιουργεί ένας υπολογιστής προς αυτή τη διεύθυνση οδηγεί ουσιαστικά πίσω στον ίδιο. Χρησιμοποιείται για επισκευές και αντιμετώπιση βλαβών αλλά και σε εφαρμογές (DNS κ.τ.λ)

3.5 Private και Public Διευθύνσεις και IPv6

Η σταθερότητα του Internet βασίζεται στη μοναδικότητα της κάθε διεύθυνσης IP που χρησιμοποιείται. Αυτό δικαιολογείται με ένα απλό παράδειγμα. Αν έχουμε δύο δίκτυα με ίδια διεύθυνση (198.150.11.0) συνδεδεμένα σε ένα δρομολογητή (router) και φτάσει ένα πακέτο με διεύθυνση προορισμού κάποιο από τα δύο, ο router σε ποιο δίκτυο θα τα προωθήσει; Σε μία τέτοια περίπτωση θα αυξανόταν πολύ η κίνηση, θα υπήρχε πρόβλημα ασφάλειας αφού και άλλοι χρήστες θα μπορούσαν να δουν τα δεδομένα, και κυρίως θα χανόταν μία πολύ βασική λειτουργία των δρομολογητών. Με κάποιο τρόπο λοιπόν, έπρεπε να διασφαλιστεί ένας μηχανισμός για τη διανομή μοναδικών public διευθύνσεων. Αυτή την ευθύνη, αρχικά την ανέλαβε το InterNIC (Internet Network Information Center). Αυτός ο οργανισμός έπαψε να υπάρχει και τον διαδέχθηκε η IANA (Internet Assigned Numbers Authority). Η Αρχή αυτή προσεκτικά διαχειρίζεται τις εναπομείναντες διευθύνσεις και τη διανομή τους διασφαλίζοντας τη μοναδικότητά τους για τη σωστή λειτουργία του Internet.

Έτσι λοιπόν, οι διευθύνσεις αυτές είναι μοναδικές και ονομάζονται public αφού είναι γνωστές σε όλους. Αποκτούνται από κάποιον Internet Service Provider ή αγοράζονται με κάποιο κόστος.

Μία λύση για ασφάλεια αλλά και για την αντιμετώπιση του πεπερασμένου αριθμού διευθύνσεων IP ήταν η χρήση ιδιωτικών (private) διευθύνσεων. Όπως αναφέρθηκε νωρίτερα η Internet Hosts χρειάζονται μία παγκοσμίως μοναδική διεύθυνση. Παρ' όλα αυτά, ιδιωτικά δίκτυα που δεν είναι συνδεδεμένα με το Internet μπορούν να χρησιμοποιούν οποιαδήποτε διεύθυνση για τους Hosts τους αρκεί να είναι μοναδική μέσα στο δίκτυό τους. Ωστόσο η χρήση οποιασδήποτε διεύθυνσης είναι δε συνίσταται γιατί το δίκτυο πρέπει να παραμείνει αναβαθμίσιμο, αφού κάποια στιγμή μπορεί να αποφασιστεί να συνδεθεί στο Internet.

Η RFC 1918 έχει ορίσει τρία σύνολα διευθύνσεων για ιδιωτική χρήση τα οποία φαίνονται στον Πίνακα 3.5.1.

IP Address Class	RFC1918 Private Address Range
Class A	10.0.0.0 έως 10.255.255.255
Class B	172.16.0.0 έως 172.31.255.255
Class C	192.168.0.0 έως 192.168.255.255

Πίνακας 3.5.1 Διαθέσιμες διευθύνσεις για Private χρήση

Ωστόσο, με τη ραγδαία ανάπτυξη κι εξάπλωση του Internet οι διαθέσιμες διευθύνσεις αρχίζουν και τελειώνουν. Έτσι έχουν συσταθεί νέα πρότυπα, το CIDR (Classless Interdomain Routing) και η IPv6. Οι διευθύνσεις IPv6 (Internet Protocol version 6 Addresses), είναι η εξέλιξη των υπαρχόντων IP διευθύνσεων (IPv4). Αντί για 32 bits που χρησιμοποιούνται στις IPv4, στις IPv6 χρησιμοποιούνται 128 bits με αποτέλεσμα μεγαλύτερο εύρος τιμών. Με τις IPv6 διευθύνσεις ο αριθμός τους μπορεί να φτάσει τις $3.4*10^{38}$. Χρησιμοποιεί δεκαεξαδικούς αριθμούς για την αναπαράσταση των 128 bits.

3.6 Subnetting

Η ιεραρχία δύο επιπέδων που είχε επικρατήσει στο Internet δεν είναι αρκετή για την επαρκή περιγραφή και σχεδιασμό των δικτύων. Σιγά, σιγά έγινε σαφές στην IETF ότι ήταν απαραίτητη η εύρεση ενός μηχανισμού για τη διαφοροποίηση των πολλαπλών δικτύων που αποτελούσαν υποδίκτυα για το Internet, αλλιώς δε θα υπήρχε επαρκής τρόπος για τη δρομολόγηση δεδομένων για συγκεκριμένα συστήματα σε αυτά.

Οι κατηγορίες (classes) δικτύων προσφέρουν ένα εύρος από 256 έως και 16,8 εκατομμύρια hosts.Για την επαρκή διαχείριση ενός περιορισμένου αριθμού IP διευθύνσεων, όλες οι classes μπορούν να διαιρεθούν σε μικρότερα υποδίκτυα (subnets).

Για τη δημιουργία ενός subnet, bits του τμήματος host της διεύθυνσης IP πρέπει να επαναπροσδιοριστούν ως subnetwork bits με τη διαίρεση των octets που αφορούν τις διευθύνσεις των hosts. Με τον τρόπο αυτό, μία διεύθυνση που αποτελούταν από δύο μέρη (network και host), τώρα αποτελείται από τρία της μορφής network.subnet.host. Η τεχνική αυτή αποκαλείται δανεισμός (borrowing) bits. Ένα δίκτυο της μορφής 172.16.0.0 μπορεί να διαιρεθεί σε υποδίκτυα π.χ 172.16.1.0 και 172.16.2.0 και 172.16.3.0.

Network	Subnet	Host
172.16	.2	.0

Μέσω της δημιουργίας υποδικτύων, ένας διαχειριστής δικτύου (network administrator) μπορεί να περιορίσει την περιοχή εκπομπής (broadcast domain) καθώς και να προσφέρει κάποια ασφάλεια χαμηλού επιπέδου. Σε τέτοιου είδους δίκτυα όλα τα υποδίκτυα είναι συνδεδεμένα σε ένα κοινό σημείο, σ' ένα δρομολογητή (router). Οτιδήποτε βρίσκεται πίσω από αυτόν δεν έχει καμία σημασία για το Internet.

Κάθε δίκτυο τέτοιου είδους είναι ορατό από τον έξω κόσμο σαν ένα απλό δίκτυο χωρίς να φαίνεται η δομή του. Με τον τρόπο αυτό το μέγεθος των πινάκων δρομολόγησης παραμένει σε χαμηλό επίπεδο. Οι εσωτερικές private addresses είναι έγκυρες μόνο μέσα σο δίκτυο αυτό.

Subnet Mask

Η μάσκα υποδικτύου ή subnet mask είναι ένα μέγεθος που χρησιμοποιείται από τον δρομολογητή και παρέχει σε αυτόν τις απαραίτητες πληροφορίες που απαιτούνται για να υπολογίσει σε ποιο subnet αντιστοιχεί ένας συγκεκριμένος host. Έχει την ίδια μορφή με τις διευθύνσεις IP. Αποτελείται δηλαδή από 32 bits και διαιρείται σε τέσσερις αριθμούς με τιμές από 1 έως 255.

Παρά την IP address Class, τα τελευταία 2 bits της τελευταίας οκτάδας (octet) δεν μπορούν να χρησιμοποιηθούν ποτέ για το subnetwork. Αυτά αποκαλούνται bits χαμηλότερης τάξης ή least significant bits. Χρησιμοποιώντας

όλα τα υπόλοιπα bits εκτός των δύο αυτών, προκύπτει ένα subnet με δύο μόνο hosts.

Για να υπολογίσει τον αριθμό των bits που θα χρησιμοποιηθούν, ο σχεδιαστής δικτύου πρέπει να υπολογίσει πόσους hosts θα περιλαμβάνει το μεγαλύτερο subnet καθώς και τον αριθμό των subnets. Για τον υπολογισμό των bits που θα ανατεθούν πρέπει να συμβουλευτούμε τον Πίνακα 3.6.1. Για παράδειγμα ας υποθέσουμε ότι απαιτούνται 30 hosts και 5 subnets. Βλέπουμε από τον Πίνακα 3.6.1 ότι για 30 hosts απαιτούνται 3 bits. Αυτό επίσης δίνει τη δυνατότητα δημιουργίας έξι υποδικτύων, κάτι που πληρεί την δεύτερη απαίτηση.

Slash Format	/25	/26	/27	/28	/29	/30	N/A	N/A
Mask	128	192	224	240	248	252	254	255
Bit	1	2	3	4	5	6	7	8
Value	128	64	32	16	8	4	2	1
Total Subnets		4	8	16	32	64		
Usable Subnets		2	6	14	30	62		
Total Hosts		64	32	16	8	4		
Usable Hosts		62	30	14	6	2		

Πίνακας 3.6.1 Subnets και Hosts

Μία εναλλακτική λύση για τον υπολογισμό μάσκας υποδικτύου είναι η παρακάτω μεθοδολογία:

Usable subnets = ($2^{\text{power of bits assigned}}$) – 2

Παράδειγμα: $2^3 - 2 = 6$

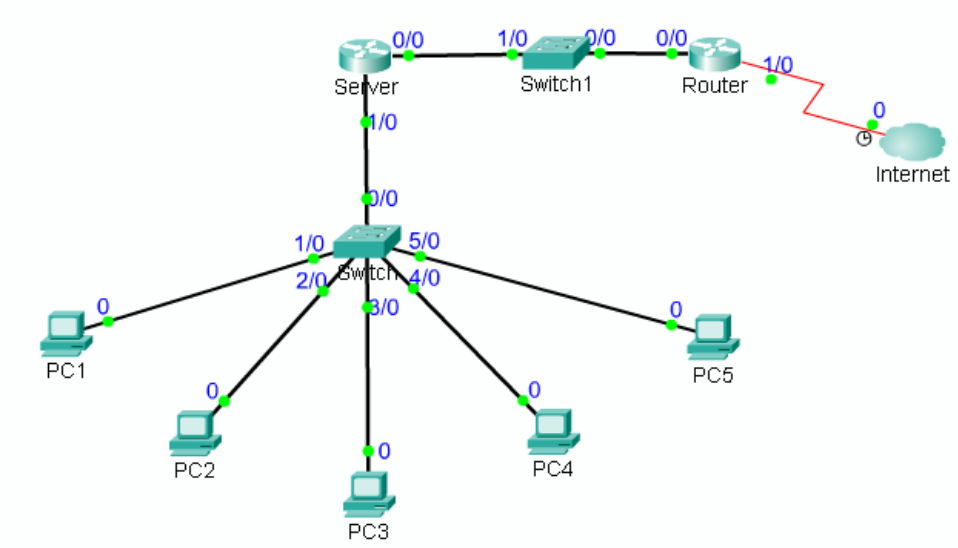
Usable hosts = ($2^{\text{power of bits assigned}}$) – 2

Παράδειγμα: $2^5 – 2 = 30$

Κάθε μάσκα υποδικτύου έχει μόνο 1 (άσσους) στο κομμάτι δικτύου και υποδικτύου και όλο 0 (μηδέν) στο κομμάτι των hosts. Από προεπιλογή (default) αν δεν οριστεί subnet mask σε ένα δίκτυο Class B η τιμή της είναι 255.255.0.0. Ωστόσο, αν δανειστούν 8 bits τότε η μάσκα γίνεται 255.255.255.0. Επειδή όμως υπάρχουν 2 οκτάδες (octets) στο πεδίο των hosts στα δίκτυα B Class, μπορούν να δανειστούν 14 bits για τη δημιουργία υποδικτύων.

Κεφάλαιο 4ο: Το Δίκτυό μας

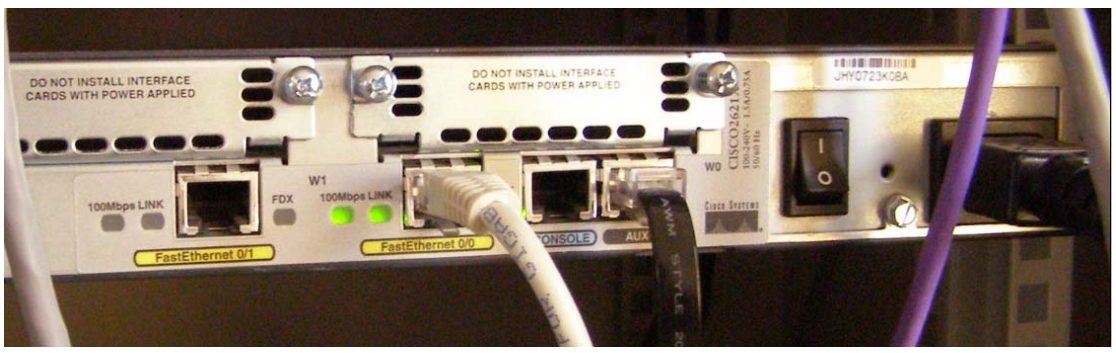
Ήρθε η στιγμή να μιλήσουμε για το δικό μας δίκτυο. Αποτελείται από ένα σύνολο clients, τον server ο οποίος είναι και το κύριο αντικείμενο ασχολίας μας, ένα switch πάνω στο οποίο είναι συνδεδεμένοι οι υπολογιστές καθώς και έναν router για την εξωτερική σύνδεση του δικτύου. Η τοπολογία είναι τύπου αστέρα και φαίνεται στο Σχήμα 4.1. Διακρίνονται οι συσκευές που απαρτίζουν το δίκτυο καθώς και τα ports που χρησιμοποιούνται. Αποτελείται από ένα δίκτυο που, όπως φαίνεται και από το σχήμα, απαρτίζεται από τον Server, ένα Switch (Switch1) και το Router για σύνδεση προς το Internet. Στον Server όμως που λειτουργεί και ως router, είναι συνδεδεμένο ένα υποδίκτυο με 5 hosts και ένα Switch στο οποίο είναι συνδεδεμένα όλα τα παραπάνω.



Σχήμα 4.1 Τοπολογία του Δικτύου

Στα σχήματα που ακολουθούν, φαίνονται τα βασικά μέλη του δικτύου εκτός των clients.

Στο Σχήμα 4.2 φαίνεται ο router του δικτύου. Είναι μοντέλο της κατασκευάστριας εταιρείας Cisco και είναι το μοντέλο **CISCO2621XM**.



Σχήμα 4.2 Router Cisco2621XM

Στο Σχήμα 4.3 φαίνεται το Switch στο οποίο συνδέεται ο Router και ο Server το οποίο είναι επίσης της Cisco και είναι από τη σειρά **Catalysts**.

Σχήμα 4.3 Κεντρικό Switch

Στο Σχήμα 4.4 φαίνεται ο Server του δικτύου που τρέχει όλες τις υλοποιημένες υπηρεσίες που αναφέρονται εδώ. Διακρίνεται καθαρά η συστοιχία των σκληρών δίσκων. Είναι τέσσερις, δύο ATA και δύο SCSI με λειτουργία mirroring.

Σχήμα 4.4 Ο Server του δικτύου

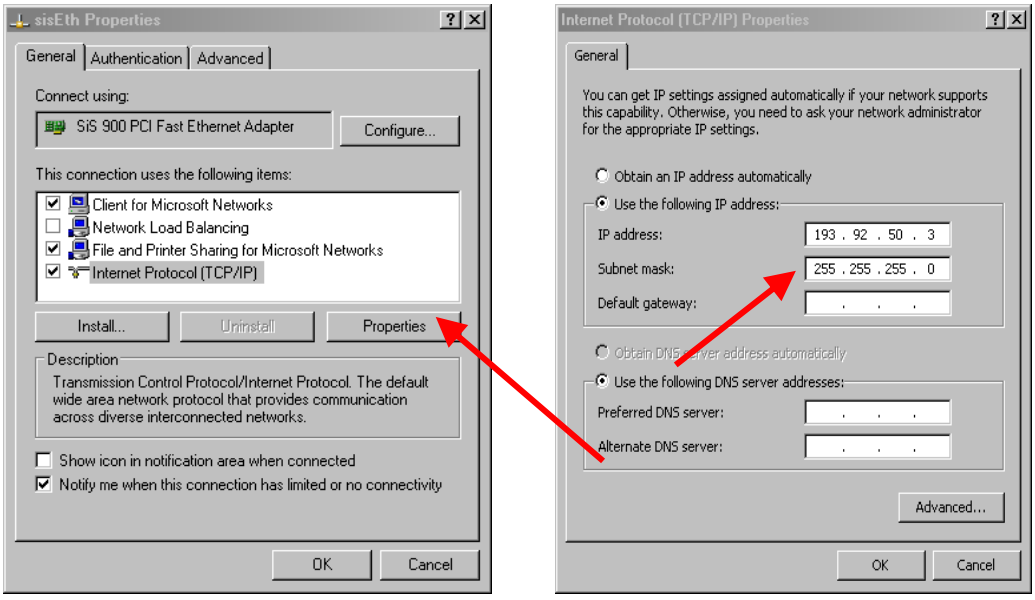
Στο Σχήμα 4.5 μπορούμε να δούμε το Switch του υποδικτύου, πάνω στο οποίο συνδέονται όλοι οι clients αλλά και ο Server.

Σχήμα 4.5 Το Switch του υποδικτύου

Δε θα αναλυθεί η συνδεσμολογία περαιτέρω από όσο φαίνεται στο Σχήμα 4.1. Αυτό που ακολουθεί είναι οι ρυθμίσεις που πρέπει να γίνουν στο δίκτυό μας. Το πρώτο βήμα είναι να ρυθμιστούν οι απαραίτητες παράμετροι για τους υπολογιστές του υποδικτύου. Το υποδίκτυο χρησιμοποιεί private διευθύνσεις και είναι δίκτυο C Class. Οι διευθύνσεις που επιλέχτηκαν για το δίκτυο είναι του συνόλου 192.168.50.x. Η διεύθυνση δηλαδή του δικτύου είναι 192.168.50.0. Πρέπει να εισάγουμε τη διεύθυνση και τη μάσκα υποδικτύου για κάθε υπολογιστή που απαρτίζει το δίκτυο. Όπως έχουμε πει, ο server ανήκει στο υποδίκτυο αφού είναι "με το ένα" πόδι μέσα σε αυτό και με το άλλο στο εξωτερικό δίκτυο. Επομένως, η μία κάρτα αυτού (eth1), η οποία ανήκει στο υποδίκτυο πρέπει να έχει διεύθυνση του συνόλου αυτού. Η διεύθυνσή του είναι η 192.168.50.200. Για να την εισάγουμε πρέπει να δώσουμε την εντολή ως root χρήστης

```
root@poultsakis:~#ifconfig eth1 192.168.50.200 netmask 255.255.255.0
```

Για κάθε MS Windows XP ρυθμίζουμε τις παραμέτρους πηγαίνοντας **Start → Control Panel → Network Connections**. Από εκεί, κάνοντας κλικ στα Properties της σύνδεσης μπορούμε να δούμε τα χαρακτηριστικά και τις ρυθμίσεις της συγκεκριμένης σύνδεσης (Σχήμα 5.1). Έχοντας επιλεγμένο το Internet Protocol (TCP/IP) και κάνοντας κλικ στο κουμπί Properties, μπορούμε να εισάγουμε τις απαραίτητες παραμέτρους στα αντίστοιχα πεδία.



Σχήμα 5.1 Ρυθμίσεις Σύνδεσης σε MS WinXP

Το εξωτερικό δίκτυο έχει διεύθυνση 10.8.1.0 και τα μέλη του 10.8.1.x. Έτσι και για τον Server, που λειτουργεί ως δρομολογητής, το εξωτερικό του interface (eth0) θα έχει μία από αυτές τις διευθύνσεις. Έχουμε επιλέξει να έχει διεύθυνση 10.8.1.10. Έτσι,

```
root@poultsakis:~#ifconfig eth0 10.8.1.10 netmask 255.255.255.0
```

Κάνοντας τα παραπάνω έχουμε ρυθμίσει τους clients να μπορούν να δουν όλο το υποδίκτυο στο οποίο ανήκουν. Ωστόσο δεν τους έχουμε ρυθμίσει ώστε να μπορούν να δουν "παραέξω", δηλαδή και το εξωτερικό δίκτυο. Πρέπει να τους εισάγουμε την προεπιλεγμένη πύλη που θα χρησιμοποιούν στην οποία θα στέλνουν τα αιτήματά τους. Αυτή η πύλη δεν είναι άλλη από το Server (πιο συγκεκριμένα της εσωτερικής κάρτας δικτύου του) που μία δουλειά του είναι να δρομολογεί πακέτα από τους clients προς τα έξω.

Στο παράθυρο των ρυθμίσεων δικτύου όπου εισάγαμε την IP και το subnet mask για κάθε client, εισάγουμε στο πεδίο default gateway την IP address της εσωτερικής κάρτας δικτύου του Server (192.168.50.200). Έτσι, όποιο αίτημα δεν είναι προς το υποδίκτυο αλλά εξωτερικό δρομολογείται προς αυτόν. Μετά αυτός, συμβουλεύοντας το routing table του, ξέρει ότι τέτοια εισερχόμενα πακέτα από το εσωτερικό interface (eth1) πρέπει να δρομολογηθούν προς το εξωτερικό (eth0). Πλέον οι υπολογιστές-μέλη του υποδικτύου μπορούν να "δουν" και άλλους που είναι μέλη του εξωτερικού δικτύου.

Το δίκτυο είναι έτοιμο να δουλέψει αλλά κάτι λείπει. Αυτό είναι η σύνδεση με το Internet. Για να γίνει κάτι τέτοιο εφικτό πρέπει να ρυθμιστεί μία προεπιλεγμένη πύλη για τον Server. Η πύλη για αυτόν είναι ο Router. Ο Router έχει ρυθμιστεί και έχει IP διεύθυνση 10.8.1.1. Για να πούμε λοιπόν στο Server να χρησιμοποιεί το Router για Gateway δίνουμε

```
root@poultsakis:~# route add default gw 10.8.1.1 eth1
```

Πλέον όλοι οι υπολογιστές του δικτύου μπορούν να βγουν και να στείλουν και να λάβουν δεδομένα από και προς το Internet.

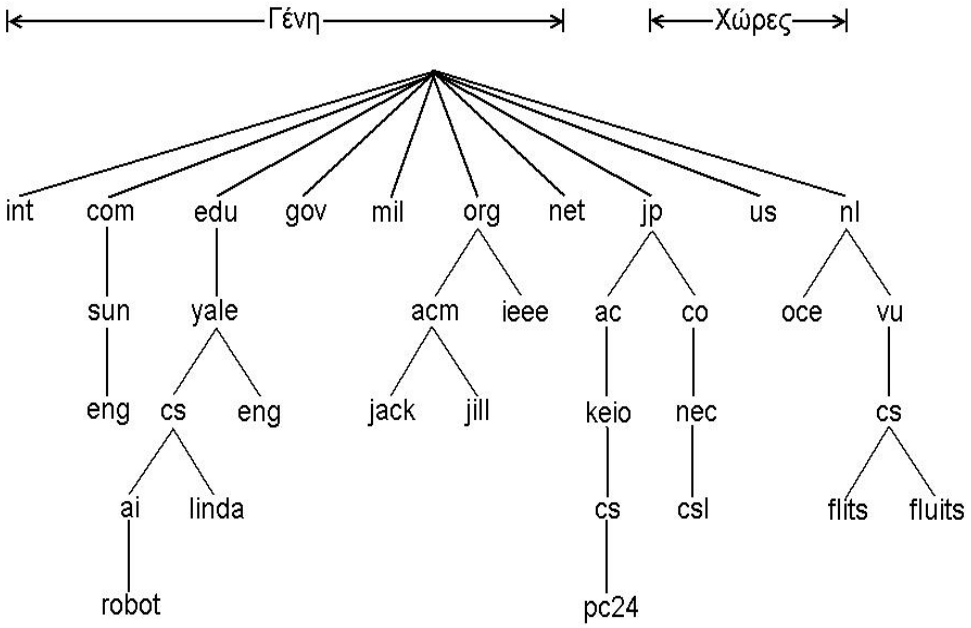
Εργαλεία διάγνωσης για τις ρυθμίσεις που έχουμε καταχωρήσει αποτελούν τα ifconfig, ping και route. Και οι δύο εντολές (ή πιο σωστά προγράμματα) χρησιμοποιήθηκαν παραπάνω.

- Μέσω της εντολής ifconfig εμφανίζονται στην οθόνη πληροφορίες και ρυθμίσεις για τις κάρτες δικτύου που λειτουργούν στο σύστημά μας. Επίσης, όπως ήδη είδαμε, μέσω αυτής μπορούμε να αλλάξουμε τις ρυθμίσεις τους.
- Με την εντολή route εμφανίζεται μπροστά μας το IP routing table ή αλλιώς ο πίνακας δρομολογήσεων του συστήματός μας. Μπορούμε να δούμε τις πύλες που χρησιμοποιούνται και προς ποια κατεύθυνση ή interface δρομολογούνται τα πακέτα ανάλογα με τον προορισμό και την προέλευση.
- Χρησιμοποιώντας την εντολή ping μπορούμε να μάθουμε αν έχουμε σύνδεση με κάποιο συγκεκριμένο σημείο. Αν σε κάποιο client κάνουμε ping 10.8.1.1 διαπιστώνουμε αν ο Router μας απαντάει. Αν ναι, σημαίνει πως έχουμε σύνδεση.

Κεφάλαιο 5ο: Services

5.1 DNS (Domain Name Server)

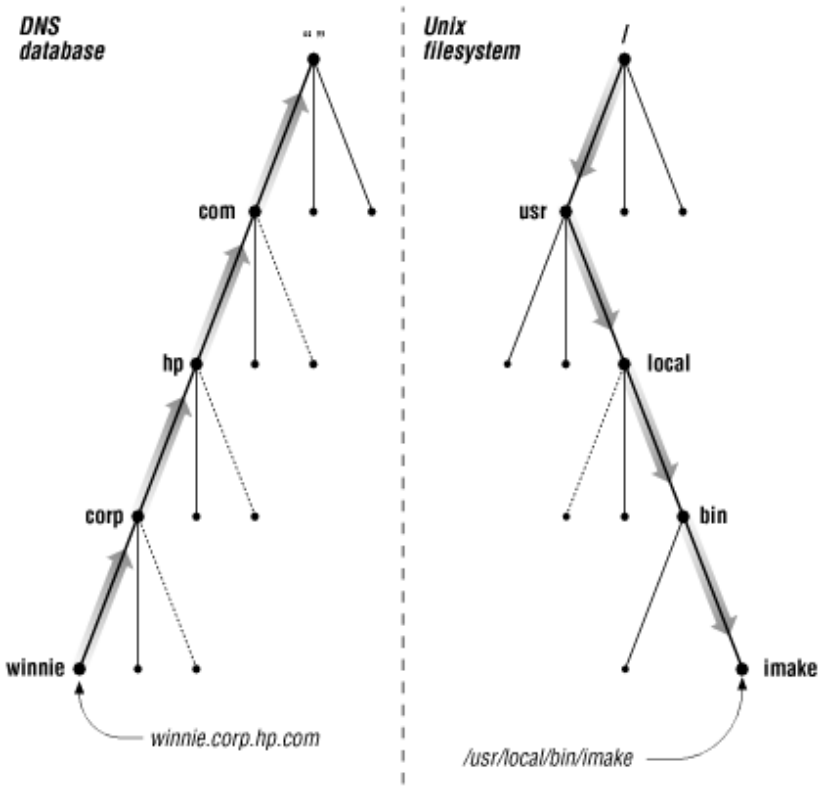
Το Internet μπορεί να αναπαρασταθεί με τη μορφή ενός δέντρου. Είναι οργανωμένο σε εκατοντάδες διαφορετικές περιοχές (**domains**) υψηλού επιπέδου, καθεμιά από τις οποίες καλύπτει πολλούς host. Κάθε περιοχή διαιρείται σε υπο-περιοχές (**sub-domains**), που διαιρούνται παραπέρα κ.ο.κ. Όπως φαίνεται και από το σχήμα 3.2 όλη η οργάνωση μπορεί να αναπαρασταθεί με τη μορφή ενός δέντρου στο οποίο τα «φύλλα» είναι περιοχές χωρίς υπο-περιοχές. Μία περιοχή-φύλλο μπορεί να περιλαμβάνει από ένα host ή και να αντιπροσωπεύει μια εταιρεία και να περιέχει χιλιάδες hosts. Οι περιοχές υψηλού επιπέδου χωρίζονται σε δύο ομάδες: γένη και χώρες. Οι περιοχές γενών είναι com (εμπορικές), edu (εκπαιδευτικοί οργανισμοί), gov (η ομοσπονδιακή κυβέρνηση των ΗΠΑ), mil (ο ομοσπονδιακός στρατός των ΗΠΑ), org (μη κερδοσκοπικοί οργανισμοί) και net (παροχείς δικτύου). Οι περιοχές χωρών περιέχουν μία καταχώρηση (υπο-ομάδα) για κάθε χώρα, κάτι που καθορίζεται από το ISO 3166.



Σχήμα 3.1. Ένα τμήμα του χώρου και της διάταξης των ονομάτων στο Internet

Κάθε υπολογιστής που θέλει να συνδεθεί στο παγκόσμιο ιστό (Internet) πρέπει να έχει μία μοναδική IP διεύθυνση (εξαιρώντας την περίπτωση των συστημάτων που βρίσκονται πίσω από ένα NAT firewall αφού δεν είναι απευθείας συνδεδεμένα στο Internet). Η υπηρεσία DNS (Domain Name Server) λειτουργεί ως υπηρεσία καταλόγου για όλα αυτά τα συστήματα δίνοντάς μας τη δυνατότητα να αναγνωρίσουμε κάποιο σύστημα μέσω του ονόματός του

(hostname). Ενώ οι υπολογιστές χρειάζονται τις δικτυακές διευθύνσεις (IP) προκειμένου να επικοινωνήσουν, οι άνθρωποι το βρίσκουν γενικά ευκολότερο να εργαστούν με ονόματα περιοχών στις δικτυακές διευθύνσεις (URLs) και στο ηλεκτρονικό ταχυδρομείο. Το DNS επομένως μεσολαβεί μεταξύ των αναγκών και των προτιμήσεων των ανθρώπων και του λογισμικού. Είναι υπηρεσία που ανήκει στο επίπεδο εφαρμογών (application layer) του δικτυακού μοντέλου OSI. Όταν κάποιος ψάχνει ένα αριθμό τηλεφώνου, ανατρέχει στον τηλεφωνικό κατάλογο, ο οποίος μέσω των ονομάτων που είναι καταχωρημένα, δίνει τον αριθμό τηλεφώνου που είναι και η μοναδική ταυτότητα για το δίκτυο του τηλεφωνικού συστήματος. Το σύστημα ονόματος περιοχών DNS είναι ένα σύστημα που αποθηκεύει τις πληροφορίες για τα διαδικτυακά ονόματα υπολογιστών και τα ονόματα δικτυακών περιοχών σε έναν τύπο κατανεμημένης βάσης δεδομένων στο διαδίκτυο. Το DNS αποθηκεύει πολλούς τύπους πληροφοριών, οι βασικότερες από τις οποίες είναι η φυσική διεύθυνση IP για κάθε όνομα περιοχής, και οι εγγραφές MX που είναι απαραίτητες για την λειτουργία του ηλεκτρονικού ταχυδρομείου. Το DNS παρέχει μια ζωτικής σημασίας υπηρεσία στο διαδίκτυο, δεδομένου ότι επιτρέπει τη μετάδοση τεχνικών πληροφοριών με έναν φιλικό προς το χρήστη τρόπο. Έτσι μας επιτρέπει να βρούμε ένα μοναδικό υπολογιστή - διακομιστή μέσω του ονόματός του και να μάθουμε την IP διεύθυνσή του, τη μοναδική του δηλαδή ταυτότητα για το Internet.



Σχήμα 3.2. Αντιστοίχιση της αναζήτησης μίας καταχώρησης μεταξύ του DNS και του file system των UNIX OS

Το να προσπαθεί κανείς να βρει ένα συγκεκριμένο υπολογιστή μέσα σε ένα παγκόσμιο δίκτυο είναι σα να προσπαθεί να βρει ένα συγκεκριμένο αρχείο μέσα σε ένα σύστημα με χιλιάδες αταξινόμητα αρχεία. Για το λόγο αυτό έχει δημιουργηθεί η ιεραρχία που αναφέρθηκε παραπάνω Το σύστημα DNS μπορεί εύκολα να παρομοιαστεί με το σύστημα αρχείων (file system) ενός λειτουργικού συστήματος UNIX (Σχήμα 3.2. O'reilly DNS and Bind, 4η έκδοση). Όπως κάθε αρχείο εντοπίζεται και ορίζεται από τη μοναδική διαδρομή σε επίπεδα - φακέλους (pathname) προς αυτό έτσι κι εδώ κάθε host εντοπίζεται από το όνομά του και τα επίπεδα – domains κάτω από τα οποία βρίσκεται. Το όνομά του κάθε σταθμού εργασίας είναι μία ακολουθία από ετικέττες με αρχή τον υπολογιστή στο τέλος της διαδρομής μέχρι την κορυφή (root) της όλης διάταξης με τα επίπεδα (sub-domains) να χωρίζονται με τελείες (dots). Στο λειτουργικό σύστημα UNIX το απόλυτο pathname είναι μία ακολουθία ονομάτων των φακέλων από τη ρίζα του δέντρου root "/" μέχρι το φύλλο – αρχείο χρησιμοποιώντας καθέτους "/" που χωρίζουν τους διαδοχικούς φακέλους – επίπεδα (αντίθετη φορά). Αξίζει να αναφερθεί οτί, σε αντίθεση με το UNIX, η ονομασία ενός domain δεν εξαρτάται από πεζά ή κεφαλαία γράμματα (case sensitive). Συνεπώς το domain με όνομα edu είναι το ίδιο με το EDU. Το μέγεθος κάθε domain δεν πρέπει να υπερβαίνει τους 63 χαρακτήρες και κάθε διαδρομή (pathname) μπορεί να φτάσει μέχρι και τους 255 χαρακτήρες.

Υπάρχουν και άλλες hostname-to-IP υπηρεσίες καταλόγου, κυρίως για τοπικά δίκτυα (LANs). Τοπικά δίκτυα σε Windows μπορούν να χρησιμοποιήσουν την υπηρεσία WINS (Windows Internet Naming Service). Συστήματα σε UNIX μπορούν να χρησιμοποιήσουν το NIS (Network Information Service). Ωστόσο επειδή η υπηρεσία DNS είναι αυτή που χρησιμοποιείται στο Internet (αλλά και σε τοπικά δίκτυα LAN) είναι η πιο ευρέως διαδεδομένη υπηρεσία καταλόγου.

Ως υπηρεσία, ο DNS, έχει καταλυτικό ρόλο στη λειτουργία του Internet . Όταν εισάγουμε www.some-domain.com σε κάποια εφαρμογή περιήγησης (Web Browser) η υπηρεσία DNS είναι αυτή που παίρνει το όνομα hostname και το μεταφράζει σε διεύθυνση δικτύου (IP address). Χωρίς τον DNS θα μπορούσαμε ακόμα να συνδεθούμε στο Internet αλλά δε θα πηγαίναμε πουθενά. Πουθενά, εκτός και αν κρατούσαμε αρχείο με τις διευθύνσεις των προορισμών που επισκεπτόμαστε και να χρησιμοποιούμε αυτές έναντι των ονομάτων (hostnames).

Έτσι λοιπόν, όταν επισκεπτόμαστε μία τοποθεσία στο Internet ουσιαστικά χρησιμοποιούμε τη διεύθυνση IP της τοποθεσίας, παρόλο που έχουμε προσδιορίσει όνομα και περιοχή (domain) στο URL. Είναι αδιαφανές στο χρήστη αλλά ο υπολογιστής που χρησιμοποιεί ρωτάει (queries) έναν εξυπηρετητή με υπηρεσία DNS (DNS server) να μάθει τη διεύθυνση IP που αντιστοιχεί στο hostname και domain name του Server στον οποίο ανήκει η τοποθεσία.

** Συνεπώς, ένα μήνυμα "cannot connect" (αδύνατη σύνδεση) δε σημαίνει απαραίτητα ότι δεν υπάρχει σύνδεση μέχρι τον προορισμό. Μπορεί κάλλιστα να υπάρχει. Το μήνυμα μπορεί να δηλώνει αποτυχία ανάλυσης και μετάφρασης του ονόματος ενός σταθμού στην IP διεύθυνσή του.*

Καθώς περιηγούμαστε στον παγκόσμιο ιστό κι επισκεπτόμαστε ιστοσελίδες ή στέλνουμε e-mail ο υπολογιστής μας ρωτάει συνεχώς έναν (ή και περισσότερους) DNS server για να αναλύει τις διευθύνσεις. Όταν επισκεπτόμαστε ιστοσελίδες, ο DNS που μας εξυπηρετεί είναι ουσιαστικά του δικού μας ISP (Internet Service Provider) αλλά θα μπορούσαμε εύκολα να χρησιμοποιούμε ένα δικό μας. Αν έχουμε τη δικιά μας ιστοσελίδα και άλλοι χρήστες την επισκέπτονται τότε χρειαζόμαστε ένα δικό μας DNS server για να απαντάει στις ερωτήσεις των δικών τους υπολογιστών. Στην περίπτωση που η σελίδα μας φιλοξενείται σε κάποια εταιρεία παροχής χώρου τότε ο DNS για τη δικιά μας τοποθεσία είναι αυτός της εταιρείας. Αν όμως θέλουμε η σελίδα να "τρέχει" στο δικό μας σύστημα είναι απαραίτητο να έχουμε τον δικό μας DNS. Στην ουσία αν στήσουμε ένα δικό μας DNS, αυτός μπορεί να απαντάει και στις εσωτερικές (internal – από το σύστημά μας) αλλά και στις εξωτερικές (external – από τους επισκέπτες μας) ερωτήσεις.

Ακόμα και στην περίπτωση που δε διαθέτουμε δικό μας domain name ή το δικό μας τοπικό δίκτυο LAN, μπορούμε να κερδίσουμε χρησιμοποιώντας ένα DNS για να έχουν άλλοι πρόσβαση στο δικό μας σύστημα. Αν έχουμε ένα σύστημα συνδεδεμένο με το Internet είτε με DSL ή καλωδιακή (cable) γραμμή είτε με ένα απλό modem, τότε μπορούμε να το μετατρέψουμε σε εξυπηρετητή που να τρέχει υπηρεσίες Web, e-mail, FTP χρησιμοποιώντας την υπηρεσία dynamic DNS.

5.2 BIND

Η πρώτη υλοποίηση του Domain Name System ονομαζόταν JEEVES και έγινε από τον Paul Mockapetris, Η δεύτερη υλοποίηση είναι το BIND, ακρωνύμιο για το "Berkeley Internet Name Domain" που δημιουργήθηκε για το λειτουργικό σύστημα του Berkeley 4.3 BSD UNIX από τον Kevin Dunlap. Πλέον το BIND το έχει αναλάβει το ISC (Internet Software Consortium). Το BIND είναι η δημοφιλέστερη υλοποίηση για Domain Name Servers και προσφέρεται με κάθε διανομή των λειτουργικών συστημάτων UNIX και Linux. Επίσης έχει εισαχθεί και στα Windows NT της Microsoft. Είναι πακέτο ανοιχτού λογισμικού (open-source) για σχεδιασμό, διαχείριση και έλεγχο ενός DNS Server και τρέχει ένα daemon με όνομα named, ο οποίος εκτός άλλων διεργασιών, απαντά σε ερωτήματα ανάλυσης διευθύνσεων (IP resolving).

Το να προσπαθεί κανείς να βρει ένα συγκεκριμένο server στο Internet είναι σα να προσπαθεί να βρει ένα συγκεκριμένο αρχείο σε ένα δίσκο με χιλιάδες αρχεία. Και στις δύο περιπτώσεις μας εξυπηρετεί η ύπαρξη κάποιας συγκεκριμένης ιεραρχίας για λογική ομαδοποίηση των αντικειμένων. Όπως είδαμε η ιεραρχία στο name system είναι παρόμοια με αυτή ενός file system αλλά με αντίθετη φορά. Όπως στο file system υπάρχει η ρίζα–αρχή με όνομα root, στο DNS υπάρχει επίσης μία ρίζα που συμβολίζεται με μία τελεία "."(dot) απ' όπου ξεκινάν οι διακλαδώσεις για τα domains υψηλού επιπέδου.

Όταν θέλουμε να δηλώσουμε ένα μονοπάτι (path) προς ένα αρχείο ενός συστήματος ξεκινώντας από τη ρίζα (root) το ορίζουμε με την παρακάτω μορφή:

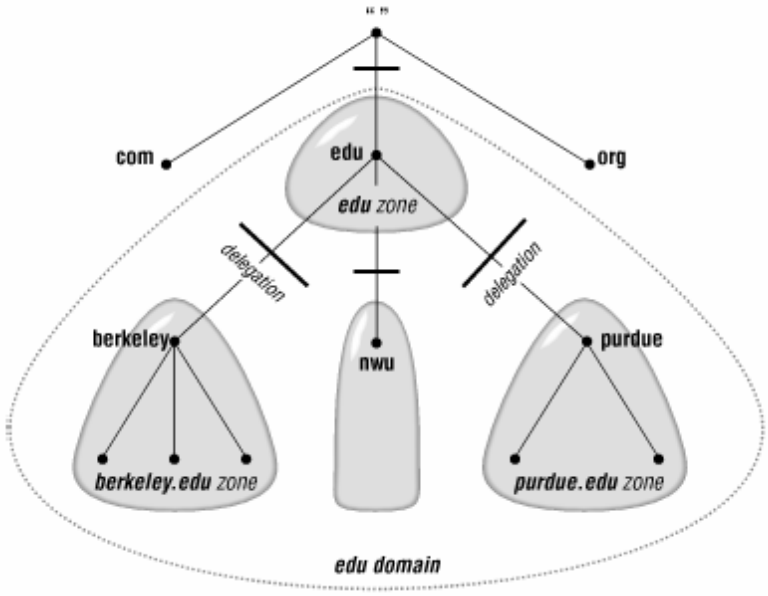
/etc/bind/named.conf

Παράλληλα στο DNS η διαδρομή για κάποιο συγκεκριμένο server ορίζεται με τη μορφή:

www.absolutepath.com.

Η παρουσία της τελείας στο τέλος του path (μετά το com) είναι απαραίτητη. Είναι ο τρόπος δήλωσης του root για το σύστημα ονομασίας (name system). Το απόλυτο μονοπάτι για το σύστημα DNS λέγεται FQDN (Fully Qualified Domain Name).

Κάθε πηγή στο Internet προσδιορίζεται από το όνομα του domain και το όνομα του σταθμού εργασίας (hostname). Τις περισσότερες φορές σε μία URL διεύθυνση ο,τι ακολουθεί το www (World Wide Web) είναι το hostname και το domain. Όπως στον τηλεφωνικό κατάλογο δεν βρίσκονται όλες οι καταχωρήσεις σε ένα συγκεκριμένο τόμο αλλά είναι χωρισμένες σε πολλούς με κάθε περιοχή ή πόλη να έχει αναλάβει την αρχειοθέτηση των δικών της καταχωρήσεων, έτσι και στο DNS οι καταχωρήσεις είναι χωρισμένες σε κομμάτια με το όνομα ζώνες (Zones) και είναι αποθηκευμένες και συντηρούνται σε διάφορες τοποθεσίες στο Internet. Ωστόσο εύκολα μπορεί να υπάρξει σύγχυση μεταξύ των εννοιών zone και domain. Χρειάζεται περαιτέρω ανάλυση για τον διαχωρισμό των δύο αυτών εννοιών αλλά συνοπτικά μπορούμε να πούμε οτί μία ζώνη (zone) μπορεί να είναι ένα σύνολο από subdomains. Όπως φαίνεται και από το παράδειγμα του σχήματος που ακολουθεί (O'reilly – DNS and Bind 4η έκδοση) το domain edu περιέχει τρία subdomains τα Berkeley, Nwu και Purdue των οποίων η διαχείριση ονομάτων έχει ανατεθεί (delegated) στους ίδιους τους οργανισμούς. Έτσι έχουν δημιουργηθεί τρεις διαφορετικές ζώνες που αποτελούνται από τρεις περιοχές – domains (τα Berkeley.edu, Nwu.edu και Purdue.edu) και τα sub-domains τους και είναι ανεξάρτητες από οτιδήποτε άλλο.



Σχήμα 3.3. Το domain edu διακλαδιζόμενο σε τρία domains που δημιουργούν τρεις διαφορετικές ζώνες,

Έτσι, κατά αντιστοιχία με τον τηλεφωνικό κατάλογο, όταν θέλουμε να βρούμε την IP διεύθυνση ενός συγκεκριμένου server σε κάποιο domain θα ανατρέξουμε στο DNS Server που έχει καταχωρημένες τις διευθύνσεις του domain που ενδιαφερόμαστε.

Οι καταχωρήσεις στη βάση δεδομένων χαρτογραφούν ένα σύνολο αντιστοιχήσεων μεταξύ hostname/domain και IP διευθύνσεις. Ακολουθεί μία απλή λογική αναπαράσταση της πληροφορίας που είναι καταχωρημένη (η ερμηνεία των A, CNAME, MX ακολουθεί παρακάτω).

A	**www.their_domain**	**172.29.183.103**
MX	**mail.their_domain.com**	**172.29.183.217**
A	**suse.my_domain.com**	**10.117.8.3**
CNAME	**www.my_domain.com**	**10.117.8.3**
MX	**suse.my_domain.com**	**10.117.8.3**

Τυπικά όταν μία εταιρεία παροχής χώρου για ιστοσελίδες και υπηρεσίες Internet (Web Hosting Service) αναλαμβάνει να μας φιλοξενήσει δεν μας παρέχει μόνο ένα Web Server για τα Web αρχεία του domain μας αλλά μας παρέχει παράλληλα και ένα DNS Server για να κρατάει τις DNS εγγραφές του domain μας. Κάπου εδώ πρέπει λοιπόν να αναφερθεί η έννοια authoritative (στα ελληνικά μία κάπως άστοχη μετάφραση είναι “έγκυρος-η”) server και απάντηση. Ονομάζουμε authoritative DNS server εκείνον που έχει τελικά την αναζητούμενη καταχώρηση και η απάντηση που δίνει στον ενδιαφερόμενο DNS Server ονομάζεται authoritative answer. Όταν ο δικός μας DNS server, ο οποίος απευθύνθηκε σε ένα DNS server που περιείχε τη ζητούμενη διεύθυνση, λάβει την authoritative απάντηση τότε τη μεταβιβάζει σε εμάς αλλά επειδή η πληροφορία που λαμβάνουμε δεν είναι απευθείας από έγκυρη πηγή τότε η απάντηση αυτή χαρακτηρίζεται non-authoritative απάντηση. Με λίγα λόγια, αν έχουμε μία δική μας τοποθεσία (π.χ Web Site) ο DNS που περιέχει τις καταχωρήσεις γι’ αυτή είναι και ο authoritative DNS Server.

Πολλές διαφορετικές ιστοσελίδες μπορούν να αντιστοιχούν σε μία μόνο IP διεύθυνση αλλά το αντίστροφο δεν ισχύει. Μία IP διεύθυνση μπορεί να αντιστοιχεί δηλαδή, σε ένα μόνο FQDN (Fully Qualified Domain Name). Η πρώτη περίπτωση (από domain name σε IP διεύθυνση) ονομάζεται forward lookup και η αντίστροφη (IP διεύθυνση σε FQDN) ονομάζεται reverse lookup. Έτσι λοιπόν, τις περισσότερες φορές, οι forward και οι reverse καταχωρήσεις δεν ταιριάζουν. Για τις reverse καταχωρήσεις είναι συνήθως υπεύθυνος ο Internet Service Provider που φιλοξενεί την ιστοσελίδα μας, γι ’αυτό είναι συνηθισμένο το reverse lookup να υποδεικνύει το domain του ISP. Αυτό δεν είναι όμως ιδιαίτερα σημαντικό για τις περισσότερες μικρές ιστοσελίδες αλλά μερικές εμπορικές (e-commerce) εφαρμογές απαιτούν καταχωρήσεις που να αντιστοιχούν σε πραγματικά δεδομένα για τη σωστή λειτουργία τους. Σε μία τέτοια περίπτωση, για να διορθωθεί αυτό χρειάζεται να έρθουμε σε επαφή με τον ISP μας για να φτιάξει μία αλλαγή στον DNS ειδικά για τις ανάγκες μας.

Όταν θέλουμε να επισκεφτούμε την τοποθεσία www.their_domain.com ο δικός μας DNS server (αυτός που έχουμε ορίσει εμείς στο TCP/IP configuration του δικού μας συστήματος) το πιθανότερο είναι να μην έχει DNS καταχώρηση για τη συγκεκριμένη τοποθεσία. Έτσι λοιπόν πρέπει να έρθει σε επαφή με τον DNS Server που την έχει. Εδώ προκύπτει το ερώτημα: Πώς γνωρίζει ο δικός μας DNS Server σε ποιόν θα απευθυνθεί; Υπάρχουν συνολικά 13 root authoritative servers παγκοσμίως στους οποίους απευθύνουν τα queries πρώτα όλοι οι DNS servers. Αυτοί οι root servers γνωρίζουν όλους τους authoritative DNS servers για όλα τα βασικά πρώτου επιπέδου domains (com, edu, net, gr κ.τ.λ). Αυτοί οι servers είναι πάντα ενήμεροι για όλους τους DNS servers τους οποίους όλοι οι Web site systems administrators έχουν αναθέσει για τα δικά τους sub-domains. Για παράδειγμα, όταν κατοχυρώνουμε (register) ένα συγκεκριμένο domain (π.χ. www.my_site.com), ουσιαστικά εισάγουμε μία εγγραφή στους .com DNS servers οι οποίοι με τη σειρά τους «δείχνουν» στον authoritative DNS server που έχουμε ορίσει εμείς να έχει καταχωρημένη την εγγραφή για τη διεύθυνση του Web server που φιλοξενεί την ιστοσελίδα μας. Ωστόσο δε θα ασχοληθούμε περαιτέρω με το πώς κανείς μπορεί να κάνει register ένα domain.

Υπάρχουν διαφορετικού τύπου εγγραφές DNS. Όλες αυτές οι πολυάριθμες εγγραφές χαρακτηρίζονται από συγκεκριμένους τύπους των οποίων οι συνηθέστερες είναι:

- **A** (address) Records: είναι ο πιο συνηθισμένος τύπος εγγραφών. Ουσιαστικά αποτελεί μία στατική χαρτογράφηση ενός hostname σε IP διεύθυνση.

- **MX** (mail eXchanger) Records: είναι τύπος εγγραφών αποκλειστικά για mail servers. Διευκρινίζει και ορίζει ένα mail server μέσα σε ένα σύνολο εγγραφών. Έτσι, αν έχουμε ένα σύστημα στο οποίο τρέχουν ταυτόχρονα οι υπηρεσίες Sendmail (mail server) και Apache (Web server) θα υπάρχουν δύο διαφορετικές εγγραφές (A και MX) που θα “δείχνουν” στην ίδια IP διεύθυνση.

- **CNAME** (Canonical Name) Records: είναι εγγραφές “ψευδωνύμων”. Είναι ένας τρόπος να έχουμε στο ίδιο σύστημα περισσότερα από ένα hostnames. Για παράδειγμα αν στο σύστημα με Sendmail και Apache θέλουμε να προσθέσουμε μία υπηρεσία WU-FTPD ώστε να λειτουργεί και ως FTP server τότε μπορούμε να προσθέσουμε μία CNAME εγγραφή με το όνομα “ftp”. Με τον τρόπο χρήστες που χρησιμοποιούν τα ftp.domain.com και www.domain.com θα έχουν πρόσβαση σε κάθε μία διαφορετική υπηρεσία ξεχωριστά.
- **NS** (Name Server) Records: οι εγγραφές αυτές ορίζουν του έγκυρους (authoritative) servers για κάποια domain.

- **SOA** (Start of Authority) Records: είναι ένας μοναδικός τύπος εγγραφής αντίθετα με τους υπόλοιπους που μπορούν να εμφανίζονται πολλές φορές μέσα σε ένα zone αρχείο. Επίσης είναι η πρώτη εγγραφή μέσα στο αρχείο. Μία τέτοια εγγραφή βρίσκεται μόνο σε zone file ενός authoritative server και περιέχει πληροφορίες όπως: ποιος είναι ο πρωτεύων authoritative server για το zone, e-mail διεύθυνση του administrator του zone, πληροφορίες χρονισμού για τους υπόλοιπους DNS servers του zone κ.τ.λ.

5.3 resolv.conf

Οι clients, όσοι δεν «τρέχουν» το BIND δηλαδή, χρησιμοποιούν το αρχείο /etc/resolv.conf για να προσδιορίσουν τη θέση του DNS που τους εξυπηρετεί αλλά και τα ονόματα των domains στα οποία ανήκουν. Το resolv.conf έχει δύο στήλες. Η πρώτη περιέχει μία λέξη κλειδί και η δεύτερη τις απαραίτητες τιμές χωρισμένες με κόμμα. Στον πίνακα που ακολουθεί φαίνεται μία αναπαράσταση της δομής του resolv.conf.

Keyword	**Value**
name server	IP διεύθυνση του DNS Server. Απαιτείται μία γραμμή για κάθε name server.
domain	Το τοπικό domain που χρησιμοποι-είται από default. Αν ο server είναι ο dns.domain.com τότε η τιμή είναι domain.com
search	Αν γίνει αναζήτηση με το όνομα μόνο ενός host παραλείποντας το domain τότε ο DNS κάνει append το όνομα σε κάθε domain name της λίστας αυτής για την αναζήτησή του. Τα domains στη λίστα αυτή πρέπει να χωρίζονται με κενό (space).

Πίνακας 4.2 Η δομή του resolv.conf

Έτσι, συμπληρώνοντας στο αρχείο resolv.conf μία καταχώρηση όπως:

```
nameserver      193.92.8.2
```

λέμε ουσιαστικά στον client ότι ο DNS Server που θα ρωτάει είναι ο 193.92.8.2. Αντίστοιχα, μπορούμε να προσθέσουμε περισσότερες εγγραφές στην περίπτωση που θέλουμε να έχουμε τη δυνατότητα να ρωτάμε πάω από ένα μόνο DNS Server.

Ένας DNS μπορεί να ρωτάει τον εαυτό του αν έχει στην cache του καταχωρήσεις αλλά και να ρωτάει άλλους. Ένας client δηλαδή μπορεί να ρωτάει ένα DNS για κάποιο συγκεκριμένο FQDN αλλά εκείνος να προωθεί (forwarding) το query σε κάποιον άλλο.

Αν θέλουμε ο DNS Server μας να ρωτάει τον εαυτό του πρέπει να υποδεικνύει και τον εαυτό του (localhost 127.0.0.1) μέσα στο resolv.conf αρχείο του. Αυτό γίνεται προσθέτοντας τη γραμμή:

```
nameserver      127.0.0.1
```

5.4 DNS Utilities

Υπάρχει ένα σύνολο εντολών σχετικά με υπηρεσίες αναζήτησης και διερεύνησης DNS (DNS lookup utilities) για όλα τα λειτουργικά συστήματα. Κάποιες είναι μόνο των UNIX και άλλες ισχύουν και στα UNIX αλλά και στα Microsoft Windows:

nslookup

Με την εντολή nslookup σε ένα Windows σύστημα μπορούμε να δούμε μέσω ποιού DNS μαθαίνουμε τη διεύθυνση του προορισμού και αν η απάντηση από αυτόν είναι authoritative. Για παράδειγμα μέσω της εντολής nslookup μπορούμε να κάνουμε μία forward αναζήτηση δίνοντας:

```
C:\>nslookup www.teicrete.gr
Server:  server1.nmc.teiher.gr
Address:  193.92.8.2

Name:    ns.admin.teiher.gr
Address:  193.92.11.2
Aliases:  www.teicrete.gr
```

Για να κάνουμε reverse αναζήτηση:

```
C:\>nslookup 193.92.11.2
Server:  server1.nmc.teiher.gr
Address:  193.92.8.2

Name:    ns.admin.teiher.gr
Address:  193.92.11.2
Aliases:  2.11.92.193.in-addr.arpa
```

Πρώτα, βλέπουμε το όνομα και την IP διεύθυνση του τοπικά ορισμένου DNS Server (locally specified DNS Server). Μετά, φαίνεται η authoritative απάντηση την οποία ο DNS Server μας έστειλε πίσω εμπεριέχοντας το όνομα

και την IP διεύθυνση του Web Server του ΤΕΙ Κρήτης. Αν ο DNS που μας εξυπηρέτησε δεν είναι ο authoritative αυτό επισημαίνεται. Π.χ:

```
C:\>nslookup www.in.gr
Server:  server1.nmc.teiher.gr
Address:  193.92.8.2

Non-authoritative answer:
Name:    www.in.gr
Address:  194.63.247.208
```

Ωστόσο πολλές φορές τα αποτελέσματα των forward και reverse αναζητήσεων δεν ταιριάζουν. Κάποιες φορές, στις reverse αναζητήσεις, μας επιστρέφεται η διεύθυνση του DNS Server ου ISP.

Αντίθετα, στα UNIX – Linux συστήματα η εντολή nslookup έχει αρχίσει εδώ και καιρό να παραμελείται και αναμένεται, μέσα σε επόμενες εκδόσεις και διανομές τους να αφαιρεθεί. Αξιόλογο είναι ότι σε πολλές διανομές δεν υπάρχει καν σελίδα – εγχειρίδιο (man page) για την εντολή αυτή. Ωστόσο:

```
root@poultsakis:~# nslookup www.teicrete.gr
Server:         193.92.8.2
Address:        193.92.8.2#53

www.teicrete.gr canonical name = ns.admin.teiher.gr.
Name:   ns.admin.teiher.gr
Address: 193.92.11.2
```

host

Έναντι της nslookup, στα UNIX συστήματα έχει επικρατήσει η εντολή host μέσω της οποίας, όπως και στην nslookup μαθαίνουμε την IP διεύθυνση ενός προορισμού ή αντίστροφα:

```
root@poultsakis:~# host www.teicrete.gr
www.teicrete.gr is an alias for ns.admin.teiher.gr.
ns.admin.teiher.gr has address 193.92.11.2
```

και για reverse:

```
root@poultsakis:~# host 193.92.11.2
2.11.92.193.in-addr.arpa  is  an  alias  for  2.0-
25.11.92.193.in-addr.arpa.
2.0-25.11.92.193.in-addr.arpa  domain  name  pointer
ns.admin.teiher.gr.
```

Όπως και για κάθε εντολή σε Linux συστήματα έτσι και η host μπορεί να δεχτεί διάφορες παραμέτρους για καλύτερα και πιο συγκεκριμένα αποτελέσματα αναζήτησης. Πληκτρολογώντας man host μας παρέχεται το εγχειρίδιό της (man page).

dig

Μία ακόμη εντολή είναι η dig. Η εντολή αυτή μας επιτρέπει να προσδιορίσουμε αν κάποιος DNS Server έχει λάβει κάποιο έγκυρο update για τη ζώνη μας. Η σύνταξη της εντολής είναι

```
dig <domain-name> <name-server> soa
```

Η επιλογή name server είναι προαιρετική. Αν ορίσουμε ένα name server τότε το dig ρωτάει (queries) εκείνο αντί για τον default που έχουμε ορίσει στο Linux σύστημά μας. Καλό είναι συνήθως να ρωτάμε και το δικό μας DNS Server όπως και ένα γνωστό για να είμαστε σίγουροι ότι οι DNS εγγραφές μας έχουν διαδοθεί σωστά. Ωστόσο η εντολή dig δουλεύει μόνο με fully qualified domain names FQDN αφού δεν ελέγχει και δε διαβάζει καθόλου το αρχείο resolv.conf.

Η παρακάτω εντολή ρωτάει τον τοπικό DNS Server. Επιστρέφει πληροφορίες για τη SOA εγγραφή και τις διευθύνσεις των DNS Servers στην ενότητα των authorities.

```
root@poultsakis:~# dig www.teicrete.gr SOA
...
...
...
;; QUESTION SECTION:
;www.teicrete.gr.          IN      SOA

;; ANSWER SECTION:
www.teicrete.gr.  172800   IN      CNAME
  ns.admin.teiher.gr.

;; AUTHORITY SECTION:
admin.teiher.gr.  172800   IN      SOA
  ns.admin.teiher.gr.       administrator.admin.teiher.gr.
    2003120558 21600 7200 172800 172800

;; Query time: 12 msec
;; SERVER: 193.92.8.2#53(193.92.8.2)
;; WHEN: Sat Oct 28 19:24:49 2006
;; MSG SIZE  rcvd: 113
```

Εδώ βλέπουμε μία περίπτωση όπου κάνουμε dig στο www.teicrete.gr χρησιμοποιώντας έναν σίγουρα (χωρίς να υπονοούμε κάτι…!) αξιόπιστο και σωστό name server, αυτόν της yahoo:

```
root@poultsakis:~# dig ns1.yahoo.com teicrete.gr  SOA
…
…
…

;; AUTHORITY SECTION:
teicrete.gr.                    172800    IN          NS
estia.csi.forth.gr.
teicrete.gr.                    172800    IN          NS
server1.nmc.teiher.gr.
teicrete.gr.                    172800    IN          NS
amaltheia.cs.teiher.gr.
teicrete.gr.                    172800    IN          NS
nic.grnet.gr.


;; ADDITIONAL SECTION:
server1.nmc.teiher.gr.    172800  IN      A     193.92.8.2
amaltheia.cs.teiher.gr.           172800    IN          A
193.92.8.76
;; Query time: 3 msec
;; SERVER: 193.92.8.2#53(193.92.8.2)
;; WHEN: Sat Oct 28 19:40:29 2006
;; MSG SIZE  rcvd: 222
```

Υπάρχουν διάφορα κριτήρια για το αν κάποιος σχεδιαστής δικτύου θα πρέπει να χρησιμοποιήσει ένα DNS server ή όχι. Ακόμα και αν καταλήξει στο συμπέρασμα ότι το δίκτυο το οποίο σχεδιάζει και διαχειρίζεται έχει την ανάγκη της παρουσίας και της λειτουργίας ενός DNS server άλλα ερωτήματα προκύπτουν.

5.5 Ο δικός μας Name Server

Στην περίπτωση του δικού μας δικτύου οι απαιτήσεις είναι οι εξής. Δεν υπάρχει ανάγκη δημιουργίας domain ή zone. Αυτό που χρειάζεται είναι τα queries για Web διευθύνσεις από τους clients του δικτύου, να γίνονται στον Server ο οποίος θα προωθεί (forwards) τα ερωτήματα αυτά σε κάποιον άλλο DNS Server ο οποίος στην περίπτωσή μας είναι ο XXX-XXX-XXX-XXX. Κατόπιν, ο δικός μας DNS Server θα κρατάει στην Cache του όποιες εγγραφές του είναι καινούργιες.

Με λίγα λόγια, ο DNS μας ονομάζεται Forwarding Name Server ο οποίος κάνει forward όλα τα requests που του γίνονται, σε κάποιον άλλο DNS Server και αποθηκεύει στην cache του τα αποτελέσματα. Κάτι τέτοιο ακούγεται ανώφελο. Ωστόσο ένας forwarding DNS μπορεί να μας ανταμείψει με δύο τρόπους όταν η σύνδεση με κάποιο εξωτερικό δίκτυο είναι αργή ή έχει υψηλό κόστος.

- **Local DNS Server Caching** – μειώνει την κίνηση προς τα έξω, αυξάνει την ταχύτητα απόκρισης και απάντησης και μειώνει την περιττή κυκλοφορία και κίνηση.
- **Remote DNS Server provides recursive query support** – Μειώνεται η κίνηση κατά μήκος του link. Συνεπάγεται: ένα μόνο query στο δίκτυο

Οι forwarding name servers μπορούν να μας απαλλάξουν από πλευράς φόρτου εργασίας στον τομέα της διαχείρισης. Αυτό γίνεται με το να παρέχουν ένα μόνο σημείο στο οποίο αλλαγές σε remote servers μπορούν να διαχειριστούν, έναντι του να πρέπει να ενημερωθεί κάθε host ξεχωριστά.

Το BIND επιτρέπει τη ρύθμιση του forwarding μέσω των παραμέτρων forward και forwarders είτε σε global επίπεδο (στον τομέα options), είτε για κάθε ζώνη στον τομέα zone του αρχείου named conf.

Παρακάτω φαίνεται το configuration του named.conf για τον forwarding name server:

```
// FORWARDING & CACHING NAME SERVER
// maintained by Poultsakis Mihalis
// 31 Oct 2006

options {
     directory "/var/named";

// Previous versions of BIND always asked
// questions using port 53, but BIND 8.1 uses an
// unprivileged
// port by default.
query-source address * port 53;

// version statement gia security gia tin apofygi gnwstwn
//adynamiwn se hacking
version "not currently available";
forwarders {194.63.237.132;193.92.8.2;};
forward only;

// apenergopoiei ola ta zone transfer requests
allow-transfer{"none";};

// Closed DNS - epitrepei mono local IPs na kanoun
// queries
// sygkekrimenou evrous timwn kathws kai ton eafto tou

allow-query {192.168.50.0/24;127.0.0.1;};
};

logging {
     channel poults_log{
      file "/var/log/named/poults.log" versions 3;
      severity info;
```

```
      print-severity yes;
      print-time yes;
      print-category yes;
        };
 category default{
 poults_log;
 };
};

// apaitoumeno local host domain

zone "localhost" in {
     type master;
     file "pri.localhost";
     allow-update{none;};
};
//localhost reverse map gia reverse lookups kai resolving
zone "0.0.127.in-addr.arpa" in {
     type master;
     file "localhost.rev";
     allow-update {none;};
};
```

Τα σχόλια μέσα στο αρχείο παρατίθενται μετά τις καθέτους (/) ή αν πρόκειται για περισσότερα από μία σειρά, σε ανά block που ξεκινάει με /* και τελειώνει με */ με κάθε ενδιάμεση σειρά να ξεκινάει με *.

Ρυθμίζοντας το named.conf όπως παραπάνω, ο Server μας είναι έτοιμος να συμπεριφέρεται ως forwarding name server. Αντίστοιχα πρέπει και οι clients να ξέρουν ότι για κάθε request, πρέπει να ρωτάνε αυτόν. Έχουμε ήδη περιγράψει ότι για Linux συστήματα, αυτό γίνεται μέσω του αρχείο resolv.conf. Για Windows συστήματα κάτι τέτοιο γίνεται μέσω των ρυθμίσεων δικτύου. Όπως έχουμε ήδη αναφέρει, πηγαίνοντας **Start → Control Panel → Network Connections** και από εκεί στα Properties του Internet Protocol (TCP/IP) μπορούμε να συμπληρώσουμε τη διεύθυνση του Server μας στο πεδίο Preferred DNS Server. Αν είχαμε υλοποιήσει άλλον ένα DNS Server για λόγους αξιοπιστίας και σταθερότητας του δικτύου μας, η διεύθυνση αυτού θα έμπαινε στο πεδίο Alternate DNS Server.

Επίσης πρέπει να ρυθμίσουμε και το /etc/resolv.conf του ίδιου του Server ώστε για κάθε εγγραφή να ρωτάει τον εαυτό του. Προσθέτουμε τη γραμμή:

```
nameserver      127.0.0.1
```

Στο φάκελο /etc/rc.d υπάρχουν συγκεκριμένου τύπου αρχεία, για κάθε service που ξεκινάει αυτόματα με την εκκίνηση του συστήματος. Υπάρχει “by default” και αρχείο με το όνομα rc.bind, το οποίο ουσιαστικά ορίζει να κάνει εκκίνηση το bind με κάθε boot του συστήματος.

5.6 Proxy

Ο Server του project το οποίο έχουμε αναλάβει πρέπει τώρα να σχεδιαστεί και να ρυθμιστεί προσφέροντας υπηρεσίες proxy cache. Το σύστημα που παρέχει την υπηρεσία αυτή μεσολαβεί ανάμεσα στο τοπικό δίκτυο και στο Internet, με τον ίδιο τρόπο που η λανθάνουσα μνήμη παρεμβαίνει μεταξύ επεξεργαστή και μνήμης. Πρώτα όμως ας ξεκαθαρίσουμε την έννοια αυτή. Ένας **proxy server** είναι ένας Web εξυπηρετητής ο οποίος αφού εξυπηρετήσει έναν client προσφέροντάς του web resources αποθηκεύει τα resources αυτά με σκοπό τη γρήγορη μελλοντική εξυπηρέτηση όταν αυτά ξαναζητηθούν. Λεπτομερέστερα, όταν ένας client ζητάει ένα web resource (web page, movie clip κ.τ.λ), η αίτησή του προωθείται σε ένα caching server. Εκείνος με τη σειρά του την προωθεί εκ μέρους του client περαιτέρω στον πραγματικό web server που περιέχει το συγκεκριμένο resource. Όταν η ζητούμενη πληροφορία επιστρέφει στον caching server εκείνος αποθηκεύει ένα αντίγραφο αυτής στην μνήμη του και κατόπιν την προωθεί στον ενδιαφερόμενο client. Με τον τρόπο αυτό, την επόμενη φορά που θα ζητηθεί το συγκεκριμένο αυτό resource από οποιοδήποτε client του δικτύου ο caching server θα του το δώσει απευθείας χωρίς να χρειαστεί να απευθυνθεί στην πραγματική πηγή. Αυτός ο caching server ονομάζεται proxy server.

Η χρήση ενός proxy server μπορεί να μειώσει πολύ σημαντικά το πλήθος των εξωτερικών αιτήσεων προς το Internet και, συνεπώς, τη σπατάλη εύρους ζώνης (bandwidth). Επίσης προσφέρει αυξημένη ταχύτητα αφού η πληροφορία κινείται εσωτερικά με τις ταχύτητες που εμείς έχουμε θέσει στο δίκτυό μας.

5.7 Squid.conf

Ας δούμε πως μπορούμε να προσδώσουμε στο σύστημά μας τις ικανότητες ενός proxy server. Για το λειτουργικό σύστημα Linux υπάρχει ένα συγκεκριμένο πακέτο με το όνομα **squid**. Μετά την εγκατάστασή του squid μέσα στο φάκελο /etc/squid/ βρίσκεται ήδη ένα configuration αρχείο με το όνομα **squid.conf**. Πριν συνεχίσουμε καλό είναι να κρατήσουμε ένα αρχικό αντίγραφο εφεδρείας δίνοντας την εντολή

cp usr/local/squid/etc/squid.conf usr/local/squid/etc/squid.conf-original

Ας εξετάσουμε το αρχείο αυτό (σχεδόν) γραμμή προς γραμμή:

- *http_port 3128*

Εδώ καθορίζουμε τη δικτυακή θύρα (tcp port) που θα χρησιμοποιεί ο proxy server, για την επικοινωνία του με τα μηχανήματα πελάτες (clients). Καλό είναι να χρησιμοποιήσουμε την προκαθορισμένη θύρα 3128 (ευρέως χρησιμοποιείται και η θύρα 8080).

- *icp_port 0*

Το **icp_port** χρησιμοποιείται για να αποστέλλονται queries τύπου Internet Cache Protocol (RFC2186) σε γειτονικούς proxy servers. Η προεπιλεγμένη θύρα είναι η 3130. Ωστόσο αν δεν έχουμε περισσότερους από ένα proxy servers μέσα στο δίκτυό μας για λόγους ασφαλείας καλό είναι να απενεργοποιηθεί ("0").

- *hierarchy_stoplist cgi-bin ?*
  acl QUERY urlpath_regex cgi-bin \?
  no_cache deny QUERY

Τα δεδομένα που περιέχουν οι περισσότερες σύγχρονες ιστοσελίδες μεταβάλλονται δυναμικά. Αυτό γίνεται με τη βοήθεια ενσωματωμένων σεναρίων. Ένα από τα πρότυπα που χρησιμοποιείται ευρύτατα για την ενσωμάτωση σεναρίων είναι το **cgi** (Common Gateway Interface). Με τις παραπάνω γραμμές υπαγορεύουμε στο squid proxy να αγνοεί όσα δεδομένα σχετίζονται με την εκτέλεση τέτοιων σεναρίων. Έτσι, δεδομένα που παράγονται δυναμικά από την εκτέλεση σεναρίων, δε θα αποθηκεύονται στον προσωρινό αποθηκευτικό χώρο του proxy μας, ούτε και θα αναζητούνται και στα περιεχόμενά του.

Πλέον περνάμε στο κομμάτι του squid που περιέχει εντολές κι επιλογές όσων αφορά το μέγεθος της cache μνήμης και αποθηκευτικού χώρου του proxy server.

- *cache_swap_low 90*
  cache_swap_hi 95

Όπως είναι προφανές, η συνεχής χρήση του squid θα έχει ως αποτέλεσμα να γεμίσει ο χώρος που χρησιμοποιείται για την αποθήκευση όλων όσων κατεβάζουμε. Για το λόγο αυτό, ο proxy ενσωματώνει μηχανισμούς με τους οποίους αποφασίζει τι θα κρατήσει και τι θα σβήσει. Οι μηχανισμοί αυτοί λειτουργούν σε δύο "σκάλες". Στην πρώτη (που ενεργοποιείται όταν ο αποθηκευτικός χώρος γεμίσει κατά το ποσοστό που δηλώνει το μέγεθος cache_swap_low) ο σχετικός μηχανισμός λειτουργεί χωρίς πρόβλημα και περιορισμούς. Ωστόσο, στη δεύτερη σκάλα (όταν το ποσοστό πλήρωσης φτάσει την τιμή της cache_swap_hi) η απελευθέρωση του χώρου πραγματοποιείται ταχύτερα και πιο "αποφασιστικά". Για παράδειγμα ορίζοντας πολύ μικρές τιμές για αυτές τις δύο μεταβλητές, ο χώρος που διαθέτει ο proxy θα μένει σε μεγάλο βαθμό ανεκμετάλλευτος.

- *maximum_object_size 4096 KB*
  minimum_object_size 0 KB

Με τα δύο παραπάνω μεγέθη καθορίζουμε την περιοχή τιμών για το μέγεθος των αντικειμένων που θα αποθηκεύει ο proxy server. Αν το άνω όριο της περιοχής αυτής είναι (αριθμητικά) μεγάλο, είναι πιθανό να συγκρατούνται ακόμα και πολύ μεγάλα αρχεία, όπως για παράδειγμα, διάφορα προγράμματα, αρχεία video ή audio που ενδέχεται να κατεβάσουμε. Κάτι τέτοιο θα ήταν άσκοπο, δε θα εξυπηρετούσε τις ανάγκες λειτουργίας τις υπηρεσίας αυτής και

θα γέμιζε πολύ γρήγορα τον αποθηκευτικό χώρο που διαχειρίζεται ο proxy. Άλλωστε, αποστολή του τελευταίου αποτελεί η επιτάχυνση της φυλλομέτρησης (browsing) ιστοσελίδων. Με βάση τις προτεινόμενες ρυθμίσεις (*maximum_object_size 4096 KB*), θα συγκρατούνται όλα τα αρχεία με μέγεθος το πολύ 4MB. Στο πεδίο minimum_object_size μπορούμε να ορίσουμε το ελάχιστο μέγεθος των αντικειμένων που θα αποθηκεύονται. Αντικείμενα με μικρότερο μέγεθος από αυτό που έχουμε ορίσει στο συγκεκριμένο πεδίο δε θα αποθηκεύονται. Στο δικό μας σχεδιασμό επιλέξαμε να μη θέσουμε κάτω όριο στο πεδίο αυτό δίνοντας τιμή μηδέν “0”.

- *cache_replacement_policy heap GDSF*
  memory_replacement_policy lru

Όπως αναφέραμε προηγουμένως, ο proxy ενσωματώνει μηχανισμούς για την αυτόματη διαγραφή όσων δεδομένων θεωρούνται άχρηστα. Για κάθε μηχανισμό όμως, διαφορετικά κριτήρια χρησιμοποιούνται για τον χαρακτηρισμό και καθορισμό ως προς το ποια δεδομένα είναι άχρηστα. Για την καλύτερη κατανόηση του ζητήματος ας δούμε ξεχωριστά τη λειτουργία του κάθε μηχανισμού.

- **lru :** Πρόκειται για το μηχανισμό που εξορισμού χρησιμοποιεί το squid. Η λογική του είναι αρκετά απλή και, με βάση αυτή, όσο πιο παλιά είναι η αποθηκευμένη πληροφορία τόσο το πιθανότερο να αντικατασταθεί. Το μοναδικό κριτήριο δηλαδή είναι η διάρκεια ζωής. Όταν γεμίσει ο αποθηκευτικός χώρος το πρώτο αρχείο που θα διαγραφεί θα είναι και το παλαιότερο.

- **heap GDSF :** Τα αντικείμενα που διατηρούνται περισσότερο είναι εκείνα για τα οποία παρατηρείται και μεγαλύτερη ζήτηση. Εκτός αυτού ο συγκεκριμένος αυτός μηχανισμός φροντίζει ώστε να διατηρούνται όσο το δυνατόν μικρότερα αντικείμενα. Το αποτέλεσμα του μηχανισμού αυτού είναι ένας αποθηκευτικός χώρος γεμάτος με πάρα πολλά μικρά αρχεία και μάλιστα δημοφιλή.

- **heap LFUDA :** Για μία ακόμη φορά ο παράγοντας που καθορίζει το αν θα διατηρηθεί ένα αντικείμενο έναντι κάποιου άλλου είναι η ζήτηση. Στην περίπτωση αυτή όμως δεν τίθεται κριτήριο ως προς το μέγεθος του αρχείου. Έτσι λοιπόν, το αποτέλεσμα είναι ο αποθηκευτικός χώρος να είναι γεμάτος από τα δημοφιλέστερα αντικείμενα αλλά να μην έχει κατά ανάγκη μεγάλο αριθμό από αυτά. Μπορεί κάλλιστα να περιέχει μικρό αριθμό αντικειμένων μεγάλου μεγέθους τα οποία όμως να είναι ιδιαίτερα δημοφιλή.

- **heap LRU :** Πρόκειται για μία εναλλακτική υλοποίηση του μηχανισμού lru η οποία κάνει χρήση της δομής δεδομένων (data structure) του σωρού (heap).

Από τους μηχανισμούς αυτούς, εμείς επιλέξαμε τον δεύτερο (**heap GDSF**). Στατιστικά προσφέρει την μεγαλύτερη πιθανότητα εύρεσης του αντικειμένου για το οποίο ενδιαφερόμαστε. Έτσι ελαχιστοποιείται η καθυστέρηση από τις συνεχής αναφορές και αιτήσεις προς το διαδίκτυο. Σε ο,τι αφορά τη διαχείριση της μνήμης που χρησιμοποιεί το squid αφήσαμε την προεπιλεγμένη ρύθμιση

- *cache_dir ufs /var/cache/squid 256 16 256*

  Με τη γραμμή αυτή ορίζουμε τη φυσική θέση του αποθηκευτικού χώρου που θα χρησιμοποιεί ο squid. Επιπρόσθετα, με το πρώτο από τα τρία νούμερα (256) είναι δυνατόν να επιλέξουμε το επιθυμητό μέγεθος σε Megabytes του εν λόγω χώρου.

- *cache_log /var/log/squid/cache.log*

Με τη γραμμή αυτή ορίζεται η διαδρομή και το αρχείο **log** στο οποίο αποθηκεύονται γενικές πληροφορίες σχετικά με τις δραστηριότητες και τη συμπεριφορά της cache. Είναι ένα (κατά την άποψή μας και σημαντικότερο) από τα διάφορα log files που έχουμε τη δυνατότητα και την επιλογή να καταγράφουν την κατάσταση των διαφόρων διεργασιών του proxy server. Το μέγεθος που μπορεί να φτάσει το log αρχείο ρυθμίζεται από τη γραμμή

- *debug_options ALL,1*

Οι επιλογές του τομέα logging (θα μπορούσε να μεταφραστεί ως «καταγραφή γεγονότων») είναι ορισμένες σε ενότητες, όπου κάθε σε κάθε αρχείο έχει ανατεθεί μία μοναδική ενότητα. Χαμηλά επίπεδα έχουν ως αποτέλεσμα λιγότερα δεδομένα εξόδου (output). Ολοκληρωτικό debugging (level 9) μπορεί να έχει ως συνέπεια αρχεία log τεραστίου μεγέθους, γι' αυτό είναι κάτι που απαιτεί προσοχή. Η λέξη ALL θέτει τα επίπεδα debugging για κάθε ενότητα. Η προτεινόμενη τιμή η "ALL,1".

- *acl all src 0.0.0.0/0.0.0.0*
  acl localhost src 127.0.0.1/255.255.255.255
  acl SSL_ports por 443 563 # ports for secure connections
  …
  acl Safe_ports port 777 #multilink http
  acl CONNECT method CONNECT
  http_access deny !safe_ports
  http_access deny CONNECT !ssl_ports

Ο proxy server μεσολαβεί για την καλύτερη διαχείριση της σύνδεσης με το Internet. Όμως οι δυνατότητες του δεν είναι απεριόριστες. Επιπλέον – αν όχι στο δικό μας δίκτυο σίγουρα σε μεγαλύτερα – προκύπτουν διάφορα ζητήματα ασφάλειας που σχετίζονται με πιθανές εκμεταλλεύσιμες αδυναμίες του. Για το λόγο αυτό, το να προσδιορίσουμε ποιος, από ποιες διευθύνσεις και από ποιες θύρες θα μπορεί να τον χρησιμοποιεί είναι μία καλή τακτική. Σε αυτή την ενότητα λοιπόν ορίζουμε αρχικά μία σειρά μεταβλητών. Στη συνέχεια, με τη

βοήθειά τους, επιτρέπουμε ή απαγορεύουμε την πρόσβαση στους πόρους του proxy server. Ο ορισμός κάθε μεταβλητής ξεκινά με το ακρωνύμιο **acl** (από το Access Control List) και συνεχίζει με το όνομα. Έπειτα με κατάλληλη χρήση των λέξεων **deny** και **allow** δίπλα από το **http_access**, περιγράφουμε την επιθυμητή συμπεριφορά (αποδοχή και άρνηση πρόσβασης στον proxy server). Παραπάνω, καθορίζονται οι δικτυακές θύρες (με τη βοήθεια των **safe_ports** και **ssl_ports**), από τις οποίες θα επιτρέπονται η σύνδεση και η επικοινωνία.

- *acl my_net src 192.168.5.0/24*

Ήρθε η ώρα να ασχοληθούμε περισσότερο με το δικό μας δίκτυο. Με τη μεταβλητή *poulchuck_net* καθορίζουμε μία περιοχή διευθύνσεων του δικού μας τοπικού δικτύου (όλες από 192.168.5.1 έως 192.168.5.254). Αργότερα μπορούμε να επιτρέψουμε ή να απαγορεύσουμε τη χρήση του proxy server από κάποιες συγκεκριμένες διευθύνσεις που θα θελήσουμε να προσθέσουμε ή να αφαιρέσουμε από το σύνολο “πρόσβασης”. Μετά την κάθετο (“/”) δηλώνουμε την μάσκα υποδικτύου (subnet mask).

- *http_access allow my_net*
  http_access allow localhost
  http_access deny all

Αφού ορίστηκαν οι απαραίτητες μεταβλητές, έφτασε η ώρα να τις χρησιμοποιηθούν. Το συγκεκριμένο απόσπασμα του configuration αρχείου squid είναι σαφές. Μέσα από τις τρεις αυτές γραμμές δηλώνεται ουσιαστικά ότι ο proxy επιτρέπει την πρόσβαση στον εαυτό του και στους υπολογιστές του τοπικού δικτύου αλλά την αρνείται σε οποιοδήποτε άλλο. Η σειρά των δηλώσεων έχει πολύ μεγάλη σημασία και μπορεί να επηρεάσει το τελικό αποτέλεσμα. Πιο συγκεκριμένα, αν κάνουμε τη δήλωση *http_access deny all* στην πρώτη σειρά ο proxy δε θα δεχόταν συνδέσεις από κανένα μηχάνημα.

- *http_reply_access allow all*

Αυτός ο κανόνας λειτουργεί συμπληρωματικά προς τους υπόλοιπους. Με τον **http_access** καθορίζουμε, όπως προαναφέρθηκε ποιοι θα μπορούν να συνδεθούν και να έχουν πρόσβαση στον proxy. Αντίθετα, με τον **http_reply_access**, υποδεικνύεται στον proxy σε ποιους επιτρέπεται να απαντά, ανεξάρτητα με το αν έχουν συνδεθεί.

- *cache_effective_user squid*

Είναι λογικό, ο proxy server, να μη γνωρίζει τι ακριβώς αποθηκεύει. Έτσι, μία επίθεση από το δίκτυο θα μπορούσε να είναι εκ φύσης παραπλανητική με αποτέλεσμα να αναγκάσει τον proxy server να γράψει στο δίσκο του κάποιο κακόβουλο πρόγραμμα. Ακριβώς γι’ αυτό το λόγο, για λόγους ασφαλείας, κατά την εγκατάσταση του πακέτου squid δημιουργείται ένας ομώνυμος λογαριασμός χρήστη (user account), με περιορισμένα δικαιώματα. Με τον τρόπο αυτό, ο χρήστης αυτός παρουσιάζεται σαν ιδιοκτήτης όλων των εγγραφών που αποθηκεύει ο proxy κατά τη λειτουργία του.

Εφαρμόζοντας τις παραπάνω εγγραφές στο αρχείο ο Server μας δουλεύει και ως Proxy. Φτάνει να δώσουμε στους Web Browsers των Clients τα απαραίτητα στοιχεία ώστε να χρησιμοποιούν αυτόν ως Proxy Server. Μέσα από το μενού των επιλογών του Web Browser εισάγουμε τη διεύθυνση του Server που είναι η 192.168.50.200 καθώς και τον αριθμό του port μέσα από το οποίο θα επικοινωνούν (3128).

Για να εκκινήσουμε τον squid δίνουμε στην κονσόλα την εντολή:

```
root@poultsakis:~# usr/local/squid/sbin/squid start
```

5.8 Samba Server

Ο Samba είναι μία υπηρεσία που δίνει τη δυνατότητα στο Linux σύστημά μας να διαμοιράζεται αρχεία, συσκευές, drives με MS Windows συστήματα. Επίσης, δίνει τη δυνατότητα δημιουργίας home directory για κάθε client.

Εδώ θα δούμε με ποιο τρόπο μπορούμε να ρυθμίσουμε τον Server ώστε να διαθέτει ένα χώρο για home directories των χρηστών για κάθε MS Windows σύστημα αλλά κυρίως θα δημιουργήσουμε ένα φάκελο στο Linux, ο οποίος θα είναι ορατός και τροποποιήσιμος μόνο από το χρήστη στον οποίο έχουν δοθεί τα απαραίτητα δικαιώματα.

5.9 smb.conf

Για το service Samba αυτό που πρέπει να ρυθμιστεί είναι το αρχείο /etc/samba/smb.conf. Το smb.conf διαθέτει μία συγκεκριμένη δομή που φαίνεται στον Πίνακα 5.6.1

Section	Description
[global]	Γενικές Samba παράμετροι
[printers]	Για sharing εκτυπωτών
[homes]	Για λογαριασμούς χρηστών
[netlogon]	Ένα sharing για αποθήκευση logon scripts
[profile]	Χώρος αποθήκευσης profile ρυθμίσεων

Πίνακας 5.6.1 Δομή του smb.conf

Στόχος είναι να δημιουργήσουμε ένα folder στο Linux σύστημά μας πάνω στο οποίο θα τοποθετήσουμε δικαιώματα μόνο σε ένα χρήστη. Μόνο ένας χρήστης δηλαδή, θα μπορεί να είτε να δει το φάκελο, είτε τα περιεχόμενά του, είτε να τροποποιήσει κάποιο από αυτά. Το σύστημα θα ελέγχει το account name των windows και αντίστοιχα θα επιτρέπει την είσοδο ή όχι στο χρήστη. Στην προκειμένη περίπτωση το όνομα του χρήστη είναι *galatis*.και το path του φακέλου /var/www/htdocs.

Επίσης, θα δημιουργηθεί κι ένας φάκελος home (home directory) για κάθε χρήστη. Τραβώντας κωδικούς ασφαλείας από ένα συγκεκριμένο password Server (ifaistos).

Ακολουθεί το τροποποιημένο αρχείο /etc/samba/smb.conf:

```
[global]
dns proxy = no
log file = /var/log/samba.%m
server string = Knossos Samba Server
password server = ifaistos
workgroup = 1sek
security = server
max log size = 50
restrict anonymous = no
domain master = no
preferred master = no
max protocol = NT
ldap ssl = No
server signing = Auto

[homes]
comment = Home Directories
browseable = no
read only = no


[htdocs]
comment = Web docs for e-class
valid users = galatis
read only = no
path = /var/www/htdocs
write list = galatis
create mask = 0777
```

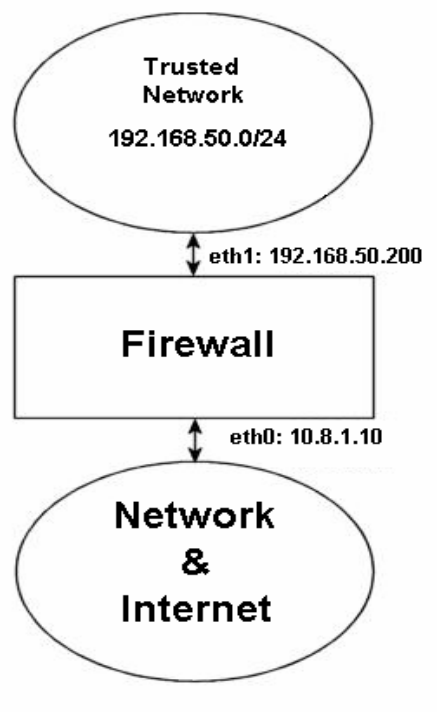
Φαίνονται καθαρά οι παράμετροι που ρυθμίζουν τη λειτουργία του Samba. Πρώτα δηλώνονται οι global παράμετροι που ισχύουν για κάθε ένα από τα τμήματα του αρχείου. Κατόπιν μπορούμε να δούμε τα άλλα δύο τμήματα, ένα για κάθε folder που δημιουργείται (homes, htdocs με path /var/www/htdocs).

Όπως και για το bind, υπάρχει αρχείο rc.samba στο φάκελο /etc/rc.d. Αυτό δίνει αυτόματα τη δυνατότητα στο Samba service να ξεκινάει με το που ξεκινάει (boots) το σύστημά μας.

5.10 Firewall

Επόμενο βήμα στην υλοποίηση του Server μας αλλά και κατά συνέπεια του δικτύου μας, είναι να του προσφέρουμε κάποια προστασία. Προστασία, είτε από εξωτερικές επιθέσεις, είτε από εσωτερικές. Θέλουμε οτιδήποτε περνάει μέσα από το Server – δρομολογητή και συνεπώς από το δίκτυό μας να φιλτράρεται προς αποφυγή δυσάρεστων καταστάσεων.

Προστασία παρέχεται μέσα σε ένα δίκτυο με την υπηρεσία firewall. Υπάρχουν πολλά προϊόντα firewall που είτε διανέμονται ελεύθερα, είτε πωλούνται στην αγορά, τα οποία όποιος χρήστης τα χρειάζεται μπορεί να τα εγκαταστήσει στο σύστημά του.



Σχήμα 5.10.1 Firewall

5.11 iptables & firewall

Ωστόσο, στην παρούσα υλοποίηση, εμείς φτιάξαμε ένα firewall πειράζοντας configuration αρχεία του συστήματός μας και συγκεκριμένα του Server. Κάθε λειτουργικό σύστημα παρέχει firewall (κατ' επιλογή) στους χρήστες του. Στο Linux τη δουλειά αυτή την κάνει το αρχείο rc.firewall. Πρόκειται για ένα αρχείο κώδικα πάνω στο οποίο βασίζονται τα υπόλοιπα scripts.

Το iptables είναι ένα εργαλείο για τον σχεδιασμό, τη συντήρηση, την επιτήρηση και τη διαχείριση των πινάκων (tables) για τα πακέτα IP που εισέρχονται στο firewall. Διάφορα tables μπορούν να οριστούν. Κάθε table περιέχει έναν αριθμό από built-in αλυσίδες (chains) αλλά και από user-defined chains.

Κάθε αλυσίδα είναι μία σειρά κανόνων και πολιτικών που αντιστοιχούν σε συγκεκριμένους τύπους πακέτων. Κάθε εισερχόμενο πακέτο, ανάλογα με τον τύπο του, αντιστοιχίζεται σε κάποιο συγκεκριμένο chain και αντίστοιχα αποφασίζεται η “τύχη” του.

Γενικότερα, υπάρχουν τέσσερις πολιτικές που ακολουθεί το firewall για την τύχη κάθε πακέτου, ανάλογα με τις προδιαγραφές και σε ποιο chain ανήκει. Αυτές είναι:

- ACCEPT
- DROP
- QUEUE
- RETURN

Το αρχείο rc.firewall απαιτεί κάποιες συγκεκριμένες παραμέτρους ή blocks από κώδικα για να γίνει σωστά compiled στον kernel. Χωρίς κάποια από αυτές, ο κώδικας θα έχει λίγο ή πολύ ελαττώματα αφού απαιτούμενες λειτουργίες δε θα τρέχουν. Οι παράμετροι αυτοί είναι οι εξής:

CONFIG_NETFILTER
CONFIG_IP_NF_CONNTRACK
CONFIG_IP_NF_IPTABLES
CONFIG_IP_NF_MATCH_LIMIT
CONFIG_IP_NF_MATCH_STATE
CONFIG_IP_NF_FILTER
CONFIG_IP_NF_NAT
CONFIG_IP_NF_TARGET_LOG

Στις επόμενες σελίδες παρατίθεται ο κώδικας του rc.firewall:

```
#PoulChuck's Firewall Settings messing up with iptables

#
# 1.1 Internet Configuration.
#

INET_IP="10.8.1.10"
INET_IFACE="eth0"
INET_BROADCAST="10.8.1.255"

# 1.2 Local Area Network configuration.
#
# your LAN's IP range and localhost IP. /24 means to only
use the first 24
# bits of the 32 bit IP address. the same as netmask
255.255.255.0
#

LAN_IP="192.168.50.200"
LAN_IP_RANGE="192.168.50.0/24"
LAN_IFACE="eth1"

#
# 1.4 Localhost Configuration.
#

LO_IFACE="lo"
LO_IP="127.0.0.1"

#
# 1.5 IPTables Configuration.
#

IPTABLES="/usr/sbin/iptables"

#
# 2. Module loading.
#
# Needed to initially load modules
/sbin/depmod -a

#
# 2.1 Required modules
#

/sbin/modprobe ip_tables
/sbin/modprobe ip_conntrack
/sbin/modprobe iptable_filter
/sbin/modprobe iptable_mangle
/sbin/modprobe iptable_nat
/sbin/modprobe ipt_LOG
/sbin/modprobe ipt_limit
/sbin/modprobe ipt_state

#
# 3.1 Required proc configuration
#
```

```
echo "1" > /proc/sys/net/ipv4/ip_forward

#
# 4.1.1 Set policies
#
$IPTABLES -F
$IPTABLES -P INPUT DROP
$IPTABLES -P OUTPUT DROP
$IPTABLES -P FORWARD DROP

#
# 4.1.2 Create userspecified chains
#
# Create chain for bad tcp packets
#

$IPTABLES -N bad_tcp_packets

#
# Create separate chains for ICMP, TCP and UDP to traverse
#

$IPTABLES -N allowed
$IPTABLES -N tcp_packets
$IPTABLES -N udp_packets
$IPTABLES -N icmp_packets

#
# 4.1.3 Create content in userspecified chains
#
# bad_tcp_packets chain
#

$IPTABLES -A bad_tcp_packets -p tcp --tcp-flags SYN,ACK
SYN,ACK \
-m state --state NEW -j REJECT --reject-with tcp-reset
$IPTABLES -A bad_tcp_packets -p tcp ! --syn -m state --
state NEW -j LOG \
--log-prefix "New not syn:"
$IPTABLES -A bad_tcp_packets -p tcp ! --syn -m state --
state NEW -j DROP

#
# allowed chain
#

$IPTABLES -A allowed -p TCP --syn -j ACCEPT
$IPTABLES -A allowed -p TCP -m state --state
ESTABLISHED,RELATED -j ACCEPT
$IPTABLES -A allowed -p TCP -j DROP

#
# TCP rules
#

$IPTABLES -A tcp_packets -p TCP -s 0/0 --dport 21 -j
allowed
```

```
$IPTABLES -A tcp_packets -p TCP -s 0/0 --dport 22 -j
allowed
$IPTABLES -A tcp_packets -p TCP -s 0/0 --dport 80 -j
allowed
$IPTABLES -A tcp_packets -p TCP -s 0/0 --dport 8080 -j
allowed
$IPTABLES -A tcp_packets -p TCP -s 0/0 --dport 113 -j
allowed

#
# UDP ports
#

#$IPTABLES -A udp_packets -p UDP -s 0/0 --destination-port
53 -j ACCEPT
#$IPTABLES -A udp_packets -p UDP -s 0/0 --destination-port
123 -j ACCEPT
#$IPTABLES -A udp_packets -p UDP -s 0/0 --destination-port
2074 -j ACCEPT
#$IPTABLES -A udp_packets -p UDP -s 0/0 --destination-port
4000 -j ACCEPT
$IPTABLES -A udp_packets -p UDP -s 0/0 --destination-port
3306 -j ACCEPT
$IPTABLES -A udp_packets -p UDP -s 0/0 --destination-port
10000 -j ACCEPT
$IPTABLES -A udp_packets -p UDP -s 0/0 --destination-port
177 -j ACCEPT

#
# ICMP rules
#

$IPTABLES -A icmp_packets -p ICMP -s 0/0 --icmp-type 8 -j
ACCEPT
$IPTABLES -A icmp_packets -p ICMP -s 0/0 --icmp-type 11 -j
ACCEPT
#
# 4.1.4 INPUT chain
#
# Bad TCP packets we don't want.
#

$IPTABLES -A INPUT -p tcp -j bad_tcp_packets

#
# Rules for special networks not part of the Internet
#

$IPTABLES -A INPUT -p ALL -i $LAN_IFACE -s $LAN_IP_RANGE -
j ACCEPT
$IPTABLES -A INPUT -p ALL -i $LO_IFACE -s $LO_IP -j ACCEPT
$IPTABLES -A INPUT -p ALL -i $LO_IFACE -s $LAN_IP -j
ACCEPT
$IPTABLES -A INPUT -p ALL -i $LO_IFACE -s $INET_IP -j
ACCEPT

#
```

```
# Special rule for DHCP requests from LAN, which are not
caught properly
# otherwise.
#

$IPTABLES -A INPUT -p UDP -i $LAN_IFACE --dport 67 --sport
68 -j ACCEPT

#
# Rules for incoming packets from the internet.
#

$IPTABLES -A INPUT -p ALL -d $INET_IP -m state --state
ESTABLISHED,RELATED \
-j ACCEPT
$IPTABLES -A INPUT -p TCP -i $INET_IFACE -j tcp_packets
$IPTABLES -A INPUT -p UDP -i $INET_IFACE -j udp_packets
$IPTABLES -A INPUT -p ICMP -i $INET_IFACE -j icmp_packets

#
# If you have a Microsoft Network on the outside of your
firewall, you may
# also get flooded by Multicasts. We drop them so we do
not get flooded by
# logs
#

#$IPTABLES -A INPUT -i $INET_IFACE -d 224.0.0.0/8 -j DROP

#
# Log weird packets that don't match the above.
#

$IPTABLES -A INPUT -m limit --limit 3/minute --limit-burst
3 -j LOG \
--log-level DEBUG --log-prefix "IPT INPUT packet died: "

#
# 4.1.5 FORWARD chain
#
# Bad TCP packets we don't want
#

$IPTABLES -A FORWARD -p tcp -j bad_tcp_packets

#
# Accept the packets we actually want to forward
#

$IPTABLES -A FORWARD -i $LAN_IFACE -j ACCEPT
$IPTABLES -A FORWARD -m state --state ESTABLISHED,RELATED
-j ACCEPT

#
# Log weird packets that don't match the above.
#
```

```
$IPTABLES -A FORWARD -m limit --limit 3/minute --limit-
burst 3 -j LOG \
--log-level DEBUG --log-prefix "IPT FORWARD packet died: "

#
# 4.1.6 OUTPUT chain
#
# Bad TCP packets we don't want.
#

$IPTABLES -A OUTPUT -p tcp -j bad_tcp_packets

#
# Special OUTPUT rules to decide which IP's to allow.
#

$IPTABLES -A OUTPUT -p ALL -s $LO_IP -j ACCEPT
$IPTABLES -A OUTPUT -p ALL -s $LAN_IP -j ACCEPT
$IPTABLES -A OUTPUT -p ALL -s $INET_IP -j ACCEPT

#
# Log weird packets that don't match the above.
#

$IPTABLES -A OUTPUT -m limit --limit 3/minute --limit-
burst 3 -j LOG \
--log-level DEBUG --log-prefix "IPT OUTPUT packet died: "

#
# Enable simple IP Forwarding and Network Address
Translation
#

$IPTABLES -t nat -A POSTROUTING -o $INET_IFACE -j SNAT --
to-source $INET_IP

#
# 4.2.6 OUTPUT chain
#
```

Μαζί με τον κώδικα υπάρχουν πολλά σχόλια που περιγράφουν το κάθε block εντολών. Κανονικά, το αρχείο rc.firewall έχει πολλά περισσότερα τμήματα, αλλά εδώ φαίνονται αυτά που αξιοποιήθηκαν. Γενικότερα, η δομή του συγκεκριμένου αρχείου ακολουθεί παρακάτω κι εξηγούνται τα σημαντικότερα σημεία του configuration.

Configuration

Internet Configuration: Στην περίπτωση του δικού μας υποδικτύου, το Internet είναι το εξωτερικό δίκτυο και ο,τι υπάρχει πέρα από αυτό. Συνεπώς, για το υποδίκτυό μας, η IP διεύθυνση του Internet είναι η 10.8.1.10 η οποία είναι η IP του εξωτερικού interface του Server (eth0).

Local Area Network Configuration: Στο πεδίο αυτό ρυθμίζεται το εύρος των διευθύνσεων του LAN που βρίσκεται πίσω από το firewall. Επίσης η διεύθυνση του interface πάνω στο οποίο είναι συνδεδεμένο το LAN (eth1).

Local Host Configuration: Ρυθμίζονται οι απαραίτητες παράμετροι για τον Server. Σαν interface με το οποίο συνδέεται για να δει τον εαυτό του ορίζουμε το loopbak (lo) και IP address 127.0.0.1.

IPtables Configuration: Ορίζεται το path για το iptables.

rules set up

Set policies: Στα system chains κάνοντας DROP σε όλα, τα απορρίπτουμε. Έτσι, μας δίνεται δυνατότητα μετά να κάνουμε ACCEPT μόνο αυτά που θέλουμε εμείς. Με αυτό τον τρόπο απαλλασσόμαστε από όλα τα ports που δε θέλουμε να χρησιμοποιούνται για είσοδο στο σύστημα από τρίτους.

Create unspecified chains: Εδώ δημιουργούμε τα chains που θα χρησιμοποιηθούν αργότερα

Create content in user specified chains: Μετά τη δημιουργία των user specified chains μπορούμε να τους εισάγουμε όλους τους κανόνες μέσα σε αυτές. Εδώ επίσης ορίζουμε τα ports που θέλουμε ανοιχτά και τα οποία απαιτούνται από τις υπηρεσίες που θέλουμε να τρέχουμε. Ορίζονται ποια πακέτα είναι κακά ποια επιτρέπονται κ.τ.λ.

INPUT chain: Έχοντας τελειώσει με το filter table θέτουμε του κανόνες για τα εισερχόμενα πακέτα από το Internet αλλά και από το υπόλοιπο εξωτερικό δίκτυο για το firewall.

Επίλογος…

Έχοντας ρυθμίσει όλα όσα αναφέρθηκαν παραπάνω, έχουμε ένα δίκτυο που λειτουργεί σε πολύ ικανοποιητικά επίπεδα.

- Έχει υλοποιηθεί η συνδεσμολογία, έχουν ρυθμιστεί οι απαραίτητες παράμετροι για κάθε μέλος του δικτύου και η μεταξύ τους επικοινωνία είναι εφικτή.

- Έχοντας τον εξυπηρετητή με δύο interfaces του δώσαμε τη δυνατότητα να είναι αυτός η γέφυρα μέσα από την οποία εισέρχονται και εξέρχονται δεδομένα προς και από το δίκτυο.

- Προσφέρεται σύνδεση με το Internet και κάθε υπολογιστής του δικτύου έχει πρόσβαση στο Internet.

- Ο Server λειτουργώντας και ως name server μέσω της υπηρεσίας BIND μπορεί να μεταφράζει hostnames σε IP διευθύνσεις με αποτέλεσμα όλοι οι clients να μπορούν να έχουν πρόσβαση σε ιστοσελίδες και web υπηρεσίες.

- Επίσης λειτουργώντας ως Proxy παρέχει ταχύτερη σύνδεση στο δίκτυο με τον εξωτερικό κόσμο.

- Με την υπηρεσία Samba παρέχεται home directory για κάθε χρήστη και, παράλληλα, ένας φάκελος με πρόσβαση αποκλειστικά σε ένα μόνο χρήστη.

- Σχεδιάζοντας ένα firewall στον Server, παρέχουμε ασφάλεια σε αυτόν αλλά και στους Clients που βρίσκονται πίσω του.

Πηγές – References

Βιβλιογραφία:

- Cisco CCNA 1 and 2 Companion Guide
  Revised Third Edition – Published by Cisco Press
  Copyright © 2005 Cisco Systems, Inc.
  ISBN: 1-58713-150-1
- Slackbook 2 – Slackware Linux Essentials, Second Edition
  Copyright © 1998, 1999, 2000, 2001, 2002, 2003, 2004, 2005
  Slackware Linux, Inc.
  ISBN: 1-57176-338-4
- O'Reilly - DNS and BIND, 4th Edition
  Copyright © 2001 O'Reilly & Associates, Inc.
  ISBN: 0-596-00158-4
- O'Reilly - Squid: The Definitive Guide – Duane Wessels
  Copyright © 2004 O'Reilly Media, Inc.
  ISBN: 0-596-00162-2
- Linux Firewalls, Third Edition
  Published by Novell Press
  Copyright © 2006 Novell Press, Inc
  ISBN-10: 0672327716
  ISBN-13: 9780672327711

Web Sites:

- http://www.linuxhomenetworking.com
  http://www.linuxhomenetworking.com/wiki/index.php/Quick _HOWTO_
- http://www.tldp.org
  http://www.tldp.org/HOWTO/DNS-HOWTO.html
  http://www.tldp.org/HOWTO/Firewall-HOWTO.html
- http://www.brennan.id.au
  http://www.brennan.id.au/08-Domain_Name_System_BIND.html
- http://www.wikipedia.com
  http://en.wikipedia.org/wiki/Ethernet
  http://el.wikipedia.org/wiki/TCP/IP
- http://www.aboutdebian.com
  http://www.aboutdebian.com/dns.htm
- http://www.squid-cache.org
  http://www.squid-cache.org/Doc/FAQ/
- http://www.faqs.org/
  http://www.faqs.org/docs/iptables/
- http://iptables-tutorial.frozentux.net
  http://iptables-tutorial.frozentux.net/iptables-tutorial.html


slackware
linux


Microsoft
Windows xp
Professional