Α.Τ.Ε.Ι. ΠΕΙΡΑΙΑ

ΤΜΗΜΑ ΗΛΕΚΤΡΟΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΙΚΩΝ ΣΥΣΤΗΜΑΤΩΝ

ΥΛΟΠΟΙΗΣΗ FIREWALL, ROUTER ΚΑΙ DNS SERVER ΣΕ ΠΛΑΤΦΟΡΜΑ LINUX/UNIX

Υπό.....

ΑΙΓΆΛΕΩ 2014

Α.Τ.Ε.Ι. ΠΕΙΡΑΙΑ

ΤΜΗΜΑ ΗΛΕΚΤΡΟΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΙΚΩΝ ΣΥΣΤΗΜΑΤΩΝ

ΥΛΟΠΟΙΗΣΗ FIREWALL, ROUTER ΚΑΙ DNS SERVER ΣΕ ΠΛΑΤΦΟΡΜΑ LINUX/UNIX

Υπό.....

ΑΙΓΑΛΕΩ 2014

ΑΤΕΙ ΠΕΙΡΑΙΑ ΣΤΕΦ
ΤΜΗΜΑ ΗΛΕΚΤΡΟΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΙΚΩΝ ΣΥΣΤΗΜΑΤΩΝ
ΑΙΓΑΛΕΩ 2014

ΥΛΟΠΟΙΗΣΗ FIREWALL, ROUTER ΚΑΙ DNS SERVER ΣΕ ΠΛΑΤΦΟΡΜΑ LINUX/UNIX

Πτυχιακή εργασία που υποβλήθηκε στο
Τ.Ε.Ι. Πειραιά για την απόκτηση του
πτυχίου

ΥΠΟ

ΠΑΠΑ ΛΑΕΡΤΙ

Επιβλέπων Καθηγητής: Δρ. Γεώργιος Ν. Πρεζεράκος

ΕΠΙΤΡΟΠΗ ΑΞΙΟΛΟΓΗΣΗΣ

_____

_____

_____

Αφιερωμένο στους γονείς μου,
Σωκράτη & Παναγιώτα

ΕΥΧΑΡΙΣΤΙΕΣ

Η παρούσα πτυχιακή εργασία εκπονήθηκε από τον φοιτητή Παπά Λαέρτη του Τμήματος Ηλεκτρονικών Υπολογιστικών Συστημάτων του Ανώτατου Τεχνολογικού Εκπαιδευτικού Ιδρύματος Πειραιά κατά το ακαδημαϊκό έτος 2013 – 2014 υπό την επίβλεψη του καθηγητή Δρ. Γεώργιου Ν. Πρεζεράκου.

Στον κύριο Πρεζεράκο οφείλω τις θερμές μου ευχαριστίες για την καθοδήγηση και την υποστήριξη του καθ' όλη τη διάρκεια διεκπεραίωσης της παρούσας πτυχιακής.

Θα ήθελα να ευχαριστήσω την οικογένεια μου για την ανυπολόγιστη ηθική υποστήριξη, την συμπαράσταση και την κατανόηση που έδειξαν όλον αυτόν τον καιρό. Σε αυτούς, που με την καθημερινή τους συμπαράσταση, την υπομονή τους και την θετική τους σκέψη, συνέβαλαν στην εκπλήρωση του στόχου μου.

ΠΕΡΙΛΗΨΗ

Στην εργασία αυτή εξετάζονται τρία κύρια στοιχεία που απαρτίζουν το Ιντερνετ σήμερα: Δρομολογητές, DNS και firewall. Γίνεται υλοποίηση τους σε λειτουργικά συστήματα Linux και Unix και αναλύουμε ποια είναι τα κύρια χαρακτηριστικά τους, ποιες δυνατότητες έχουμε και γιατί καταφεύγουμε σε μια λύση ανοιχτού λογισμικού.

Αρχικά εξετάζουμε το DNS πρωτόκολλο και αναλύουμε αναλυτικά τις προδιαγραφές και την ανάγκη ύπαρξης του. Εξετάζουμε το λειτουργικό σύστημα Linux, τα πλεονεκτήματα του και το TCP/IP πρωτόκολλο στο Linux. Γίνεται ανάλυση στα εσωτερικά πρωτόκολλα δρομολόγησης και βλέπουμε τις διαφορές τους μέσα από παραδείγματα (RIP, OSPF). Εξετάζουμε την ανάγκη για τη προστασία του δικτύου μας και γίνεται εισαγωγή στις έννοιες proxy servers και firewalls.

Γίνεται αναφορά στη τεχνολογία virtualization μιας και στη πτυχιακή αυτή ολόκληρο το πειραματικό μέρος θα γίνει με εικονικές μηχανές πάνω σε έναν ESX server της VMware η οποία είναι και πρωτοπόρος στη τεχνολογία virtualization.

Τέλος βλέπουμε τα χαρακτηριστικά των διανομών που επιλέξαμε να υλοποιήσουμε, το δρομολογητή (vyatta), το firewall (pfsense) και το DNS (named), τις δυνατότητες τους και στο τελευταίο κεφάλαιο γίνονται δοκιμές για τις λειτουργίες των κάθε στοιχείων. Υλοποιούμε λειτουργίες όπως RIP, VPN, Proxy server, URL φιλτράρισμα κτλ.

KEYWORDS

DNS, Domains, Name Server, DHCP, router, firewall, Linux, UNIX, RIP, OSPF, IGP, EGP, VMware, ESX, virtualization, IP routing, Proxies, NAT, DMZ, Virtual Machine, Vyatta, pfsense, named, VLANS, φιλτράρισμα πακέτων, VPN

ABSTRACT

In this paper we examine three main elements that make up Internet today: Routers, DNS and firewall. We see their implementation in operating systems such as Linux and UNIX and analyse what are their main characteristics, what possibilities we have and why we resort to an open source solution.

Initially we examine the DNS protocol and analyse in detail the requirements and the need for its existence. Examine the operating system Linux, its advantages and the TCP / IP protocol in Linux. We are analysing internal routing protocols and describe the differences through examples (RIP, OSPF). We review the need for the protection of our network and make an introduction to the concepts of proxy servers and firewalls.

Reference is made to virtualization technology as the implementation of our network topology will be done with virtual machines on a VMware ESX server which is market leader in virtualization.

Finally we see the characteristics of the distributions we chose to implement the router (vyatta), firewall (pfsense) and DNS (named), their capabilities and in the last chapter we are testing the functions of each component. We implement functions such as RIP, VPN, Proxy server, packet filtering, URL filtering, VLANs, NAT, DHCP etc.

KEYWORDS

DNS, Domain, Name Server, DHCP, router, firewall, Linux, UNIX, RIP, OSPF, IGP, EGP, VMware, ESX, virtualization, IP routing, Proxies, NAT, DMZ, Virtual Machine, Vyatta, pfsense, named, VLANs, packet filtering, VPN

ΠΕΡΙΕΧΟΜΕΝΑ

ΕΙΚΟΝΕΣ

1 ΕΙΣΑΓΩΓΗ

Στην εργασία αυτή εξετάζονται τρία κύρια στοιχεία που απαρτίζουν το Ιντερνετ σήμερα: Δρομολογητές, DNS και firewall. Γίνεται υλοποίηση τους σε λειτουργικά συστήματα Linux και Unix και αναλύουμε ποια είναι τα κύρια χαρακτηριστικά τους, ποιες δυνατότητες έχουμε και γιατί καταφεύγουμε σε μια λύση ανοιχτού λογισμικού.

Αρχικά στο 2° κεφάλαιο κάνουμε εισαγωγή στην ιστορία του Ιντερνετ και εξηγούμε ποιες ανάγκες μας οδήγησαν στην ανάπτυξη του DNS πρωτοκόλλου. Γίνεται ανάλυση της λειτουργίας του DNS: με ποιες προδιαγραφές σχεδιάστηκε, η λειτουργία της ανάλυσης, πως χωρίζεται ο χώρος ονομάτων του DNS και εξηγούμε αναλυτικά την ανάθεση ζωνών και ποια η διαφορά τους με τα domains.

Στο 3° κεφάλαιο βλέπουμε το λειτουργικό σύστημα Linux. Κάνουμε εισαγωγή στην έννοια του συγκεκριμένου λειτουργικού συστήματος, και βλέπουμε ποια είναι τα πλεονεκτήματα του. Εξηγούμε για ποιο λόγο κάποιος πρέπει να προτιμήσει το λειτουργικό σύστημα Linux

Στο 4° κεφάλαιο κάνουμε μια μικρή εισαγωγή στα τέσσερα επίπεδα του TCP/IP πρωτοκόλλου και εξηγούμε τρία βασικά πρωτόκολλα που χρησιμοποιούνται για την επικοινωνία κόμβων σε ένα TCP/IP δίκτυο. Τα πρωτόκολλα αυτά είναι ι τα ARP, ICMP και IGMP. Τέλος βλέπουμε ένα μικρό παράδειγμα πως χρησιμοποιούνται τα πρωτόκολλα αυτά για την επικοινωνία δύο απομακρυσμένων δικτύων.

Στο κεφάλαιο 5 βλέπουμε το πρωτόκολλο IP το οποίο χρησιμοποιείται για τη δρομολόγηση των πακέτων μας σε ολόκληρο το Ιντερνετ. Περιγράφουμε τα πρωτόκολλα εσωτερικής δρομολόγησης (IGP) και φέρνουμε μερικά παραδείγματα για τα πρωτόκολλα OSPF και RIP. Τα παραδείγματα γίνονται σε δρομολογητές της Cisco, ενώ αργότερα στο κεφάλαιο 10 χρησιμοποιούμε τις έννοιες αυτές για την υλοποίηση του δικτύου μας, με δύο δρομολογητές της Linux διανομής Vyatta. Τέλος αναφέρονται τα πρωτόκολλα εξωτερικής δρομολόγησης (EGP) τα οποία όμως δεν αναλύονται περισσότερο.

Στο κεφάλαιο 6 γίνεται εισαγωγή στην ασφάλεια δικτύων και εξετάζουμε την ανάγκη ύπαρξης των firewall και των proxy servers. Εξετάζουμε γιατί χρησιμοποιούνται τα firewall, ποιες τεχνολογίες χρησιμοποιούνται σήμερα και βλέπουμε κάποιες αρχιτεκτονικές firewall. Αναλύουμε τους κανόνες που πρέπει να έχει ένα firewall, πόσο σημαντική είναι η διαχείριση τους αλλά και ο σχεδιασμός τους. Τέλος βλέπουμε ένα παράδειγμα για σχεδιασμό και τη ρύθμιση των κανόνων firewall για μια DMZ ζώνη: τι επιλογές έχουμε, ποια πακέτα πρέπει να απορρίπτουμε και ποια να επιτρέπουμε να περνάνε στο δίκτυο μας.

Στο κεφάλαιο 7, αν και δεν ήταν στα πλαίσια της πτυχιακής εργασίας αυτής, αναφερόμαστε στη τεχνολογία virtualization. Η αναφορά έγινε, διότι το δίκτυο που θα υλοποιήσουμε στο κεφάλαιο 10 , υλοποιείται με τη τεχνολογία αυτή σε έναν ESX server της VMware. Αυτό σημαίνει ότι είναι σημαντικό να γνωρίζουμε τη λειτουργία του και τα πλεονεκτήματα που μας προσφέρουν. Εξηγούμε τη λειτουργία του virtualization και ποια προβλήματα

αντιμετωπίζει. Η τεχνολογία αυτή είναι ένας σημαντικός παράγοντας για τις τεχνολογίες Cloud, SaaS (Software as a Service), PaaS(Platform as a Service) κτλ.

Στα κεφάλαια 8 και 9 βλέπουμε τα λογισμικά που θα χρησιμοποιήσουμε για το δρομολογητή και το firewall (pfsense και vyatta). Για το DNS το λογισμικό named μελετάται στο κεφάλαιο 2. Αναφέρονται οι λειτουργίες και οι δυνατότητες τους και οι εφαρμογές που χρησιμοποιούνται.

Στο κεφάλαιο 10 αναφερόμαστε στην υλοποίηση του δικτύου μας. Συγκεκριμένα βλέπουμε το περιβάλλον του ESX, τους δρομολογητές για το εσωτερικό μας δίκτυο (intranet), και το firewall μας σε αρχιτεκτονική three-way. Βλέπουμε τη λειτουργία της υλοποίησης μας σε επίπεδο δρομολόγησης μεταξύ των VLANs που δημιουργούμε, την ανάλυση και την επικοινωνία των hosts μέσω των ονομάτων τους, έτσι ώστε να γίνει επαλήθευση του εσωτερικού DNS μας και τη λειτουργία του firewall. Ο DNS χρησιμοποιείται και σαν DHCP server για τα VLANs και η ενημέρωση των εγγραφών του DNS (RR) γίνεται αυτόματα και δυναμικά από τον DHCP για κάθε IP που παίρνει ο κάθε host. Τέλος υλοποιούμε το firewall και αναλύουμε τις λειτουργίες του με βάση τη τοπολογία που σχεδιάζουμε. Ρυθμίζουμε κάποιους βασικούς και απαραίτητους κανόνες και δίνουμε μερικά παραδείγματα για κάποιες εφαρμογές που χρησιμοποιείται (VPN, Dual WAN, Proxy Server κτλ).

2 DOMAIN NAME SYSTEM

2.1 ΕΙΣΑΓΩΓΗ ΣΤΗΝ ΙΣΤΟΡΙΑ ΤΟΥ INTEPNET

Στα τέλη της δεκαετίας του 1960 το Department of Defense's Advanced Research Projects Agency, ARPA (αργότερα DARPA), άρχισε τη χρηματοδότηση του ARPAnet, ένα πειραματικό δίκτυο υπολογιστών που σύνδεε σημαντικούς οργανισμούς στις ΗΠΑ. Ο αρχικός στόχος του ARPAnet ήταν να επιτρέπει σε κυβερνητικούς αναδόχους να μοιράζονται ακριβούς ή σπάνιους υπολογιστικούς πόρους. Ωστόσο από την αρχή οι χρήστες του ARPAnet χρησιμοποιούσαν το δίκτυο για συνεργασία. Η συνεργασία αυτή κυμαινόταν από τη κοινή χρήση αρχείων και λογισμικού και να ανταλλάσουν μηνύματα μέσω ηλεκτρονικού ταχυδρομείου.

Η σουίτα TCP/IP αναπτύχτηκε στις αρχές του 1980 και πολύ σύντομα έγινε το πρωτόκολλο δικτύωσης του ARPAnet. Η συμπερίληψη του TCP/IP πρωτοκόλλου στο λειτουργικό σύστημα BSD UNIX έπαιξε σημαντικό ρόλο στον εκδημοκρατισμό της δικτύωσης. Αυτό σήμαινε ότι η διαδικτύωση και η συνδεσιμότητα στο ARPAnet ήταν ξαφνικά διαθέσιμο σε ακόμα περισσότερους οργανισμούς. Πολλοί από τους Η/Υ που συνδεόντουσαν στο ARPAnet ήταν συνδεδεμένοι και σε ένα τοπικό δίκτυο(LAN), και πολύ σύντομα και οι υπόλοιποι υπολογιστές του LAN δικτύου είχαν πρόσβαση στο ARPAnet.

Το δίκτυο μεγάλωσε από μερικούς χρήστες σε δεκάδες χιλιάδες χρήστες. Το αρχικό ARPAnet δίκτυο, έγινε η ραχοκοκαλιά της συνομοσπονδίας των τοπικών και περιφερειακών δικτύων βασισμένο στο TCP/IP, που ονομάστηκε INTEPNET.

Το 1988 ωστόσο, το DARPA αποφάσισε ότι το πείραμα τελείωσε. Το υπουργείο άμυνας, άρχισε τη διάλυση του ARPAnet. Ένα άλλο δίκτυο το NSFNET, που ιδρύθηκε από το NSF(National Science Foundation), αντικατέστησε το ARPAnet και έγινε η ραχοκοκαλιά του INTEPNET. Στα τέλη της δεκαετίας του '80 όλο και περισσότερες χώρες συνδεόντουσαν στο NSFNET(Καναδάς, Γαλλία, Σουηδία, Αυστραλία κ.α.). Χιλιάδες πανεπιστήμια και οργανισμοί δημιουργούν τα δικά τους δίκτυα και τα συνδέουν πάνω στο παγκόσμιο αυτό δίκτυο (Η Ελλάδα συνδέεται το 1990) το οποίο γίνεται γνωστό ως INTEPNET και να εξαπλώνεται με τρομερούς ρυθμούς σε ολόκληρο το κόσμο. Το 1990 το ARPAnet καταργείται.

2.2 Η ΙΣΤΟΡΙΑ ΤΟΥ DOMAIN NAME SYSTEM (DNS)

Κατά τη διάρκεια της δεκαετίας του '70, το ARPAnet ήταν μια πολύ μικρή και φιλική κοινότητα μερικών εκατοντάδων host. Ένα απλό αρχείο κειμένου HOSTS.TXT περιείχε αντιστοιχήσεις από ονόματα σε διευθύνσεις για κάθε host που ήταν συνδεδεμένος στο δίκτυο του ARPAnet. Ο γνωστός πίνακας hosts στο UNIX (/etc/hosts) δημιουργήθηκε από αυτό το αρχείο (κυρίως διαγράφοντας πεδία που το UNIX δεν τα χρησιμοποιούσε).

Το αρχείο HOSTS.TXT διατηρούνταν από το Network Information Center(NIC) του SRI. Οι διαχειριστές του ARPAnet συνήθως στέλνανε με mail τις αλλαγές στο NIC, και περιοδικά με το πρωτόκολλο FTP παίρνανε το τρέχων HOSTS.TXT αρχείο από το NIC. Οι αλλαγές τους μεταγλωττίζονταν σε ένα καινούργιο HOSTS.TXT αρχείο. Όσο το ARPAnet όμως μεγάλωνε αυτό ο τρόπος άρχισε να γίνεται αδύνατος. Το αρχείο μεγάλωνε πολύ σε μέγεθος και ακόμη αυτό δημιουργούσε περισσότερη κίνηση από την διαδικασία της ανανέωσης. Ένας χρήστης για παράδειγμα ήθελε μια γραμμή στο αρχείο HOSTS.TXT αλλά και ένας άλλος χρήστης να ενημερώνεται από το SRI-NIC.

[*] SRI = Stanford Research Institute, Menlo Park, California.

Τα προβλήματα στο HOSTS.TXT με την αύξηση των χρηστών στο ARPAnet δίκτυο, ήταν:

1. ***Κίνηση και Φορτίο***
   Το τίμημα στο SRI-NIC από την άποψη της κίνηση στο δίκτυο και το φορτίο του επεξεργαστή όσο διανέμουν το αρχείο, ήταν ανυπόφορο.
2. ***Συγκρούσεις ονομασίας***
   Δύο host στο HOSTS.TXT δεν μπορούσαν να έχουν το ίδιο όνομα
3. ***Συνοχή***
   Η διατήρηση της συνοχής του αρχείου όσο αυξανόταν το δίκτυο, γινόταν όλο και ποιο δύσκολη.

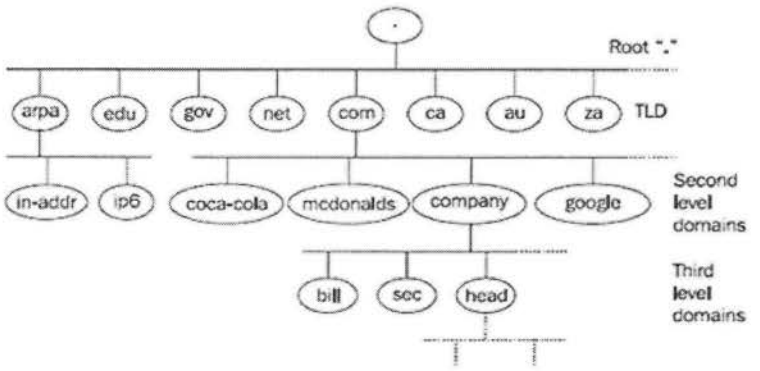
Μια λύση έπρεπε να βρεθεί στη οποία έπρεπε να φτιαχτεί ένα σύστημα το οποίο να επέτρεπε τη τοπική διαχείριση των δεδομένων και ταυτόχρονα να έκανε τα δεδομένα διαθέσιμα σε παγκόσμιο επίπεδο. Η αποκέντρωση της διαχείρισης θα εξέλειπε τη συμφόρηση που προκαλεί ένας μόνο χρήστης και θα απελευθέρωνε το πρόβλημα με τη μεγάλη κίνηση δικτύου (network traffic). Επίσης η τοπική διαχείριση μας βοηθά να κρατάμε τα δεδομένα ενημερωμένα πολύ ποιο εύκολα και με λιγότερο κόπο. Το νέο σύστημα θα έπρεπε να χρησιμοποιεί ιεραρχικό χώρο ονομάτων έτσι ώστε. Αυτό θα εξασφάλιζε την μοναδικότητα των ονομάτων.

Ο Paul Mockapetris από το Information Science Institute, ήταν υπεύθυνος στο να σχεδιάσει την αρχιτεκτονική του νέου συστήματος. Το 1984 κυκλοφόρησε το RFC 882 και 883, που περιέγραφε το Domain Name System (DNS). Τα δύο τελευταία αντικαταστήθηκαν από τα RFC 1034 και 1035 αντίστοιχα, τις τρέχων προδιαγραφές για το Domain Name System.

2.3 DOMAINS ΚΑΙ SUBDOMAINS

Ολόκληρο το Ιντερνετ διαιρείται σε **περιοχές (domains)**, για παράδειγμα ομάδες ονομάτων που λογικά ανήκουν μαζί. Τα domain προσδιορίζουν, αν τα ονόματα ανήκουν σε μια συγκεκριμένη εταιρία, χώρα, και ούτω καθεξής. Είναι πιθανό να δημιουργήσεις υπό ομάδες μέσα σε ένα domain και ονομάζονται **subdomains (υποτομείς)**. Για παράδειγμα είναι πιθανό να δημιουργήσουμε subdomain για κάθε τμήμα μιας εταιρίας. Το όνομα του domain αντανακλά τη συμμετοχή ενός host σε μια ομάδα και υποομάδα. Κάθε ομάδα έχει ένα όνομα συνδεδεμένο με αυτό. Το όνομα domain ενός host αποτελείται από τα μοναδικά

ονόματα ομάδων. Για παράδειγμα ο host με το όνομα bob.company.com αποτελείτε από έναν host Που ονομάζεται Bob που είναι μέσα σε ένα subdomain που ονομάζεται company και το οποίο είναι ένα subdomain του domain com. Το όνομα του domain αποτελείται από αλφαριθμητικά χωριζόμενα με τελείες. Το όνομα επεξεργάζεται από αριστερά στα δεξιά. Η υψηλότερη αρμόδια αρχή είναι το domain root και εκφράζεται με μια τελεία (.) στο τέλος δεξιά (συνήθως παραλείπεται). Τα **Top Level Domain (TLD)** ορίζονται στο root domain και έχουμε δύο ειδών TLD, Generic Top Level Domain (gTLD) και Country Code Top Level Domain (ccTLD). Πολύ γνωστά gTLD όπως θα δούμε παρακάτω είναι τα edu, com και mil τα οποία χρησιμοποιούνται κυρίως στις ΗΠΑ. Σύμφωνα με το ISO 3166, έχουμε ακόμη 2 γράμματα ccTLD για τις επιμέρους χώρες. Για παράδειγμα το gr είναι συνδεδεμένο με την Ελλάδα. Τα TLD domain χωρίζονται σε subdomains (υποπεριοχές) για συγκεκριμένους οργανισμούς, για παράδειγμα cocacola.com, google.com, mcdonalds.com. Επίσης η εταιρία Company Ltd για παράδειγμα μπορεί να έχει το subdomain της company.com αλλά και χαμηλότερα επίπεδα όπως για παράδειγμα το τμήμα πωλήσεων sales.comapny.com, τμήμα ασφάλειας το sec.company.com κ.α. Τα ονόματα φτιάχνουν μια δομή δέντρου όπως φαίνεται στην εικόνα 2-1.



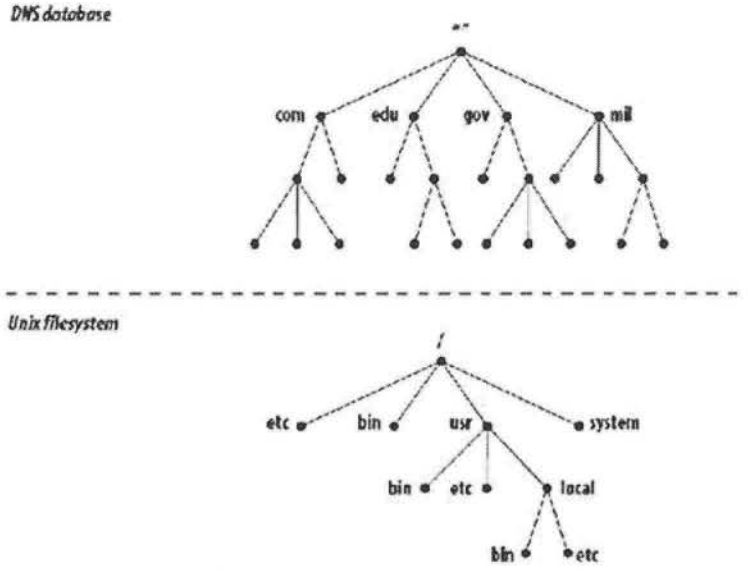
Εικόνα 2-1: Τα ονόματα σε ένα σύστημα DNS δημιουργούν μια δομή δέντρο

Η δομή της βάσης δεδομένων του DNS που φαίνεται στο εικόνα 2-2, είναι παρόμοιο με το σύστημα αρχείων ενός *NIX συστήματος. Ολόκληρη η βάση δεδομένων (ή αρχείο συστήματος) φαίνεται σαν ένα ανεστραμμένο δέντρο με το root κόμβο στη κορυφή. Κάθε κόμβος στο δέντρο έχει μια ετικέτα (κείμενο) που προσδιορίζει το κόμβο σε σχέση με το γονέα του (parent). Στο σύστημα DNS ο root συμβολίζεται με (.) ενώ στο Unix με (/).

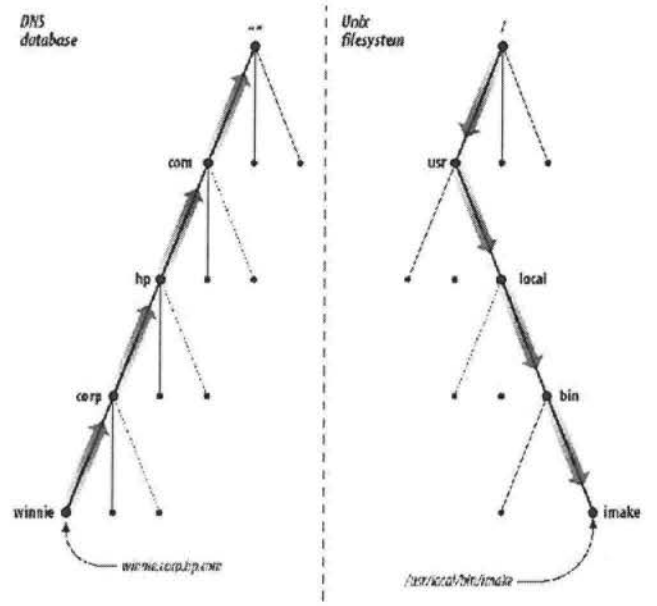
Κάθε domain έχει ένα μοναδικό όνομα όπως κάθε διαδρομή καταλόγου. Το όνομα ενός domain προσδιορίζει τη θέση του στη βάση δεδομένων, λίγο πολύ όπως το absolute pathname ενός καταλόγου προσδιορίζει τη θέση του στο σύστημα αρχείων. Στο DNS το όνομα του domain είναι η ακολουθία των κειμένων των ετικετών από το κόμβο root του συγκεκριμένου domain στο root όλου του δέντρου με τελείες να χωρίζουν τις ετικέτες κειμένου. Στο UNIX file system ισχύει το αντίθετο (αρχίζεις από το root "/" και φτάνεις στο κατάλογο που θέλεις) για τη σχετική διεύθυνση ενός καταλόγου (εικόνα 2-3). Στο DNS όπως είπαμε κάθε domain μπορεί να «σπάσει» σε άλλα subdomains και υπεύθυνος για αυτά τα subdomain μπορεί να ανατεθεί ένας τρίτος οργανισμός όπως θα δούμε παρακάτω.

Στο DNS, κάθε domain μπορεί να χωριστεί σε ένα αριθμό από subdomains, και για την ευθύνη αυτών των subdomains μπορούν να τη διαμοιράσουμε σε διαφορετικούς

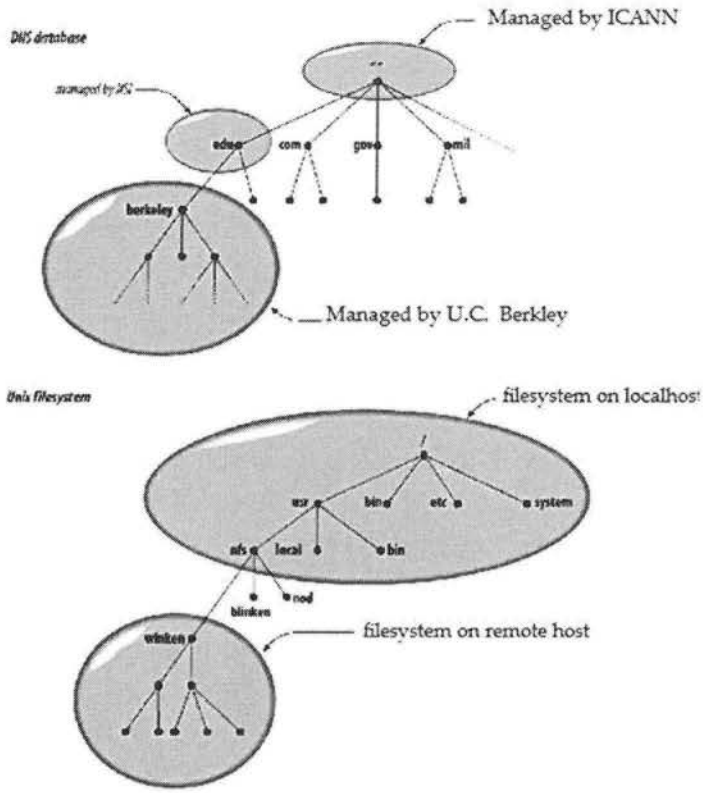
οργανισμούς. Για παράδειγμα ένας οργανισμός με το όνομα EDUCAUSE διαχειρίζεται το edu (educational) domain αλλά μεταβιβάζει την ευθύνη για το berkley.edu domain στο U.C. Berkley (εικόνα 2-4). Αυτό είναι παρόμοιο με το απομακρυσμένο mounting ενός συστήματος αρχείων: ορισμένοι κατάλογοι σε ένα σύστημα αρχείων μπορεί στη πραγματικότητα να είναι σύστημα αρχείων σε άλλους υπολογιστές που έχουν τοποθετηθεί από απομακρυσμένους υπολογιστές. Ο διαχειριστής του host winken για παράδειγμα (εικόνα 2-4), είναι υπεύθυνος για το σύστημα αρχείων που εμφανίζεται στον τοπικό υπολογιστή ως το κατάλογο /usr/nfs/winken.



Εικόνα 2-2: Η βάση δεδομένων του DNS και το σύστημα αρχείων UNIX

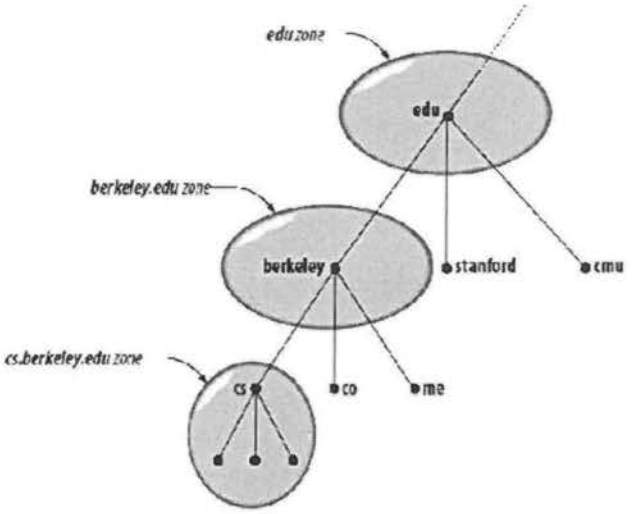


Εικόνα 2-3: Διάβασμα Ονομάτων από Βάση Δεδομένων DNS και σύστημα αρχείων UNIX



Εικόνα 2-4: Απομακρυσμένη διαχείριση subdomain και συστήματος αρχείων

Αναθέτοντας την ευθύνη για το Berkley.edu στον οργανισμό U.C Berkley δημιουργείται μια καινούργια ζώνη, ένα αυτόνομο διαχειριστικό κομμάτι του namespace (χώρος ονομάτων). Η ζώνη Berkley.edu είναι τώρα ανεξάρτητη από το edu domain και περιέχει πλέον όλα τα domain ονόματα που τελειώνουν σε Berkley.edu. Η ζώνη edu από την άλλη μεριά περιέχει μόνο τα domain ονόματα που τελειώνουν σε edu αλλά δεν είναι σε ανατεθειμένες (delegated) ζώνες όπως το berkley.edu. Το berkley.edu μπορεί να διαιρεθεί σε άλλες υποπεριοχές (subdomains) όπως το cs.berkley.edu και μερικά από αυτά τα subdomain μπορεί να είναι ξεχωριστές ζώνες από μόνα τους, αν και μόνο αν οι διαχειριστές του Berkley.edu αναθέσουν την ευθύνη για αυτές τις ζώνες σε άλλους οργανισμούς. Αν το cs.berkley.edu είναι μια ξεχωριστή ζώνη, η ζώνη Berkley.edu δεν περιέχει πλέον ονόματα domain που τελειώνουν σε cs.berkley.edu (εικόνα 2-5).

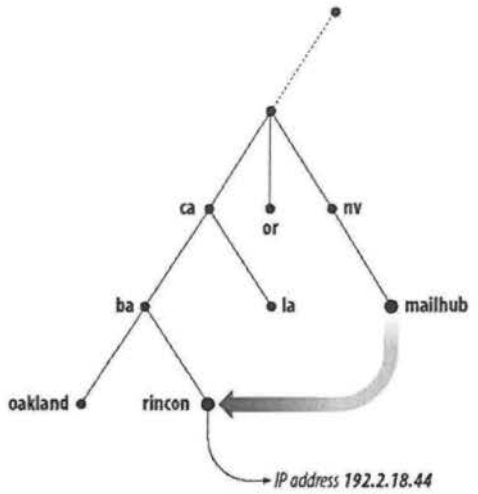


Εικόνα 2-5: Οι edu, berkley.edu και cs.berkley.edu ζώνες

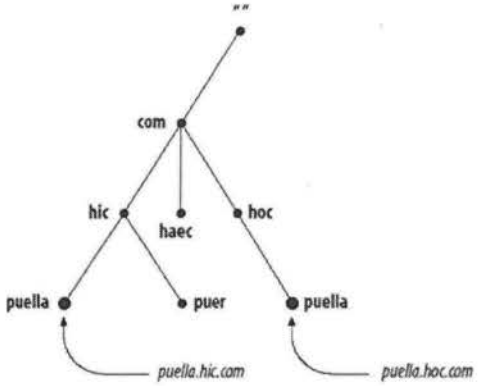
Τα ονόματα domain χρησιμοποιούνται σαν ευρετήριο στη βάση δεδομένων του DNS. Τα domain μπορούν να περιέχουν και subdomain αλλά και hosts. Ένα domain περιέχει όλους τους host και subdomains των οποίων τα domain ονόματα είναι εντός του υποδέντρου του domain του χώρου ονομάτων (namespace).

Κάθε host στο δίκτυο έχει ένα όνομα domain, που δείχνει σε πληροφορίες που σχετίζονται με τον host (εικόνα 2-6). Αυτές οι πληροφορίες μπορεί να περιέχουν IP διεύθυνση, πληροφορίες σχετικά με τη δρομολόγηση του mail κ.α. Οι host μπορούν επίσης να έχουν ένα οι περισσότερα ψευδώνυμα domain (domain name aliases), που απλά είναι δείκτες από το ένα domain (ψευδώνυμο - alias) σε ένα άλλο (στο επίσημο ή κανονικό domain). Στην εικόνα 2-6, το mailhub.tv, είναι ένα alias (ψευδώνυμο) για το επίσημο όνομα Rincon.ba.ca.

Όλη αυτή η πολύπλοκη δομή γίνεται για να λύσουμε το πρόβλημα που είχε το HOSTS.TXT του ARPAnet. Για παράδειγμα κάνοντας τα ονόματα ιεραρχικά εξαλείφει το πρόβλημα που είχαμε με τη σύγκρουση ονομάτων. Κάθε domain έχει ένα μοναδικό όνομα, έτσι ο οργανισμός που τρέχει το συγκεκριμένο domain είναι ελεύθερος να ονομάσει τους host και τα subdomain όπως θέλει μέσα στα πλαίσια της περιοχής του. Ότι όνομα διαλέξει για κάποιο host ή subdomain δεν θα έρχεται σε σύγκρουση με ονόματα domain άλλων οργανισμών γιατί θα τελειώνει στο μοναδικό τους όνομα domain. Για παράδειγμα ο οργανισμός που έχει το hic.com μπορεί να δημιουργήσει ένα host puella (εικόνα 2-7) επειδή ξέρει ότι θα είναι μοναδικό, αφού το domain όνομα του host θα τελειώνει σε hic.com, το οποίο είναι μοναδικό όνομα domain.



Εικόνα 2-6: Ένα ψευδώνυμο στο DNS που δείχνει στο επίσημο όνομα



Εικόνα 2-7: Λύνοντας το πρόβλημα της σύγκρουσης ονομάτων

2.4 ΠΡΕΠΕΙ ΝΑ ΧΡΗΣΙΜΟΠΟΙΗΣΩ DNS;

Αν και το DNS μας λύνει τα χέρια σε πολλά προβλήματα, υπάρχουν περιπτώσεις που δεν αξίζει να το χρησιμοποιήσουμε. Εκτός από το DNS υπάρχουν και άλλοι μηχανισμοί μερικοί από τους οποίους μπορεί να είναι μέρος του λειτουργικού συστήματος μας. Μερικές φορές η επιβάρυνση που προκαλείται στη διαχείριση των ζωνών και των διακομιστών ονομάτων (name servers) δεν αξίζει. Από την άλλη μεριά, υπάρχουν περιπτώσεις όπου δεν έχεις άλλη λύση από το να εγκαταστήσεις και να διαχειριστείς διακομιστές ονομάτων. Γενικά:

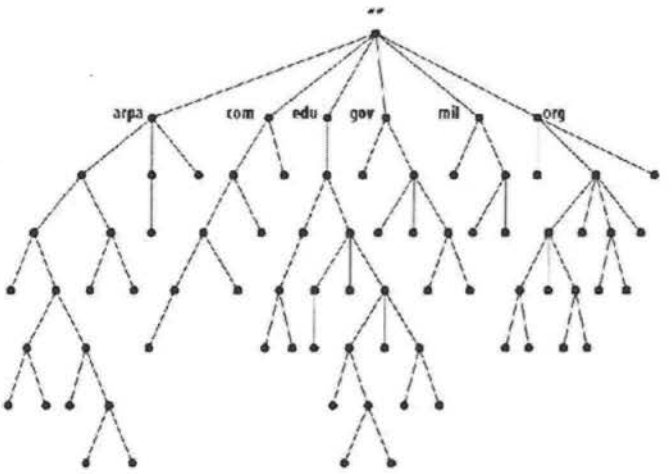
- **Αν είμαστε συνδεδεμένοι στο Ιντερνετ**
  Το DNS είναι απαραίτητο, αφού όλες οι διαδικτυακές εφαρμογές και υπηρεσίες χρησιμοποιούν το DNS, όπως τα Πρωτόκολλα WEB, ηλεκτρονικό ταχυδρομείο, Απομακρυσμένη τερματική πρόσβαση, μεταφορά αρχείων κ.α. Από την άλλη μεριά αυτό δεν σημαίνει ότι πρέπει να εγκαταστήσουμε και να διαχειριστούμε μόνοι μας ζώνες. Αν έχουμε πολλούς host μπορούμε να ενωθούμε με μια υπάρχουσα ζώνη ή να βρούμε κάποιον άλλον να μας φιλοξενήσει της ζώνες μας ή ακόμα και ο ISP (Internet Service Provider – Πάροχος internet) μας. Σε περίπτωση που έχουμε

αρκετούς host όμως, πιθανόν να χρειαστούμε τη δικιά μας ζώνη και αν θέλουμε άμεσο έλεγχο στη ζώνη μας και στους διακομιστές ονομάτων, θα πρέπει να μάθουμε να τις διαχειριζόμαστε μόνοι μας.

- **Αν έχεις το δικό σου τοπικό δίκτυο ή Ιστοσελίδα**
  Και αυτό το δίκτυο δεν είναι συνδεδεμένο σε μεγαλύτερα δίκτυα, πιθανόν να μην σου χρειάζεται το DNS. Θα μπορούσες να χρησιμοποιήσεις το WINS (Windows Internet Name Service) ή το NIS (Sun's Network Information Service). Αλλά αν χρειαζόμαστε μια κατανεμημένη διαχείριση ή έχουμε πρόβλημα διατηρώντας τη συνοχή των δεδομένων του δικτύου μας, το DNS είναι η λύση. Αν το δίκτυο μας θα συνδεθεί με άλλα δίκτυα θα ήταν σοφό να έχουμε τις δικές μας ζώνες.

2.5 DOMAIN NAMESPACE

Η βάση δεδομένων του DNS περιέχει δείκτες σε domain ονόματα. Κάθε domain είναι ένα μονοπάτι σε ένα μεγάλο ανεστραμμένο δέντρο, που ονομάζεται domain namespace (Χώρος ονομάτων domain). Το βάθος του δέντρου περιορίζεται σε 127 επίπεδα (ένας περιορισμός που δύσκολα μπορούμε να φτάσουμε).

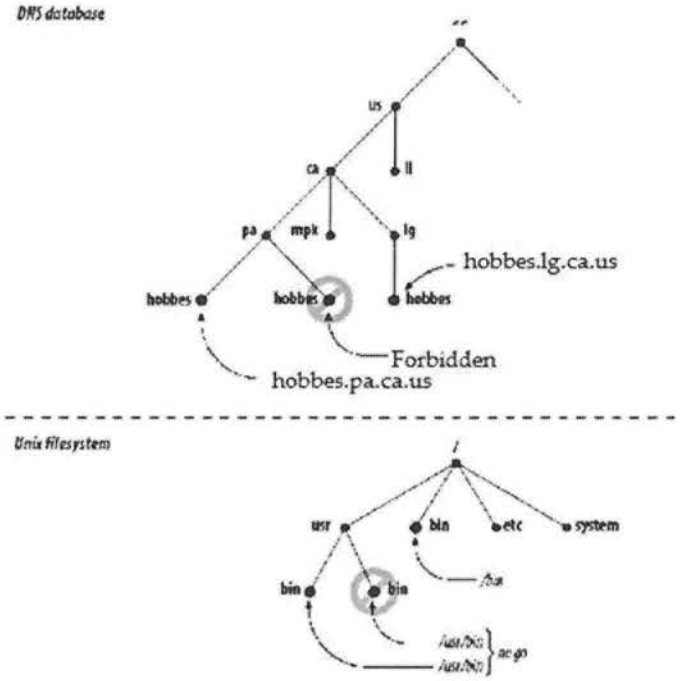


Εικόνα 2-8: Δομή του DNS namespace

2.5.1 ONOMATA DOMAIN

Κάθε κόμβος σε ένα δέντρο έχει μια ετικέτα (χωρίς τελεία) η οποία μπορεί να είναι μέχρι 63 χαρακτήρες. Ένας χαρακτήρας τύπου null είναι δεσμευμένος για το root κόμβο. Το πλήρες όνομα του κάθε κόμβου στο δέντρο είναι ο συνδυασμός των ετικετών στο μονοπάτι από το κόμβο μέχρι το root (""). Τα domain ονόματα διαβάζονται πάντα από το κόμβο προς την αρχή του δέντρου (root), με τελείες να διαχωρίζουν τα ονόματα στο μονοπάτι.

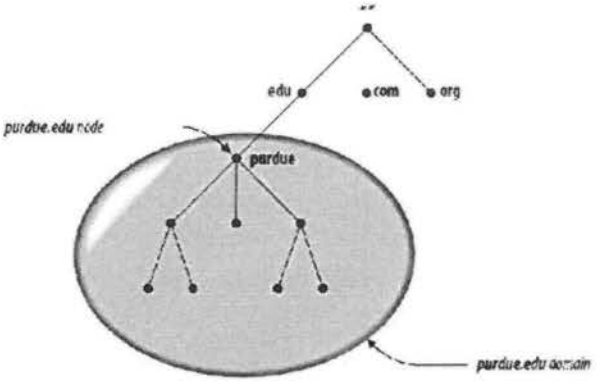
Το DNS απαιτεί τους κόμβους αδέρφια, οι κόμβοι που είναι παιδιά του ίδιου γονέα, να έχουν διαφορετικές ετικέτες. Ο περιορισμός αυτός εγγυάται ότι ένα όνομα domain προσδιορίζει μοναδικά ένα μόνο κόμβο στο δέντρο. Ο περιορισμός δεν αποτελεί πρόβλημα, επειδή οι επιγραφές (ετικέτες) πρέπει να είναι μοναδικές μόνο μεταξύ των παιδιών, και όχι μεταξύ όλων των κόμβων στο δέντρο. Ο ίδιος περιορισμός ισχύει και στο σύστημα αρχείων του κάθε λειτουργικού συστήματος: Δεν μπορείς να δώσεις σε δυο καταλόγους αδέρφια ή σε δύο αρχεία που βρίσκονται στον ίδιο κατάλογο το ίδιο όνομα. Όπως απεικονίζεται στην εικόνα 2-9, δεν μπορείς να έχεις δύο hobbes.pa.ca.us κόμβους στο χώρο ονομάτων (namespace) όπως δεν μπορείς να έχεις δύο /usr/bin καταλόγους. Μπορείς ωστόσο να έχεις έναν hobbes.pa.ca.us κόμβο και έναν hobbes.lg.ca.us κόμβο όπως μπορείς να έχεις ένα /bin κατάλογο και ένα /usr/bin κατάλογο.



Εικόνα 2-9: Εξασφαλίζοντας μοναδικότητα στα domain και στο σύστημα αρχείων του Unix

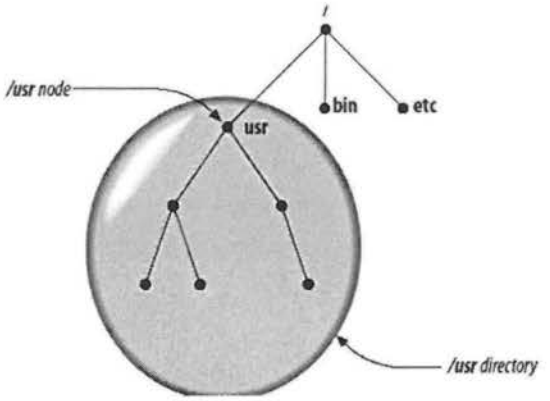
2.5.2 DOMAINS

Ένα domain είναι απλά ένα υποδέντρο του χώρου ονομάτων domain (domain namespace). Το όνομα ενός domain είναι ίδιο με το όνομα του domain του κόμβου στη κορυφή της περιοχής. Έτσι για παράδειγμα η κορυφή του purdue.edu domain είναι ένας κόμβος που ονομάζεται purdue.edu όπως φαίνεται στην εικόνα 2-10.



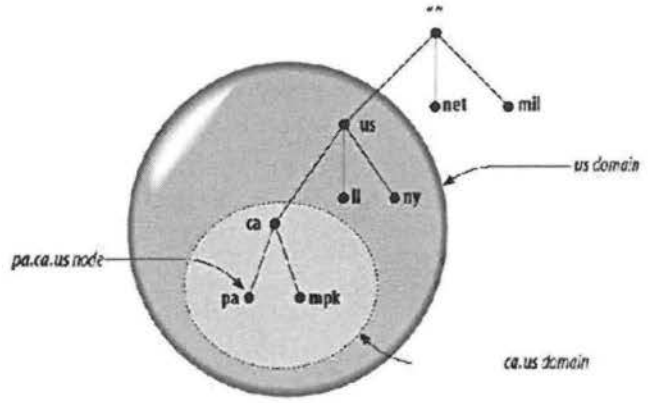
Εικόνα 2-10: Το purdue.edu domain

Ομοίως όπως σε ένα κατάλογο σε ένα σύστημα αρχείων, στη κορυφή του καταλόγου /usr, έχουμε ένα κόμβο που ονομάζεται /usr όπως φαίνεται στην εικόνα 2-11



Εικόνα 2-11: Ο κατάλογος /usr

Το όνομα ενός domain σε ένα υποδέντρο θεωρείται μέρος του domain. Επειδή το όνομα ενός domain μπορεί να είναι σε πολλά υποδέντρα, το όνομα ενός domain μπορεί επίσης να βρίσκεται σε πολλά domains. Για παράδειγμα το domain pa.ca.us είναι μέρος του ca.us domain και επίσης μέρος του us domain όπως φαίνεται στην εικόνα 2-12.



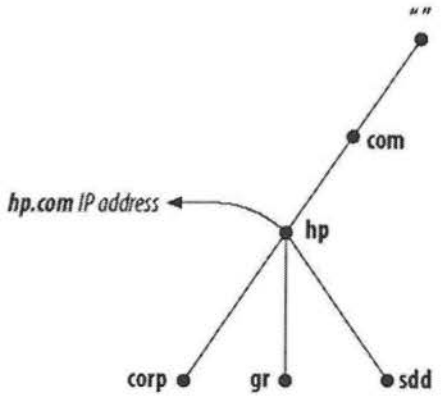
Εικόνα 2-12: Ένας κόμβος σε πολλαπλά domain

Άρα ένα domain είναι απλά ένα υποδέντρο του χώρου ονομάτων του domain (domain namespace). Αλλά αν ένα domain έχει φτιαχτεί απλά από domain ονόματα και άλλα domains, το ερώτημα είναι που βρίσκονται οι hosts;

Οι hosts βρίσκονται εκεί και αντιπροσωπεύονται από domain ονόματα. Είπαμε ότι τα domain ονόματα είναι απλά δείκτες στη βάση δεδομένων του DNS. Οι hosts είναι τα domain ονόματα που δείχνουν στη πληροφορία σχετικά με το κάθε μεμονωμένο host και το domain περιέχει όλους τους hosts όπου τα domain ονόματα τους βρίσκονται μέσα στο domain. Οι hosts μεταξύ τους σχετίζονται λογικά, συνήθως με βάση τη γεωγραφία που βρίσκονται ή την οργανωτική υπαγωγή τους και όχι απαραίτητα με βάση το δίκτυο στο οποίο ανήκουν. Θα μπορούσαμε να έχουμε 10 διαφορετικούς host και ο καθένας τους να είναι σε διαφορετικό δίκτυο αλλά και σε διαφορετικές χώρες, και όλοι μεταξύ τους στο ίδιο domain.

Τα domain ονόματα μπορούν να δείχνουν σε διευθύνσεις δικτύου, πληροφορίες υλικού και πληροφορίες δρομολόγησης e-mail. Τα domain ονόματα στο εσωτερικό του δέντρου μπορούν να ονομάσουν και να δείχνουν σε πληροφορίες σχετικές με το domain. Τα εσωτερικά domain ονόματα μπορούν να αντιπροσωπεύουν το domain που αντιπροσωπεύουν αλλά και ειδικότερα ένα host στο δίκτυο. Για παράδειγμα, η hp.com είναι ταυτόχρονα το όνομα του domain της εταιρίας Hewlett-Packard και το domain όνομα που αναφέρεται στο host που τρέχει το κύριο διακομιστή ιστού (web server) της HP.

Το τι είδους πληροφορία θα πάρεις όταν χρησιμοποιείς ένα domain όνομα εξαρτάται από την εφαρμογή που χρησιμοποιείς. Στέλνοντας ένα e-mail στην hp.com παίρνουμε πληροφορίες για τη δρομολόγηση του mail ενώ αν θέλουμε να ανοίξουμε μια σύνδεση με το SSH (Secure Shell) παίρνουμε πληροφορίες για το host (στην εικόνα 2-13 για παράδειγμα παίρνουμε την IP της hp.com)

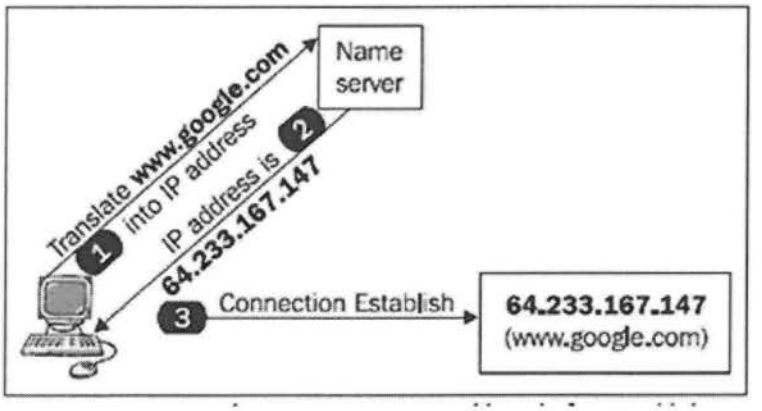


Εικόνα 2-13: Εσωτερικός κόμβος με δεδομένα για το Host αλλά και δεδομένα για το domain

Ένας απλός τρόπος για να καταλάβουμε αν ένα domain είναι subdomain ενός άλλου domain είναι αν συγκρίνουμε τα ονόματα τους. Το domain όνομα ενός subdomain τελειώνει με το domain όνομα του γονέα του. Για παράδειγμα το domain la.tyrell.com πρέπει να είναι subdomain του tyrell.com domain, επειδή το la.tyrell.com τελειώνει με tyrell.com. Επίσης είναι και subdomain του com όπως και ο tyrell.com.

2.6 INTERNET DOMAIN NAMESPACE

Όλες οι εφαρμογές που παρέχουν επικοινωνία ανάμεσα σε υπολογιστές στο INTEPNET χρησιμοποιούν διευθύνσεις IP για να προσδιορίσουν την επικοινωνία μεταξύ των hosts. Αφού οι IP διευθύνσεις είναι δύσκολο για τους ανθρώπους να τις θυμούνται, χρησιμοποιούμε το όνομα μιας δικτυακής διεπαφής και όχι IP. Για κάθε IP διεύθυνση υπάρχει ένα όνομα μιας δικτυακής επαφής και για να είμαστε ποιο σαφείς ένα domain όνομα. Αυτό το domain όνομα μπορεί να χρησιμοποιηθεί σε οποιαδήποτε εντολή που χρειάζεται IP. Για παράδειγμα ping my.work.bob. Εξαίρεση αποτελεί η προδιαγραφή ενός name server. Μια IP μπορεί να αντιστοιχίζεται σε πολλά domains. Η σχέση μεταξύ της IP και του ονόματος ενός HY καθορίζεται στη βάση δεδομένων του Domain Name System (DNS). Η βάση δεδομένων του DNS διανέμεται παντού σε όλο το κόσμο. Η βάση δεδομένων όπως περιέχει αρχεία που ονομάζονται **Resource Records(RR)**. Τα μεμονωμένα τμήματα που είναι οι ζώνες τοποθετούνται σε συγκεκριμένους name servers. Το DNS είναι μια παγκόσμια διανυμένη βάση δεδομένων. Αν θέλουμε να πλοηγηθούμε στο παγκόσμιο ιστό, γράφουμε για παράδειγμα στον browser μας www.google.com με IP 64.233.167.147 (Εικόνα 2-14). Πριν την σύνδεση μας με το διακομιστή ιστού www.google.com, το www.google.com DNS όνομα μεταφράζεται σε μια IP διεύθυνση και μόνο τότε εγκαθίσταται η σύνδεση. Ο name server (διακομιστής ονόματος) είναι απλά μια εξειδικευμένη βάση δεδομένων που μεταφράζει ονόματα σε IP διευθύνσεις και το αντίστροφο (reverse DNS). Οι name servers κάνουν το δίκτυο ποιο εύκολα διαχωρίσιμο ποιο δυναμικό και να ανταποκρίνεται πολύ γρήγορα σε αλλαγές. Ωστόσο η λύση αυτή δημιουργεί και προβλήματα. Αν ο name server μας δεν είναι διαθέσιμος, τότε ο host μας δεν μπορεί να έχει πρόσβαση σε κανένα πόρο στο δίκτυο. Η λύση στο πρόβλημα αυτό έφεραν ο πρωτεύων και δευτερεύων name server.



Εικόνα 2-14: Είναι απαραίτητο να μεταφραστεί το όνομα σε μια IP διεύθυνση πριν δημιουργηθεί η σύνδεση

Αν ο πρωτεύον name server δεν ανταποκριθεί σε ένα αίτημα, ο host θα ξαναπροσπαθήσει χρησιμοποιώντας το δευτερεύων name server. Τόσο κρίσιμο είναι το θέμα με τους name servers που είναι πολύ συχνό να βλέπουμε δύο τρείς ή και περισσότερους name servers. Παρόλα αυτά όσο αυξάνει το δίκτυο μας, και δίνουμε καινούργια ονόματα για καταχωρήσεις στους name servers μας, δημιουργούνται τρία νέα προβλήματα:

- **Οργάνωση:** Για να βρούμε μια καταχώρηση στη βάση δεδομένων, γίνεται όλο και ποιο αργό αφού αναζητούμε ανάμεσα σε εκατομμύρια καταχρήσεις. Χρειαζόμαστε μια μέθοδο ευρετηρίου για να οργανώσουμε τις καταχωρήσεις μας.
- **Επεκτασιμότητα:** Αν κάθε host έχει πρόσβαση στο name server μας, ο φόρτος γίνεται πολύ μεγάλος. Χρειαζόμαστε μια μέθοδο για να μοιράσουμε το φόρτο σε πολλούς name servers.
- **Διαχείριση:** Με πολλές καταχωρήσεις στη βάση δεδομένων μας το πρόβλημα της διαχείρισης αυξάνεται, όσο πολλαπλοί διαχειριστές προσπαθούν να ενημερώσουν τη βάση δεδομένων ταυτόχρονα. Χρειαζόμαστε μια μέθοδο, όπως είδαμε προηγουμένως να ξεχωρίσουμε (delegating – ανάθεση) τη διαχείριση αυτών των ονομάτων.

Η ανάγκη να λύσουμε όλα αυτά τα προβλήματα οδήγησε στην δημιουργία και στην εξέλιξη του Internet Domain Name Systems (DNS).

2.6.1 TOP LEVEL DOMAINS

Τα αρχικά τα top-level domains διαιρούσαν το χώρο ονομάτων του INTEPNET οργανωτικά σε εφτά domains:

- ***Com***: Εμπορικοί οργανισμοί, όπως την Hewlett-Packard (hp.com), Sun Microsystems (sun.com) και IBM (ibm.com)
- ***Edu***: Εκπαιδευτικοί οργανισμοί όπως η U.C. Berkley (Berkley.edu)
- ***Gov***: Κυβερνητικοί οργανισμοί, όπως η NASA (nasa.gov)
- ***Mil***: Στρατιωτικοί οργανισμοί, όπως η U.S. Army (armu.mil)
- ***Net:*** Πρώην οργανισμοί που παρέχουν δικτυακή υποδομή, όπως το NSFNEY (nsf.net). Από το 1996 έχει ανοίξει για κάθε εμπορικό οργανισμό όπως η com.
- ***Org:*** Πρώην μη εμπορικοί οργανισμοί. Όπως το net domain οι περιορισμοί καταργήθηκαν το 1996.
- ***Int:*** Διεθνής οργανισμοί όπως το NATO (nato.int)

Ένα άλλο ανώτατου επιπέδου domain (top-level domain) που ονομάζεται arpa είχε αρχικά χρησιμοποιηθεί στη μετάβαση του ARPAnet από τους πίνακες των host στο DNS. Αρχικά όλοι οι host του ARPAnet είχαν ονόματα κάτω από το ARPA domain. Αργότερα μετακινήθηκαν σε ένα από τα παραπάνω domain (com, net κτλ). Παρόλα αυτά το ARPAnet χρησιμοποιείται ακόμα σήμερα όπως θα δούμε παρακάτω. Σήμερα όπως είδαμε τα 7 αυτά domain ονομάζονται generic top-level domains ή gTLD.

2.6.1.1 COUNTRY CODE TOP-LEVEL DOMAINS

Προκειμένου να αντιμετωπισθεί η αυξανόμενη διεθνοποίηση του Internet, οι φορείς υλοποίησης των ονομάτων του Internet αντί να επιμένουν ότι όλοι οι τομείς ανώτατου επιπέδου περιγράφουν τους οργανωτικούς οργανισμούς, αποφάσισαν να επιτρέψουν και γεωγραφικές ονομασίες. Νέα top-level domains είχαν κρατηθεί (αλλά όχι κατ 'ανάγκη δημιουργηθεί) για να αντιστοιχούν σε μεμονωμένες χώρες. Τα ονόματα αυτά ακολούθησαν

ένα υπάρχον διεθνές πρότυπο το ISO 3166. Το ISO 3166 καθορίζει επίσημα, δύο γράμματα συντομεύσεις για όλες τις χώρες του κόσμου [1].

2.6.1.2 NEW TOP LEVEL DOMAINS

Στα τέλη του 2000, ο οργανισμός που διαχειρίζεται το DNS, ο Internet Corporation for Assigned Names and Numbers (ICANN), δημιούργησε άλλα 7 νέα gTLD για να φιλοξενήσει τη ταχεία επέκταση του Ιντερνετ και για την ανάγκη για περισσότερο χώρο για domains.

Παρακάτω δίνονται τα νέα gTLDs:

- Aero, Για την αεροναυτική βιομηχανία
- Biz, Γενικό
- Coop, Για συνεταιρισμούς
- Info, Γενικό
- Museum, Για μουσεία
- Name, Γενικό, για ανεξάρτητους
- Pro, γενικό, για επαγγελματίες

2.6.1.3 RESOURCE RECORDS

Τα δεδομένα που συνδέονται με τα ονόματα των domain περιέχονται σε εγγραφές πόρων (Resource Records - RR). Οι εγγραφές διαιρούνται σε κλάσεις, καθένα από τα οποία σχετίζεται με ένα είδος δικτύου ή λογισμικού. Επί του παρόντος υπάρχουν κλάσεις για TCP/IP δίκτυα (κλάσεις του σημερινού INTERNET), δίκτυα που βασίζονται στο Chaosnet [2] πρωτόκολλο και δίκτυα που χρησιμοποιούν το Hesiod λογισμικό. Οι κλάσεις του Ιντερνετ είναι τα ποιο δημοφιλή.

Μέσα σε μια κλάση, οι εγγραφές είναι σε πολλά είδη, τα οποία αντιστοιχούν σε διάφορα είδη δεδομένων που αποθηκεύονται στο domain namespace. Διαφορετικές κλάσεις μπορούν να καθορίσουν διαφορετικές εγγραφές, αν και ορισμένα είδη είναι κοινά σε περισσότερες από μια κλάση. Για παράδειγμα, κάθε κλάση καθορίζει ένα τύπο διεύθυνσης. Κάθε είδος εγγραφής σε μια συγκεκριμένη κλάση ορίζει μια συγκεκριμένη σύνταξη εγγραφής για το οποίο όλες οι εγγραφές των πόρων αυτής της κλάσης και τύπου πρέπει να τηρούν.

[1] Εκτός από την Μεγάλη Βρετανία. Σύμφωνα με το ISO 3166 και παράδοση του Διαδικτύου, το top-level domain όνομα της Μεγάλης Βρετανίας θα πρέπει να είναι gb. Αντ 'αυτού, οι περισσότερες οργανώσεις της Μεγάλης Βρετανίας και της Βόρειας Ιρλανδίας (δηλαδή, το Ηνωμένο Βασίλειο) χρησιμοποιούν το top-level domain name uk.

[2]Το Chaosnet είναι ένα παλιό δίκτυο με ιστορική σημασία πια. Δεν ξέρουμε στα σίγουρα αν κάποιος πια χρησιμοποιεί κλάσεις Chaosnet και η χρησιμοποίηση της Hedoid κλάσης είναι κυρίως περιορισμένο στο MIT.

2.6.2 ΑΝΑΓΝΩΣΗ DOMAIN NAMES

Παρακάτω θα δούμε μερικά παραδείγματα για το πως διαβάζονται τα domain ονόματα.

- ***Lithium.cchem.berkley.edu***
  Το Berkley.edu είναι το domain του U.C. Berkley, cchem είναι το subdomain του Berkley.edu και το lithium είναι το όνομα ενός συγκεκριμένου host στο domain αυτό και πιθανότατα είναι ένα από τα εκατοντάδες host που μπορεί να έχουν.
- ***Winnie.corp.hp.com***
  Το hp.com domain είδαμε ότι ανήκει στην Hewlett-Packard. Το corp subdomain είναι σίγουρα τα κεντρικά της γραφεία. Και το Winnie είναι απλά ένα όνομα για κάποιο host που έβαλε κάποιος.
- ***Daphne.ch.apollo.hp.com***
  Apollo.hp.com είναι το πρώην Apollo Computer subdomain του hp.com domain. Το ch.apollo.hp.com(ch = Chelmsford) και το daphne είναι ένας host στο Chelmsford.

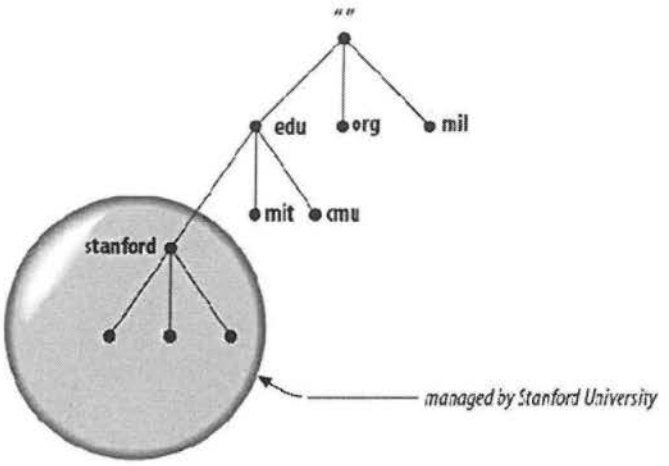
2.7 ΑΝΑΘΕΣΗ – DELEGATION

Ένας από τους στόχους όταν σχεδιάστηκε το DNS ήταν όπως είπαμε να αποκεντρώσουμε τη διαχείριση. Αυτό γίνεται μέσω της ανάθεσης (delegation). Η ανάθεση στα domain δουλεύει όπως η ανάθεση στη δουλειά μας. Ο μάνατζερ του έργου σπάει το έργο σε μικρά κομμάτια και αναθέτει σε κάθε υπάλληλο μια εργασία. Ο κάθε υπάλληλος πλέον είναι υπεύθυνος για το δικό του κομμάτι εργασίας που του έχει ανατεθεί.

Ομοίως, ένας οργανισμός που διαχειρίζεται ένα domain μπορεί να το διαιρέσει σε subdomains, όπου κάθε subdomain να ανατεθεί σε διαφορετικούς οργανισμούς, που σημαίνει ότι ο κάθε οργανισμός γίνεται υπεύθυνος στο να διατηρήσει όλα τα δεδομένα στο αντίστοιχο δικό της subdomain. Ο κάθε οργανισμός μπορεί ποια ελευθέρα να αλλάξει τα δεδομένα αλλά ακόμα και να διαιρέσει το subdomain του σε άλλα subdomains και να τα αναθέσει σε τρίτους οργανισμούς. Ο domain γονέας διατηρεί μόνο δείκτες σε πηγές δεδομένων του subdomain, έτσι ώστε να μπορεί να κάνει ερωτήματα σε αυτά. Το domain Stanford.edu για παράδειγμα, έχει ανατεθεί στα άτομα του πανεπιστημίου στο Stanford και είναι τα άτομα εκείνα που τρέχουν και διαχειρίζονται το δίκτυο του πανεπιστημίου (Εικόνα 2-15).Επομένως με τον όρο ανάθεση (delegation) αναφερόμαστε στο να αντιστοιχίσουμε ευθύνη για ένα subdomain σε έναν άλλον οργανισμό.

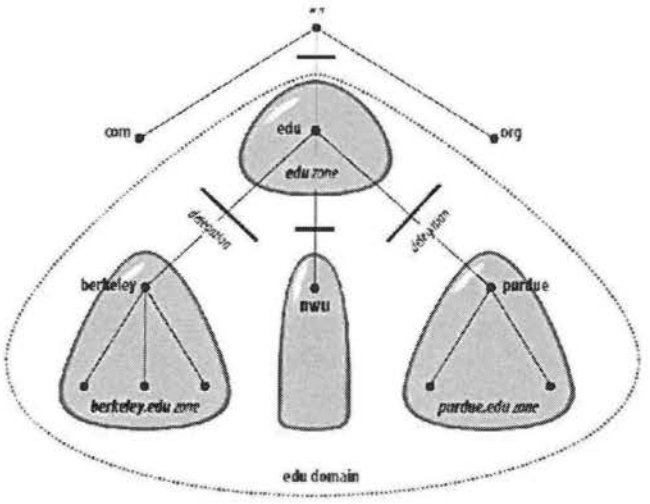
2.8 NAME SERVERS ΚΑΙ ΖΩΝΕΣ

Τα προγράμματα που κρατάνε πληροφορίες για το domain namespace ονομάζονται name servers. Οι name servers γενικά έχουν όλη τη πληροφορία σχετικά για κάποιο κομμάτι του domain namespace(ζώνη), που φορτώνουν από ένα αρχείο ή από άλλον name server. Ο name server λέμε τότε ότι είναι αρμόδιος για τη συγκεκριμένη ζώνη. Οι name servers μπορούν να είναι αρμόδιοι ακόμα και για πολλές ζώνες.



Εικόνα 2-15: Η περιοχή stanford.edu έχει ανατεθεί στο Πανεπιστήμιο του Stanford

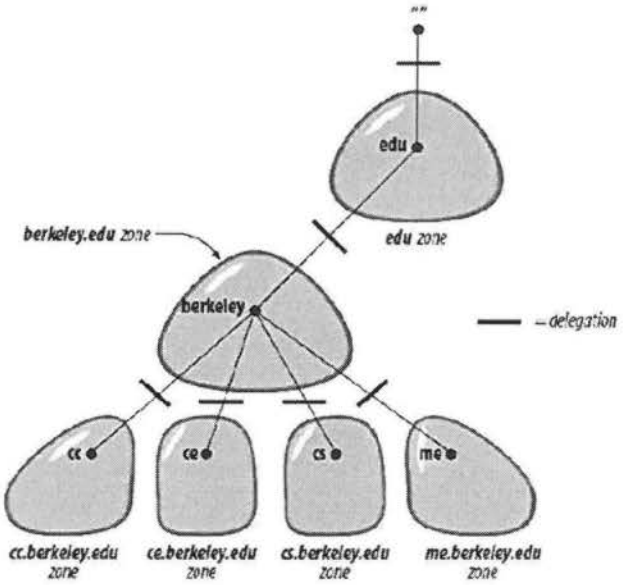
Η διαφορά ανάμεσα σε μια ζώνη και σε ένα domain είναι σημαντική αλλά πολύ λεπτή. Όλα τα top-level domain αλλά και δεύτερου και μεγαλύτερου επιπέδου, όπως το Berkley.edu και hp.com, σπάνε όπως είδαμε σε μικρότερα, περισσότερο ποιο διαχειριστικά κομμάτια με τη διαδικασία της ανάθεσης. Αυτά τα κομμάτια ονομάζονται ζώνες. Το domain edu όπως φαίνεται στην εικόνα 2-16, έχει διαιρεθεί σε πολλές ζώνες, την purdue.edu ζώνη και nwu.edu ζώνη. Στη κορυφή του domain υπάρχει επίσης και μια edu ζώνη. Είναι φυσικό ότι τα άτομα που ευθύνονται για το edu domain θα πρέπει να διαιρέσουν το edu domain αλλιώς το Berkley.edu subdomain θα έπρεπε να το διαχειριστούν οι ίδιοι. Έχει περισσότερο νόημα να αναθέσεις την ευθύνη για το Berkley.edu στον οργανισμό Berkley. ***Επομένως για τα άτομα που τρέχουν το edu domain έχει απομείνει η ζώνη edu, που κυρίως περιέχει πληροφορίες ανάθεσης για τα subdomain του edu.***



Εικόνα 2-16: Το domain edu σπασμένο σε ζώνες

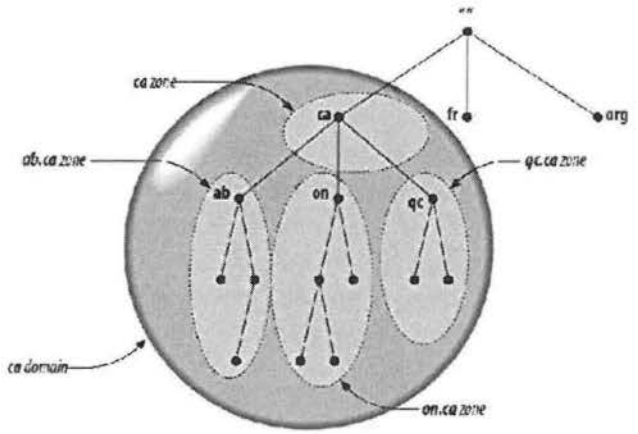
Το subdomain Berkley.edu με τη σειρά του διαιρείται σε πολλαπλές ζώνες με τη διαδικασία της ανάθεσης όπως φαίνεται στην εικόνα 2-17. Υπάρχουν ανατεθειμένα subdomain που ονομάζονται cc, cs, me κ.α. Κάθε subdomain έχει ανατεθεί σε ένα σύνολο από name

servers, μερικοί από τους οποίους είναι επίσημοι για το Berkley.edu. Ωστόσο, οι ζώνες είναι ακόμα χωριστές και μπορεί να έχουν τελείως διαφορετικές ομάδες από name servers.



Εικόνα 2-17: Το domain berkley.edu "σπασμένο" σε ζώνες

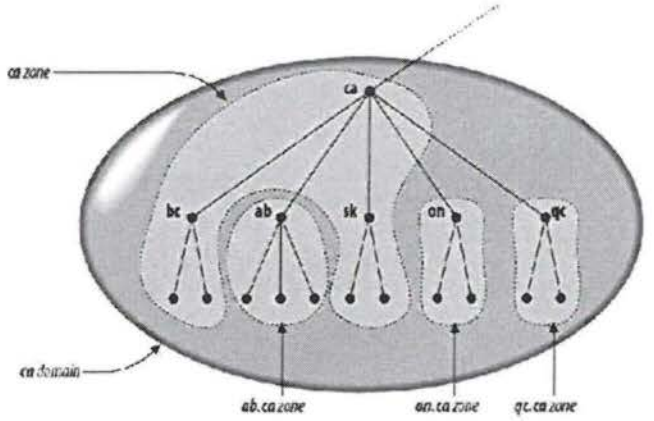
Μία ζώνη περιέχει όλα τα domain ονόματα όπου το domain με το ίδιο domain όνομα περιέχει, εκτός από τα domain ονόματα σε ανατεθειμένα subdomain. Για παράδειγμα το ανώτερο επίπεδο ca domain (Canada) έχει subdomain τα ab.ca, on.ca και qc.ca για τους Νομούς Alberta, Ontario και Quebec. Ο αρμόδιος για το ab.ca, on.ca και qc.ca domain μπορεί να ανατεθεί σε name servers σε κάθε νομό. Το domain ca περιέχει όλα τα δεδομένα στο ca συν όλα τα δεδομένα στο ab.ca, on.ca και qc.ca. Ωστόσο, η ζώνη ca περιέχει μόνο τα δεδομένα στο ca (Εικόνα 2-18) που είναι πιθανόν δείκτες στα ανατεθειμένα subdomains. Τα ab.ca, on.ca και qc.ca είναι ξεχωριστές ζώνες από την ca ζώνη.



Εικόνα 2-18: Το domain ca

Η ζώνη επίσης περιέχει τα domain ονόματα και δεδομένα που δεν έχουν ανατεθεί ακόμα. Για παράδειγμα, το bc.ca και το sk.ca (British Columbia και Saskatchewan) subdomain του ca domain μπορεί να υπάρχουν αλλά να μην έχουν ανατεθεί σε κανέναν. Σε αυτή τη

περίπτωση, η ζώνη ca έχει ένα τραχύ κάτω άκρο, που περιέχει το bc.ca και sk.ca αλλά όχι τα ανατεθειμένα subdomain, όπως φαίνεται στην εικόνα 2-19.



Εικόνα 2-19: Η ζώνη ca

Τώρα μπορούμε πια να δούμε το γιατί οι name servers φορτώνουν ζώνες και όχι domains: Ένα domain μπορεί να περιέχει περισσότερες πληροφορίες απ όσες χρειάζεται ένας name server, επειδή μπορεί να περιέχει δεδομένα που είναι ανατεθειμένα σε άλλους name servers. Από τη στιγμή που μια ζώνη περιορίζεται στην ανάθεση, ποτέ δεν θα περιλαμβάνει ανατεθειμένα δεδομένα. *Αν ένας root name server φόρτωνε το root domain αντί για τη root ζώνη. Θα φόρτωνε ολόκληρο το χώρο ονομάτων.* Επίσης αν αρχίσουμε ένα καινούργιο domain και δεν έχουμε κανένα subdomain, δηλαδή δεν έχουμε καμιά ανάθεση, τότε το domain και η ζώνη μας θα περιέχουν τα ίδια δεδομένα.

2.8.1 ΑΝΑΘΕΣΗ SUBDOMAINS

Η ανάθεση γενικά περιλαμβάνει τη ανάθεση των ευθυνών κάποιων κομματιών ενός domain σε έναν άλλον οργανισμό. Στη πραγματικότητα, αυτό που συμβαίνει είναι η ανάθεση της εξουσιοδότησης κάποιου subdomain σε διαφορετικούς ή διαφορετικό name server.

Τα δεδομένα της ζώνης μας, αντί να περιέχουν πληροφορίες για το subdomain που μόλις αναθέσαμε, περιέχει δείκτες σε name servers που είναι πλέον υπεύθυνοι για αυτό το subdomain. Τώρα αν κάποιος από τους name servers μας ρωτηθεί για τα δεδομένα του subdomain, μπορεί να απαντήσει με μια λίστα που περιέχει τους αρμόδιους name servers που πρέπει να επικοινωνήσει.

2.8.2 PRIMARY KAI SLAVE NAME SERVERS

Οι προδιαγραφές του DNS περιέχουν δύο είδη name server: **primary master** και **secondary master**. Ένας *primary master name server* για μια ζώνη διαβάζει τα δεδομένα της ζώνης από ένα αρχείο που βρίσκεται στο host που ανήκει. Ένας secondary master name server για μια ζώνη παίρνει τα δεδομένα της ζώνης από έναν άλλον name server που είναι υπεύθυνος για

τη ζώνη αυτή και ονομάζεται ως master server του. Πολύ συχνά, ο master server είναι ο primary master της ζώνης, αλλά αυτό δεν ισχύει πάντα: Ένας secondary master μπορεί να φορτώνει δεδομένα μιας ζώνης από έναν άλλον secondary master. Κατά την εκκίνηση του secondary, επικοινωνεί με το master name server του και αν είναι απαραίτητο, παίρνει τα δεδομένα της ζώνης από εκείνον. Αυτό ονομάζεται μεταφορά ζώνης. Σήμερα ο προτιμότερος τρόπος για να λέμε έναν secondary master name server είναι slave, αν και πολύ ακόμα χρησιμοποιούν την παλιά ορολογία. Στην εργασία αυτή θα καλείται ως slave.

Παρά το κάπως υποτιμητικό όνομα, οι slaves δεν είναι δεύτερης κατηγορίας name servers. Το DNS παρέχει αυτά τα δύο είδη name server, για να κάνει τη διαχείριση ακόμα πιο εύκολη. Όταν δημιουργείς τα δεδομένα για μια ζώνη και δημιουργήσεις ένα primary master name server, δεν χρειάζεσαι να αντιγράψεις τα δεδομένα από host σε host για να δημιουργήσεις καινούργιους name servers για αυτή τη ζώνη. Απλά δημιουργείς slave name servers που φορτώνουν τα δεδομένα από τον πρωτεύον name server για αυτή τη ζώνη. Οι slaves που εγκαθιστάς θα μεταφέρουν καινούργια δεδομένα από το primary όταν είναι απαραίτητο.

Οι slave name servers είναι σημαντικοί γιατί είναι καλή ιδέα να δημιουργήσεις περισσότερους από έναν υπεύθυνο name server για μια ζώνη. Θέλεις περισσότερο από έναν για πλεονασμό (redundancy), να διαδίδουν το φόρτο γύρο τους και για να είσαι σίγουρος ότι όλοι οι hosts στη ζώνη έχουν name servers γύρο τους. Χρησιμοποιώντας name servers, κάνει αυτή τη διαχείριση εφικτή.

Επίσης ένας name server μπορεί για μια ζώνη να είναι master ενώ για μια άλλη ζώνη να είναι slave. Οι περισσότεροι name server όμως είναι είτε primary για τις περισσότερες ζώνες που φορτώνου ή slave για τις περισσότερες ζώνες που φορτώνουν.

2.8.3 ΑΡΧΕΙΑ ΔΕΔΟΜΕΝΩΝ ΖΩΝΗΣ

Τα αρχεία από τα οποία οι primary master name servers φορτώνουν τα δεδομένα μιας ζώνης ονομάζονται αρχεία δεδομένων ζώνης. Συνήθως αναφερόμαστε σε αυτά σαν αρχεία δεδομένων παραλείποντας το «ζώνης». Οι slave name servers μπορούν και αυτοί να φορτώνουν δεδομένα ζώνης από αρχεία δεδομένων, αλλά οι slaves συνήθως χρησιμοποιούνται για να κρατάνε αντίγραφα των δεδομένων μιας ζώνης που μεταφέρουν από έναν primary master name server σε αρχεία δεδομένων. Αν ο slave κάνει επανεκκίνηση, πρώτα διαβάζει τα αρχεία δεδομένων που έχει κρατήσει αντίγραφα και μετά ελέγχει να δει αν τα δεδομένα του είναι τα τρέχοντα. Αυτό εξαλείφει την ανάγκη να μεταφέρονται τα δεδομένα ζώνης εάν δεν έχουν αλλάξει και παρέχει μια πηγή δεδομένων σε περίπτωση που ο master έχει πέσει για κάποιο λόγο.

Τα αρχεία δεδομένων περιέχουν εγγραφές πόρων (RR) που περιγράφουν τη ζώνη. Οι εγγραφές πόρων περιγράφουν όλους τους hosts στη ζώνη και σημαδεύουν οποιαδήποτε ανάθεση έχει γίνει για κάποιο subdomain.

2.9 RESOLVERS - ΑΝΑΛΥΤΕΣ

Οι αναλυτές πελάτες (clients) που έχουν πρόσβαση σε name servers. Προγράμματα που τρέχουν σένα host και χρειάζονται πληροφορίες για το namespace χρησιμοποιούν τον αναλυτή. Ο αναλυτής χειρίζεται:

- Ερωτήσεις σε έναν name server
- Ερμηνεύει απαντήσεις (που μπορεί να είναι ή εγγραφές πόρων ή σφάλμα)
- Να γυρίσει τη πληροφορία στην διεργασία που τη ζήτησε.

Στο λογισμικό BIND οι αναλυτές είναι ένα σύνολο από συναρτήσεις βιβλιοθηκών που συνδέονται σε προγράμματα όπως το ssh και το ftp. Δεν είναι καν μια ξεχωριστή διεργασία. Ο αναλυτής βασίζεται σχεδόν ολόκληρος στο name server που ρωτάει: Είναι αρκετά «έξυπνος» ώστε να συγκεντρώσει την ερώτηση, να τη στείλει και να περιμένει μια απάντηση και να ξαναστείλει την ερώτηση σε περίπτωση που δεν έχει πάρει απάντηση. Το περισσότερο βάρος στο να βρεις μια απάντηση στην ερώτηση βρίσκεται στο name server.

2.10 RESERVED DOMAINS ΚΑΙ PSEUDO DOMAINS

Στο RFC 2606 αποφασίστηκε, ότι τα παρακάτω domain δεν μπορούν να χρησιμοποιηθούν:

- Το test domain για test
- Το example domain για τη δημιουργία τεκμηριώσεων και παραδειγμάτων
- Το invalid domain για αναφορές σε σφάλματα
- Το localhost domain για βρόγχους λογισμικού (loopback address)

Τα domains που δεν συνδέονται κατευθείαν στο διαδίκτυο, μπορούν να υπάρχουν, για παράδειγμα υπολογιστές που δεν χρησιμοποιούν το TCP/IP πρωτόκολλο επομένως δεν έχουν IP. Αυτά τα domain λέγονται ψευδό-domains. Είναι ιδιαίτερα σημαντικοί για το ηλεκτρονικό ταχυδρομείο. Είναι δυνατό να στείλεις ένα mail σε άλλα δίκτυα και μετά στο Ιντερνετ με τη βοήθεια ενός ψευδό-domain (όπως το DECnet πρωτόκολλο ή το MS Exchange). Στο εσωτερικό δίκτυο της μια εταιρία μπορεί να χρησιμοποιήσει αρχικά το TCP/IP και στη συνέχεια το DECnet πρωτόκολλο. Ένας χρήστης που χρησιμοποιεί το εσωτερικό δίκτυο, για παράδειγμα (για παράδειγμα hr@computer.company.com) απευθύνεται από το διαδίκτυο και για να απευθυνθούμε σε χρήστες που χρησιμοποιούν το DECnet πρωτόκολλο, βάζουμε ένα φανταστικό dnet ψευδό-domain στη διεύθυνση. Άρα ο χρήστης στο χρήστη hr απευθυνόμαστε ως hr@computer.dnet.company.com. Με τη βοήθεια του DNS, ολόκληρο το μήνυμα ηλεκτρονικού ταχυδρομείου που απευθυνόταν στο domain dnet.company.com ανακατευθύνεται σε μια πύλη του DECnet πρωτοκόλλου (Η πύλη του company.com domain) που κάνει το μετασχηματισμό από TCP/IP (SMTP) σε DECnet (για Mail-11).

2.11 RESOLUTION - ΑΝΑΛΥΣΗ

Οι name servers όχι μόνο μπορούν να απαντήσουν σε ερωτήματα απομακρυσμένων υπολογιστών για τα δεδομένα που οι ίδιοι κρατάνε, αλλά έχουν την ικανότητα να ψάξουν το Internet για να βρούνε κάποια πληροφορία την οποία δεν την γνωρίζουν – δεν είναι δηλαδή μέσα στις ζώνες που εξυπηρετούν. Οι εξυπηρετητές χρειάζονται μόνο μία σημαντική πληροφορία για να αρχίσουν την έρευνα για κάποιο στοιχείο το οποίο δεν έχουν: τους ριζικούς εξυπηρετητές ονομάτων (**Root Name Servers**). Οι τελευταίοι διαθέτουν πληροφορίες για κάθε top level domain αλλά μόνο αυτές – δεν έχουν ποτέ πληροφορίες για το πού βρίσκεται κάποιο συγκεκριμένο δεδομένο. Κάθε εξυπηρετητής ονομάτων λοιπόν, πρέπει να έχει ένα αρχείο με όλους τους Root Name Servers (είναι δεκατρείς σε όλο τον κόσμο). Τι ακριβώς συμβαίνει κάθε φορά που ζητάμε να δούμε μια σελίδα, για παράδειγμα την www.nasa.gov στον Παγκόσμιο Ιστό;

Ο πλοηγός (browser) στον οποίο έχουμε δώσει την παραπάνω διεύθυνση δεν μπορεί να βρει τον συγκεκριμένο υπολογιστή αφού δεν γνωρίζει την IP διεύθυνση του. Στον υπολογιστή μας έχουμε δηλώσει κάποια τετράδα αριθμών ως DNS server. Ο browser, λοιπόν, παίρνει την πρωτοβουλία και ρωτάει τον συγκεκριμένο server για την IP διεύθυνση του υπολογιστή με το όνομα www.nasa.gov.

Ο DNS server όμως γνωρίζει στοιχεία μόνο για τη ζώνη edu-net.gr. Αυτό που κάνει είναι να ρωτήσει έναν από τους ριζικούς εξυπηρετητές ονομάτων για το που βρίσκονται στοιχεία για το top level domain gov. (προσέξτε ότι δεν ρωτάει την IP διεύθυνση του ονόματος www.nasa.gov). Ο ριζικός εξυπηρετητής του απαντάει ότι το μηχάνημα υπεύθυνο για το domain gov είναι το A.ROOT-SERVERS.NET. Αμέσως μετά ο server που προσπαθεί να βρει την IP διεύθυνση ρωτάει τον τελευταίο (A.ROOT-SERVERS.NET) για το που μπορεί να βρει το domain nasa. Ο υπολογιστής A.ROOT-SERVERS.NET ο οποίος γνωρίζει ποια μηχανήματα είναι υπεύθυνα για τα subdomains του απαντάει ότι είναι ο υπολογιστής NS1.JPL.nasa.gov. Το τελευταίο βήμα είναι να ρωτήσει τον προηγούμενο υπολογιστή (NS1.JPL.nasa.gov) για την IP διεύθυνση που αντιστοιχεί στο όνομα www.nasa.gov. Εφόσον πάρει απάντηση (την IP Που ζητάμε) ο πλοηγός θα αποκαλύψει τα περιεχόμενα της τοποθεσίας www.nasa.gov.

Προσέξτε ότι υπάρχουν δύο είδη ερωτημάτων: Το ένα είναι 'βρες μου την IP διεύθυνση που αντιστοιχεί στο τάδε όνομα και το κάνουν πάντα οι πελάτες (ονομάζονται resolvers στην ορολογία του DNS) των εξυπηρετητών ονομάτων. Αυτού του είδους οι ερωτήσεις ονομάζονται recursive. Το δεύτερο είναι ξέρεις που μπορώ να βρω περισσότερες πληροφορίες για αυτό που ψάχνω; Και είναι τα ερωτήματα που ανταλλάσσουν μεταξύ τους οι Name Servers κάθε φορά που χρειάζεται να αντιστοιχίσουν ένα όνομα σε μία IP διεύθυνση. Κάθε φορά λοιπόν, που δίνεται μια διεύθυνση στον browser σας ή προσπαθείτε να στείλετε email ο Name Server που σας εξυπηρετεί πρέπει να κάνει τον γύρο του κόσμου για να βρει την απάντηση; Τα πράγματα ευτυχώς δεν είναι ακριβώς έτσι.

Ο κάθε εξυπηρετητής ονομάτων διαθέτει μνήμη (cache) που καταγράφει τα στοιχεία που έχει ήδη μάθει. Έτσι δεν χρειάζεται να καταφεύγει κάθε φορά στους root Name servers για να βρει μια πληροφορία – αν και αυτό συμβαίνει πολλές φορές!

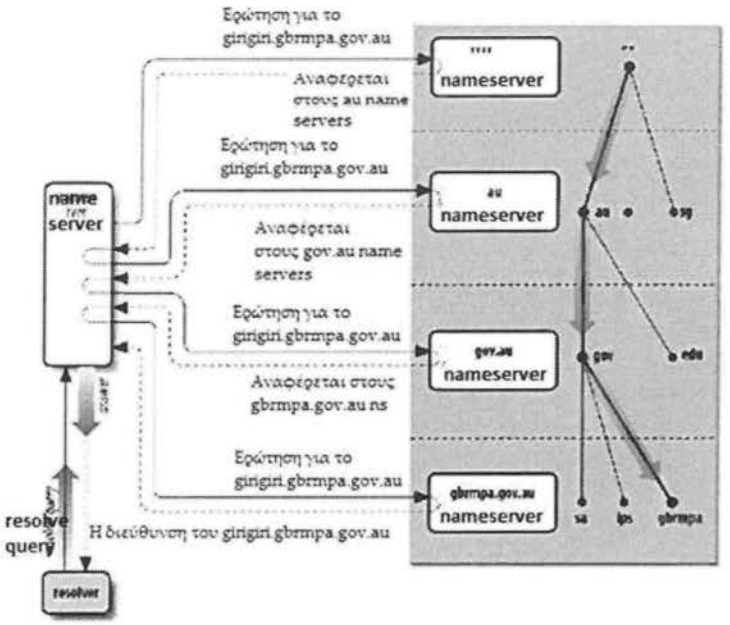
Οι εξυπηρετητές ονομάτων όμως δεν αντιστοιχίζουν μόνο ονόματα υπολογιστών σε IP διευθύνσεις. Μπορούν να βρουν και που ακριβώς βρίσκεται ένας υπολογιστής με μια δεδομένη IP διεύθυνση. Κι αυτό γιατί πολλά προγράμματα που λειτουργούν μέσα από το δίκτυο χρειάζονται πληροφορίες βασισμένες σε IP διευθύνσεις και όχι σε ονόματα υπολογιστών. Για αυτήν τη διαδικασία ,δηλαδή, την εύρεση του δρόμου που πρέπει να ακολουθηθεί για να βρεθεί ένας υπολογιστής στο δίκτυο, υπάρχει μια ξεχωριστή ιεραρχική δομή παρόμοια με αυτήν των domains και zones: είναι η ιεραρχία του **in-addr.arpa domain**.

2.11.1 ROOT NAME SERVERS

Οι root name servers ξέρουν που βρίσκονται οι υπεύθυνοι name servers για κάποια ζώνη ανωτάτου επιπέδου (top-level domain). Κάνοντας μια ερώτηση για κάποιο domain name, ο root name server μπορεί να απαντήσει και να μας δώσει τα ονόματα και τις διευθύνσεις των name server που είναι υπεύθυνοι για τις ζώνες ανώτερου επιπέδου που τελειώνει το domain που ζητάμε (π.χ. .com , .net κτλ). Με τη σειρά τους οι ανώτερου επιπέδου (TLD – Top Level Domain) name servers μπορούν να μας δώσουν τους υπεύθυνους name server για της δεύτερης επιπέδου ζώνης όπου το domain όνομα τελειώνει (για παράδειγμα *company*.com). Κάθε name server που ρωτείται, ή θα μας δώσει την πληροφορία για το πως θα φτάσουμε ποιο κοντά στην απάντηση που ψάχνουμε, ή θα μας δώσει την απάντηση ο ίδιος. Οι root name server είναι πολύ σημαντική στην ανάλυση. Επειδή είναι τόσο σημαντικοί, το DNS παρέχει έναν μηχανισμό προσωρινής αποθήκευσης όπως θα δούμε παρακάτω. Επομένως οι root name servers έχουν ζωτική σημασία για τη λειτουργία του DNS. Αν όλοι οι root διακομιστές ονομάτων (root name servers) ήταν απρόσιτοι για κάποιο χρονικό διάστημα, όλη η ανάλυση στο Ιντερνετ θα αποτύγχανε. Για την αποφυγή αυτού του προβλήματος, το Ιντερνετ έχει 13 (λογικούς) root name servers, γιατί στη πραγματικότητα είναι περισσότεροι φυσικοί servers. Οι περισσότεροι είναι είτε εξισορροπώντας το φόρτο του δικτύου πίσω από μια IP, μια ομάδα από κατανεμημένους servers που χρησιμοποιούν την ίδια IP διεύθυνση (Unicast) ή συνδυασμός των δύο.

Το να είσαι το κεντρικό σημείο για τόσα πολλά ερωτήματα κρατά τους root name servers πολύ απασχολημένους και ακόμα και με 13 λογικούς που έχουμε, η κίνηση σε καθένα από αυτούς είναι πολύ μεγάλη (δεκάδες χιλιάδες ερωτήματα ανά δευτερόλεπτο). Παρά το φορτίο που υπάρχει στους root name servers, η ανάλυση στο Ιντερνετ δουλεύει αρκετά καλά. Η εικόνα 2-20 δείχνει την διαδικασία της ανάλυσης για τη διεύθυνση ενός πραγματικού host σε ένα πραγματικό domain και βλέπουμε πως η διαδικασία διέρχεται στο δέντρο του Internet Domain Namespace.

Ο τοπικός name server κάνει ένα ερώτημα σε ένα root name server για τι διεύθυνση girigiri.gbrmpa.gov.au και του λένε να αναφερθεί στους au name servers . Ο τοπικός name server ρωτά τους au name servers την ίδια ερώτηση και ύστερα αναφέρεται στου .gov.au name servers. Ο gov.au name server παραπέμπει το τοπικό name server στους gbrmpa.gov.au name servers. Τελικά, ο τοπικός name server είναι ο gbrmpa.gov.au name server για την διεύθυνση και παίρνει την τελική απάντηση που είναι η IP διεύθυνση του giri.gbrmpa.gov.au.



Εικόνα 2-20: Ανάλυση του girigiri.gbrmpa.gov.au στο Internet

2.11.2 ΑΝΑΔΡΟΜΙΚΟΤΗΤΑ - RECURSION

Στο προηγούμενο παράδειγμα είδαμε μια μεγάλη διαφορά στον όγκο της εργασίας που γίνεται από τους name servers. Οι τέσσερις name servers απλά γύρισαν την καλύτερη απάντηση που είχαν κυρίως σε παραπομπές σε άλλους servers από τα ερωτήματα που δέχτηκαν. Δεν χρειάστηκε να στείλουν και εκείνη τα δικά τους ερωτήματα για να βρουν τα δεδομένα που ζητήθηκαν. Αλλά ο name server που έλαβε ένα ερώτημα από τον αναλυτή, έπρεπε να ακολουθήσει διαδοχικές παραπομπές μέχρι να λάβει μια απάντηση.

Ο τοπικός name server ήξερε ότι δεν έπρεπε να απαντήσει πίσω στον αναλυτή (στην πρώτη αίτηση) γιατί ο αναλυτής έστειλε ένα αναδρομικό ερώτημα. Τα ερωτήματα διακρίνονται σε δύο είδη τα αναδρομικά και τα επαναληπτικά γνωστά και ως μη αναδρομικά. (recursive και iterative). Τα αναδρομικά ερωτήματα όπως και στον προγραμματισμό με τους αναδρομικούς αλγορίθμους, ο name server εκτελεί την ίδια βασική λειτουργία (ρωτώντας έναν απομακρυσμένο name server μέχρι να βρει μια απάντηση) μέχρι να βρει μια απάντηση.

Στην αναδρομή, ο αναλυτής στέλνει ένα αναδρομικό ερώτημα σε έναν name server για πληροφορίες σχετικά για ένα συγκεκριμένο domain name. Ο ρωτούμενος name server είναι υποχρεωμένος να απαντήσει με τα ζητούμενα δεδομένα ή με ένα σφάλμα δείχνοντας ότι ο ζητούμενος τύπος δεδομένων δεν υπάρχει ή ότι το όνομα του domain δεν υπάρχει.

Αν ο ρωτούμενος name server δεν είναι υπεύθυνος για τα ζητούμενα δεδομένα, θα πρέπει να ρωτήσει άλλους name servers για να βρει την απάντηση. Θα μπορούσε να στείλει αναδρομικά ερωτήματα, υποχρεώνοντας τους να βρουν την απάντηση, ή θα μπορούσε να στείλει επαναληπτικά ερωτήματα και πιθανόν να αναφερθεί σε άλλους name servers που

είναι ποιο «κοντά» στο όνομα του domain που ψάχνουμε. Οι σύγχρονες υλοποιήσεις κάνουν τη δεύτερη επιλογή ακολουθώντας άλλους name servers μέχρι να βρούμε την απάντηση μας.

Ένας name server που λαμβάνει ένα αναδρομικό ερώτημα και δεν μπορεί να το απαντήσει μόνος του, θα ρωτήσει το ποιο κοντινό name server. Οι ποιο «κοντινοί» είναι εκείνοι που είναι υπεύθυνοι για τη ζώνη που είναι ποιο κοντά στο όνομα του domain Που ψάχνουμε. Για παράδειγμα αν ένας name server λαμβάνει ένα αναδρομικό ερώτημα τη διεύθυνση του domain lapis.gbrmpa.gov.au, πρώτα ελέγχει αν ξέρει ποιοι name servers είναι υπεύθυνοι για το lapis.gbrmpa.gov.au. Αν ξέρει, στέλνει το ερώτημα σε έναν από αυτούς, αλλιώς αν δεν ξέρει, ελέγχει αν ξέρει τους υπεύθυνους name servers για το gbrmpa.gov.au. Αν ξέρει στέλνει το ερώτημα σε έναν από αυτούς, αλλιώς ελέγχει ομοίως τους υπεύθυνους για το gov.au και μετά au. Το προεπιλεγμένο σημείο που είμαστε σίγουροι ότι ο έλεγχος θα σταματήσει, είναι η ζώνη της ρίζας (root zone), επειδή κάθε name server ξέρει το όνομα domain και τη διεύθυνση του κάθε root name server.

Χρησιμοποιώντας το «κοντινότερο» γνωστό name server εγγυάται ότι η διαδικασία της ανάλυσης γίνεται όσο ποιο σύντομη γίνεται. Ένας name server του Berkley.edu λαμβάνοντας ένα ερώτημα για τη διεύθυνση του lapis.ce.berkley.edu δεν θα πρέπει να συμβουλευτεί το root name server: μπορεί απλά να ακολουθήσει πληροφορίες ανάθεσης (delegation) για να φτάσει κατευθείαν στους ce.berkley.edu name servers. Επίσης ένας name server που έχει αναζητήσει ένα domain name στο ce.berkley.edu δεν θα πρέπει να αρχίσει την ανάλυση στο root name server για να βρει ένα άλλο ce.berkley.edu domain. Θα δούμε πως δουλεύει αυτή η τεχνική παρακάτω στη προσωρινή αποθήκευση μνήμης (caching).

2.11.3 ΕΠΑΝΑΛΗΨΗ - ITERATION

Η επαναληπτική ανάλυση δεν απαιτεί πολύ δουλειά εκ μέρους του ρωτούμενου name server. Στην επαναληπτική ανάλυση, ένας name server δίνει την καλύτερη απάντηση που είδη ξέρει πίσω στον ρωτούνται. Ο ρωτούμενος name server συμβουλεύεται τα τοπικά του δεδομένα (και τη προσωρινή του μνήμη), ψάχνοντας τα δεδομένα που ζητήθηκαν. Αν δεν βρει την απάντηση εκεί, βρίσκει τα ονόματα και τις διευθύνσεις των name servers που είναι «κοντινότεροι» στο όνομα του domain, στα τοπικά του δεδομένα και γυρνάει αυτό σαν απάντηση για παραπομπή, για να βοηθήσει εκείνον που ρώτησε, να συνεχίσει την διαδικασία της ανάλυσης. Σημειώνουμε ότι στην απάντηση παραπομπή, περιέχονται όλοι οι name server των τοπικών δεδομένων. Είναι κρίση εκείνου που ρωτάει ποιον θα ρωτήσει στη συνέχεια.

2.11.4 ΕΠΙΛΟΓΗ ΜΕΤΑΞΥ ΕΠΙΣΗΜΩΝ NAME SERVER

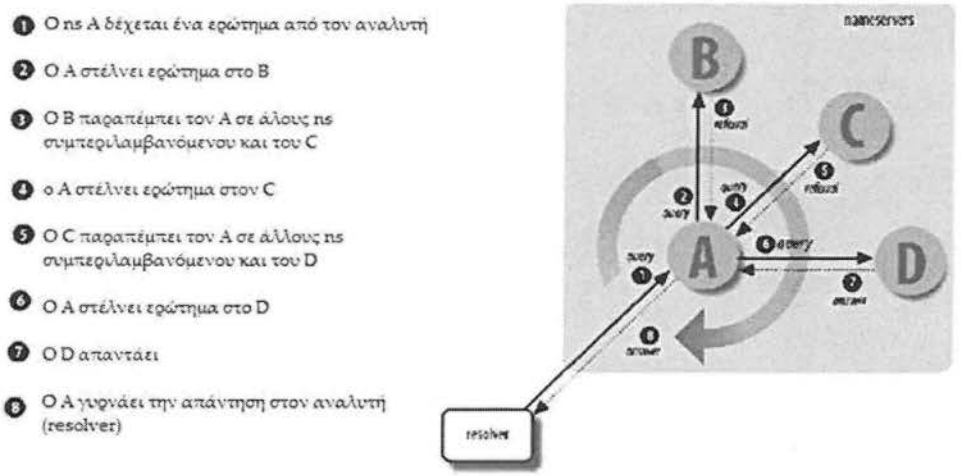
Απ' όλα όσα είπαμε τίθεται ένα βασικό ερώτημα: πως ένας name server που λαμβάνει ένα αναδρομικό ερώτημα, διαλέγει το κατάλληλο name server για μια ζώνη; Οι name server

που χρησιμοποιούν το λογισμικό BIND, χρησιμοποιούν μια μέτρηση που ονομάζεται χρόνος μετ 'επιστροφής, ή RTT (roundtrip time), για να διαλέξουν έναν από τους name servers που είναι υπεύθυνοι για την ίδια ζώνη. Ο χρόνος RTT είναι μια μέτρηση του πόσο καιρό κάνει ένας απομακρυσμένος name server να απαντήσει σε ερωτήματα. Κάθε φορά που ένας name server στέλνει ένα ερώτημα σε έναν απομακρυσμένο name server ξεκινά ένα εσωτερικό χρονόμετρο. Όταν λαμβάνει μια απάντηση σταματάει το χρονόμετρο και σημειώνει πόσο καιρό ήθελε ο απομακρυσμένος διακομιστής να απάντηση. Όταν ένας name server πρέπει να επιλέξει σε ποιο name server από την ομάδα να ρωτήσει επιλέγει εκείνον με το χαμηλότερο RTT χρόνο.

Σε γενικές γραμμές, αυτός ο απλός αλλά κομψός αλγόριθμος επιτρέπει στους BIND name servers να "κλειδώνουν" στο πλησιέστερο name server γρήγορα και χωρίς την επιβάρυνση ενός πολύπλοκου μηχανισμού για τη μέτρηση των επιδόσεων.

2.11.5 ΟΛΟΚΛΗΡΗ Η ΔΙΑΔΙΚΑΣΙΑ ΤΗΣ ΑΝΑΛΥΣΗΣ

Στην εικόνα 2-21 βλέπουμε τη διαδικασία της ανάλυσης.



Εικόνα 2-21: Διαδικασία Ανάλυσης

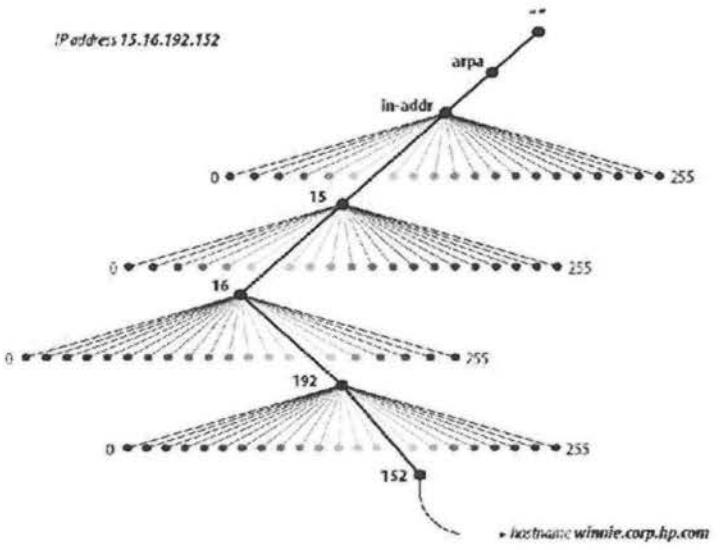
Ο αναλυτής (resolver) στέλνει ένα ερώτημα στο τοπικό name server, ο οποίος στέλνει επαναληπτικά ερωτήματα σε άλλους name servers κυνηγώντας μια απάντηση για τον αναλυτή. Κάθε name server που ρωτάει το παραπέμπει σε ένα άλλο name server που είναι υπεύθυνος για μα ζώνη πιο κάτω στο χώρο ονομάτων και ποιο κοντά στο όνομα του domain που επιδιώκεται. Τελικά ο τοπικός name server ρωτάει τον υπεύθυνο name server (D) που έχει την απάντηση που ψάχνει ο αναλυτής και ο D πια γυρνάει την απάντηση στον τοπικό name server και εκείνος με τη σειρά του στον αναλυτή. Τέλος σε όλο αυτό το διάστημα ο τοπικός name server χρησιμοποιεί κάθε απάντηση που λαμβάνει είτε είναι παραπομπή, είτε είναι απάντηση για την ενημέρωση του χρόνου RTT της ανταπόκρισης του name server που θα το βοηθήσει να αποφασίσει ποιούς name servers να ρωτήσει στο μέλλον.

2.11.6 ΑΝΤΙΣΤΟΙΧΗΣΗ ΔΙΕΥΘΥΝΣΕΩΝ ΣΕ ΟΝΟΜΑΤΑ

Ένα πολύ σημαντικό κομμάτι που λείπει από την διαδικασία της ανάλυσης που είδαμε είναι πως οι διευθύνσεις αντιστοιχίζονται σε domain ονόματα. Οι αντιστοιχίσεις διευθύνσεων σε ονόματα παράγουν έξοδο που είναι ποιο εύκολη από τους ανθρώπους να διαβαστούν και να ερμηνευτούν (σε αρχεία καταγραφής για παράδειγμα). Το αρχείο του UNIX hosts αντιστοιχίζει τις διευθύνσεις σε domain ονόματα για να συγκρίνει καταχωρήσεις στο .rhosts και hosts.equiv αρχεία. Όταν χρησιμοποιούμε πίνακες host η αντιστοίχηση είναι ασήμαντη. Το μόνο που χρειάζεται είναι μια απλή διαδοχική αναζήτηση στον πίνακα hosts για μια διεύθυνση. Η αναζήτηση επιστρέφει το επίσημο όνομα στη λίστα. Στο DNS ωστόσο, η αντιστοίχηση σε ονόματα δεν είναι τόσο απλή. Τα δεδομένα, περιλαμβανομένων και των διευθύνσεων, στο χώρο ονομάτων των domain δεικτοδοτούνται μέσω του ονόματος. Δεδομένου ότι έχεις ένα όνομα domain, το να βρεις τη διεύθυνση είναι εύκολο. Αλλά να βρεις το όνομα του domain που αντιστοιχεί σε μια δεδομένη διεύθυνση απαιτεί μια εξαντλητική αναζήτηση των δεδομένων, που βρίσκονται σε κάθε domain στο δέντρο.

Στην πραγματικότητα υπάρχει μια καλύτερη λύση που είναι και έξυπνη αλλά και αποτελεσματική. Επειδή είναι εύκολο να βρει κανείς τα δεδομένα όταν μας δίνεται το domain όνομα, δημιουργούμε ένα μέρος του χώρου ονομάτων (namespace) που χρησιμοποιεί διευθύνσεις ως ετικέτες. Στο χώρο ονομάτων του Ιντερνετ, αυτό το τμήμα είναι το domain in-addr.arpa.

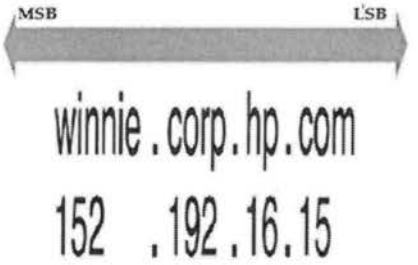
Οι κόμβοι στο in-addr.arpa domain επισημαίνονται με τους αριθμούς όπως μια διεύθυνση IP. Το in-addr.arpa domain για παράδειγμα μπορεί να έχει μέχρι και 256 subdomains, όπου το καθένα αντιστοιχεί σε κάθε πιθανή τιμή στην πρώτη οκτάδα της διεύθυνσης IP. Κάθε subdomain μπορεί να έχει 256 subdomains που αντιστοιχούν στις πιθανές τιμές της δεύτερης οκτάδας. Τέλος, στο τέταρτο επίπεδο κάτω, υπάρχουν εγγραφές πόρων (RR) που συνδέονται με την τελική οκτάδα δίνοντας ολόκληρο το domain name του host σε αυτή την IP. Αυτό κάνει ένα πολύ μεγάλο domain, το in-addr.arpa όπως φαίνεται στην Εικόνα 2-22 είναι αρκετά ευρύχωρο για κάθε διεύθυνση IP στο Ιντερνετ.



Εικόνα 2-22: Το domain in-addr.arpa

Παρατηρήστε ότι όταν διαβάζουμε το όνομα ενός domain, η IP εμφανίζεται ανάποδα διότι, το όνομα διαβάζεται από αριστερά στα δεξιά. Για παράδειγμα, αν το Winnie.corp.hp.com έχει IP 15.16.192.152, ο αντίστοιχος κόμβος στο domain in-addr.arpa είναι 152.192.16.15.in-addr.arpa, το οποίο αντιστοιχίζεται πίσω πάλι στο domain όνομα Winnie.corp.hp.com.

Οι IP διευθύνσεις θα μπορούσαν να είχαν παρασταθεί με τον αντίθετο τρόπο, με την πρώτη οκτάδα της IP να είναι στο κάτω μέρος του in-addr.arpa domain. Με αυτό το τρόπο οι διευθύνσεις να διαβάζονται σωστά (προς τα εμπρός) στο domain. Αλλά οι διευθύνσεις IP είναι και αυτές ιεραρχικά όπως και τα domain. Η διαφορά όμως είναι ότι οι IP διευθύνσεις, γίνονται ποιο συγκεκριμένα από αριστερά προς τα δεξιά ενώ τα domain γίνονται λιγότερο συγκεκριμένα από αριστερά προς τα δεξιά (LSB - MSB) όπως φαίνεται στην εικόνα 2-23.



Εικόνα 2-23: Ιεραρχικά ονόματα και διευθύνσεις

Κάνοντας την πρώτη οκτάδα να εμφανίζεται υψηλότερη στο δέντρο, επιτρέπει στους διαχειριστές να αναθέσουν ευθύνη για τις ζώνες του in-addr.arpa κατά μήκος των γραμμών του δικτύου. Για παράδειγμα η 15.in-addr.arpa ζώνη, που περιέχει πληροφορίες της αντίστροφης αντιστοίχησης, για όλους τους hosts που η IP τους αρχίζει από 15, μπορεί να ανατεθεί στους διαχειριστές του δικτύου 15/8 (15.x.x.x με μάσκα 255.0.0.0). Αυτό θα ήταν αδύνατο αν οι οκτάδες εμφανίζονταν στην αντίστροφη σειρά. Αν οι IPs ήταν αναπαριστάμενες στην αντίστροφη σειρά, το 15.in-addr.arpa θα αποτελούσε κάθε host που η IP του τελείωνε σε 15!

2.11.7 ΠΡΟΣΩΡΙΝΗ ΑΠΟΘΗΚΕΥΣΗ

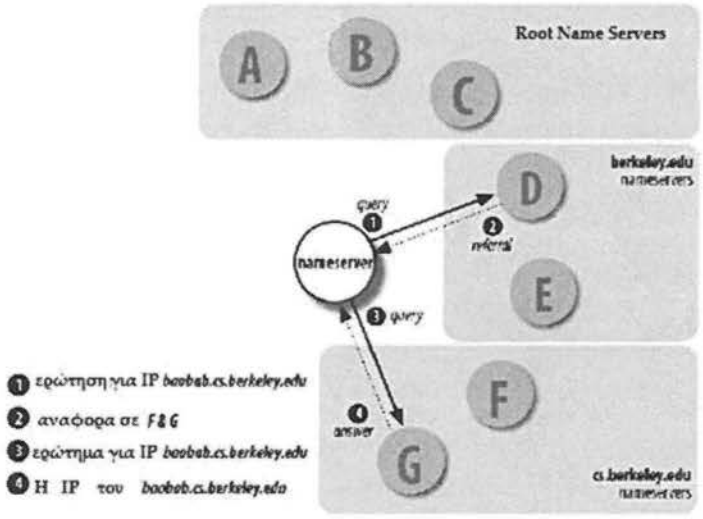
Η όλη διαδικασία της ανάλυσης μπορεί να φαίνεται απαίσια, περίπλοκη και δυσκίνητη σε κάποιον που έχει συνηθίσει σε απλές αναζητήσεις μέσω του πίνακα host. Στην πραγματικότητα, όμως, είναι συνήθως αρκετά γρήγορη. Ένα από τα χαρακτηριστικά που επιταχύνει σημαντικά είναι το caching (προσωρινή αποθήκευση).

Ένας name server που επεξεργάζεται ένα αναδρομικό ερώτημα μπορεί να χρειαστεί να στείλει πολλά ερωτήματα για να βρει μια απάντηση. Ωστόσο ανακαλύπτει πολλές πληροφορίες σχετικά με το χώρο ονομάτων των domain όσο το κάνει αυτό. Κάθε φορά που αναφέρεται σε μια άλλη λίστα από name servers, μαθαίνει ότι εκείνοι οι name servers είναι υπεύθυνοι για κάποιες ζώνες και μαθαίνει τις διευθύνσεις αυτών των διακομιστών. Στο τέλος της ανάλυσης όταν βρει τα δεδομένα που το αρχικό ερώτημα ζήτησε, μπορεί να κρατήσει τα δεδομένα για μελλοντική χρήση. Ο BIND name server υλοποιεί ακόμα και

αρνητική προσωρινή αποθήκευση: Αν ένας name server απαντήσει σε ένα ερώτημα με μια απάντηση που λέει ότι το domain όνομα ή το τύπος δεδομένων στο ερώτημα δεν υπάρχουν, ο τοπικός name server αποθηκεύει προσωρινά αυτή τη πληροφορία.

Οι name servers αποθηκεύουν όλη αυτή την πληροφορία για να επιταχύνουν τη διαδικασία εύρεσης επιτυχών ερωτημάτων. Την επόμενη φορά που ο αναλυτής θα ρωτήσει έναν name server για ένα domain το οποίο ξέρει κάποιες πληροφορίες, όλη η διαδικασία θα μειωθεί αρκετά. Ο name server μπορεί να έχει αποθήκευση την απάντηση θετικά ή αρνητικά οπότε απλά επιστρέφει την απάντηση στον αναλυτή (resolver). Ακόμα και αν δεν έχει τη πληροφορία αποθηκευμένη, μπορεί να έχει μάθει τις ταυτότητες των name server που ξέρουν (είναι υπεύθυνοι) για την ζώνη που είναι το domain όνομα και θα είναι σε θέση να τους ρωτήσει κατευθείαν.

Για παράδειγμα, ας υποθέσουμε ότι ο name server μας έχει είδη εξετάσει τη διεύθυνση eecs.berkley.edu. Κατά τη διάρκεια της διαδικασίας αποθηκεύονται προσωρινά τα ονόματα και οι διευθύνσεις των eecs.berkley.edu και Berkley.edu name server συν την IP του eecs.berkley.edu . Τώρα αν ο αναλυτής μας θέλει να κάνει ένα ερώτημα στο name server μας για τη διεύθυνση baobabc.cs.berkley.edu, ο name server θα μπορούσε να παραλείψει ρωτώντας τους root name servers. Αναγνωρίζοντας ότι το Berkley.edu είναι ποιο κοντά στον baobab.berkley.edu ο name server μας θα αρχίσει να στείλει ένα ερώτημα σε έναν από του Berkley.edu name servers όπως φαίνεται στην εικόνα 2-24.



Εικόνα 2-24: Αναλύοντας το baobab.cs.berkley.edu

Από την άλλη πλευρά, αν ο name server μας ανακάλυπτε ότι δεν υπήρχε καμία διεύθυνση για τον eecs.berkeley.edu, την επόμενη φορά που θα λάβει ένα ερώτημα για τη διεύθυνση, θα μπορούσε απλά να ανταποκριθεί κατάλληλα με βάση την προσωρινή μνήμη της.

2.11.8 TIME TO LIVE - TTL

Οι name server δεν μπορούν να αποθηκεύουν προσωρινά δεδομένα για πάντα. Ο διαχειριστής της ζώνης που περιέχει τα δεδομένα, αποφασίζει ένα χρόνο ζωής (Time To Live) για τα δεδομένα. Ο χρόνος ζωής είναι ο χρόνος που ένας name server κρατά τα προσωρινά αποθηκευμένα δεδομένα. Όταν ο χρόνος λήξει, ο name server πρέπει να διαγράφει τα δεδομένα και να πάρει καινούργια δεδομένα από τον αρμόδιο name server. Το ίδιο ισχύει και για την αρνητική προσωρινή αποθήκευση.

Από τη μια μεριά με το να έχεις μικρό TTL έχεις ποιο ενημερωμένους τους name servers σου, αλλά από την άλλη έτσι αυξάνεις το φόρτο που θα έχουν. Επίσης μακραίνει ο μέσος όρος χρόνου ανάλυσης για πληροφορίες στις ζώνες σου.

2.12 ΛΟΓΙΣΜΙΚΟ DNS

Υπάρχουν διαφορετικές επιλογές λογισμικού DNS ανάλογα με τις απαιτήσεις των χρηστών. Το Berkley Internet Name Domain–BIND–είναι μια ανοιχτού λογισμικού υλοποίηση που αναπτύσσεται από την Internet Systems Consortium (www.isc.org). Είναι πιθανόν η ποιο γνωστή και αναπτυσσόμενη DNS υλοποίηση. Το BIND ωστόσο, δεν είναι η μοναδική λύση που υπάρχει. Το λογισμικό BIND μπορεί να τρέξει τόσο σε windows όσο και σε Linux και UNIX συστήματα. Στην εργασία αυτή θα τρέξουμε το BIND στο λειτουργικό σύστημα FreeBSD και Ubuntu Linux όπως θα δούμε στο κεφάλαιο 10.

Το BIND λογισμικό περιέχει τρία μέρη:

- **Ένα DNS διακομιστή.** Αυτό το λογισμικό ονομάζεται "named" (name daemon). Απαντάει σε ερωτήσεις που απευθύνονται σε αυτό, ακολουθώντας τις προδιαγραφές του DNS πρωτοκόλλου. Μπορούμε να παρέχουμε DNS υπηρεσίες εγκαθιστώντας το λογισμικό αυτό σε κάποιο δημόσιο διακομιστή μας και να το ρυθμίσουμε σωστά για τα domain ονόματα μας.
- Μια DNS βιβλιοθήκη "resolver". Ένας resolver είναι ένα πρόγραμμα που αναλύει ερωτήματα σχετικά με κάποιο όνομα, στέλνοντας τα ερωτήματα αυτά στους κατάλληλους διακομιστές και απαντώντας κατάλληλα με την απάντηση του διακομιστή. Είναι μια βιβλιοθήκη που περιέχει κομμάτια λογισμικού που ένας προγραμματιστής μπορεί να προσθέσει σε λογισμικό που αναπτύσσεται, για να δώσει στο λογισμικό αυτό την ικανότητα να αναλύει ονόματα. Για παράδειγμα ένας προγραμματιστής που αναπτύσσει έναν καινούργιο web browser δεν χρειάζεται να δημιουργήσει ένα κομμάτι από αυτό που θα ψάχνει ονόματα στο DNS. Μπορεί απλά να χρησιμοποιήσει τη βιβλιοθήκη και να στέλνει ερωτήματα στα κομμάτια λογισμικού που περιέχει. Αυτό εξοικονομεί χρόνο και βοηθάει στο ότι ο καινούργιος browser ακολουθεί σωστά τις DNS προδιαγραφές.
- **Εργαλεία λογισμικού για διακομιστές υπό δοκιμή.** Αυτά είναι εργαλεία που χρησιμοποιούμε για δοκιμές και υπάρχουν για να βεβαιωθεί ο διαχειριστής ότι ο διακομιστής δουλεύει σωστά.

Το DNS λογισμικό αντανακλά πλήρως τις απαιτήσεις που έχει ένα τόσο μεγάλο δίκτυο σαν το Ιντερνετ. Ειδικότερα:

- Πηγές Δεδομένων.
- Πολυπλοκότητα.
- Διαχείριση.
- Δυναμικά Δεδομένα.

Αν σκοπεύουμε να εγκαταστήσουμε τις δικές μας ζώνες, και να τρέξουμε name servers για τις ζώνες αυτές, τότε πρώτα πρέπει να εγκαταστήσουμε ένα λογισμικό DNS. Οι περισσότερες διανομές Unix μαζί με το λειτουργικό σύστημα δίνουν και το BIND μαζί με τα υπόλοιπα βασικά TCP/IP δικτυακά λογισμικά. Ακόμη το δικτυακό λογισμικό περιλαμβάνεται μαζί με το λειτουργικό σύστημα. Σε περίπτωση που δεν έχουμε το BIND όμως μπορούμε να κατεβάζουμε το πηγαίο κώδικα της ποιο ενημερωμένης έκδοσης του BIND. Η μεταγλώττιση σχεδόν σε κάθε Unix πλατφόρμα είναι μια απλή διαδικασία. Σε περίπτωση που έχουμε μια παλιά έκδοση του BIND, καλό θα ήταν να ενημερώσουμε το λογισμικό στην ποιο πρόσφατη έκδοση για πολλούς λόγους, όπως διορθώσεις ασφαλείας και λειτουργίες ασφαλείας. Ακόμη οι εκδόσεις 8 και 9 του BIND υποστηρίζουν δυναμικές ενημερώσεις που επιτρέπει τους πράκτορες (agents) να ενημερώσουν τα δεδομένα μιας ζώνης στέλνοντας ειδικά ενημερωτικά μηνύματα για να προσθέσουν ή να διαγράψουν κάποιες εγγραφές πόρων. Στις ίδιες εκδόσεις BIND έχουμε αυξητική ζώνη μεταφοράς, όπου επιτρέπει σε ένα slave name server να ζητήσει μόνο τις αλλαγές σε μια ζώνη από τους master servers του. Αυτό κάνει τη μεταφορά ζωνών γρήγορη και ποιο αποτελεσματική, το οποίο είναι πολύ σημαντικό για μεγάλες και δυναμικές ζώνες. Εμπειρικά το BIND 9 είναι ποιο ισχυρό από την έκδοση 8. Βέβαια αυτό δεν σημαίνει ότι όλοι πρέπει να εγκαταστήσουν την έκδοση 9 από την 8. Για να καταλήξουμε σε ποια έκδοση θα πάμε πρέπει πρώτα να μελετήσουμε τις ανάγκες μας και ύστερα να δούμε τα χαρακτηριστικά της κάθε έκδοσης για να πάρουμε την κατάλληλη απόφαση.

2.12.1 ΒΡΙΣΚΟΝΤΑΣ IP ΔΙΕΥΘΥΝΣΕΙΣ

Παρακάτω θα δούμε πως μπορούμε να χρησιμοποιήσουμε κάποιο name server ενός τρίτου οργανισμού για να για να βρούμε πληροφορίες για κάποιο host. Όσο έχουμε μια σύνδεση στο Ιντερνετ και το πρόγραμμα nslookup μπορούμε να πάρουμε πληροφορίες για το χώρο ονομάτων του Ιντερνετ. Για να βρούμε την IP για τον ftp.isc.org για παράδειγμα μπορούμε να πούμε:

```
% NSLOOKUP FTP.ISC.ORG. 193.92.150.3
```

Αυτή η εντολή λέει στο πρόγραμμα nslookup να ρωτήσει τον name server που τρέχει στο host με IP διεύθυνση 193.92.150.3 να βρει την IP διεύθυνση του ftp.isc.org, και παίρνουμε το αποτέλεσμα της εντολής:

```
SERVER:  NSHER.FORTHNET.GR
ADDRESS:  193.92.150.3
NAME:    FTP.ISC.ORG
ADDRESS: 204.152.184.110
```

Η IP διεύθυνση του ftp.isc.org είναι η 204.152.184.110. Η IP του name server που χρησιμοποιήσαμε την πήραμε από τον ISP που έχουμε για την DSL σύνδεση μας. Μπορούμε να χρησιμοποιήσουμε άλλους DNS servers όπως το δημόσιο της google 8.8.8.8 ή τον 4.2.2.2 που είναι κάποιου πανεπιστημίου στην Αμερική.

2.13 ΔΙΑΛΕΓΟΝΤΑΣ ΤΟ ΟΝΟΜΑ ΓΙΑ ΤΟ DOMAIN ΜΑΣ

Τα πρώτα βήματα για να διαλέξεις ένα domain όνομα, είναι να βρεις που στο υπάρχων domain name space ανήκεις. Είναι ποιο εύκολο να αρχίσεις από τη κορυφή και να προχωρήσεις προς τα κάτω: να αποφασίσεις σε ποιο από τα ανώτερα domain ανήκεις και σε ποιο subdomain ταιριάζει καλύτερα.

2.13.1 REGISTRARS ΚΑΙ REGISTRIES

Αρχικά πρέπει να ορίσουμε μερικές ορολογίες: registry (Μητρώο), registrar (Καταχωρητής) και registration (Εγγραφή). Αυτοί οι όροι δεν αναφέρονται στις προδιαγραφές του DNS αλλά χρησιμοποιούνται στο τρόπο με τον οποίο ο χώρος ονομάτων του Ιντερνετ διαχειρίζεται σήμερα.

Το μητρώο (registry) είναι ένας οργανισμός που είναι υπεύθυνος για τη διατήρηση των αρχείων δεδομένων ενός ανώτερου (top-level) domain, τα οποία περιέχουν τις αναθέσεις για κάθε subdomain του κάθε ανώτερου domain. Στη σημερινή δομή του Ιντερνετ, ένα ανώτερο domain δεν μπορεί να έχει πάνω από ένα μητρώο. Ένας καταχωρητής (registrar) λειτουργεί ως διεπαφή ανάμεσα στο στους πελάτες και το μητρώο, παρέχοντας εγγραφές και υπηρεσίες προστιθέμενης αξίας. Όταν ένας πελάτης διαλέξει ένα subdomain μιας υψηλού επιπέδου ζώνης, ο καταχωρητής του πελάτη υποβάλει στο αρμόδιο μητρώο τα δεδομένα της ζώνης που είναι απαραίτητα για να αναθέσουν αυτό το subdomain στους name servers που καθόρισε ο πελάτης. Τα μητρώα λίγο πολύ είναι σαν τους εμπόρους χονδρικής ανάθεσης στις υψηλού επιπέδου ζώνες τους. Ενώ οι καταχωρητές (registrars) λειτουργούν ως έμποροι λιανικής πώλησης. Συνήθως μεταπωλούν την ανάθεση σε περισσότερα από ένα μητρώο. Η εγγραφή (registration) είναι η διαδικασία με την οποία ένας πελάτης λέει στο καταχωρητή σε ποιους name servers να αναθέσει ένα subdomain και παρέχει στο καταχωρητή πληροφορίες επικοινωνίας και χρέωσης. Για παράδειγμα το Public Interest Registry διευθύνει το org μητρώο. Η VeriSign λειτουργεί αυτή τη περίοδο ως το μητρώο για την com και net ανώτατα domain. Υπάρχουν δεκάδες καταχωρητές για com net και org όπως η GoDaddy.com, Regisry.com και η Network Solutions. Ένας οργανισμός που ονομάζεται EDUCAUSE λειτουργεί ως το μητρώο για το edu domain και είναι ο μόνος καταχωρητής για το edu domain.

Εφόσον είμαστε στην Ελλάδα έχουμε το δικαίωμα να ενταχθούμε είτε στο gr domain αλλά και σε ένα από τα ανώτατα γενικά domain όπως τα biz, com, info, net και org. Έστω για παράδειγμα ότι θέλουμε ένα όνομα για έναν Οργανισμό με το όνομα Lapis και επιλέγουμε

ότι το όνομα lapis.com είναι κατάλληλο για όνομα domain. Αρχικά ελέγχουμε αν το όνομα lapis.com είναι διαθέσιμο:

```
% NSLOOKUP
DEFAULT SERVER:  NS.UNET.UMN.EDU
ADDRESS:  128.101.101.101

> SET TYPE=ANY              // ΚΟΙΤΑ ΓΙΑ ΟΠΟΙΑΔΗΠΟΤΕ ΕΓΓΡΑΦΗ
> LAPIS.COM.                    // ΓΙΑ ΤΟ LAPIS.COM

SERVER:  NS.UNET.UMN.EDU
ADDRESS:  128.101.101.101
LAPIS.COM   NAMESERVER = NS3.WORLDNIC.COM
LAPIS.COM   NAMESERVER = NS4.WORLDNIC.COM
```

Βλέπουμε ότι το όνομα lapis.com δεν είναι διαθέσιμο. Επιλέγουμε ένα άλλο όνομα: lapis-institute.com

```
% NSLOOKUP
DEFAULT SERVER:  NS.UNET.UMN.EDU
ADDRESS:  128.101.101.101

> SET TYPE=ANY
> LAPIS-INSTITUTE.COM.

SERVER:  NS.UNET.UMN.EDU
ADDRESS:  128.101.101.101

*** NS.UNET.UMN.EDU CAN'T FIND LAPIS-INSTITUTE.COM.: NON-
EXISTENT HOST/DOMAIN
```

Το lapis-institute.com είναι διαθέσιμο επομένως μπορούμε να πάμε στο επόμενο βήμα και να διαλέξουμε έναν καταχωρητή (registrar).

2.13.2 ΔΙΑΛΕΓΟΝΤΑΣ REGISTRAR

Πριν το 1999 μια εταιρία μόνο η Network Solutions ήταν το μητρώο αλλά και ο καταχωρητής για τα net, com, org αλλά και edu. Για να καταχωρήσουμε ένα subdomain κάτω από ένα οποιοδήποτε από αυτά τα ανώτατα subdomain θα έπρεπε να απευθυνθούμε στην Network Solutions. Τον Ιούνιο του 1999, η ICANN, ο οργανισμός που διαχειρίζεται το χώρο ονομάτων του Ιντερνετ εισήγαγε τον ανταγωνισμό στη καταχώρηση του com, net και org. Υπάρχουν σήμερα δεκάδες com, net, org καταχωρητές που μπορεί κάθε πελάτης να διαλέξει (για παράδειγμα godaddy.com).

Προτού προχωρήσουμε παραπέρα θα πρέπει να βεβαιωθούμε ότι το IP δίκτυο μας είναι εγγεγραμμένο. Σήμερα οι ISP δεσμεύουν χώρο για τα δικά τους δίκτυα, και τα IP αυτά τα δίνουν σε πελάτες. Οι ISP 99% έχουν καταχωρημένες τις IP που δεσμεύουν. Οι οργανισμοί που εγγράφεις τις IP λέγονται Περιφερειακά Μητρώα Ιντερνετ—regional Internet registries (RIR). Για παράδειγμα στην βόρεια Αμερική ο οργανισμός American Registry of Internet Numbers –ARIN (http://www.arin.net) δίνει χώρο διευθύνσεων και εγγράφει δίκτυα. Στην

Ευρώπη υπεύθυνος είναι ο Οργανισμός RIPE Network Coordination Center (http://www.ripe.net).

Αν δεν είμαστε σίγουροι για το αν το δίκτυο μας είναι εγγεγραμμένο, ο καλύτερος τρόπος να μάθουμε είναι να χρησιμοποιήσουμε μια υπηρεσία whois που παρέχονται από τους αρμόδιους οργανισμούς. Για παράδειγμα για το RIPE:

http://www.ripe.net/perl/whois/

Παρακάτω θα κάνουμε ένα ερώτημα για την IP 195.251.90.226 που είναι η IP για το διακομιστή ιστού της κεντρικής σελίδας του ΤΕΙ Πειραιά (http://www.teipir.gr). Παρακάτω φαίνονται τα αποτελέσματα του whois ερωτήματος μας:

```
% THIS IS THE RIPE DATABASE QUERY SERVICE.
% THE OBJECTS ARE IN RPSL FORMAT.
%
% THE RIPE DATABASE IS SUBJECT TO TERMS AND CONDITIONS.
% SEE HTTP://WWW.RIPE.NET/DB/SUPPORT/DB-TERMS-CONDITIONS.PDF

% NOTE: THIS OUTPUT HAS BEEN FILTERED.
%       TO RECEIVE OUTPUT FOR A DATABASE UPDATE, USE THE "-
B" FLAG.

% INFORMATION RELATED TO '195.251.64.0 - 195.251.95.255'
% ABUSE CONTACT FOR '195.251.64.0 - 195.251.95.255' IS
'ABUSE@GRNET.GR'


INETNUM:        195.251.64.0 - 195.251.95.255
NETNAME:        TEIPIR
DESCR:          TECHNOLOGICAL EDUCATION INSTITUTE OF PIRAEUS
COUNTRY:        GR
ADMIN-C:        NTP1-RIPE
TECH-C:         NTP1-RIPE
STATUS:         ASSIGNED PA
MNT-BY:         GRNET-NOC
MNT-DOMAINS:    MNT-GRNET-DNS
SOURCE:         RIPE #FILTERED

ROLE:           NOC TEIPIR
ADDRESS:        250 THIVON AV. & P. RALLI STR.
ADDRESS:        122 44 EGALEO
ADDRESS:        ATHENS GREECE
PHONE:          +30 210 5381304
FAX-NO:         +30 210 5381261
REMARKS:        ------------------------------------
REMARKS:        FOR COMPLAINS ABOUT ABUSE, SPAM ETC:
ABUSE-MAILBOX:  ABUSE@TEIPIR.GR
REMARKS:        ------------------------------------
ADMIN-C:        MK3060-RIPE
ADMIN-C:        SL1001-RIPE
TECH-C:         MK3060-RIPE
MNT-BY:         GRNET-NOC
```

```
NIC-HDL:        NTP1-RIPE
SOURCE:         RIPE #FILTERED

% INFORMATION RELATED TO '195.251.64.0/19AS5408'

ROUTE:          195.251.64.0/19
DESCR:          TEIPIR
ORIGIN:         AS5408
MNT-BY:         GRNET-NOC
SOURCE:         RIPE #FILTERED
```

Βλέπουμε ότι ο χώρος διευθύνσεων 195.251.64.0/19 δηλαδή 195.251.64.0 – 195.251.95.255 είναι εγγεγραμμένος στο δίκτυο teipir. Στο σχόλιο:

% Information related to '195.251.64.0/19AS5408'

Το AS αναφέρεται στο *Autonomous System* του πρωτοκόλλου BGP. Το πρωτόκολλο BGP είναι ένα πρωτόκολλο δρομολόγησης και θα το δούμε στο κεφάλαιο 4. Αξίζει να σημειωθεί ότι είναι το πρωτόκολλο του διαδικτύου και κάθε AS νούμερο είναι πολύ σημαντικό γιατί το κάθε νούμερο αναγνωρίζει το κάθε δίκτυο στο Διαδίκτυο.

Αν η IP μας δεν είναι εγγεγραμμένη πρέπει να την καταχωρήσουμε προτού εγκαταστήσουμε τις in-addr.arpa ζώνες μας. Μόλις όλοι οι host μας είναι σε εγγεγραμμένα δίκτυα, μπορούμε μετά να καταχωρήσουμε τις ζώνες μας.

3 ΑΝΑΠΤΥΞΗ ΕΝΣΩΜΑΤΩΜΕΝΩΝ ΣΥΣΤΗΜΑΤΩΝ LINUX

3.1 ΕΙΣΑΓΩΓΗ

Παρά την ευρεία χρήση του Linux σε κινητά, οχήματα, αεροσκάφη, αμυντικά όπλα, διαστημικές αποστολές κλπ, υπάρχει πολύ μικρή βάση τεκμηριωμένης γνώσης για τη δημιουργία, την εγκατάσταση και τον έλεγχο του Linux Kernel και των εργαλείων που χρησιμοποιούνται κατά την ανάπτυξη ενός ενσωματωμένου συστήματος Linux (Embedded Linux System). Έτσι, πριν προχωρήσουμε στην ανάπτυξη ενός τέτοιου συστήματος θα πρέπει πρώτα να έχουμε κατανοήσει μια γενικότερη εικόνα. Σε αυτό το κεφάλαιο παρατίθενται κάποιες απαραίτητες γενικές γνώσεις που καλύπτουν το βασικό θεωρητικό υπόβαθρο ανάπτυξης ενσωματωμένων συστημάτων Linux και οι οποίες βοηθούν στην κατανόηση των παρακάτω θεμάτων:

- Τι είναι το Linux
- Το ενσωματωμένο Linux
- Οι τύποι των ενσωματωμένων συστημάτων Linux
- Γιατί να προτιμήσουμε το Linux

3.2 ΤΙ ΕΙΝΑΙ ΤΟ LINUX

Το Linux είναι ένα Λειτουργικό Σύστημα το οποίο δημιουργήθηκε αρχικά από τον Linus Torvalds και τον οργανισμό FSF (Free Software Foundation) υπό την άδεια GNU (General Public License). Ο πυρήνας του Linux παρέχει μια μεγάλη ποικιλία βασικών λειτουργιών οι οποίες απαιτούνται από κάθε σύστημα για να λειτουργήσει σωστά. Ένα επίπεδο πιο «πάνω» από τον πυρήνα βρίσκεται το λογισμικό εφαρμογών το οποίο βασίζεται σε συγκεκριμένες λειτουργίες του πυρήνα. Οι λειτουργίες αυτές αφορούν τον χειρισμό των συσκευών και την παροχή μιας ποικιλίας επιπέδων αφαίρεσης (abstraction layers), όπως είναι η εικονική μνήμη (virtual memory), οι διεργασίες (tasks ή processes), τα sockets, τα συστήματα αρχείων κλπ. Για την εκκίνηση του πυρήνα του Linux χρησιμοποιείται συνήθως κάποιο εξειδικευμένο λογισμικό εκκίνησης.

Στις μέρες μας, ο όρος "Linux" είναι κάπως συγκεχυμένος λόγω της ολοένα και αυξανόμενης δημοτικότητάς του καθώς και της καθημερινής του χρήσης από ανθρώπους οι οποίοι δεν είναι ειδικοί. Συνήθως ο όρος Linux χρησιμοποιείται εναλλακτικά όταν κάποιος θέλει να αναφερθεί είτε στον πυρήνα του Linux, είτε σε ένα σύστημα Linux, είτε σε μια εφαρμογή που έχει στηθεί επάνω στον πυρήνα του Linux.

Οι διανομές Linux διαφέρουν στο σκοπό, το κόστος και το μέγεθος αλλά έχουν τον ίδιο στόχο. Ο στόχος είναι να παρέχουν στον τελικό χρήστη ένα προ-συσκευασμένο και συμπυκνωμένο σετ αρχείων και έναν μηχανισμό εγκατάστασης έτσι ώστε να μπορεί να εγκατασταθεί ο πυρήνας και οι εφαρμογές, σε διάφορες αρχιτεκτονικές και για διάφορους σκοπούς.

3.3 ΤΟ ΕΝΣΩΜΑΤΩΜΕΝΟ LINUX

Το ενσωματωμένο Linux τυπικά αναφέρεται σε ένα πλήρες σύστημα ή σε μια διανομή που είναι στοχευόμενη για ενσωματωμένα συστήματα. Παρ' όλο που ο όρος "ενσωματωμένο" υποδηλώνει μια ειδική έκδοση Linux, δεν υπάρχει κάποιος ειδικός τύπος του Linux kernel για εφαρμογές σε ενσωματωμένα συστήματα. Ο ίδιος πηγαίος κώδικας του πυρήνα που χρησιμοποιείται σε PCs ή σε Servers, μεταγλωττίζεται και για όλα τα είδη των διάφορων συστημάτων που τον χρησιμοποιούν. Υπάρχουν όμως κάποιες παράμετροι που μπορούν να τροποποιούνται κατά τη μεταγλώττιση ώστε να αφαιρούνται περιττά χαρακτηριστικά και να προσθέτονται άλλα που είναι χρήσιμα. Για παράδειγμα, η υποστήριξη για terabytes μνήμης σε ένα ενσωματωμένο σύστημα είναι εντελώς περιττή και μπορεί να αφαιρεθεί.

Στα πλαίσια της ανάπτυξης ενσωματωμένων συστημάτων Linux, γίνεται χρήση μιας σειράς από λογισμικά. Εκτός από τις δωρεάν εκδοχές, υπάρχει και μία αρκετά μεγάλη ποικιλία εμπορικών διανομών ενσωματωμένου Linux που σχεδιάζονται ειδικά για ενσωματωμένα συστήματα. Οι διανομές αυτές παράγονται από κάποιους εξειδικευμένους προμηθευτές. Οι πιο σημαντικοί από αυτούς είναι οι: MontaVista, Wind River, Lynuxworks, Timesys και Denx. Τα εργαλεία που αναπτύσσονται από αυτές τις εταιρείες είναι: cross-compilers, debuggers, εφαρμογές διαχείρισης έργων (projects), boot image builders κλπ. Αυτό είναι ουσιαστικά και αυτό που πληρώνουμε σε αυτές τις εταιρείες όταν στρεφόμαστε σε κάποια έτοιμη λύση. Το αν θα χρησιμοποιήσουμε κάποια έτοιμη λύση βέβαια, εξαρτάται καθαρά από τις οικονομικές μας δυνατότητες και τις ειδικές μας γνώσεις στο αντικείμενο.

3.4 ΟΙ ΤΥΠΟΙ ΤΩΝ ΕΝΣΩΜΑΤΩΜΕΝΩΝ ΣΥΣΤΗΜΑΤΩΝ LINUX

Η κατηγοριοποίηση των ενσωματωμένων συστημάτων Linux δεν είναι εύκολο να γίνει με βάση την εφαρμογή τους. Για να εντοπιστούν πραγματικές διαφορές θα πρέπει να βρούμε κάποια κριτήρια που θα παρέχουν πληροφορίες σχετικά με τη δομή του κάθε συστήματος.

Τα κριτήρια αυτά είναι:

- Το μέγεθος
- Οι χρονικοί περιορισμοί
- Η δυνατότητα δικτύωσης
- Ο βαθμός αλληλεπίδρασης του χρήστη με το σύστημα.

Το μέγεθος ενός ενσωματωμένου συστήματος προσδιορίζεται από έναν αριθμό διαφορετικών παραγόντων που αφορά κυρίως τις φυσικές δυνατότητες των ολοκληρωμένων που υπάρχουν σε αυτό. Οι κυριότεροι παράγοντες είναι η ταχύτητα του μικροελεγκτή, η χωρητικότητα της κύριας μνήμης RAM και η χωρητικότητα των μέσων μόνιμης αποθήκευσης. Έτσι ανάλογα με το μέγεθός τους, τα ενσωματωμένα συστήματα χωρίζονται σε μικρά, μεσαία και μεγάλα. Τα μικρά συστήματα χαρακτηρίζονται από έναν

μικροελεγκτή των 32bit χαμηλής κατανάλωσης και μνήμη ROM των 4MB. Η μνήμη συνήθως δεν είναι πραγματική ROM αλλά Flash και μπορεί να φτάσει μέχρι και τα 32MB.

Στα μεσαία συστήματα, τα χαρακτηριστικά που εντοπίζουμε είναι: μικροελεγκτής μεσαίας κατανάλωσης, με 32MB ή και μεγαλύτερη ROM (σχεδόν πάντα NOR Flash, ή ακόμη και NAND όταν υπάρχει δυνατότητα εκτέλεσης κώδικα από block-addressable NAND FLASH μνήμες) και 64 – 128 MB RAM. Μεσαία συστήματα θεωρούνται τα mp3 players, τα PDAs, οι συσκευές δικτύωσης όπως είναι τα routers κλπ. Πρέπει να πούμε ότι κάποια από τα παραπάνω συστήματα μπορούν να υποστηρίξουν (προς το παρόν) μέχρι και 32GB NAND Flash βοηθητικής μνήμης σε δευτερεύοντες αποθηκευτικούς χώρους.

Τα μεγάλα συστήματα χαρακτηρίζονται από έναν δυνατό επεξεργαστή ή από μια ομάδα επεξεργαστών σε συνδυασμό με μεγάλα μεγέθη μνήμης RAM και μόνιμο αποθηκευτικό χώρο. Τα συστήματα αυτά χρησιμοποιούνται συνήθως σε περιβάλλοντα στα οποία εκτελούνται μεγάλοι αριθμοί απαιτητικών υπολογισμών ώστε να επιτευχθούν συγκεκριμένες διεργασίες. Οι μεγάλοι τηλεπικοινωνιακοί σταθμοί και οι προσομοιωτές πτήσης αποτελούν παραδείγματα τέτοιων μεγάλων συστημάτων. Γι' αυτά τα συστήματα το κόστος και οι πόροι που απαιτούν είναι δευτερεύοντα ζητήματα. Το ζητούμενο είναι η επίτευξη ενός στόχου με κάθε θυσία. Ένα τέτοιο παράδειγμα είναι και το αμυντικό σύστημα μιας χώρας.

Ας δούμε τώρα πως διαχωρίζονται τα ενσωματωμένα συστήματα Linux ως προς τους χρονικούς περιορισμούς. Υπάρχουν δύο τύποι χρονικών περιορισμών. Οι αυστηροί και οι ήπιοι.

Στους αυστηρούς περιορισμούς απαιτείται η ανάδραση του συστήματος να γίνεται σε ένα προκαθορισμένο χρονικό πλαίσιο, αλλιώς κάτι πολύ ανεπιθύμητο μπορεί να συμβεί. Ας πάρουμε για παράδειγμα, ένα μηχάνημα κοπής ξυλείας όπου το χέρι ενός εργάτη πλησιάζει επικίνδυνα στην κορδέλα κοπής. Αν ο αισθητήρας που έχει τοποθετηθεί για τέτοιες περιπτώσεις, στείλει το μήνυμα για το συμβάν στο σύστημα και εκείνο δεν προχωρήσει άμεσα στην ακινητοποίηση της κορδέλας, τότε σίγουρα κάποια στιγμή θα προκληθεί σοβαρό εργατικό ατύχημα. Ένα τέτοιο σύστημα λοιπόν πρέπει να δουλεύει αυστηρά και σε πραγματικό χρόνο (real time).

Τα συστήματα ήπιων χρονικών περιορισμών τα οποία είναι και τα πιο συνηθισμένα, δεν χρειάζεται να λειτουργούν σε πραγματικό χρόνο. Για παράδειγμα, ένα μηχάνημα αυτόματης ανάληψης χρημάτων δεν θα θεωρηθεί αναξιόπιστο αν αργήσει λίγο παραπάνω για να ολοκληρώσει μια εντολή συναλλαγής που του αναθέτουμε. Φυσικά ακόμη και στα συστήματα αυτά, τα χρονικά όρια πρέπει να κινούνται σε κάποια λογικά πλαίσια διαφορετικά δίνεται στον χρήστη η εντύπωση ότι δε λειτουργούν σωστά.

Συνεχίζοντας την κατηγοριοποίηση των ενσωματωμένων συστημάτων Linux θα ασχοληθούμε με το κριτήριο δυνατότητας δικτύωσης. Με τον όρο "δυνατότητα δικτύωσης" καθορίζεται αν ένα ενσωματωμένο σύστημα μπορεί να συνδεθεί σε κάποιο δίκτυο ή όχι. Στις μέρες μας, περιμένουμε σχεδόν από κάθε συσκευή που αγοράζουμε να μπορεί να είναι προσβάσιμη μέσω δικτύου. Αυτό ορισμένες φορές ισχύει ακόμα και για τις "λευκές" οικιακές συσκευές (ψυγεία, πλυντήρια, κουζίνες κλπ). Όλα αυτά προσδίδουν νέες

απαιτήσεις σε κάθε σύστημα που σχεδιάζεται. Επομένως, ένας ακόμη παράγοντας για τον οποίο επιλέγεται το Linux είναι και οι δυνατότητες δικτύωσης που παρέχει ο πυρήνας του.

Ολοκληρώνοντας την αναφορά μας στους τύπους των ενσωματωμένων συστημάτων Linux θα εξετάσουμε την κατηγοριοποίησή τους ως προς τον βαθμό αλληλεπίδρασής τους με τον τελικό χρήστη. Ο βαθμός αυτός είναι διαφορετικός για κάθε σύστημα. Κάποια συστήματα όπως είναι τα tablet PCs και τα PDAs βασίζονται σχεδόν ολοκληρωτικά στην αλληλεπίδρασή τους με τον χρήστη παρέχοντάς ένα πλούσιο User Interface με οθόνες αφής, πλούσια μενού και ηχητικές εντολές, ενώ άλλα, όπως είναι για παράδειγμα τα βιομηχανικά συστήματα ελέγχου παραγωγής, παρέχουν μόνο κάποια LEDs ενδείξεων και κουμπιά.

3.5 ΓΙΑΤΙ ΝΑ ΠΡΟΤΙΜΗΣΟΥΜΕ ΤΟ LINUX

Υπάρχει μεγάλο φάσμα κινήτρων για την ενσωμάτωση του Linux σε ένα ενσωματωμένο σύστημα. Πολλά από αυτά τα κίνητρα είναι ίδια με εκείνα που μας κάνουν να επιλέγουμε το Linux ως Λειτουργικό Σύστημα στους προσωπικούς υπολογιστές, τους servers και στους χώρους των επιχειρήσεων, ενώ άλλα είναι πιο εξειδικευμένα και αφορούν αποκλειστικά την φύση των ενσωματωμένων συστημάτων. Γενικά οι σημαντικότεροι λόγοι που μας κάνουν να θέλουμε να χρησιμοποιούμε το Linux είναι οι παρακάτω:

- Ποιότητα και αξιοπιστία του κώδικα
- Διαθεσιμότητα του κώδικα
- Ευρεία υποστήριξη υλικού
- Standards για πρωτόκολλα επικοινωνίας και λογισμικό
- Διαθέσιμα εργαλεία
- Υποστήριξη από την κοινότητα
- Άδειες χρήσης λογισμικού
- Ανεξαρτησία από τον προμηθευτή
- Κόστος

Ποιοτικός κώδικας είναι ο κώδικάς που προσφέρει επεκτασιμότητα, έχει σωστή δομή, είναι ευανάγνωστος και παραμετροποιείται εύκολα. Η επεκτασιμότητα έχει να κάνει κυρίως με τη δυνατότητα εύκολης προσθήκης νέων λειτουργιών. Για να μπορεί όμως να είναι εύκολο κάτι τέτοιο θα πρέπει μέσα στον κώδικα κάθε ξεχωριστή λειτουργία να έχει τη δική της ξεχωριστή, ευδιάκριτη και ευανάγνωστη ενότητα. Η εύκολη παραμετροποίηση προκύπτει από τη δυνατότητα που παρέχει ο κώδικάς μας για την επιλογή των χαρακτηριστικών τα οποία θα είναι ή όχι, διαθέσιμα στην τελική εφαρμογή.

Από την άλλη, αξιόπιστος κώδικας είναι ο κώδικας που παρέχει προβλεπτικότητα, ανοχή σε σφάλματα και βιωσιμότητα. Η προβλεπτικότητα αφορά τη συμπεριφορά της τελικής εφαρμογής η οποία θα πρέπει να βρίσκεται μέσα στα πλαίσια που είχαν οριστεί εξ' αρχής από εμάς. Η ανοχή σε σφάλματα αφορά την ομαλή ανάδραση σε προβληματικές περιστάσεις όπου επιπλέον θα πρέπει μέσω κατάλληλων μηνυμάτων να ειδοποιείται ο προγραμματιστής σχετικά με τη θέση μέσα στον κώδικα αλλά και το λόγο που προέκυψε το κάθε ξεχωριστό σφάλμα.

Τέλος, η βιωσιμότητα αφορά την αδιάκοπη και ακέραια λειτουργία της εφαρμογής χωρίς κάποια υποβοήθηση από τον χρήστη για μεγάλα χρονικά διαστήματα. Οι περισσότεροι προγραμματιστές – μηχανικοί που έχουν ασχοληθεί με τον πυρήνα του Linux, πιστεύουν ότι ο κώδικάς του πληροί όλες τις παραπάνω προϋποθέσεις και μπορεί να χαρακτηριστεί ποιοτικός και αξιόπιστος. Ως προς τη διαθεσιμότητα του κώδικα, είναι φανερό ότι το Linux υπερέχει κατά πολύ. Τόσο ο πηγαίος κώδικας, όσο και τα εργαλεία για να τον μεταγλωττίσουμε, είναι διαθέσιμα χωρίς περιορισμούς στην πρόσβασή τους από εμάς. Τα πιο σημαντικά στοιχεία του Linux συμπεριλαμβανομένου του kernel, διανέμονται υπό την άδεια χρήσης λογισμικού, GNU GPL (General Public License).

Όταν κατά καιρούς προκύπτουν προβλήματα στην πρόσβαση του πηγαίου κώδικα, η κοινότητα προσπαθεί άμεσα να τον αντικαταστήσει με κάποιον άλλο αντίστοιχων ιδιοτήτων. Επίσης, οι διορθώσεις για προβλήματα ασφαλείας είναι άμεσα διαθέσιμες και μπορούμε να αναβαθμίζουμε εύκολα και γρήγορα με αυτές το σύστημά μας. Τα κυριότερα πλεονεκτήματα που προκύπτουν από τη διαθεσιμότητα του πηγαίου κώδικα, είναι η δυνατότητα που μας παρέχεται να μπορούμε να τον διορθώνουμε, να τον τροποποιούμε και να ψάχνουμε βαθύτερα σε αυτόν έτσι ώστε να καταλαβαίνουμε ευκολότερα τις λειτουργίες του και τις ιδιαιτερότητές του.

Ένας άλλος λόγος ο οποίος μας οδηγεί στην επιλογή του Linux, είναι η ευρεία υποστήριξη υλικού που παρέχει. Το Linux υποστηρίζει πολλούς διαφορετικούς τύπους πλατφορμών υλικού και συσκευών. Επειδή οι περισσότεροι drivers γράφονται από την κοινότητα, μπορούμε να τους χρησιμοποιήσουμε με τη σιγουριά ότι δεν θα πάψουν να υποστηρίζονται στο μέλλον όπως πιθανόν θα συνέβαινε σε περίπτωση που είχαν δημιουργηθεί από κάποια εταιρεία.

Ευρεία υποστήριξη υλικού, σημαίνει επίσης ότι το Linux τρέχει σε δεκάδες διαφορετικές αρχιτεκτονικές μικροελεγκτών. Έτσι, βλέποντας κάποιον μικροελεγκτή μπορούμε να σκεφτούμε ότι πιθανότατα κάποιος έχει ήδη μπει στη διαδικασία προσαρμογής και παραμετροποίησης του πυρήνα ώστε να τον υποστηρίζει. Μπορούμε επίσης να περιμένουμε ότι η εφαρμογή που γράφουμε σε κάποια πλατφόρμα θα μπορεί εύκολα να μεταφερθεί σε μια άλλη με πολύ μικρές αλλαγές. Αυτό ισχύει και για τους drivers.

Ως προς τα πρότυπα του λογισμικού και των πρωτοκόλλων επικοινωνίας, το Linux παρέχει ευρεία υποστήριξη. Κάτι τέτοιο καθιστά εύκολη την ενσωμάτωση του σε ήδη υπάρχοντα frameworks καθώς και την ενσωμάτωση παλιότερων εκδόσεων λογισμικού σε αυτό. Έτσι, για παράδειγμα μπορεί εύκολα κάποιος να συνδέσει κάποιο σύστημα Linux σε ένα ήδη υπάρχον δίκτυο Windows και να περιμένει από αυτό να εξυπηρετεί αιτήματα μέσω του πρωτοκόλλου SAMBA.

Το Linux μοιάζει με το Unix και έτσι μπορούμε να μεταφέρουμε παλιές εφαρμογές του δεύτερου σε αυτό. Στη πραγματικότητα, πολλές εφαρμογές που υπάρχουν εγκατεστημένες στις διάφορες διανομές, έχουν αρχικά γραφτεί για εμπορικές εκδόσεις του Unix και αργότερα μεταφέρθηκαν (ported) σε συστήματα Linux. Σήμερα αρκετός κώδικας γράφεται για Linux πάντα με το κριτήριο της μεταφερσιμότητας. Μεταφερσιμότητα ακόμα και για συστήματα που δεν είναι Linux, αφού είναι δυνατόν να τρέχουμε εφαρμογές Linux και σε Windows μέσω κάποιων βιβλιοθηκών συμβατότητας όπως είναι για παράδειγμα το Cygwin.

Από τα παραπάνω αντιλαμβανόμαστε ότι υπάρχουν πολλά διαθέσιμα εργαλεία που έχουν γραφτεί για Linux και το γεγονός αυτό το κάνει ένα πολύ ευέλικτο λειτουργικό σύστημα. Αν σκεφτούμε κάποια εφαρμογή την οποία χρειαζόμαστε είναι σχεδόν σίγουρο ότι κάποιοι από την κοινότητα θα έχουν ήδη νιώσει την ανάγκη να τη δημιουργήσουν και να τη διαθέσουν δωρεάν στο Internet. Για να το αντιληφθεί καλύτερα αυτό κάποιος, αρκεί να επισκεφτεί τις ιστοσελίδες freshmeat.net και sourcefourge.net .

Η υποστήριξη του Linux από την κοινότητά του είναι ένα σημαντικότατο πλεονέκτημα που έχει σε σχέση με άλλα Λειτουργικά Συστήματα. Σε αυτή την κοινότητα μπορούμε να νιώσουμε απόλυτα το πνεύμα του δωρεάν και ελεύθερου λογισμικού. Επίσης μέσω των αδειών χρήσης λογισμικού, μπορούμε να κάνουμε πράγματα που ούτε θα μπορούσαμε να τα φανταστούμε με βάση του τι ισχύει στην υπόλοιπη αγορά. Στην ουσία, μπορούμε να χρησιμοποιήσουμε, να τροποποιήσουμε και να αναδιανείμουμε το λογισμικό μας με μοναδικό περιορισμό την παροχή των ίδιων ακριβώς δικαιωμάτων και στους αποδέκτες του.

Είδαμε έως τώρα αρκετά από τα κυριότερα πλεονεκτήματα που μας δίνουν σημαντικά κίνητρα και λόγους έτσι ώστε να θέλουμε να χρησιμοποιήσουμε το Linux. Στη συνέχεια θα αναφέρουμε δύο ακόμα. Το πλεονέκτημα της ανεξαρτησίας σε σχέση με τον προμηθευτή (vendor) και το πλεονέκτημα του κόστους του Linux.

Ανεξαρτησία από τον προμηθευτή σημαίνει ότι δε χρειάζεται να βασιστούμε σε κάποιον για να προμηθευτούμε το Linux ή για να το χρησιμοποιήσουμε. Αν όμως έχουμε επιλέξει ήδη κάποια εμπορική διανομή ενός προμηθευτή και είμαστε δυσαρεστημένοι, μπορούμε να τον αλλάξουμε αφού στην ουσία έχουμε τα ίδια δικαιώματα με αυτόν. Ορισμένοι προμηθευτές παρέχουν επιπλέον λογισμικό στις διανομές τους, που δεν είναι open source. Για το κομμάτι αυτό θα πρέπει να βρεθεί μια δική μας λύση ή κάποιος άλλος προμηθευτής. Τέτοια ζητήματα πρέπει να λαμβάνονται σοβαρά υπ' όψιν όταν επιλέγουμε διανομή για το ενσωματωμένο μας σύστημα.

Αφήσαμε το κόστος για το τέλος μιας και δεν έχει να κάνει με το τεχνικό κομμάτι του Linux. Είναι όμως, ίσως το σημαντικότερο πλεονέκτημα του Linux, σε σχέση με άλλες λύσεις που υπάρχουν στην αγορά λογισμικού. Ιδιαίτερα τώρα που αυτές οι γραμμές γράφονται σε περίοδό βαθειάς οικονομικής κρίσης. Γενικά, υπάρχουν τρία τμήματα λογισμικού που κοστίζουν κατά την ανάπτυξη ενός κλασσικού ενσωματωμένου συστήματος:

- το αρχικό περιβάλλον ανάπτυξης
- τα επιπρόσθετα εργαλεία
- τα δικαιώματα χρήσης

Το «μηδενικό» κόστος του Linux είναι αποτέλεσμα των αδειών χρήσης ανοικτού λογισμικού και διαφέρει από οποιοδήποτε άλλο ενσωματωμένο λειτουργικό σύστημα. Με τη χρήση του Linux τα περισσότερα εργαλεία ανάπτυξης και τα τμήματα του λειτουργικού είναι δωρεάν και οι άδειες υπό τις οποίες προστατεύονται, προστατεύουν την οικονομική εκμετάλλευσή τους.

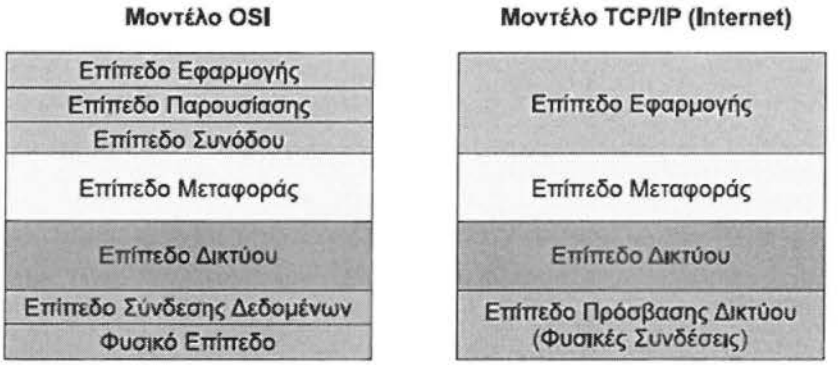
4 ΜΟΝΤΕΛΟ ΑΝΑΦΟΡΑΣ TCP/IP

4.1 ΣΧΕΣΗ OSI ΚΑΙ TCP/IP

Το TCP/IP και το OSI ουσιαστικά αναπτύχθηκαν ταυτόχρονα. Δεν υπάρχει στην πραγματικότητα σύγκρουση μεταξύ των δύο προτύπων, ωστόσο υπάρχουν κάποιες ουσιαστικές διαφορές.

Το TCP/IP χρησιμοποιεί επίσης το ίδιο μοντέλο. Μια από τις βασικές διαφορές των δύο είναι ότι το OSI χρησιμοποιεί επτά επίπεδα ενώ το TCP/IP μόνο τέσσερα. Αυτό σημαίνει ότι δεν υπάρχει αντιστοιχία των επιπέδων ένα – προς –ένα. Όπως μπορείτε να δείτε στην εικόνα 4-1, πλήρης αντιστοιχία υπάρχει στα επίπεδα μεταφοράς και δικτύου. Τα επίπεδα εφαρμογής, παρουσίασης και συνόδου του OSI συνδυάζονται στο επίπεδο εφαρμογής του TCP/IP, ενώ και τα επίπεδα σύνδεσης δεδομένων και φυσικό συνδυάζονται στο επίπεδο πρόσβασης δικτύου. Ο συνδυασμός των επιπέδων σύνδεσης δεδομένων και φυσικού στο TCP/IP είναι απαραίτητος, καθώς βασική αρχή της τεχνολογίας TCP/IP είναι η υλοποίηση πρωτοκόλλου χωρίς σύνδεση.

Στην πραγματικότητα ωστόσο, ακόμα και στο μοντέλο OSI το επίπεδο σύνδεσης δεδομένων και το φυσικό επίπεδο συνδυάζονται σε ένα έξυπνο ελεγκτή (κάρτα) δικτύου.



Εικόνα 4-1: Μοντέλο OSI και TCP/IP

Στην εικόνα 4-2 παρουσιάζονται τα επίπεδα του TCP/IP σε σχέση με τα επίπεδα του OSI ενώ παρουσιάζονται και τα πρωτόκολλα που χρησιμοποιούνται για την υλοποίηση κάθε επιπέδου. Πάνω από τα πρωτόκολλα TCP/IP, βρίσκονται τα πρωτόκολλα που χρησιμοποιούνται στο επίπεδο εφαρμογής. Τα πρωτόκολλα αυτά έχουν δημιουργηθεί με τέτοιο τρόπο ώστε να χρησιμοποιούν για την επικοινωνία είτε το *Πρωτόκολλο Ελέγχου Μετάδοσης TCP* είτε το *Πρωτόκολλο Αυτοδύναμων Πακέτων Χρήστη, User Datagram Protocol, UDP* στο επίπεδο μεταφοράς. Στο επίπεδο δικτύου χρησιμοποιείται το *Πρωτόκολλο Διαδικτύου, IP* καθώς και το *Πρωτόκολλο Μηνύματος Ελέγχου Διαδικτύου, Internet Control Message Protocol, ICMP*.

	Εφαρμογές	Εφαρμογές
Επίπεδο Εφαρμογής	Telnet, FTP, SMTP	TFTP
Επίπεδο Μεταφοράς	TCP	UDP
Επίπεδο Δικτύου	IP/ICMP	

Εικόνα 4-2: Στοίβα Πρωτοκόλλων TCP/IP

Τα πρωτόκολλα εφαρμογής που φαίνονται στην αριστερή στήλη χρησιμοποιούν το πρωτόκολλο TCP στο επίπεδο μεταφοράς. Τα πρωτόκολλα εφαρμογής της δεξιάς στήλης χρησιμοποιούν το πρωτόκολλο UDP στο επίπεδο μεταφοράς. Και στις δύο περιπτώσεις, στο επίπεδο δικτύου χρησιμοποιούνται τα πρωτόκολλα IP και ICMP. Τα πρωτόκολλα που αναφέρονται στην εικόνα είναι:

4.1.1 ΕΠΙΠΕΔΟ ΠΡΟΣΒΑΣΗΣ ΔΙΚΤΥΟΥ

Το επίπεδο πρόσβασης δικτύου παρέχει την πρόσβαση στο φυσικό μέσο στο οποίο η πληροφορία μεταδίδεται με την μορφή πακέτων. Το επίπεδο πρόσβασης δικτύου αντιπροσωπεύει το χαμηλότερο επίπεδο λειτουργικότητας που απαιτείται από ένα δίκτυο και περιλαμβάνει όλα τα στοιχεία της φυσικής σύνδεσης: καλώδια, κάρτες δικτύου, πρωτόκολλα πρόσβασης τοπικών δικτύων. Όπως κάθε επίπεδο στο TCP/IP (αλλά και στο OSI), το επίπεδο αυτό παρέχει τις υπηρεσίες του στο αμέσως ανώτερο επίπεδο, το επίπεδο δικτύου. Στην τεχνολογία TCP/IP δεν υπάρχουν προδιαγραφές για τα χαμηλότερα επίπεδα του επιπέδου δικτύου και έτσι μπορούν να χρησιμοποιούνται εντελώς διαφορετικές τεχνολογίες. Αυτό πρακτικά σημαίνει ότι το TCP/IP μπορεί να χρησιμοποιηθεί σε διαφορετικά φυσικά μέσα και τεχνολογίες (Ethernet, Token ring κλπ).

4.1.2 ΕΠΙΠΕΔΟ ΔΙΚΤΥΟΥ

Το επίπεδο αυτό είναι υπεύθυνο για τη μετάδοση στο φυσικό δίκτυο των πακέτων που δημιουργούνται από τα πρωτόκολλα TCP και UDP που βρίσκονται στο αμέσως ανώτερο επίπεδο (Μεταφοράς). Το βασικό πρωτόκολλο που χρησιμοποιείται σε αυτό το επίπεδο είναι το IP ή πρωτόκολλο Διαδικτύου και είναι αυτό που μας εξασφαλίζει την παγκόσμια συνδεσιμότητα. Το πρωτόκολλο IP είναι υπεύθυνο για την παροχή λογικών διευθύνσεων (IP) στα σημεία διεπαφής του με το φυσικό δίκτυο (σε κάθε δηλ. συσκευή του δικτύου που διαθέτει δική της διεύθυνση). Είναι επίσης υπεύθυνο για την αντιστοίχηση των λογικών (IP) διευθύνσεων με τις φυσικές διευθύνσεις.

Οι φυσικές διευθύνσεις παρέχονται από το επίπεδο πρόσβασης δικτύου (φυσικό επίπεδο) ή από το υπό-επίπεδο ελέγχου προσπέλασης μέσου MAC Media Access Control) του OSI. Το πρωτόκολλο IP παρέχει λογικές διευθύνσεις στα σημεία διεπαφής του με το φυσικό δίκτυο ενώ υπάρχει και αντιστοίχηση των λογικών διευθύνσεων με

φυσικές. Για τις εργασίες αυτές χρησιμοποιούνται τα πρωτόκολλα *ARP(Address Resolution Protocol)* και *RARP (Reverse Address Resolution Protocol).*

- ARP: Πρωτόκολλο Μετατροπής Διευθύνσεων
- RARP: Πρωτόκολλο Ανάστροφης Μετατροπής Διευθύνσεων

Στο επίπεδο δικτύου λειτουργεί επίσης και το πρωτόκολλο *ICMP, Internet Control Message Protocol* ή *Πρωτόκολλο Ελέγχου Μεταφοράς Μηνυμάτων*. Αυτό χρησιμοποιείται για να αναφέρει προβλήματα και ασυνήθιστες καταστάσεις που σχετίζονται με το πρωτόκολλο IP. Συνήθως δημιουργεί και μεταφέρει μηνύματα που έχουν να κάνουν με την κατάσταση λειτουργίας των συσκευών του δικτύου. Δημιουργεί επίσης και μεταφέρει μηνύματα που σχετίζονται με την ίδια τη λειτουργία του TCP/IP και όχι από κάποια εφαρμογή που εκτελεί ο χρήστης. Για παράδειγμα όταν κάποιος προσπαθεί να συνδεθεί σε ένα υπολογιστή ο οποίος δεν είναι διαθέσιμος τη δεδομένη στιγμή (π.χ. γιατί δεν είναι ενεργός ή γιατί υπάρχει πρόβλημα στο συγκεκριμένο τμήμα του δικτύου) θα λάβει ένα μήνυμα ότι ο υπολογιστής είναι "απρόσιτος".

4.1.3 ΕΠΙΠΕΔΟ ΜΕΤΑΦΟΡΑΣ

Το επίπεδο μεταφοράς είναι υπεύθυνο για την υλοποίηση των συνδέσεων μεταξύ των υπολογιστών ενός δικτύου. Το βασικό πρωτόκολλο εδώ είναι το TCP (πρωτόκολλο με σύνδεση) ενώ μπορεί να χρησιμοποιηθεί και το UDP (πρωτόκολλο χωρίς σύνδεση). Το TCP είναι υπεύθυνο για την αποκατάσταση αξιόπιστων ταυτόχρονων συνδέσεων διπλής κατεύθυνσης.

Η έννοια του *αξιόπιστου* είναι ότι το TCP αναλαμβάνει να διορθώσει τα λάθη που τυχόν παρουσιάζονται στη μετάδοση (π.χ. μεταδίδοντας ξανά ένα πακέτο που χάθηκε ή αλλοιώθηκε). Το TCP παρέχει τις υπηρεσίες του στο αμέσως ανώτερο επίπεδο (Εφαρμογής). Καθώς θεωρείται ότι οι συνδέσεις που παρέχει είναι αξιόπιστες, τα προγράμματα στο επίπεδο εφαρμογής δεν κάνουν κανένα έλεγχο για ορθότητα των δεδομένων που προέρχονται από το TCP.

Η έννοια του *ταυτόχρονου* είναι ότι ένας υπολογιστής μπορεί σε μια δεδομένη στιγμή να διατηρεί πλήθος διαφορετικών συνδέσεων TCP οι οποίες να λειτουργούν όλες μαζί αλλά καμιά να μην επηρεάζει την άλλη. *Επικοινωνία διπλής κατεύθυνσης* σημαίνει ότι μέσω μιας σύνδεσης μπορούν ταυτόχρονα να μεταδίδονται και να λαμβάνονται δεδομένα.

Το πρωτόκολλο αυτοδύναμων πακέτων UDP είναι ένα πρωτόκολλο χωρίς σύνδεση. Δεν είναι ιδιαίτερα αξιόπιστο, αλλά επειδή είναι λιγότερο πολύπλοκο χρησιμοποιείται σε περιπτώσεις που η αξιοπιστία δεν είναι κρίσιμη και δεν είναι η επιθυμητή η χρήση του TCP.

Παραδείγματα κατανόησης UDP: Μια μετάδοση ραδιοφώνου μέσω Internet μπορεί να χρησιμοποιεί μετάδοση με πακέτα UDP. Αν κάποια πακέτα χαθούν ή αλλοιωθούν θα έχει σαν αποτέλεσμα την προσωρινή διακοπή ή παραμόρφωση του ήχου.

Ωστόσο στη συγκεκριμένη εφαρμογή αυτό δεν είναι κρίσιμο. Από την άλλη δεν θα μπορούσαμε να κατεβάσουμε αρχεία μέσω UDP χωρίς έξτρα έλεγχο λαθών (ο οποίος θα πρέπει προφανώς να γίνει πλέον στο επίπεδο εφαρμογής). Διαφορετικά τα περιεχόμενα τους θα μπορούσαν να είναι κατεστραμμένα, χωρίς να μπορούμε να το αντιληφθούμε άμεσα.

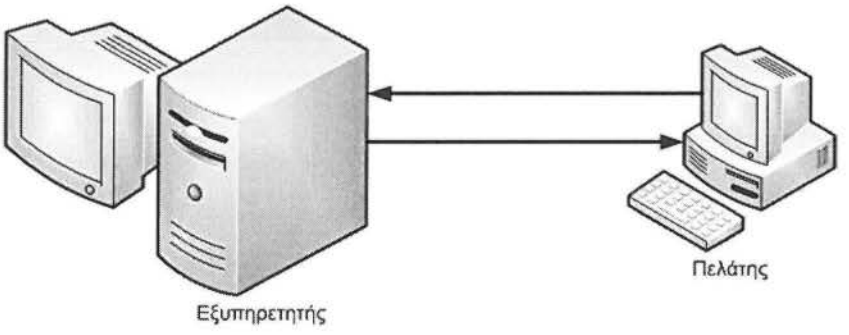
4.1.4 ΕΠΙΠΕΔΟ ΕΦΑΡΜΟΓΗΣ

Το επίπεδο εφαρμογής παρέχει τις εφαρμογές (προγράμματα) που χρησιμοποιούν τα πρωτόκολλα του επιπέδου μεταφοράς. Παραδείγματα δώσαμε στην προηγούμενη ενότητα (μεταφορά αρχείων, ηλεκτρονικό ταχυδρομείο, απομακρυσμένη πρόσβαση). Το επίπεδο εφαρμογής είναι και το σημείο που ο τελικός χρήστης έρχεται σε επαφή με την στοίβα πρωτοκόλλων της τεχνολογίας TCP/IP.

Στην εικόνα 4-3 φαίνεται το βασικό μοντέλο επικοινωνίας που χρησιμοποιείται στις περισσότερες εφαρμογές TCP/IP και το οποίο δεν είναι άλλο από το μοντέλο *πελάτη – εξυπηρετητή*. Ο εξυπηρετητής είναι μια διεργασία (πρόγραμμα) η οποία εκτελείται σε ένα υπολογιστή (γνωστός ως server) και ελέγχει τις εισερχόμενες αιτήσεις πελατών για να δει αν κάποια απευθύνεται προς αυτήν. Αν υπάρχει κάποια τέτοια αίτηση, ο εξυπηρετητής αναλαμβάνει να βρει τα δεδομένα που ζητούνται και να τα στείλει στον πελάτη.

Ο πελάτης είναι πάλι αντίστοιχα το πρόγραμμα που χρησιμοποιείται (συνήθως από τον τελικό χρήστη) για να ζητήσει τα δεδομένα από τον εξυπηρετητή. Ο πελάτης στέλνει την αντίστοιχη αίτηση και περιμένει να λάβει τα δεδομένα που ζήτησε. Με το τέλος της εξυπηρέτησης ενός πελάτη, ο εξυπηρετητής επιστρέφει ξανά σε κατάσταση αναμονής, περιμένοντας νέα αίτηση (Σημείωση: Τυπικά ένας εξυπηρετητής είναι σε θέση να εξυπηρετήσει ταυτόχρονα περισσότερες από μια αιτήσεις).

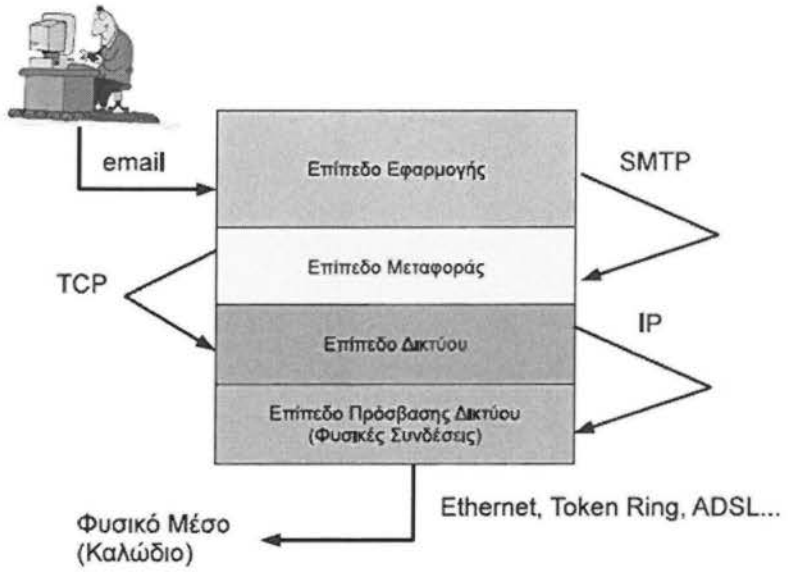
Όταν χρησιμοποιούμε έναν web browser για να συνδεθούμε σε μια ιστοσελίδα, το πρόγραμμα αυτό λειτουργεί ως πελάτης. Ζητάει τα δεδομένα της ιστοσελίδας από τον αντίστοιχο *εξυπηρετητή ιστοσελίδων (Web Server)* ο οποίος εκτελείται στο μηχάνημα που προσπαθούμε να συνδεθούμε.



Εικόνα 4-3: Πρότυπο Πελάτη - Εξυπηρετητή

4.2 ΒΑΣΙΚΕΣ ΑΡΧΕΣ ΕΠΙΚΟΙΝΩΝΙΑΣ ΣΤΟ TCP/IP

Για να αντιληφθούμε την επικοινωνία σύμφωνα με το μοντέλο TCP/IP, αρκεί να κατανοήσουμε την εικόνα 4-4. Όπως βλέπουμε, στο υψηλότερο επίπεδο (εφαρμογών) του TCP/IP βρίσκονται οι εφαρμογές οι οποίες χρησιμοποιούν τα επίπεδα που βρίσκονται κάτω από αυτό, δηλ. τα μεταφοράς, δικτύου και πρόσβασης δικτύου.



Εικόνα 4-4: Εικόνα 4-4 Επικοινωνία επιπέδων TCP/IP

Το επίπεδο εφαρμογών του δικού μας υπολογιστή μπορεί να θεωρηθεί ότι επικοινωνεί με το αντίστοιχο επίπεδο εφαρμογών του απομακρυσμένου προκειμένου να ολοκληρωθεί μια εργασία (για παράδειγμα η αποστολή ενός μηνύματος Ηλεκτρονικού Ταχυδρομείου με βάση το πρωτόκολλο SMTP το οποίο ανήκει στο επίπεδο εφαρμογής). Τα ενδιάμεσα επίπεδα προσαρμόζουν και μεταφέρουν τα δεδομένα που παράγονται από το επίπεδο εφαρμογής. Στο παράδειγμα μας χρησιμοποιούμε το πρωτόκολλο SMTP:

- Τα αρχικά δεδομένα παράγονται από την εφαρμογή του χρήστη και παραδίδονται στο πρωτόκολλο που εκτελείται στο επίπεδο εφαρμογής.
- Το SMTP προσθέτει τις εντολές και τα μηνύματα που απαιτούνται για να γίνει η επικοινωνία με τον απομακρυσμένο εξυπηρετητή SMTP.
- Τα δεδομένα από το επίπεδο εφαρμογής παραδίδονται στο επίπεδο μεταφοράς στο οποίο μετατρέπονται σε πακέτα TCP ή UDP ανάλογα με την εφαρμογή.
- Τα πακέτα από το επίπεδο μεταφοράς εισέρχονται στο επίπεδο δικτύου όπου προστίθενται οι πληροφορίες διεύθυνσης IP που απαιτούνται για τη δρομολόγηση.
- Τέλος, μεταφέρονται στο επίπεδο πρόσβασης δικτύου όπου προσαρμόζονται στο πρωτόκολλο του φυσικού μέσου (Ethernet, ADSL, token ring κλπ) και παραδίδονται μέσα από τη δικτυακή συσκευή (π.χ. την κάρτα δικτύου) στο φυσικό μέσο.

Προφανώς, στην μεριά του παραλήπτη ακολουθείται η αντίστροφη διαδικασία. Τα δεδομένα εισέρχονται από το φυσικό μέσο (επίπεδο πρόσβασης δικτύου) και ανεβαίνουν τα επίπεδα προς τα πάνω, όπου διαδοχικά ανασυνθέτονται μέχρι να φτάσουν στο επίπεδο εφαρμογής και να παραληφθούν από το πρωτόκολλο SMTP. Το πρωτόκολλο SMTP θεωρεί ότι η μετάδοση είναι αξιόπιστη (μη ξεχνάμε ότι γίνεται μέσω TCP, το οποίο είναι αξιόπιστο πρωτόκολλο). Ο έλεγχος λαθών (π.χ. πακέτα που χάθηκαν ή αλλοιώθηκαν) γίνεται στο επίπεδο μεταφοράς από το πρωτόκολλο TCP.

Οι εφαρμογές που χρησιμοποιούν τα πρωτόκολλα TCP/IP χρησιμοποιούν γενικά τέσσερα επίπεδα:

- *Πρωτόκολλο εφαρμογής:* Π.χ. SMTP, FTP, HTTP. Ανάλογα με το πρωτόκολλο εφαρμογής θα επιλεχθεί και το κατάλληλο πρωτόκολλο μεταφοράς (TCP ή UDP).
- *Πρωτόκολλο μεταφοράς:* TCP ή UDP. Έχουμε ήδη πει τις διαφορές τους. Παρέχουν τις υπηρεσίες τους στα πρωτόκολλα εφαρμογών.
- *Πρωτόκολλο δικτύου:* Το IP που παρέχει τις υπηρεσίες για τη μεταφορά των πακέτων στον προορισμό τους.
- *Πρωτόκολλα πρόσβασης δικτύου (φυσικού μέσου):* Απαιτούνται για τη διαχείριση του φυσικού μέσου (π.χ. Ethernet).

Η τεχνολογία TCP/IP βασίζεται σε μοντέλο που θεωρεί ότι οι υπολογιστές συνδέονται μεταξύ τους διαμέσου ενός μεγάλου αριθμού δικτύων. Με λίγα λόγια, τα δεδομένα από τον υπολογιστή πηγής θα περάσουν από ένα αριθμό ενδιάμεσων μηχανημάτων μέχρι να φτάσουν στον υπολογισμό προορισμού. Τα δίκτυα αυτά συνδέονται μεταξύ τους με τη βοήθεια ειδικών μηχανημάτων που ονομάζονται *δρομολογητές*.

Η αποστολή των πακέτων πρέπει να γίνεται με τέτοιο τρόπο ώστε ο χρήστης να μην αντιλαμβάνεται την διαδικασία (πρέπει να είναι *διάφανη*). Έτσι ο χρήστης δεν χρειάζεται να γνωρίζει από ποια ενδιάμεσα μηχανήματα και δρομολογητές θα περάσουν τα πακέτα για να φτάσουν στον προορισμό τους. Το μόνο που χρειάζεται να γνωρίζει πρακτικά, είναι η διεύθυνση IP του παραλήπτη.

Έστω ότι μια εφαρμογή στον υπολογιστή πηγής θέλει να επικοινωνήσει με την αντίστοιχη στον υπολογιστή προορισμού:

- Τα δεδομένα δημιουργούνται στο επίπεδο εφαρμογής του υπολογιστή αποστολής και κατεβαίνουν τα επίπεδα προς τα κάτω, σχηματίζοντας το πακέτο που πρόκειται τελικά να μεταδοθεί. Φτάνοντας στο επίπεδο πρόσβασης δικτύου, το πακέτο μεταβιβάζεται στο τοπικό δίκτυο του υπολογιστή αποστολής.
- Το πακέτο κατευθύνεται στο δρομολογητή του τοπικού δικτύου ο οποίος αναγνωρίζει ότι έχει προορισμό το Internet και το προωθεί (Ο τοπικός δρομολογητής είναι συνδεδεμένος με κάποιο δρομολογητή στο Διαδίκτυο).
- Το πακέτο κινείται από δρομολογητή σε δρομολογητή μέσω του επικοινωνιακού υποδικτύου (των ενδιάμεσων δρομολογητών) μέχρι να φτάσει στο δίκτυο προορισμού. Ο κάθε δρομολογητής από τον οποίο περνάει το πακέτο, αναλύει την επικεφαλίδα του και βρίσκει αν προορίζεται για το δικό

του δίκτυο. Αν αυτό δεν συμβαίνει το στέλνει σε άλλο δρομολογητή, ανάλογα με τη διεύθυνση IP που βρήκε στην επικεφαλίδα.

- Όταν το πακέτο βρεθεί στο δίκτυο προορισμού, παραλαμβάνεται από τον αντίστοιχο δρομολογητή και παραδίδεται στο τοπικό δίκτυο. Από εκεί οδηγείται στον υπολογιστή προορισμού όπου και ανεβαίνει ανάποδα τα επίπεδα μέχρι να φτάσει στο επίπεδο εφαρμογής. Τελικά, το επίπεδο εφαρμογής θα δώσει το πακέτο στην κατάλληλη εφαρμογή ολοκληρώνοντας έτσι τη διαδικασία μεταφοράς του πακέτου.

4.2.1 ARP

Το Address Resolution Protocol (ARP) (πρωτόκολλο επίλυσης διευθύνσεων) ορίστηκε στο RFC 826 και χρησιμοποιείται για να βρεθεί μια διεύθυνση του επιπέδου συνδέσμου (link layer) ή διεύθυνση υλικού (hardware address) ενός ξένιου υπολογιστή με βάση μια διεύθυνση του επιπέδου επικοινωνίας (network layer). Αν και το συναντάμε κυρίως με τα πρωτόκολλα IPv4 και Ethernet (το RFC 826 το ονομάζει πρωτόκολλο επίλυσης διευθύνσεων Ethernet (Ethernet Address Resolution Protocol)), το ARP μπορεί να χρησιμοποιηθεί με το IP πάνω στο ATM ή το FDDI.

Η λειτουργία του ARP μπορεί να χωριστεί σε 4 κατηγορίες:

1. Όταν ένας ξένιος υπολογιστής θέλει να στείλει ένα πακέτο σ' έναν άλλο ξένιο υπολογιστή που βρίσκεται στο ίδιο δίκτυο
2. Όταν οι δυο ξένιοι υπολογιστές βρίσκονται σε διαφορετικά δίκτυα και επικοινωνούν μέσω μιας πύλης/δρομολογητή (gateway/router): π.χ. A → B
3. Όταν ένας δρομολογητής πρέπει να προωθήσει ένα πακέτο ενός host μέσω άλλου δρομολογητή: π.χ. B → C
4. Όταν ένας δρομολογητής πρέπει να προωθήσει ένα πακέτο ενός ξένιου υπολογιστή προς έναν άλλο, ο οποίος βρίσκεται στο ίδιο δίκτυο: π.χ. C → D

Εικόνα 4-5: Παράδειγμα χρήσεις του πρωτοκόλλου ARP

Η πρώτη περίπτωση ισχύει όταν δυο host βρίσκονται στο ίδιο φυσικό δίκτυο (physical network, π.χ. συνδεδεμένοι με ένα καλώδιο Ethernet), κατά συνέπεια επικοινωνούν απευθείας, χωρίς την μεσολάβηση δρομολογητή. Οι υπόλοιπες τρεις είναι οι πιο κοινές στο Διαδίκτυο εφόσον δυο ξένιοι υπολογιστές χωρίζονται σχεδόν πάντα από πάνω από τρεις κόμβους.

Κάθε ξένιος υπολογιστής που είναι συνδεδεμένος σε ένα δίκτυο που βασίζεται στο ARP κρατάει έναν κατάλογο (ARP table) ζευγών του τύπου Διεύθυνση πρωτοκόλλου → Αντίστοιχη διεύθυνση υλικού (π.χ. ο δρομολογητής μπορεί να έχει το ζεύγος 192.168.0.30 → 30:30:30:30:30:30. Στην περίπτωση που, για ένα συγκεκριμένο

χρονικό διάστημα, δεν υπάρχει επικοινωνία με έναν ξένιο υπολογιστή που βρίσκεται στον κατάλογο, το ζεύγος που τον αναφέρει αφαιρείται ερωτήματα ARP στέλνονται με broadcast, που σημαίνει πως όλοι οι hosts στο ίδιο broadcast domain τα λαμβάνουν.

Γενικά το πρωτόκολλο ARP αντιστοιχίζει λογικές διευθύνσεις IP σε διευθύνσεις υλικού MAC και έχει δημιουργηθεί για να χρησιμοποιείται από το πρωτόκολλο Ethernet σε τοπικά δίκτυα LAN.

4.2.2 ICMP

Το πρωτόκολλο Internet Control Message Protocol (ICMP) είναι ένα από τα βασικά πρωτόκολλα του διαδικτύου. Χρησιμοποιείται κυρίως από τα λειτουργικά συστήματα των ηλεκτρονικών υπολογιστών ενός δικτύου για την ανταλλαγή μηνυμάτων λάθους, όπως για παράδειγμα την έλλειψη κάποιας υπηρεσίας από έναν server ή την απουσία ενός υπολογιστή από το δίκτυο.

Το πρωτόκολλο ICMP διαφέρει από τα πρωτόκολλα TCP και UDP διότι συνήθως δεν χρησιμοποιείται από τις εφαρμογές που εκτελούνται σε κάποιον υπολογιστή, αλλά από το λειτουργικό του σύστημα. Εξαίρεση σε αυτό τον κανόνα αποτελεί το εργαλείο ping, το οποίο στέλνει μηνύματα ICMP Echo Request σε κάποιον υπολογιστή του δικτύου για να διαπιστώσει εάν ο υπολογιστής αυτός υπάρχει ή όχι και επίσης πόσο χρόνο χρειάζεται το μήνυμα να φτάσει σε αυτόν. Εάν ο υπολογιστής αυτός υπάρχει, θα απαντήσει με μηνύματα Echo Response.

Το πρωτόκολλο ICMP έχει τυποποιηθεί στα έγγραφα RFC 792 και RFC 1122. Η έκδοση του πρωτοκόλλου που χρησιμοποιείται πιο συχνά είναι η έκδοση 4, η οποία ονομάζεται και ICMPv4 και αποτελεί μέρος του IPv4. Το IPv6 διαθέτει ένα αντίστοιχο πρωτόκολλο το οποίο ονομάζεται ICMPv6.

Τα μηνύματα ICMP κατασκευάζονται στο επίπεδο δικτύου και αποτελούν κανονικά πακέτα IP. Όπως και το πρωτόκολλο UDP, το ICMP δεν εγγυάται ότι το πακέτο θα φτάσει αξιόπιστα στον προορισμό του. Μερικές από τις πιο συνηθισμένες δικτυακές εφαρμογές χρησιμοποιούν πακέτα ICMP, όπως για παράδειγμα η εντολή traceroute. Η εντολή αυτή χρησιμοποιείται για την εύρεση όλων των κόμβων ενός δικτύου από τους οποίους πρέπει να περάσει ένα πακέτο για να φτάσει στον τελικό προορισμό του. Αυτό που κάνει ουσιαστικά είναι να στέλνει πακέτα UDP με συγκεκριμένο χρόνο ζωής (TTL - Time To Live) και να περιμένει πακέτα ICMP που να περιέχουν μήνυμα σφάλματος "ο χρόνος ζωής τελείωσε" (Time To Live exceeded in transit) ή "ο προορισμός δεν βρέθηκε" (Destination unreachable). Στο σημείο αυτό αξίζει να αναφερθεί ότι ο χρόνος ζωής (TTL - Time To Live) ενός πακέτου είναι ο μέγιστος αριθμός των κόμβων του δικτύου από τους οποίους θα πρέπει να περάσει έως ότου φτάσει στον προορισμό του. Εάν ένα πακέτο κατά την πορεία του στο δίκτυο περάσει από περισσότερους κόμβους απ' ότι αναγράφεται στο πεδίο TTL, τότε το πακέτο αυτομάτως απορρίπτεται και ο υπολογιστής ο οποίος διαπίστωσε το σφάλμα στέλνει ένα ICMP μήνυμα σφάλματος στον υπολογιστή που δημιούργησε το πακέτο. Τέλος, η εντολή ping χρησιμοποιεί επίσης το πρωτόκολλο ICMP για την λειτουργία της και συγκεκριμένα τα ICMP μηνύματα "Echo request" και "Echo reply".

4.2.3 IGMP

Το πρωτόκολλο IGMP (Internet Group Management Protocol) υπάρχει για να ανταλλάσσει μηνύματα διαχείρισης δρομολόγησης πολλαπλών παραληπτών (multicast routing) και μετάδοσης μηνυμάτων. Ουσιαστικά ο σκοπός του είναι να αντιστοιχίσει μια ομάδα διευθύνσεων (unicast) σε μια διεύθυνση (multicast) κλάσης D. Η ανταλλαγή πληροφοριών που αφορούν αυτές τις διευθύνσεις γίνεται ανάμεσα σε συστήματα που βρίσκονται συνδεδεμένα στο δίκτυο και σε routers. Υπάρχουν τρεις εκδόσεις του πρωτοκόλλου IGMP. Η έκδοση 1, η 2 και η 3. Κάθε έκδοση επιτυγχάνει το δικό της συγκεκριμένο σκοπό. Επειδή το Linux δεν υποστηρίζει αυτόματη ενημέρωση των πινάκων δρομολόγησης κατά τη λήψη αναφορών IGMP, έτσι ώστε να μπορεί να πραγματοποιηθεί δρομολόγηση multicast, γράφονται ειδικές εφαρμογές προκειμένου κάτι τέτοιο να είναι δυνατό.

4.3 Η ΔΙΑΔΙΚΑΣΙΑ ΤΗΣ ΔΡΟΜΟΛΟΓΗΣΗΣ ΣΕ ΕΝΑ ΔΙΚΤΥΟ

Μέχρι τώρα έχουμε εξετάσει το πρωτόκολλο TCP/IP και έχουμε αντιληφθεί ότι το πρωτόκολλο IP είναι υπεύθυνο για την μεταφορά των αυτοδύναμων πακέτων στο προορισμό τους (όπως δηλώνεται από τη διεύθυνση προορισμού), αλλά δεν έχουμε πει ακόμα με ποιο τρόπο πραγματοποιείται η δρομολόγηση.

Θα πρέπει καταρχήν να διευκρινίσουμε ότι σε ένα δίκτυο TCP/IP δεν είναι όλοι οι κόμβοι υπεύθυνοι να εκτελούν υπηρεσίες δρομολόγησης. Γενικά μπορούμε να διακρίνουμε δύο είδη κόμβων:

- Τους τελικούς υπολογιστές - hosts: Οι υπολογιστές αυτοί παίρνουν αποφάσεις δρομολόγησης μόνο για τα δικά τους αυτοδύναμα πακέτα. Όταν λαμβάνουν πακέτα που δεν προορίζονται για αυτούς, δεν εκτελούν καμιά διαδικασία για να τα προωθήσουν στον πραγματικό προορισμό τους.

- Τους δρομολογητές - routers: Τα μηχανήματα αυτά παίρνουν αποφάσεις δρομολόγησης για όλα τα πακέτα που λαμβάνουν και τα προωθούν στον προορισμό τους.

Να σημειώσουμε εδώ ότι η παραπάνω διάκριση έχει να κάνει με το σκοπό και λειτουργία του μηχανήματος και όχι τη φυσική του υπόσταση: Ένας κανονικός υπολογιστής μπορεί να λειτουργήσει ως δρομολογητής αν τον εξοπλίσουμε με το κατάλληλο λογισμικό. Σε πολλές περιπτώσεις χρησιμοποιούμε ως δρομολογητές εξειδικευμένα μηχανήματα (*routers*). Όταν χρησιμοποιούμε κανονικό υπολογιστή για δρομολόγηση, είναι δυνατόν το ίδιο μηχάνημα να έχει και το ρόλο του τελικού μηχανήματος (αυτό συνήθως συμβαίνει σε μικρά δίκτυα). Αντίστοιχα, σε πολύ μεγάλα δίκτυα ένας εξειδικευμένος δρομολογητής μπορεί να είναι απλώς ένας πολύ ισχυρός γενικός υπολογιστής με κατάλληλο πρόγραμμα.

Βασικό ρόλο στη διαδικασία δρομολόγησης έχει ο *πίνακας δρομολόγησης*. Το πρωτόκολλο IP χρησιμοποιεί αυτό τον πίνακα για να πάρει όλες τις αποφάσεις που έχουν να κάνουν με την δρομολόγηση πακέτων στον προορισμό τους.

Σε μεγάλα επικοινωνιακά κέντρα, υπάρχουν συνήθως δρομολογητές που διασυνδέουν πολλά δίκτυα μεταξύ τους. Στο IP, η δρομολόγηση συνήθως βασίζεται στην διεύθυνση του δικτύου προορισμού. Κάθε υπολογιστής διαθέτει ένα πίνακα με διευθύνσεις δικτύων, για καθένα από τα οποία αντιστοιχεί ένας δρομολογητής. Όταν δημιουργείται ένα αυτοδύναμο πακέτο προς κάποιο δίκτυο, ο υπολογιστής συμβουλεύεται αυτό τον πίνακα για να τα στείλει στον αντίστοιχο δρομολογητή ο οποίος και θα τα προωθήσει τελικά στο δίκτυο προορισμού. Σημειώστε εδώ ότι ο δρομολογητής αυτός δεν είναι απαραίτητο να είναι συνδεδεμένος απευθείας με το δίκτυο προορισμού: αρκεί να αποτελεί την καλύτερη επιλογή για διασύνδεση με το συγκεκριμένο δίκτυο σε σχέση με τους υπόλοιπους δρομολογητές του πίνακα. Ο δρομολογητής θα αναλάβει να στείλει το πακέτο σε άλλο δρομολογητή μέχρις ότου να φτάσει σε ένα δρομολογητή ο οποίος να είναι συνδεδεμένος απευθείας με το δίκτυο προορισμού.

Ο αλγόριθμος δρομολόγησης που χρησιμοποιείται από το πρωτόκολλο IP για τη δρομολόγηση αυτοδύναμων πακέτων διακρίνει δύο περιπτώσεις:

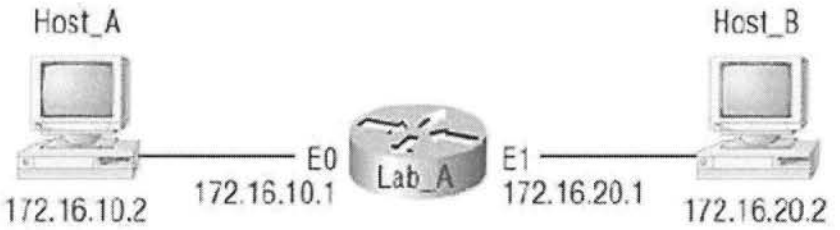
- **Άμεση Δρομολόγηση (direct routing)**: Στην περίπτωση αυτή ο υπολογιστής προορισμού βρίσκεται στο ίδιο δίκτυο με τον υπολογιστή αποστολής. Το πακέτο μπορεί να σταλεί απευθείας χωρίς άλλα βήματα, και άρα δεν γίνεται καμιά προώθηση του πακέτου. Πρόκειται για την απλούστερη μορφή δρομολόγησης.
- **Έμμεση Δρομολόγηση (indirect routing)**: Στην περίπτωση αυτή ο υπολογιστής προορισμού βρίσκεται σε διαφορετικό δίκτυο από τον υπολογιστή αποστολής. Θα πρέπει το πακέτο να δρομολογηθεί μέσω των κατάλληλων δρομολογητών για να φτάσει στον προορισμό του. Προφανώς για το σκοπό αυτό θα χρησιμοποιηθούν οι πίνακες δρομολόγησης που αναφέραμε προηγουμένως.

Όταν ένας υπολογιστής δημιουργήσει και πρόκειται να στείλει ένα αυτοδύναμο IP πακέτο, ελέγχει πρώτα αν η διεύθυνση προορισμού του βρίσκεται στο ίδιο τοπικό δίκτυο με την δική του. Για παράδειγμα, αν ο υπολογιστής αποστολής έχει διεύθυνση 192.168.0.42 και ο προορισμού 192.168.0.31, βρίσκονται και οι δύο στο ίδιο δίκτυο, το 192.168.0. Στην περίπτωση αυτή το πακέτο μπορεί να σταλεί απευθείας και δεν απαιτούνται επιπλέον βήματα. Σε αντίθετη περίπτωση, το σύστημα θα βρει μια εγγραφή στον πίνακα δρομολόγησης που να αναφέρει σε ποιο δρομολογητή πρέπει να σταλεί το πακέτο για να προωθηθεί στο δίκτυο προορισμού.

Καθώς το Διαδίκτυο (Internet) αναπτύσσεται με ραγδαίους ρυθμούς και διασυνδέει πολλά εκατομμύρια υπολογιστών, είναι φανερό ότι το μέγεθος ενός τέτοιου πίνακα δρομολόγησης αυξάνει επικίνδυνα και η διαχείριση του γίνεται προβληματική. Για το σκοπό αυτό έχουν αναπτυχθεί τεχνικές για τη μείωση του μεγέθους των πινάκων δρομολόγησης. Μια τέτοια τεχνική είναι η χρήση ενός και μόνο ορισμένου από πριν προεπιλεγμένου δρομολογητή. Σε πολλά δίκτυα υπάρχει ένας και μόνο δρομολογητής που συνδέει το δίκτυο με τον έξω κόσμο (stub network). Ένας τέτοιος δρομολογητής τυπικά συνδέει ένα τοπικό δίκτυο σε κάποιο δίκτυο κορμού.

Στην παραπάνω περίπτωση, ο πίνακας δρομολόγησης κάθε υπολογιστή του τοπικού δικτύου είναι ιδιαίτερα απλός, αφού δεν χρειάζεται μια εγγραφή για κάθε δίκτυο προορισμού. Απλώς δηλώνεται ο προεπιλεγμένος δρομολογητής ο οποίος και θα

αναλάβει κάθε κίνηση που προορίζεται για τον εξωτερικό κόσμο, ανεξάρτητα από το δίκτυο προορισμού. Προεπιλεγμένος δρομολογητής μπορεί να υπάρχει και σε δίκτυο το οποίο περιέχει περισσότερους από ένα δρομολογητή. Σε αυτό το δρομολογητή αυτό προωθούνται τα αυτοδύναμα πακέτα τα οποία στην επικεφαλίδα τους δεν καθορίζουν κάποιον από τους άλλους διαθέσιμους δρομολογητές.



Εικόνα 4-6: Δρομολόγηση μεταξύ δύο χρηστών χρησιμοποιώντας έναν δρομολογητή

Σε αυτό το παράδειγμα (εικόνα 4-2) ένας χρήστης Α εκτελεί την εντολή ping στην IP του χρήστη Β. Είναι μια απλή δρομολόγηση αλλά περιλαμβάνει πολλά στάδια:

1. Το πρωτόκολλο ICMP δημιουργεί ένα ωφέλιμο φορτίο (payload) echo request (όπου στο πεδίο των δεδομένων περιέρχεται η αγγλική αλφάβητος).
2. Το ICMP παραδίδει το ωφέλιμο φορτίο στο IP πρωτόκολλο το οποίο με τη σειρά του δημιουργεί ένα IP πακέτο. Στο ελάχιστο αυτό το IP πακέτο περιέχει μια διεύθυνση προορισμού και μια διεύθυνση αποστολέα και στο πεδίο protocol της IP επικεφαλίδας την τιμή 0x01 η οποία δείχνει ότι το IP πακέτο περιέχει ICMP πληροφορία έτσι ώστε όταν παραληφθεί το πακέτο να παραδοθεί στο ICMP πρωτόκολλο.
3. Μόλις φτιαχτεί το πακέτο, το IP πρωτόκολλο καθορίζει αν η διεύθυνση του παραλήπτη είναι στο τοπικό δίκτυο/υποδίκτυο ή σε ένα απομακρυσμένο. Η εύρεση γίνεται με ένα λογικό και μεταξύ IP παραλήπτη και μάσκας υποδικτύου.
4. Εφόσον το IP πρωτόκολλο καθορίσει ότι πρόκειται για ένα απομακρυσμένο δίκτυο, το πακέτο πρέπει να προωθηθεί στην προεπιλεγμένη πύλη έτσι ώστε να μπορέσει να προωθηθεί στο απομακρυσμένο δίκτυο.
5. Η προεπιλεγμένη πύλη για τον χρήστη Α (172.16.10.2) είναι η IP 172.16.10.1. Για να σταλεί το πακέτο όμως στο τοπικό δίκτυο, πρέπει να ξέρουμε την φυσική διεύθυνση (MAC) της προεπιλεγμένης πύλης (της διεπαφής του δρομολογητή), έτσι ώστε το πακέτο να παραδοθεί στο επίπεδο ζεύξης, να δημιουργηθεί ένα πλαίσιο και να σταλεί στη διεπαφή του δρομολογητή που είναι συνδεδεμένο στο 172.16.10.0 δίκτυο.
6. Μετά ελέγχεται η ARP cache του χρήστη Α για να ελέγξουμε αν είδη η IP του δρομολογητή έχει είδη επιλυθεί σε κάποια MAC διεύθυνση του δρομολογητή.
    a. Αν έχει επιλυθεί τότε το πακέτο είναι ελεύθερο να πάει στο επίπεδο ζεύξης.
    b. Αν όχι τότε δημιουργείτε ένα ARP broadcast πακέτο το οποίο παραλαμβάνεται από όλα τα τερματικά στο τοπικό δίκτυο και μόνο ο

δρομολογητής απαντάει με ένα ARP reply μήνυμα το οποίο περιέχει την IP και την MAC του δρομολογητή για τη διεπαφή E0.

7. Επόμενο βήμα είναι το πακέτο να παραδοθεί στο επίπεδο ζεύξης. Εκεί ο οδηγός της κάρτας δικτύου χρησιμοποιείται για να δώσει πρόσβαση στο μέσω του δικτύου που χρησιμοποιείτε (Ethernet). Τότε δημιουργείτε ένα πλαίσιο το οποίο ενθυλακώνει την IP πληροφορία. Στην επικεφαλίδα του πλαισίου, υπάρχουν οι φυσικές διευθύνσεις αποστολέα και παραλήπτη και ένα πεδίο Ethertype το οποίο καθορίζει ποιο είναι το επόμενο επίπεδο στην ενθυλάκωση όταν παραληφθεί το πλαίσιο το οποίο στο παράδειγμα μας είναι το IP.
8. Όταν το πλαίσιο δημιουργηθεί, παραδίδεται στο φυσικό επίπεδο για να μπει στο φυσικό μέσο ένα bit κάθε φορά.
9. Κάθε συσκευή στο τομέα σύγκρουσης παραλαμβάνει αυτά τα bit και δημιουργεί ένα πλαίσιο. Τρέχουν ένα CRC αλγόριθμο και επιβεβαιώνουν το αποτέλεσμα με το FCS πεδίο του πλαισίου που μόλις έφτιαξαν. Αν το CRC είναι ίδιο με το FCS τότε ελέγχεται και η φυσική διεύθυνση παραλήπτη αν είναι ίδια επίσης. Αν είναι ίδια, τότε το Ethertype πεδίο ελέγχεται για να βρεθεί ότι το επόμενο επίπεδο που πρέπει να παραδοθεί το πλαίσιο, είναι το IP.
10. Το πακέτο απονθυλακώνεται από το πλαίσιο και το πακέτο παραδίδεται στο IP πρωτόκολλο.
11. Το IP παραλαμβάνει το πακέτο και ελέγχει την IP διεύθυνση προορισμού. Κοιτάει το πίνακα δρομολόγησης του για να δει που θα σταλεί το πακέτο που μόλις παρελήφθη από εκείνον.
12. Ο πίνακας δρομολόγησης πρέπει να έχει μια καταχώρηση για το 172.16.20.0 δίκτυο αλλιώς το πακέτο θα απορριφτεί κατευθείαν και ένα ICMP μήνυμα θα σταλεί πίσω στον χρήστη A:
    destination network unreachable
13. Αν ο δρομολογητής βρει μια καταχώρηση για το δίκτυο του παραλήπτη, το πακέτο μετάγεται στην διεπαφή E1.

```
Lab_A>sh ip route

Codes:C - connected,S - static,I - IGRP,R - RIP,M - mobile,B -

[output cut]

Gateway of last resort is not set


     172.16.0.0/24 is subnetted, 2 subnets

C       172.16.10.0 is directly connected, Ethernet0

C       172.16.20.0 is directly connected, Ethernet1
```

14. Όταν ο δρομολογητής αποφασίσει ότι πρέπει να φύγει το πακέτο από την διεπαφή E1 τότε το πακέτο παραδίδεται στο buffer της E1 διεπαφής.

15. Ο buffer πρέπει να ξέρει τη φυσική διεύθυνση του παραλήπτη και πρώτα κοιτάει την ARP cache του. Αν η διεύθυνση του Β γνωρίζεται τότε το πακέτο παραδίδεται στο επίπεδο ζεύξης αλλιώς δημιουργείτε ένα ARP μήνυμα από τη διεπαφή E1 για να βρεθεί η MAC διεύθυνση του Β. Ο Β απαντάει με τη φυσική του διεύθυνση και το πακέτο και η φυσική διεύθυνση παραδίδονται στο πεδίο ζεύξης.

```
Lab_A#sh ip arp

Protocol  Address          Age(min)  Hardware Addr   Type   Interface
Internet  172.16.20.1         -     00d0.58ad.05f4  ARPA   Ethernet1
Internet  172.16.20.2         3     0030.9492.a5dd  ARPA   Ethernet1
Internet  172.16.10.1         -     00d0.58ad.06aa  ARPA   Ethernet0
Internet  172.16.10.2        12     0030.9492.a4ac  ARPA   Ethernet0
```

16. Το πεδίο ζεύξης δημιουργεί το πλαίσιο με τις φυσικές διευθύνσεις, το πεδίο Ethertype και το FCS στο τέλος του πλαισίου. Μετά το πλαίσιο παραδίδεται στο φυσικό επίπεδο για να σταλεί το φυσικό μέσο επικοινωνίας ένα bit τη φορά.
17. Ο χρήστης Β παραλαμβάνει το πλαίσιο και τρέχει έναν CRC αλγόριθμο. Αν το αποτέλεσμα είναι ίδιο με το πεδίο FCS ελέγχει αν είναι ίδια ι φυσική διεύθυνση παραλήπτη. Αν και αυτό περάσει τότε το ethertype πεδίο ελέγχεται για να καθοριστεί ποιο είναι το επόμενο επίπεδο που πρέπει να παραλάβει το πλαίσιο – στο παράδειγμα μας IP.
18. Στο επίπεδο δικτύου, το IP πρωτόκολλο ελέγχει την IP διεύθυνση προορισμού. Αφού είναι για εκείνον, ελέγχεται το πεδίο protocol για να βρεθεί σε ποιο επόμενο πρωτόκολλο θα πρέπει να παραδοθεί το ωφέλιμο φορτίο
19. Το ωφέλιμο φορτίο παραδίδεται στο ICMP πρωτόκολλο το οποίο με τη σειρά του καταλαβαίνει ότι πρόκειται για ένα echo request μήνυμα. Το ICMP απαντάει στο μήνυμα αυτό απορρίπτοντας το πακέτο και δημιουργεί ένα καινούργιο ωφέλιμο φορτίο σαν echo reply.
20. Ένα πακέτο τότε φτιάχνεται αλλά με τις IPs να είναι αντίστροφα. Ο παραλήπτης αυτή τη φορά είναι ο χρήστης Α.
21. Το IP τότε ελέγχει να δει αν ο παραλήπτης είναι στο τοπικό ή σε απομακρυσμένο δίκτυο. Αφού ο παραλήπτης είναι σε ένα απομακρυσμένο δίκτυο, το πακέτο πρέπει να παραδοθεί στην προεπιλεγμένη πύλη.
22. Ομοίως Η προεπιλεγμένη πύλη για τον χρήστη Β (172.16.20.2) είναι η IP 172.16.20.1. Για να σταλεί το πακέτο όμως στο τοπικό δίκτυο, πρέπει να ξέρουμε την φυσική διεύθυνση (MAC) της προεπιλεγμένης πύλης (της διεπαφής του δρομολογητή), έτσι ώστε το πακέτο να παραδοθεί στο επίπεδο ζεύξης, να δημιουργηθεί ένα πλαίσιο και να σταλεί στη διεπαφή του δρομολογητή που είναι συνδεδεμένο στο 172.16.10.0 δίκτυο. Η φυσική διεύθυνση γνωρίζεται όμως αφού προηγουμένως είχε γίνει επικοινωνία μεταξύ της διεπαφής E1 του δρομολογητή και του χρήστη Β.

23. Αφού βρεθεί η διεύθυνση της προεπιλεγμένης πύλης από την ARP cache του λειτουργικού συστήματος, το πακέτο και η φυσική διεύθυνση παραλήπτη παραδίδονται στο επίπεδο ζεύξης.
24. Στο επίπεδο ζεύξης δημιουργείτε το πλαίσιο με όλες τις απαραίτητες πληροφορίες
25. Το πλαίσιο τώρα παραδίδεται στο φυσικό επίπεδο και να σταλεί.
26. Η διεπαφή E1 συντάσσει το πλαίσιο από τα bit που παραλαμβάνει. Ελέγχει το FCS αν είναι ίδιο με το CRC που υπολογίζει και στη συνέχεια ελέγχει το φυσική διεύθυνση παραλήπτη που είναι ίδια.
27. Αφού είναι όλα σωστά το πακέτο απονθυλακώνεται και παραδίδεται στο επίπεδο δικτύου.
28. Εδώ ο δρομολογητής ελέγχει το πίνακα δρομολόγησης για να δει αν έχει κάποια καταχώρηση για το δίκτυο 172.16.10.0. Αφού έχει το πακέτο μετάγεται στην διαπαφή E0.
29. Ο δρομολογητής κοιτάει την ARP cache του για να δει αν η φυσική διεύθυνση του χρήστη 172.16.10.2 υπάρχει.
30. Αφού η φυσική διεύθυνση του παραλήπτη υπάρχει από την προηγούμενη αντίθετη επικοινωνία, το πακέτο και η φυσική διεύθυνση του χρήστη B παραδίδονται στο επίπεδο ζεύξης.
31. Το πλαίσιο συντάσσεται και παραδίδεται στο φυσικό επίπεδο όπου και στέλνεται ένα bit κάθε φορά.
32. Ο χρήστης B παραλαμβάνει το πλαίσιο, τρέχει ένα CRC αλγόριθμο, ελέγχει τη φυσική διεύθυνση προορισμού και κοιτάει το ether type πεδίο για να δει σε ποιον θα παραδώσει το πακέτο. K
33. Το IP παραλαμβάνει το πακέτο και ελέγχει το πεδίο protocol.
34. Το IP τότε καταλαβαίνει ότι πρέπει να δώσει το ωφέλιμο φορτίο στο ICMP και το ICMP καταλαβαίνει ότι πρόκειται για ένα ICMP echo reply μήνυμα.
35. Το ICMP επιβεβαιώνει ότι έχει λάβει ένα echo reply στέλνοντας ένα θαυμαστικό (!) στη διεπαφή του χρήστη. Το ICMP τότε ξαναπροσπαθεί στέλνοντας 4 ακόμα echo request μηνύματα στο χρήστη B.

5 ΠΡΩΤΟΚΟΛΛΑ ΔΡΟΜΟΛΟΓΗΣΗΣ IP

5.1 ΕΙΣΑΓΩΓΗ

Ο αλγόριθμος δρομολόγησης ανήκει στο επίπεδο δικτύου και σκοπός του είναι να κατευθύνει ένα πακέτο από την πηγή στον προορισμό του. Ο όρος "δρομολόγηση" αναφέρεται στη διαδικασία εύρεσης της διαδρομής που πρέπει να ακολουθήσει ένα πακέτο για να φτάσει στον προορισμό του. Η διαδικασία αυτή δεν είναι πάντα εύκολη, τη στιγμή που γνωρίζουμε ότι ένα σύνθετο δίκτυο (όπως το Internet) μπορεί να διαθέτει πολλές εναλλακτικές διαδρομές που να οδηγούν το πακέτο στον ίδιο προορισμό.

Γενικά μπορείτε να φανταστείτε ότι ένα αντίστοιχο πρόβλημα είναι να βρει ένα παιδί τη διαδρομή ανάμεσα σε τραπέζια ενός εστιατορίου για να κατευθυνθεί στο τραπέζι των γονιών του. Αν και ενδεχομένως ένας ενήλικας μπορεί να λύσει αυτό το πρόβλημα εύκολα (κρίνοντας πολύ γρήγορα ποια διαδρομή είναι η βέλτιστη), το παιδί δεν έχει ακόμα την πλήρη εποπτεία του χώρου και την απαιτούμενη εμπειρία. Φανταστείτε ότι ο αλγόριθμος δρομολόγησης θα πρέπει να κρίνει με βάση αρκετά κριτήρια ποια διαδρομή θα πρέπει να επιλεχθεί για ένα πακέτο που κατευθύνεται προς ένα συγκεκριμένο προορισμό. Η χρονική στιγμή κατά την οποία λαμβάνονται οι αποφάσεις δρομολόγησης εξαρτάται από το δίκτυο και ειδικότερα από το αν χρησιμοποιούνται *νοητά κυκλώματα* ή *αυτοδύναμα πακέτα*.

Αν χρησιμοποιούνται νοητά κυκλώματα, η εγκαθίδρυση της σύνδεσης γίνεται στην αρχή της επικοινωνίας και υποχρεωτικά όλα τα πακέτα ακολουθούν την ίδια διαδρομή (νοητό κύκλωμα). Στην περίπτωση αυτή, η επιλογή της διαδρομής γίνεται στην αρχή, κατά την εγκατάσταση της σύνδεσης.

Αν χρησιμοποιούνται αυτοδύναμα πακέτα, δεν είναι απαραίτητο τα πακέτα που ανήκουν στην ίδια σύνδεση να ακολουθούν την ίδια διαδρομή. Στην περίπτωση αυτή, η απόφαση για τη διαδρομή που θα ακολουθήσει κάθε πακέτο, λαμβάνεται για καθένα από αυτά, ξεχωριστά.

Ανεξάρτητα από τα παραπάνω, ένας αλγόριθμος δρομολόγησης είναι γενικά επιθυμητό να διαθέτει τις παρακάτω ιδιότητες:

- Απλότητα: Ο αλγόριθμος πρέπει να είναι απλός - να περιέχει σαφείς και κατανοητούς κανόνες που να διέπουν τη λειτουργία του.
- Ορθότητα: Ο αλγόριθμος πρέπει να επιλύει σωστά το πρόβλημα της δρομολόγησης.
- Ανθεκτικότητα: Ο αλγόριθμός πρέπει να είναι σε θέση να αντιμετωπίζει αλλαγές στην τοπολογία του δικτύου - π.χ. στην περίπτωση που κάποιος ενδιάμεσος κόμβος ή γραμμή σύνδεσης σταματήσουν να λειτουργούν.

- Δικαιοσύνη: Τα πακέτα που προέρχονται από διαφορετικές συνδέσεις θα πρέπει να αντιμετωπίζονται με δίκαιο τρόπο. Για παράδειγμα δεν θα πρέπει τα πακέτα μιας σύνδεσης να καθυστερούν συνέχεια για να μεταδοθούν με μεγαλύτερη ταχύτητα τα πακέτα κάποιας άλλης. Ωστόσο αυτό μπορεί να έρχεται σε αντίθεση με την ιδιότητα της βελτιστοποίησης.
- Βελτιστοποίηση: Στοχεύει στην καλύτερη δυνατή αξιοποίηση των πόρων του δικτύου. Για παράδειγμα στην μεγιστοποίηση της συνολικής κίνησης που εξυπηρετείται από το δίκτυο.

Το έργο της δρομολόγησης είναι ιδιαίτερα πολύπλοκο καθώς χρειάζεται συντονισμός και συνεργασία όλων των ενδιάμεσων κόμβων του δικτύου – και όχι μόνο των γειτονικών όπως απαιτείται από τα πρωτόκολλα των χαμηλότερων επιπέδων του OSI και του TCP/IP (π.χ. από το επίπεδο σύνδεσης δεδομένων). Τυπικά, για τη δρομολόγηση σε ένα δίκτυο συνεργάζονται μεταξύ τους πολλοί αλγόριθμοι οι οποίοι ως ένα σημείο λειτουργούν μεταξύ τους ανεξάρτητα.

Οι δύο βασικές λειτουργίες ενός αλγόριθμου δρομολόγησης είναι:

- Η επιλογή της διαδρομής για τη μεταφορά των δεδομένων από την πηγή στον προορισμό τους.
- Η παράδοση των πακέτων στον προορισμό τους όταν πλέον έχει καθοριστεί η διαδρομή.

Η παράδοση των πακέτων στον προορισμό τους γίνεται με τη χρήση των *πινάκων δρομολόγησης*. Η επιλογή της διαδρομής και η ενημέρωση των πινάκων δρομολόγησης αποτελεί ένα δύσκολο πρόβλημα το οποίο επηρεάζει την απόδοση του δικτύου. Τα βασικά μέτρα απόδοσης που επηρεάζονται από τον αλγόριθμο δρομολόγησης είναι:

- Η ρυθμοαπόδοση (δηλ. ο ρυθμός μετάδοσης που επιτυγχάνεται)
- Η μέση καθυστέρηση (ο χρόνος δηλ. που χρειάζεται για να γίνει η δρομολόγηση των πακέτων στον προορισμό τους – κατά μέσο όρο)

Είναι προφανές ότι η μέση καθυστέρηση που υφίστανται τα πακέτα, εξαρτάται από την διαδρομή που θα ακολουθήσουν μέχρι τον προορισμό τους. Η διαδρομή αυτή ωστόσο αποφασίζεται από τον αλγόριθμο δρομολόγησης. Οι αποφάσεις του αλγορίθμου έχουν κατά συνέπεια άμεση επίδραση στη μέση καθυστέρηση. Όταν η καθυστέρηση αυξάνεται ιδιαίτερα, σημαίνει ότι η εισερχόμενη κίνηση δεν μπορεί να εξυπηρετηθεί. Μπορείτε να φανταστείτε τις γραμμές ενός δικτύου σαν μια οδική αρτηρία. Όταν η κίνηση είναι αυξημένη, τα αυτοκίνητα κινούνται με μικρότερη ταχύτητα. Αν η κίνηση αυξηθεί ακόμα περισσότερο θα δημιουργηθεί κυκλοφοριακή συμφόρηση (μποτιλιάρισμα) και η κίνηση σχεδόν θα σταματήσει. Αντίστοιχα φαινόμενα παρατηρούνται και στα δίκτυα δεδομένων.

Όταν η μέση καθυστέρηση αυξάνεται πάνω από ένα όριο, ενεργοποιείται ένας μηχανισμός προστασίας που ονομάζεται *έλεγχος ροής*. Ο έλεγχος ροής εμποδίζει την είσοδο νέου φορτίου στο δίκτυο. Σκοπός του είναι να εξισορροπήσει την ρυθμοαπόδοση με την καθυστέρηση. Όσο πιο αποτελεσματικός είναι ο αλγόριθμος στην διατήρηση χαμηλής καθυστέρησης, τόσο περισσότερη κίνηση μπορεί να δεχθεί το

δίκτυο και άρα επιτυγχάνει και μεγαλύτερη ρυθμοαπόδοση. Οι αλγόριθμοι δρομολόγησης διακρίνονται σε:

- Πρώτον σε Κατανεμημένους και Συγκεντρωτικούς
- Δεύτερον σε Στατικούς και Προσαρμοζόμενης Δρομολόγησης

Στους *συγκεντρωτικούς αλγόριθμους* οι αποφάσεις δρομολόγησης λαμβάνονται εξ' ολοκλήρου από ένα κεντρικό κόμβο. Ο κόμβος αυτός πρέπει να γνωρίζει πλήρως την κατάσταση του δικτύου και άρα οι πίνακες δρομολόγησης που θα διατηρεί θα έχουν αρκετά μεγάλο μέγεθος. Έτσι ο κόμβος πρέπει να έχει μεγάλες δυνατότητες τοπικής αποθήκευσης δεδομένων αλλά και πολύ ισχυρή κεντρική μονάδα επεξεργασίας ώστε η αναζήτηση στους πίνακες να γίνεται με μεγάλη ταχύτητα.

Αντίθετα στους *κατανεμημένους αλγόριθμους* οι αποφάσεις δρομολόγησης λαμβάνονται κατανεμημένα μεταξύ των κόμβων του δικτύου. Όταν χρειάζεται, οι κόμβοι αυτοί επικοινωνούν μεταξύ τους και ανταλλάσσουν πληροφορίες για να λάβουν σωστότερες αποφάσεις (π.χ. μαθαίνουν το φορτίο που αντιμετωπίζει τη δεδομένη στιγμή κάποιο συγκεκριμένο τμήμα του δικτύου, ώστε αν χρειάζεται και είναι εφικτό να αποφεύγουν να χρησιμοποιήσουν τη συγκεκριμένη διαδρομή).

Οι *στατικοί αλγόριθμοι* δρομολόγησης χρησιμοποιούν σταθερές διαδρομές για τη μεταφορά δεδομένων. Οι διαδρομές είναι ανεξάρτητες από τις συνθήκες κίνησης που επικρατούν στο δίκτυο και αλλάζουν μόνο αν ένας κόμβος ή μια γραμμή σύνδεσης τεθούν εκτός λειτουργίας. Οι στατικοί αλγόριθμοι χρησιμοποιούνται συνήθως σε σχετικά απλά δίκτυα καθώς δεν μπορούν να επιτύχουν μεγάλες ρυθμοαποδόσεις και είναι ακατάλληλοι για δίκτυα που το φορτίο έχει μεγάλες διακυμάνσεις.

Οι *αλγόριθμοι προσαρμοζόμενης δρομολόγησης* έχουν τη δυνατότητα να τροποποιούν τις διαδρομές ανάλογα με το φορτίο των γραμμών του δικτύου. Για παράδειγμα, όταν αντιληφθούν ότι ένα τμήμα του δικτύου έχει υποστεί συμφόρηση λόγω μεγάλης εισερχόμενης κίνησης, έχουν τη δυνατότητα να τροποποιήσουν τις διαδρομές τους ώστε τα πακέτα να ακολουθούν διαδρομή που δεν περνάει από αυτό το τμήμα. Για να αποφασίσουν για τις διαδρομές, οι αλγόριθμοι αυτοί μετρούν ή εκτιμούν έμμεσα την κίνηση του δικτύου με βάση την τοπολογία του (μπορούν επίσης να ενημερώνονται με μηνύματα σχετικά με την κίνηση του δικτύου από αντίστοιχους αλγόριθμους απομακρυσμένων κόμβων).

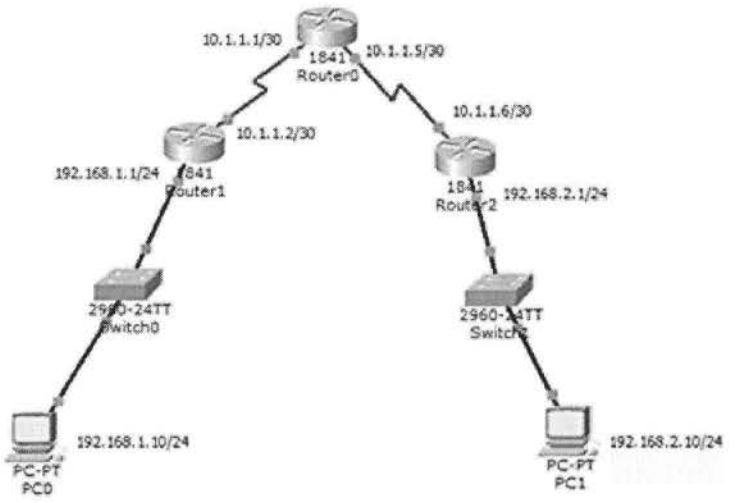
Τα κριτήρια με τα οποία λαμβάνουν τις αποφάσεις τους οι αλγόριθμοι δρομολόγησης είναι:

- Η *Συντομότερη Διαδρομή* η οποία καθορίζεται με βάση:
  - είτε τον αριθμό των χωριστών τμημάτων που την αποτελούν
  - είτε τη μέση καθυστέρηση (ουράς και μετάδοσης) που εισάγει
  - είτε τη χρησιμοποίηση τους εύρους ζώνης τη γραμμής του δικτύου
- Τον *Αριθμό Πακέτων* που περιμένουν προς μετάδοση στην ουρά εξόδου
- Το *Κόστος Γραμμής*. Είναι μια συνάρτηση στην οποία συμμετέχουν με διαφορετικούς συντελεστές βαρύτητας οι παράγοντες: μέση καθυστέρηση, μέσο μήκος ουράς, χρήση εύρους ζώνης.

Τα πρωτόκολλα δρομολόγησης IP χωρίζονται σε δύο κατηγορίες: στα πρωτόκολλα εσωτερικής δρομολόγησης (Interior Gateway Protocols, IGPs) και τα πρωτόκολλα εξωτερικής δρομολόγησης (Exterior Gateway Protocols, EGPs). Τα IGPs χρησιμοποιούνται για δρομολόγηση πακέτων σε δίκτυα που είναι διαχωρίσιμα από έναν οργανισμό, που έχουν δηλαδή κοινή δικτυακή διαχείριση. Παραδείγματα IGPs είναι τα πρωτόκολλα δρομολόγησης RIP (Routing Information Protocol), OSPF (Open Shortest Path First), IGRP (Interior Gateway Routing Protocol), EIGRP, IS-IS κτλ. Αντίθετα από τα IGPs, τα EGPs χρησιμοποιούνται για την ανταλλαγή πληροφορίας δρομολόγησης μεταξύ δικτύων που δεν μοιράζονται την ίδια διαχειριστική αρχή (τα ονομαζόμενα Αυτόματα Συστήματα, Autonomous System - AS). Το ποιο χαρακτηριστικό πρωτόκολλο είναι το Border Gateway Protocol (BGP) και θεωρείτε το πρωτόκολλο του «Ιντερνετ». Τέλος τα πρωτόκολλα δρομολόγησης διακρίνονται σε στατικής και δυναμικής δρομολόγησης.

5.2 ΠΡΩΤΟΚΟΛΛΑ ΣΤΑΤΙΚΗΣ ΔΡΟΜΟΛΟΓΗΣΗΣ

Τα πρωτόκολλα στατικής δρομολόγησης δεν ανταλλάσουν πληροφορίες σχετικά με τη κατάσταση του δικτύου με άλλους δρομολογητές. Ο πίνακας δρομολόγησης δημιουργείται κάθε φορά που ενεργοποιείτε μια διεπαφή. Ο πίνακας δρομολόγησης δημιουργείται χειροκίνητα σε κάθε δρομολογητή από το διαχειριστή του δικτύου και αποθηκεύεται στα μέσα αποθήκευσης, ώστε να δημιουργείται αυτόματα πλέον σε κάθε επανεκκίνηση του συστήματος. Ας εξετάσουμε τη παρακάτω απλή τοπολογία της εικόνας 5-1:



Εικόνα 5-1: Τοπολογία δικτύου δύο απομακρυσμένων δικτύων

Θέλουμε ο Host1 να μπορεί να επικοινωνεί με το Host2. Όλοι οι δρομολογητές δεν έχουν κανένα configuration προς το παρών. Συνεπώς ο κάθε δρομολογητής γνωρίζει μόνο για τα τοπικά δίκτυα που είναι συνδεδεμένος. Επομένως ο δρομολογητής Router1 γνωρίζει μόνο για τα δίκτυα 10.1.1.0/30 και 192.168.1.0/24. Αν από το χρήστη PC0 έρθει ένα πακέτο στο

δρομολογητή Router1 που προορίζεται για κάποιο δίκτυο εκτός από τα 10.1.1.0/30 το πακέτο θα απορριφθεί αμέσως και ένα ICMP μήνυμα θα γυρίσει πίσω στο PC0 με μήνυμα destination host unreachable. Πράγματι προσπαθώντας να κάνουμε ping στο χρήστη PC1 με IP 192.168.2.10 παίρνουμε το αποτέλεσμα της εικόνας 5-2:

```
Packet Tracer PC Command Line 1.0
PC>ping 192.168.2.10

Pinging 192.168.2.10 with 32 bytes of data:

Reply from 192.168.1.1: Destination host unreachable.
Reply from 192.168.1.1: Destination host unreachable.
Reply from 192.168.1.1: Destination host unreachable.
Reply from 192.168.1.1: Destination host unreachable.

Ping statistics for 192.168.2.10:
    Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),

PC>
```

Εικόνα 5-2: Το PC0 δεν επικοινωνεί με το απομακρυσμένο δίκτυο. Ένα ICMP μήνυμα στέλνεται από το δρομολογητή Router1 στο PC0 με κείμενο Destination host unreachable

Το ICMP μήνυμα αυτό οφείλεται στο γεγονός ότι ο δρομολογητής Router1 δεν γνωρίζει το δίκτυο 192.168.2.0. Πράγματι κοιτώντας το πίνακα δρομολόγησης του Router1 παίρνουμε τα παρακάτω αποτελέσματα όπως φαίνεται στην εικόνα 5-3:

```
Router#show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
       * - candidate default, U - per-user static route, o - ODR
       P - periodic downloaded static route

Gateway of last resort is not set

     10.0.0.0/30 is subnetted, 1 subnets
C       10.1.1.0 is directly connected, Serial0/0/0
C    192.168.1.0/24 is directly connected, FastEthernet0/0
Router#
```

Εικόνα 5-3: Πίνακας δρομολόγησης του Router1

Βλέπουμε λοιπόν ότι ο δρομολογητής Router1 βλέπει σαν connected δίκτυα τα δίκτυα που είναι συνδεδεμένα σε αυτόν (10.1.1.0/30 και 192.168.1.0/24). Αντίστοιχα ο δρομολογητής Router0 γνωρίζει μόνο για τα τοπικά συνδεδεμένα δίκτυα 10.1.1.0/30 και 10.1.1.4/30 όπως φαίνεται από το πίνακα δρομολόγησης του.

```
Router0(config)# do show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
       * - candidate default, U - per-user static route, o - ODR
       P - periodic downloaded static route

Gateway of last resort is not set

     10.0.0.0/30 is subnetted, 2 subnets
C       10.1.1.0 is directly connected, Serial0/0/0
C       10.1.1.4 is directly connected, Serial0/0/1
Router0(config)#
```

Εικόνα 5-4: Πίνακας δρομολόγησης του Router0

Επομένως για να μπορέσει ο χρήστης στον PC0 να επικοινωνήσει με το χρήστη στο PC1 θα πρέπει να ξέρει ο δρομολογητής Router1 ότι πρέπει να προωθήσει το πακέτο στον Router0 και με την σειρά του ο Router0 να προωθήσει το πακέτο στο Router2. Ας εφαρμόσουμε λοιπόν στατική δρομολόγηση στο κάθε δρομολογητή την ύπαρξη κάθε δικτύου. Για το κάθε δρομολογητή έχουμε αντίστοιχα τις παρακάτω ρυθμίσεις (σημείωση: Οι εντολές ισχύουν για Cisco δρομολογητές και μόνο):

- **Για τον δρομολογητή Router0**

```
Router0(config)#ip route 192.168.1.0 255.255.255.0 s0/0/0
Router0(config)#ip route 192.168.2.0 255.255.255.0 s0/0/1
Router0(config)#^Z
```

- **Για τον δρομολογητή Router1**

```
Router1(config)#
Router1(config)#ip route 192.168.2.0 255.255.255.0 s0/0/0
Router1(config)#
```

- **Για τον δρομολογητή Router2**

```
Router2(config)#
Router2(config)#ip route 192.168.1.0 255.255.255.0 s0/0/0
Router2(config)#
Router2(config)#
```

Προγραμματίζουμε λοιπόν τον δρομολογητή Router1 για την ύπαρξη του απομακρυσμένου δικτύου 10.1.2.0/24 ότι μπορεί να φτάσει σε αυτό το δίκτυο μέσω του δρομολογητή Router0. Επίσης πρέπει να προγραμματίσουμε και τον δρομολογητή Router2 για την ύπαρξη του δικτύου 192.168.1.0/24. Τέλος ο δρομολογητής Router0 θα πρέπει να προγραμματιστεί ότι μπορεί να φτάσει τα δίκτυα 192.168.1.0/24 μέσω τους δρομολογητή Router1 και το δίκτυο 192.168.2.0/24 μέσω του δρομολογητή Router2. Μετά τον προγραμματισμό των δρομολογητών ο πίνακας δρομολόγησης για το κάθε δρομολογητή φαίνεται στην εικόνα 5-5.

```
Router0#show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
       * - candidate default, U - per-user static route, o - ODR
       P - periodic downloaded static route

Gateway of last resort is not set

     10.0.0.0/30 is subnetted, 2 subnets
C       10.1.1.0 is directly connected, Serial0/0/0
C       10.1.1.4 is directly connected, Serial0/0/1
S    192.168.1.0/24 is directly connected, Serial0/0/0
S    192.168.2.0/24 is directly connected, Serial0/0/1
```

```
Router1#show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
       * - candidate default, U - per-user static route, o - ODR
       P - periodic downloaded static route

Gateway of last resort is not set

     10.0.0.0/30 is subnetted, 1 subnets
C       10.1.1.0 is directly connected, Serial0/0/0
C    192.168.1.0/24 is directly connected, FastEthernet0/0
S    192.168.2.0/24 is directly connected, Serial0/0/0
Router1#

Router2#show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
       * - candidate default, U - per-user static route, o - ODR
       P - periodic downloaded static route

Gateway of last resort is not set

     10.0.0.0/30 is subnetted, 1 subnets
C       10.1.1.4 is directly connected, Serial0/0/0
S    192.168.1.0/24 is directly connected, Serial0/0/0
C    192.168.2.0/24 is directly connected, FastEthernet0/0
Router2#
```

Εικόνα 5-5: Πίνακας δρομολόγησης σε κάθε δρομολογητή μετά την χειροκίνητη ρύθμιση τους για κάθε δίκτυο χωριστά

Βλέπουμε δηλαδή ότι ο κάθε δρομολογητής γνωρίζει για την ύπαρξη των δικτύων 192.168.1.0/24 και 192.168.2.0/24.

Δοκιμάζοντας τώρα την εντολή ping από το PC0 για το PC1 παίρνουμε τα αποτελέσματα της εικόνας 5-6.

```
PC>ping 192.168.2.10

Pinging 192.168.2.10 with 32 bytes of data:

Reply from 192.168.2.10: bytes=32 time=2ms TTL=125
Reply from 192.168.2.10: bytes=32 time=10ms TTL=125
Reply from 192.168.2.10: bytes=32 time=2ms TTL=125
Reply from 192.168.2.10: bytes=32 time=2ms TTL=125

Ping statistics for 192.168.2.10:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 2ms, Maximum = 10ms, Average = 4ms

PC>
```

Εικόνα 5-6: Το τερματικό PC0 επικοινωνεί πλέον με το απομακρυσμένο δίκτυο

Που σημαίνει ότι το τερματικό PC0 μπορεί πλέον και επικοινωνεί με το τερματικό PC1.

Το δίκτυο του παραδείγματος μας είναι πολύ μικρό σε σχέση με ένα δίκτυο που μπορεί να έχει ένας οργανισμός πόσο μάλλον το Ιντερνετ ολόκληρο. Η στατική δρομολόγηση δεν είναι ο καταλληλότερος τρόπος για να διαφημίσεις τα δίκτυα σε όλους τους δρομολογητές. Έχει πολλά μειονεκτήματα διότι ο διαχειριστής του δικτύου θα πρέπει να ρυθμίζει κάθε

δρομολογητή ξεχωριστά για το κάθε απομακρυσμένο δίκτυο. Επιπλέον η διαχείριση του δικτύου γίνεται ποιο δύσκολη.

Η στατική δρομολόγηση χρησιμοποιείτε σε ειδικές και συγκεκριμένες περιπτώσεις μόνο όπως για παράδειγμα σε stub δίκτυα, για gateway of last resort κτλ. Τα stub δίκτυα είναι τα δίκτυα τα οποία τα πακέτα μπορούν να φύγουν από μια διεπαφή και μόνο για να επικοινωνήσουν με όλα τα άλλα δίκτυα. Stub δίκτυο μπορεί να θεωρηθεί το τοπικό δίκτυο στο σπίτι μας ή στο προηγούμενο παράδειγμα μας, το δίκτυο 192.168.1.0/24 και 192.168.2.0/24.

5.3 ΠΡΩΤΟΚΟΛΛΑ ΔΥΝΑΜΙΚΗΣ ΔΡΟΜΟΛΟΓΗΣΗΣ

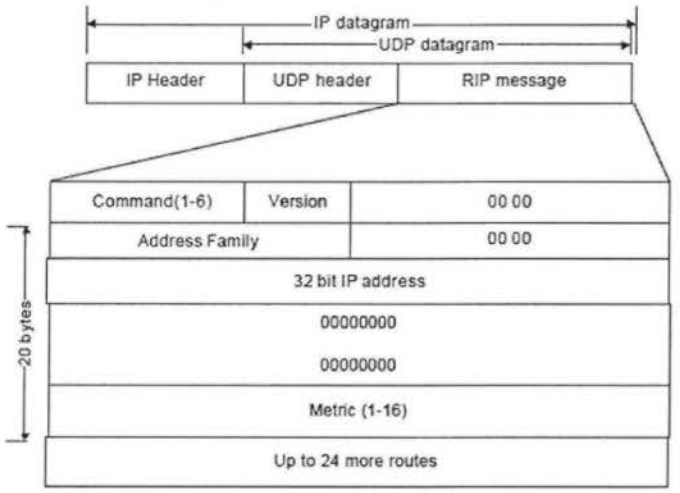
Με τον όρο δυναμική δρομολόγηση εννοούμε την ανταλλαγή πληροφοριών, μεταξύ γειτονικών δρομολογητών, οι οποίες πληροφορίες είναι μηνύματα που περιέχουν τα δίκτυα τα οποία γνωρίζει ο κάθε δρομολογητής. Η διαδικασία η οποία χρησιμοποιείται για την επικοινωνία μεταξύ των γειτονικών δρομολογητών ονομάζεται δαίμονας δρομολόγησης. Ο δαίμονας δρομολόγησης αναλαμβάνει την ενημέρωση των πινάκων βασισμένος στην πληροφορία που λαμβάνει από πίνακες γειτονικών δρομολογητών. Ο μηχανισμός τις δρομολόγησης όπως είδαμε στο προηγούμενο κεφάλαιο δεν αλλάζει αυτό που αλλάζει είναι η πληροφορία που εγγράφεται στο πίνακα δρομολόγησης του κάθε δρομολογητή. Ο δαίμονας δρομολόγησης δηλαδή καθορίζει την πολιτική δρομολόγησης αποφασίζοντας με κάποια κριτήρια ανάλογα με τον αλγόριθμο που εφαρμόζεται, ποιο είναι το καλύτερο μονοπάτι προς ένα προορισμό (στην περίπτωση ύπαρξης περισσότερων του ενός μονοπατιού) και εγγράφοντας το στον πίνακα.

Τα πρωτόκολλα αυτά όπως είπαμε διακρίνονται σε εσωτερικής και εξωτερικής δρομολόγησης. Το διαδίκτυο είναι οργανωμένο σε αυτόνομα συστήματα. Το κάθε αυτόνομο σύστημα έχει το δικό του διαχειριστή, ο οποίος καθορίζει ποιο εσωτερικό πρωτόκολλο θα χρησιμοποιηθεί για την επικοινωνία μεταξύ των δρομολογητών του αυτόνομου συστήματος. Αυτά τα εσωτερικά πρωτόκολλα δρομολόγησης όπως είπαμε ονομάζονται Πρωτόκολλα Εσωτερικής Δρομολόγησης (IGP). Για την επικοινωνία μεταξύ δρομολογητών διαφορετικών αυτόνομων συστημάτων χρησιμοποιούνται τα πρωτόκολλα εξωτερικής δρομολόγησης (EGP) το οποίο πλέον είναι ένα και είναι το BGP. Υπάρχουν διαφορετικά είδη πρωτοκόλλων εσωτερικής δρομολόγησης (RIP, OSPF, IGRP, EIGRP, IS-IS) όπου το καθένα έχει τα πλεονεκτήματα του και τα μειονεκτήματα του.

5.3.1 ROUTING INFORMATION PROTOCOL (RIP)

Το εσωτερικό πρωτόκολλο δρομολόγησης RIP ήταν το ποιο δημοφιλές πρωτόκολλο δρομολόγησης. Όλο το Internet χρησιμοποιούσε το RIP. Τα μηνύματα του RIP μεταδίδονται χρησιμοποιώντας τη UDP πόρτα 520. Στην εικόνα 5-7 παρουσιάζεται η μορφή του πακέτου UDP που μεταφέρει ένα μήνυμα RIP. Η πληροφορία για κάθε μονοπάτι έχει μέγεθος 20 bytes και σε κάθε μήνυμα μπορεί να μεταφερθεί πληροφορία από 1 έως 25 μονοπάτια. Το

όριο των 25 μονοπατιών τέθηκε έτσι ώστε το συνολικό μέγεθος του μηνύματος να είναι μικρότερο από 512 bytes. Με τον περιορισμό αυτό είναι πολύ συχνό το φαινόμενο να χρειάζονται περισσότερα του ενός μηνύματα για την αποστολή ολόκληρου του πίνακα δρομολόγησης. Η επικεφαλίδα του μηνύματος περιέχει τα πεδία command και version. Το πεδίο command μπορεί να έχει τις τιμές 1 (αίτηση παροχής πληροφορίας) ή 2 (απάντηση στην αίτηση). Μπορεί επίσης να περιέχει τις τιμές 5 και 6 που σημαίνουν αίτηση παροχής πληροφορίας για ένα μέρος του πίνακα και απάντηση στην αίτηση αυτή. Το πεδίο version περιγράφει την έκδοση του πρωτοκόλλου RIP που χρησιμοποιείται. Μετά την επικεφαλίδα ακολουθεί η πληροφορία των διαφόρων μονοπατιών η οποία για κάθε μονοπάτι έχει μέγεθος 20 bytes. Το πεδίο metric περιέχει τον αριθμό των διαδοχικών δρομολογητών (hop count) μέχρι το προορισμό.



Εικόνα 5-7: UDP πακέτο που μεταφέρει ένα RIP μήνυμα

Το metric είναι ο αριθμός που χρησιμοποιείται από τον αλγόριθμο δρομολόγησης για να καθορίσει πότε ένα μονοπάτι θα πρέπει να προτιμηθεί από ένα άλλο. Ο πίνακας δρομολόγησης περιέχει μόνο τα καλύτερα μονοπάτια. Το RIP χρησιμοποιεί το hop count για να βρει τη καλύτερη διαδρομή ενώ το OSPF χρησιμοποιεί το κόστος. Το κόστος σε Cisco δρομολογητές για το OSPF θεωρείτε το bandwidth κάθε διεπαφής.

Κατά την εκκίνηση ενός συστήματος που χρησιμοποιεί RIP, το σύστημα αναζητά τις ενεργές διεπαφές με τις οποίες είναι συνδεδεμένο. Κατόπιν στέλνει πακέτα RIP ζητώντας τους πλήρεις πίνακες δρομολογήσεις των γειτονικών δρομολογητών. Για την αίτηση παροχής του πλήρους πίνακα δρομολόγησης από γειτονικούς δρομολογητές τα πεδία command, address family και metric της επικεφαλίδας είναι 1,0 και 16 αντίστοιχα. Μετά την άφιξη της αίτησης από γειτονικούς δρομολογητές αποστέλλεται μέσω ενός η διαδοχικών μηνυμάτων ο πλήρης πίνακα του κάθε γειτονικού δρομολογητή στο δρομολογητή που ζήτησε τη παροχή. Στην περιγραφή του πρωτοκόλλου καθορίζεται ότι για κάθε σύστημα είναι απαραίτητες οι μεταδόσεις ολόκληρων των καταχωρήσεων στο πίνακα δρομολόγησης που έχουν ρυθμιστεί να διαφημίζονται με το RIP πρωτόκολλο. Η περιοδική μετάδοση γίνεται κάθε 30 δευτερόλεπτα από προεπιλογή. Η πληροφορία που περιέχεται σε κάθε πίνακα έχει περιορισμένη διάρκεια ζωής. Αν ένας δρομολογητής διαπιστώσει ότι για κάποιο συγκεκριμένο μονοπάτι δεν έχει έρθει καμιά ενημέρωση για διάστημα πέρα των 180

δευτερολέπτων , το πεδίο metric του συγκεκριμένου μονοπατιού τίθεται στη τιμή 16. Αν περάσουν ακόμα 60 δευτερόλεπτα και δεν έχει έρθει καμιά ενημέρωση, τότε διαγράφεται η καταχώρηση από το πίνακα δρομολόγησης για το συγκεκριμένο μονοπάτι. Οι προδιαγραφές του RIP (RFC 1058) καθορίζουν επίσης πως εκτός από την ενημέρωση κατά τακτά χρονικά διαστήματα, υπάρχει και η εξαναγκασμένη ενημέρωση, που συμβαίνει όταν αλλάζει το πεδίο metric για κάποιο μονοπάτι, ή όταν μια διεπαφή μεταθέτεται από ενεργή κατάσταση σε ανενεργή. Σε αυτή τη περίπτωση η εγγραφή που αφορά το μονοπάτι αυτό θα πρέπει να διαδοθεί στο υπόλοιπο δίκτυο μέσω του πρωτοκόλλου RIP. Το πεδίο metric έχει την τιμή ένα για όλα τα δίκτυα που είναι συνδεδεμένα απευθείας στον δρομολογητή.

Το πρωτόκολλο RIP αν και είναι πολύ απλό πρωτόκολλο και από τα πρώτα που χρησιμοποιήθηκαν έχει αρκετά σημαντικά μειονεκτήματα. Πρώτον ο μέγιστος αριθμός δρομολογητών που μπορεί να διασχίσει ένα RIP μήνυμα είναι 15 (hop count) και έτσι δεν είναι δυνατή η χρήση του πρωτοκόλλου σε μεγάλα δίκτυα. Ένα άλλο σημαντικό μειονέκτημα το οποίο υπάρχει μόνο στην πρώτη έκδοση του πρωτοκόλλου RIP είναι ότι δεν περιέχει πληροφορία διευθυνσιοδότησης υποδικτύων. Έτσι δεν μπορεί να διακρίνει αν μια διεύθυνση αντιστοιχεί σε υποδίκτυο ή τερματικό ούτε να κάνουμε χρήση του CIDR. όλα τα δίκτυα δηλαδή διαφημίζονται με την class full IP τους. Για παράδειγμα αν έχουμε το τοπικό δίκτυο 10.1.1.0 με μάσκα 255.255.255.0, το RIP θα το διαφημίσει στην Class full μορφή του που είναι η 10.0.0.0 με μάσκα 255.0.0.0 (IP από 1 – 127 στο πρώτο octet κλάση A). Για την αντιμετώπιση των προβλημάτων αυτών παρουσιάστηκε μια νέα έκδοση του πρωτοκόλλου τι RIPv2. Η νέα έκδοση δεν αλλάζει το πρωτόκολλο απλά περιέχει περισσότερη πληροφορία (πχ μάσκα υποδικτύου). Το RIPv2 περιέχει επίσης και ένα πεδίο που ονομάζεται router Tag το οποίο διευκολύνει την επικοινωνία με πρωτόκολλα εξωτερικής δρομολόγησης. Παρόλο που τα παραπάνω προβλήματα αντιμετωπίστηκαν στην νέα έκδοση, ο αλγόριθμος που υλοποιείται στο πρωτόκολλο RIP είναι πλέον ξεπερασμένος και δεν είναι σε θέση να χρησιμοποιηθεί το πρωτόκολλο αυτό σε μεγάλα δίκτυα.

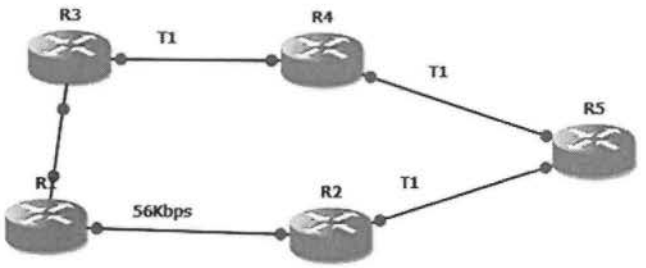
Ποιο συγκεκριμένα μεγάλο μειονέκτημα είναι ο μεγάλος χρόνος που απαιτείται μέχρι να ισορροπήσει (convergence) το δίκτυο μετά την αστοχία ή την απενεργοποίηση μιας ζεύξης. Ακόμη το ότι το RIP έχει μόνο σαν κριτήριο επιλογής την απόσταση, για τη εύρεση της βέλτιστης διαδρομής αποτελεί μεγάλο πρόβλημα. Για την εικόνα 5-8 ο δρομολογητής R1 για να φτάσει στο υποδίκτυο του R5 θα χρησιμοποιήσει τη διαδρομή R1->R2->R5 με βάση το πρωτόκολλο RIP αφού είναι μόλις 2 hops μακριά. Αυτή η διαδρομή όμως δεν είναι και η καλύτερη αφού το πρωτόκολλο RIP δεν εξετάζει ότι το μονοπάτι ανάμεσα στο R1 και R2 είναι μια με χωρητικότητα 56Kbps. Άρα για τη διαδρομή R1->R2->R5 έχουμε συνολική ταχύτητα:

1.544MBS + 0.056MBPS = 1.6MBS

Αν επιλέξουμε όμως τη διαδρομή R1 -> R3 -> R4 -> R5, η συνολική χωρητικότητα μας είναι:

1,544 * 3 = 4,632MBS

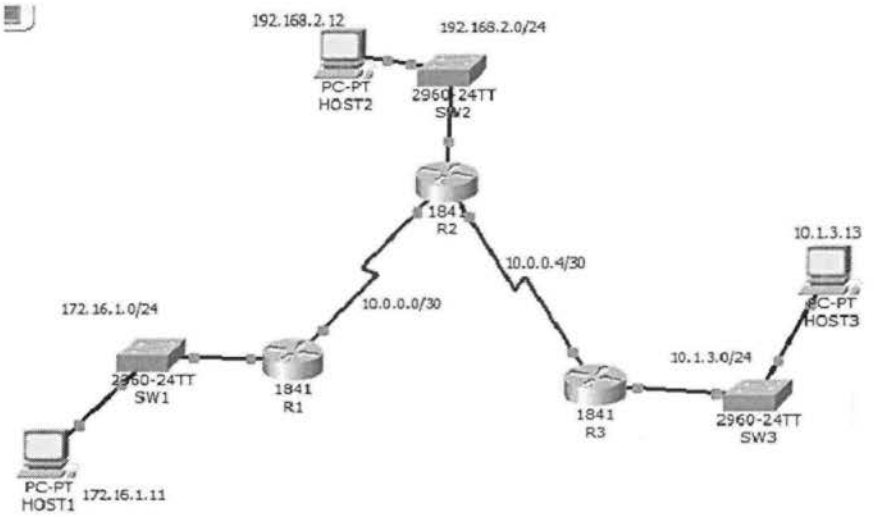
Σχεδόν τριπλάσια από το πρώτο επιλεγμένο μονοπάτι. Βλέπουμε λοιπόν ότι το RIP αδυνατεί να υπολογίζει τη βέλτιστη διαδρομή σε τέτοιες περιπτώσεις διότι στον αλγόριθμο που χρησιμοποιεί (Bellman - Ford) δεν περιέχεται σαν μεταβλητή η χωρητικότητα.



Εικόνα 5-8: Το RIP αδυνατεί να βρει τη καλύτερη διαδρομή. Ο R1 για να φτάσει στον R5 θα χρησιμοποιήσει την διαδρομή R1->R2->R5 που είναι και η ποιο αργή

5.3.1.1 ΠΑΡΑΔΕΙΓΜΑ RIP

Για την παρακάτω τοπολογία (Εικόνα 5-9) θα χρησιμοποιήσουμε το πρωτόκολλο RIP σαν πρωτόκολλο δυναμικής δρομολόγησης.



Εικόνα 5-9: Τοπολογία δικτύου για παράδειγμα RIP

Για να ενεργοποιηθεί το πρωτόκολλο RIP σε κάθε δρομολογητή αρκούν οι παρακάτω εντολές σε ένα Cisco δρομολογητή (Εικόνα 5-10):

```
R1(config)#router rip
R1(config-router)#version
R1(config-router)#version 2
R1(config-router)#ne
R1(config-router)#network 172.16.1.0
R1(config-router)#ne
R1(config-router)#network 10.0.0.0
R1(config-router)#
```

Εικόνα 5-10: Εντολές για ενεργοποίηση του πρωτοκόλλου RIP στον δρομολογητή R1

Με τις παραπάνω εντολές λέμε στο δρομολογητή R1 να χρησιμοποιήσει το εσωτερικό πρωτόκολλο δρομολόγησης RIP την έκδοση 2 και σε όποια διεπαφή έχει οποιαδήποτε δίκτυα ή υποδίκτυα των δικτύων 172.16.1.0 και 10.0.0.0 να τα διαφημίσει. Στη περίπτωση του δρομολογητή R1 θα διαφημίσει στον R2 τα υποδίκτυα 172.16.1.0/24 και 10.0.0.0/30.

Ομοίως ο δρομολογητής R2 μόλις ρυθμιστεί να χρησιμοποιεί το πρωτόκολλο RIP θα διαφημίζει στους δρομολογητές R1 και R3 το τοπικό του υποδίκτυο (192.168.2.0/24). Στη παρακάτω εικόνα (Εικόνα 5-11), βλέπουμε το πίνακα δρομολόγησης του R2.

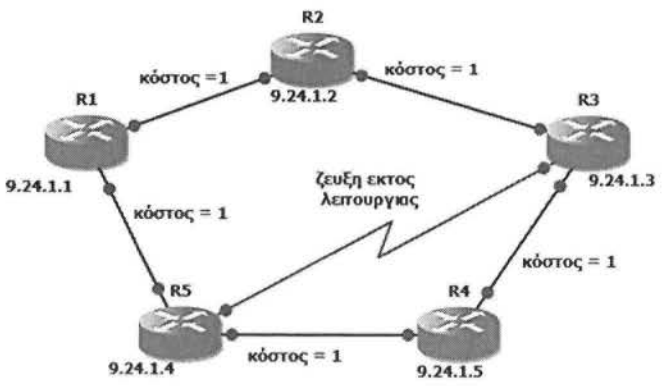
```
R2#show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
       * - candidate default, U - per-user static route, o - ODR
       P - periodic downloaded static route

Gateway of last resort is not set

     10.0.0.0/8 is variably subnetted, 3 subnets, 2 masks
C       10.0.0.0/30 is directly connected, Serial0/0/0
C       10.0.0.4/30 is directly connected, Serial0/0/1
R       10.1.3.0/24 [120/1] via 10.0.0.6, 00:00:11, Serial0/0/1
R    172.16.0.0/16 [120/1] via 10.0.0.2, 00:00:08, Serial0/0/0
C    192.168.2.0/24 is directly connected, FastEthernet0/0
R2#
```

Εικόνα 5-11: Πίνακας δρομολόγησης του δρομολογητή R2

Ο R2 εκτός από τα συνδεδεμένα σε αυτόν δίκτυα που έχει (10.0.0.0/30, 10.0.0.4/30 και 192.168.2.0/24) έχει μάθει πλέον μέσω του πρωτοκόλλου RIP (αριστερά στη καταχώρηση έχει το γράμμα R που σημαίνει έχει μαθευτεί η καταχώρηση από το RIP πρωτόκολλο το C σημαίνει connected δηλαδή ότι το δίκτυο είναι συνδεδεμένο στο δρομολογητή) τα δύο απομακρυσμένα τοπικά δίκτυα των δρομολογητών R1 και R3. Τα δίκτυα αυτά είναι τα: 10.1.3.0 και 172.16.0.0. Τέλος αξίζει να σημειωθεί ότι στην πρώτη έκδοση ο πίνακας δρομολόγησης που στέλνεται ανά τακτά χρονικά διαστήματα, γίνεται σε broadcast διεύθυνση ενώ στην δεύτερη έκδοση μεταδίδονται στην multicast διεύθυνση 224.0.0.9.

5.3.2 OPEN SHORTEST PATH FIRST (OSPF)

Ένα πρωτόκολλο εσωτερικής δρομολόγησης για μεγάλα δίκτυα είναι το OSPF. Πρόκειται για ένα πρωτόκολλο δυναμικής δρομολόγησης που σημαίνει ότι ο πίνακας δρομολόγησης ενημερώνεται σε τακτά χρονικά διαστήματα, χωρίς παρέμβαση από τον διαχειριστή μέσω πληροφοριών που καταφθάνουν από γειτονικούς γείτονες. Στην εικόνα 5-12 το δίκτυο αποτελείται από 5 δρομολογητές.



Εικόνα 5-12: Παράδειγμα τοπολογίας δικτύου OSPF

Σε κάθε ζεύξη μεταξύ των δρομολογητών καθορίζεται ένα κόστος το οποίο μπορεί να καθορίζεται από παράγοντες όπως η ικανότητα ρυθμού διέλευσης της ζεύξης, η αξιοπιστία κλπ. Τα κόστη κάθε ζεύξης γίνονται γνωστά μέσω του δικτύου σε όλους τους δρομολογητές. Αρχικά όλες οι ζεύξεις είναι ενεργές και έχουν κόστος 1. Έτσι για το δεδομένο δίκτυο και τη δεδομένη στιγμή όλοι οι δρομολογητές έχουν την παρακάτω βάση δεδομένων.

Δρομολογητής	9.24.1.1	9.24.1.2	9.24.1.3	9.24.1.4	9.24.1.5
9.24.1.1		1	2	1	2
9.24.1.2	1		1	2	2
9.24.1.3	2	1		1	1
9.24.1.4	1	2	1		1
9.24.1.5	2	2	1	1	

Εικόνα 5-13: κόστη ζεύξεων

Τα πακέτα δεδομένων που διέρχονται από τον 9.24.1.2 και πηγαίνουν στον 9.24.1.4 πηγαίνουν είτε μέσω του .1.3 είτε μέσω του .1.1 αφού και οι δύο διαδρομές έχουν το ίδιο κόστος. Στη πραγματικότητα το φορτίο θα διανεμηθεί ομοιόμορφα και στις δυο διαδρομές.

Στο OSPF κάθε δρομολογητής στέλνει προς τους γειτονικούς του το μήνυμα HELLO (Hello Protocol) για να επισημάνει ότι είναι ενεργός. Έστω ότι το δίκτυο λειτουργεί κανονικά και ότι για κάποια χρονική στιγμή η ζεύξη μεταξύ 9.24.1.3 και 9.24.1.4 τίθεται εκτός λειτουργίας. Οι δρομολογητές 9.24.1.3 και 9.24.1.4 αντιλαμβάνονται τη δυσλειτουργία και μεταδίδουν στο δίκτυο την πληροφορία πως η ζεύξη αυτή δεν είναι ποια ενεργή. Οι υπόλοιποι δρομολογητές του δικτύου ενημερώνουν τους πίνακες του μόλις λάβουν την πληροφορία και κατασκευάζεται ένας πίνακας αρκετά διαφορετικός από τον πίνακα 1. Θα υπάρχουν δύο διαδρομές .1.2 στον .1.4 εκ τον οποίων η μία θα έχει κόστος 2 μέσω του .1.1 και η άλλη κόστος 3 (μέσω των .1.3 και .1.5).Τώρα πλέον η ροή της πληροφορίας θα πραγματοποιείται μέσω της διαδρομής που εμφανίζει το χαμηλότερο κόστος.

Ο ποιο συνηθισμένος τρόπος υπολογισμού κόστους, είναι βασισμένος στο ρυθμό διέλευσης μιας ζεύξης. Ένας τύπος υπολογισμού του κόστους είναι ο παρακάτω:

ΚΟΣΤΟΣ = 100,000,000 /ΡΥΘΜΟΣ ΔΙΕΛΕΥΣΗΣ ΣΕ BITS/SEC

Σημειώνεται ότι ο τύπος αυτός ισχύει για Fast Ethernet πρωτόκολλο μόνο. Σε Gigabit Ethernet ο τύπος βγάζει κόστος μηδέν. Για Gigabit Ethernet ο τύπος γίνεται:

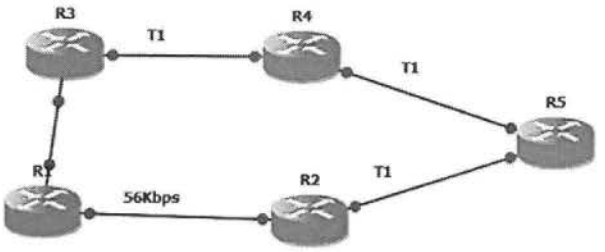
ΚΟΣΤΟΣ = 100,000,000,000 /ΡΥΘΜΟΣ ΔΙΕΛΕΥΣΗΣ ΣΕ BITS/SEC

Για παράδειγμα το κόστος σε μια γραμμή Ethernet (10Mbps) είναι:

100,000,000/10,000,000 = 10

Ενώ σε μια γραμμή E1 είναι 49.

Έτσι λοιπόν το OSPF έρχεται να λύσει πολλά προβλήματα που δημιουργούνται στο RIP. Για παράδειγμα σύμφωνα με το τη παρακάτω τοπολογία:



Το RIP επιλέγει από τον R1 στον R5 τη διαδρομή R1 – R2 – R5 όπως εξηγήσαμε. Ας δούμε το OSPF ποια διαδρομή θα διαλέξει. Αρχικά το OSPF θα κοιτάξει τα κόστη για τον υπολογισμό του καλύτερου μονοπατιού. Επομένως έχει δύο πιθανά μονοπάτια ή R1-R2-R5 ή R1-R3-R4-R5.

Υπολογίζοντας τα κόστη και για τις δύο διαδρομές έχουμε:

Για τη Διαδρομή R1 – R2 – R5, το κόστος είναι:

K(R1,R2,R5) = 100.000.000 / 56.000 + 100.000.000/1.544.000 = 1849

Για τη Διαδρομή R1 - R3 - R4 - R5

K(R1,R3,R4,R5) = 100.000.000/1.544.000 + 100.000.000/1.544.000 + 100.000.000/1.544.000 = 64 + 64 + 64 = 192

Βλέπουμε λοιπόν ότι με το OSPF η καλύτερη βέλτιστη διαδρομή είναι αντίθετη με εκείνη του πρωτοκόλλου RIP και η ποιο σύντομη.

Η πληροφορία σχετικά με τα κόστη των διάφορων ζεύξεων γίνεται γνωστή στους δρομολογητές ενός δικτύου OSPF μέσω του πρωτοκόλλου HELLO. Το HELLO εκτός από τη μετάδοση της πληροφορίας σχετικής με τις ζεύξεις, συγχρονίζει και τα ρολόγια των συστημάτων που συμμετέχουν σε ένα δίκτυο OSPF. Ακόμη όπως είδαμε το RIP χρησιμοποιεί το UDP πρωτόκολλο ενώ το OSPF χρησιμοποιεί το TCP που σημαίνει ότι η επικοινωνία είναι ποιο αξιόπιστη.

Το OSPF είναι πιθανώς το πιο διαδεδομένο πρωτόκολλο εσωτερικών πυλών (IGP) σε μεγάλα δίκτυα. Μπορεί να λειτουργήσει με ασφάλεια, χρησιμοποιώντας MD5 για να πιστοποιήσει τους ομότιμούς του πριν να σχηματίσει γειτνιάσεις και πριν αποδεχτεί διαφημίσεις κατάστασης σύνδεσης (*link-state advertisement*). Μια νεότερη έκδοση του OSPF, (η OSPFv3), υποστηρίζει επίσης και το IPv6. Επεκτάσεις πολυεκπομπής για το OSPF, όπως τα πρωτόκολλα multiport open shortest path first (MOSPF) έχουν μεν οριστεί, αλλά δεν χρησιμοποιούνται ευρέως προς το παρόν. Το OSPF μπορεί να βάλει «ετικέτες» στις διαδρομές και να τις μεταδώσει και αυτές μαζί με τις διαδρομές.

Στην εικόνα 5-14 φαίνεται το γενικό σχήμα του OSPF μηνύματος. Το πεδίο version περιέχει την τρέχουσα έκδοση του πρωτοκόλλου. Το type χρησιμοποιείται για την αναγνώριση του τύπου του μηνύματος που ακολουθεί. Υπάρχουν οι τύποι HELLO, Database Description, Link

Status Request (LSR), Link Status Update (LSU) και Link Status Acknowledgement (LSA). Το τελευταίο χρησιμοποιείται για επαλήθευση των μηνυμάτων Link Status Update. Το πεδίο message length δίνει το μήκος του μηνύματος ενώ στο Router ID περιέχεται η διεύθυνση αποστολέα του μηνύματος. Ακόμη στο Area ID περιέχεται ο αριθμός της περιοχής (περιγράφεται παρακάτω). Το checksum παρέχει τον τρόπο για να επαληθεύσουμε αν το μήνυμα περιέχει λάθη ενώ τέλος, το authentication type και το authentication μας παρέχει τρόπο να διαχωρίσουμε τις εκπομπές των πιστοποιημένων δρομολογητών από ενδεχόμενες εκπομπές κάποιων κακόβουλων που θα προκαλούσαν προβλήματα στη διαδικασία της δρομολόγησης.

1	1	2	4	4	2	2	8	Variable
Version number	Type	Packet length	Router ID	Area ID	Check sum	Authenti-cation type	Authentication	Data

Εικόνα 5-14: Επικεφαλίδα μηνύματος OSPF

- ***Το μήνυμα HELLO:*** Το μήνυμα αυτό αποστέλλεται περιοδικά για να διαπιστωθεί αν υπάρχει επικοινωνία μεταξύ των γειτονικών δρομολογητών σε ένα δίκτυο OSPF. Στο πεδίο Type εισάγεται η τιμή 1 και έχει τη μορφή που παρουσιάζεται στην παρακάτω εικόνα. Το πεδίο Network Mask περιέχει τη μάσκα του υποδικτύου από το οποίο εστάλη το μήνυμα. Αν το χρονικό διάστημα το οποίο περιέχεται στο Dead Time παρέλθει χωρίς να απαντήσει ο δρομολογητής στον οποίο απευθύνεται, τότε ο δρομολογητής αυτός δεν θα ληφθεί υπό όψιν για τη διαδικασία δρομολόγησης. Το επόμενο πεδίο δείχνει το διάστημα που μεσολαβεί μεταξύ της αποστολής διαδοχικών μηνυμάτων του τύπου αυτού. Τα πεδία Designated Gateway και Backup Designated Gateway. Περιέχουν της διευθύνσεις των προεπιλεγμένων δρομολογητών μιας περιοχής σε Broadcast δίκτυα (όπως το Ethernet) ή NBMA (Non Broadcast Multi-access Network, όπως το Frame Relay).

8	4	2	2	8	8	Variable
Network Mask	Dead Timer	Hello Int	GWAY Pr.	Designated Gateway	Backup Designated Gateway	Neighbor IP adress

Μήνυμα HELLO

- ***Το μήνυμα Databases Description:*** Χρησιμοποιείται για την ανταλλαγή πληροφοριών που περιέρχονται στις βάσεις δεδομένων των δρομολογητών. Το περιεχόμενο των μηνυμάτων αυτών βοηθάει κάποιον δρομολογητή να καταλάβει την τοπολογία του δικτύου στο οποίο βρίσκεται. Κατά την έναρξη της διαδικασίας δρομολογητής master απαιτεί από κάποιον άλλο δρομολογητή slave να του παράσχει πληροφορίες από τη βάση δεδομένων του. Ο slave δρομολογητής απαντάει με μήνυμα όπως φαίνεται στο παρακάτω σχήμα.

1	3			8	8	8	8	8	4	4
Must be zero	I	M	S	Database Sequence Number	Link Type	Link ID	Advertising Gateway	Link Sequence Number	Link Checksum	Link Age

Μήνυμα Database Descriptor

- ***Το μήνυμα Link Status Request:*** Μετά την απόκτηση της πληροφορίας σχετικής με την πληροφορία του δικτύου, κάποιος δρομολογητής είναι πιθανόν να ανακαλύψει ότι μέρος της βάσης που αφορά κάποιες ζεύξεις του δεν έχει ενημερωθεί για μεγάλο διάστημα. Χρησιμοποιεί το μήνυμα, για να απαιτήσει από κάποιον γειτονικό δρομολογητή πληροφορία σχετική με τις ζεύξεις αυτές.
- ***Το μήνυμα Link Status Update:*** Σε περιοδικά χρονικά διαστήματα ή όταν η κατάσταση μιας ζεύξης αλλάξει οι δρομολογητές χρησιμοποιούν το μήνυμα αυτό για να ενημερώσουν τους γειτονικούς τους. Το μήνυμα είναι μεταβλητού μήκους και στην επικεφαλίδα του περιέχει το πλήθος των εγγραφών που περιέχονται σε αυτό. Η μορφή της κάθε εγγραφής είναι ίδια με την περίπτωση του μηνύματος Database Description και περιέχει τα πεδία που περιγράφουν αν η ζεύξη είναι προς άλλο δίκτυα, άλλη περιοχή κτλ.

Το OSPF έχει τα εξής χαρακτηριστικά:

- Περιλαμβάνει περιοχές και αυτόνομα συστήματα
- Επιτρέπει επεκτασιμότητα
- Υποστηρίζει VLSM/CIDR
- Δεν περιορίζεται στο hop count σε αντίθεση με το RIP
- Είναι ανοιχτού προτύπου επομένως επιτρέπει να χρησιμοποιηθεί από συσκευές δρομολογητές διαφορετικών εταιριών (πχ Cisco - Juniper)

Το OSPF είναι το πρώτο link-state πρωτόκολλο δρομολόγησης που οι περισσότεροι έρχονται σε επαφή και μάθηση έτσι λοιπόν είναι καλό να δούμε τις διαφορές με πρωτόκολλα δρομολόγησης όπως το RIPv1 και RIPv2 τα οποία είναι πρωτόκολλα διανύσματος απόστασης. Ο παρακάτω πίνακας (Εικόνα 5-15) δείχνει τις διαφορές των τριών αυτών πρωτοκόλλων:

Χαρακτηριστικά	OSPF	RIPv1	RIPv2
Είδος πρωτοκόλλου	Link State	Distance vector	Distance Vector
Classless	Ναι	Ναι	Όχι
Υποστήριξη VLSM	Ναι	Ναι	Όχι
Αυτόματη Περίληψη Δικτύων	Ναι	Ναι	Όχι
Χειροκίνητη Περίληψη Δικτύων	Όχι	Ναι	Ναι
Υποστήριξη Discontiguous	Ναι	Ναι	Όχι

Διάδοση Δρομολόγησης	Multicast triggered	Multicast περιοδικά	broadcast
Metric διαδρομής	Bandwidth	Hops	Hops
Όριο hop count	κανένα	15	15
Convergence (Σύγκληση)	Γρήγορα	Αργά	Αργά
Authentication	Ναι	Ναι	Όχι
Ιεραρχική Δομή Δικτύου	Ναι (περιοχές)	όχι	Όχι
Ενημερώσεις	Triggered Events	Ενημέρωση όλου του πίνακα δρομολόγησης	Ενημέρωση όλου του πίνακα δρομολόγησης
Αλγόριθμος	Dijkstra	Bellman-Ford	Bellman-Ford

Εικόνα 5-15: Διαφορές πρωτόκολλων RIPv1, RIPv2 και OSPF

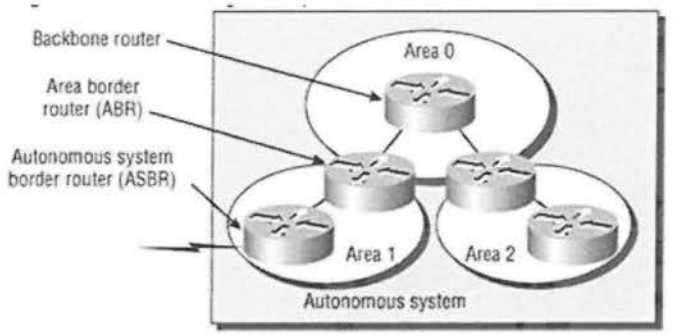
Το OSPF έχει πολλά πλεονεκτήματα πέρα από αυτά του παραπάνω πίνακα και όλα αυτά οδηγούν σε ένα γρήγορο, επεκτάσιμο και ισχυρό πρωτόκολλο που μπορεί να χρησιμοποιηθεί σε χιλιάδες παραγωγικά δίκτυα.

Το OSPF έχει δημιουργηθεί για να σχεδιάζεται σε ιεραρχική δομή, που σημαίνει ότι μπορείς να χωρίσεις το μεγάλο δίκτυο σε μικρότερες μικρότερα διαδίκτυα που ονομάζονται περιοχές. Αυτός είναι και ο καλύτερος σχεδιασμός του OSPF. Παρακάτω είναι και οι λόγοι για να δημιουργήσεις το OSPF σε ιεραρχική δομή:

- Για να μειωθεί ο φόρτος δρομολόγησης
- Για να επιταχύνουμε τη σύγκληση (convergence) ενός δικτύου. Δηλαδή αν μια δρομολόγηση χαθεί από το πίνακα που συνεπάγεται να έχει πέσει ένα δίκτυο, ο χρόνος που θα κάνει ο δρομολογητής για να βρει ένα άλλο μονοπάτι.
- Για να περιοριστεί η αστάθεια του δικτύου σε μεμονωμένες περιοχές του διαδικτύου μας.

Οι λόγοι αυτοί βέβαια δεν κάνουν το OSPF ποιο εύκολο για να ρυθμιστεί και να διαχειριστεί, αλλά ποιο δύσκολο και πολύπλοκο.

Η εικόνα 5-16 δείχνει ένα τυπικό και απλό OSPF δίκτυο. Όλοι οι δρομολογητές ενώνονται στο δίκτυο κορμού (backbone) το οποίο ονομάζεται περιοχή 0. Το OSPF πρέπει να έχει μια περιοχή 0 και όλες οι άλλες περιοχές να ενώνονται σε αυτήν. Οι δρομολογητές που ενώνουν άλλες περιοχές στη περιοχή 0 μέσα σε ένα αυτόνομο σύστημα καλούνται Area Border Routers (ABR) και τουλάχιστον μια διεπαφή σε ένα ABR δρομολογητή πρέπει να ανήκει στην περιοχή 0.



Εικόνα 5-16: OSPF δίκτυο

Το OSPF τρέχει μέσα σε ένα αυτόνομο σύστημα αλλά μπορεί ακόμη να ενώνει περισσότερα αυτόνομα συστήματα μαζί. Ο δρομολογητής που ενώνει αυτά τα αυτόνομα συστήματα μαζί ονομάζεται Autonomous Systems Boundary Router (ASBR).

5.3.2.1 ΟΡΓΑΝΩΣΗ ΕΝΟΣ ΔΙΚΤΥΟΥ OSPF

Σε ένα δίκτυο που αποτελείται από K δρομολογητές το συνολικό πλήθος των μηνυμάτων HELLO θα ήταν K^2, εφόσον ο καθένας από αυτούς θα μεταδώσει HELLO μήνυμα προς όλους τους γειτονικούς του δρομολογητές (το πρόβλημα αυτό υπάρχει σε Broadcast δίκτυα και NBMA δίκτυα σε Point-to-Point δίκτυα δεν υπάρχει). Το γεγονός αυτό είναι δυνατόν να μειώσει το αξιοποιήσιμο για μετάδοση δεδομένων εύρος ζώνης σε χαμηλά επίπεδα. Η περίπτωση αυτή έχει προβλεφθεί από το OSPF και για το λόγο αυτό σε κάθε φυσικό δίκτυο (broadcast ή Non Broadcast Multi Access Network) υπάρχει ένας προεπιλεγμένος δρομολογητής στον οποίο όλοι στέλνουν τα μηνύματα τους. Αυτός είναι και υπεύθυνος για την μετάδοση των μηνυμάτων προς όλους τους άλλους δρομολογητές «εκπροσωπώντας» τους. Με τον τρόπο αυτό το συνολικό πλήθος των μηνυμάτων μειώνεται σε 2K. Επειδή υπάρχει η πιθανότητα δυσλειτουργίας του ή αστοχίας των ζεύξεων που τον συνδέουν με το υπόλοιπο δίκτυο υπάρχει και ο εφεδρικός προεπιλεγμένος δρομολογητής ο οποίος αναλαμβάνει τον ρόλο του μόλις διαπιστωθεί ότι η επικοινωνία με τον πρωτεύοντα δεν είναι δυνατή. Αυτό επιτυγχάνεται μέσω των περιοδικών μηνυμάτων HELLO.

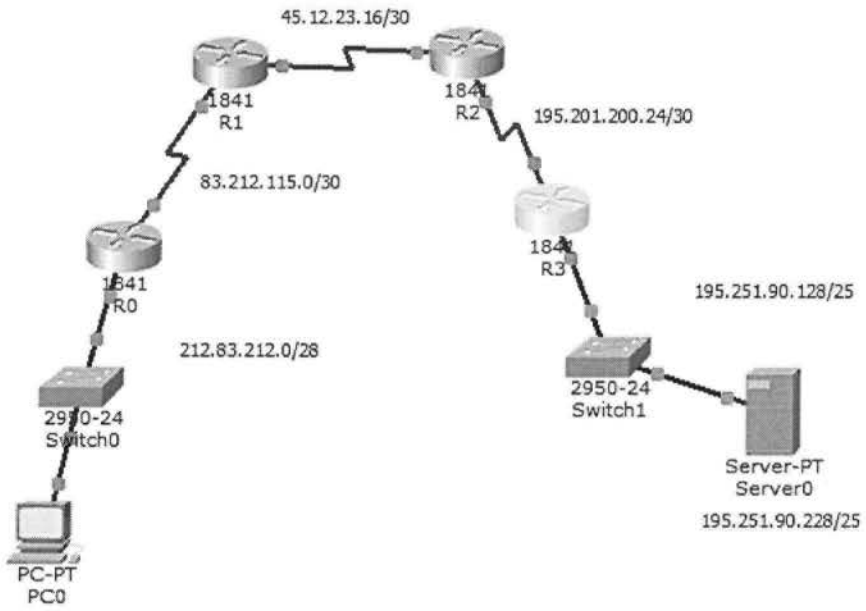
Όπως είπαμε το OSPF χρησιμοποιεί ιεραρχική δομή για να χωρίσει ένα δίκτυο σε πολλές περιοχές. Μια περιοχή είναι ένα σύνολο δικτύων μέσα σε ένα αυτόνομο σύστημα. Κάθε δρομολογητής διατηρεί ένα μια βάση δεδομένων που περιέχει πληροφορίες σχετικές με την κατάσταση των ζεύξεων μεταξύ των γειτονικών του δρομολογητών καθώς και την τοπολογία ολόκληρου του δικτύου. Σε μια περιοχή, όλοι οι δρομολογητές έχουν κοινή βάση δεδομένων (όχι κοινούς πίνακες δρομολόγησης).

Όλες οι περιοχές συνδέονται με το δίκτυο κορμού ή περιοχή 0, το οποίο είναι υπεύθυνο για τη μετάδοση της πληροφορίας δρομολόγησης σε όλες τις περιοχές του αυτόνομου συστήματος. Μια περιοχή για παράδειγμα περιοχή 1 μπορεί να μην συνδέεται με την περιοχή 0 αλλά με κάποια άλλη περιοχή για παράδειγμα με τη περιοχή 2. Η σύνδεση που πραγματοποιεί η περιοχή 1 με το δίκτυο κορμού ονομάζεται εικονική σύνδεση αφού διέρχεται μέσω άλλης περιοχής (περιοχή διέλευσης).

Ο διαχωρισμός των δρομολογητών σε ένα OSPF δίκτυο (ABR, ASBR κτλ) έγινε για να μειωθεί ο όγκος της διακινούμενης πληροφορίας καθώς και των πινάκων δρομολόγησης που διατηρούν οι δρομολογητές σε κάθε περιοχή. Ποιο συγκεκριμένο ένας δρομολογητής σε μια περιοχή, διατηρεί πληροφορία που είναι σχετική μόνο με την περιοχή στην οποία ανήκει. Ένας ABR δρομολογητής που ενώνει 2 ή περισσότερες περιοχές μέσα σε ένα αυτόνομο σύστημα, διατηρεί βάση δεδομένων που περιέχει την τοπολογία όλων των περιοχών στις οποίες συνδέεται ανταλλάσσοντας πληροφορία με τους δρομολογητές όλων των περιοχών αυτών. Τέλος οι ASBR δρομολογητές που ενώνουν δύο ή περισσότερα αυτόνομα συστήματα χρησιμοποιούν πρωτόκολλα εξωτερικής δρομολόγησης (BGP) για να επικοινωνήσουν με δρομολογητές άλλων αυτόνομων συστημάτων. Είναι υπεύθυνοι να μεταδίδουν στην περιοχή με την οποία ανήκουν πληροφορία σχετική με εξωτερικές συνδέσεις, αλλά και για την μετάδοση σε εξωτερικά δίκτυα πληροφορίας σχετικής με την κατάσταση των ζεύξεων του αυτόνομου συστήματος

5.3.2.2 ΠΑΡΑΔΕΙΓΜΑ OSPF ΔΙΚΤΥΟΥ

Ας δούμε τη παρακάτω τοπολογία:



Εικόνα 5-17: Τοπολογία δικτύου II

Αρχικά θα εξετάσουμε ότι η επικοινωνία από το PC0 στο Server0 με IP 195.251.90.228 είναι εφικτή:

```
PC>ping 195.251.90.228

Pinging 195.251.90.228 with 32 bytes of data:

Reply from 195.251.90.228: bytes=32 time=4ms TTL=124
Reply from 195.251.90.228: bytes=32 time=3ms TTL=124
Reply from 195.251.90.228: bytes=32 time=3ms TTL=124
Reply from 195.251.90.228: bytes=32 time=4ms TTL=124

Ping statistics for 195.251.90.228:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 3ms, Maximum = 4ms, Average = 3ms
```

Στη συνέχεια ελέγχουμε ότι στον δρομολογητή τρέχει το πρωτόκολλο OSPF.

```
R1(config)#do show ip protocols

Routing Protocol is "ospf 1"
  Outgoing update filter list for all interfaces is not set
  Incoming update filter list for all interfaces is not set
  Router ID 83.212.115.1
  Number of areas in this router is 1. 1 normal 0 stub 0 nssa
  Maximum path: 4
  Routing for Networks:
    0.0.0.0 255.255.255.255 area 0
  Routing Information Sources:
    Gateway         Distance      Last Update
    83.212.115.1         110      00:23:43
    195.201.200.25       110      00:22:45
    195.251.90.229       110      00:10:07
    212.83.212.1         110      00:05:15
  Distance: (default is 110)
```

Από τα παραπάνω αποτελέσματα βλέπουμε ότι τρέχει το OSPF πρωτόκολλο στον δρομολογητή R1. Εικόνα 5-16 φαίνεται ο πίνακας δρομολόγησης του δρομολογητή R1:

```
R1#show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
       * - candidate default, U - per-user static route, o - ODR
       P - periodic downloaded static route

Gateway of last resort is not set

     45.0.0.0/30 is subnetted, 1 subnets
C       45.12.23.16 is directly connected, Serial0/0/1
     83.0.0.0/30 is subnetted, 1 subnets
C       83.212.115.0 is directly connected, Serial0/0/0
     195.201.200.0/30 is subnetted, 1 subnets
O       195.201.200.24 [110/128] via 45.12.23.17, 00:27:07, Serial0/0/1
     195.251.90.0/25 is subnetted, 1 subnets
O       195.251.90.128 [110/129] via 45.12.23.17, 00:13:45, Serial0/0/1
     212.83.212.0/28 is subnetted, 1 subnets
O       212.83.212.0 [110/65] via 83.212.115.2, 00:08:50, Serial0/0/0
```

Εικόνα 5-18: Πίνακας δρομολόγησης του δρομολογητή R1 για OSPF δίκτυο

Παρατηρούμε ότι ο δρομολογητής εκτός από τα δύο συνδεδεμένα δίκτυα που έχει (45.12.23.16/30 και 83.212.115.0/30, περιγράφονται με το γράμμα C) έχει μάθει και για όλα τα υπόλοιπα δίκτυα στην τοπολογία μας (195.201.200/30, 195.251.90.128/128 και 212.83.212.0/28).

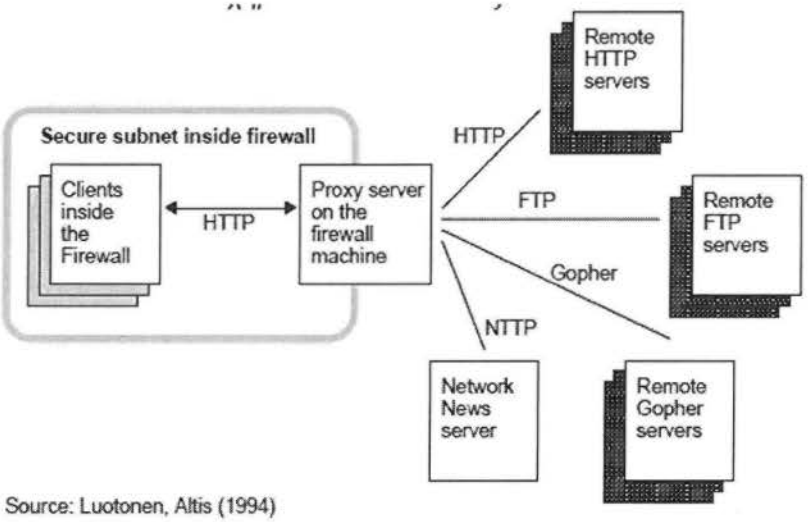
5.4 ΠΡΩΤΟΚΟΛΛΑ ΕΞΩΤΕΡΙΚΗΣ ΔΡΟΜΟΛΟΓΗΣΗΣ

Τα πρωτόκολλα εξωτερικής δρομολόγησης επιτρέπουν την επικοινωνία μεταξύ δρομολογητών που βρίσκονται σε διαφορετικά αυτόνομα συστήματα. Το ποιο διαδεδομένο και χρησιμοποιούμενο πρωτόκολλο σήμερα είναι το BGP. Δεν θα αναλύσουμε περισσότερο το πρωτόκολλο αυτό στην εργασία.

6 ΠΕΡΙ FIREWALL ΚΑΙ PROXY ΔΙΑΚΟΜΙΣΤΩΝ

6.1 WWW PROXIES

Η κύρια χρήση των proxies είναι να επιτρέψουν πρόσβαση στον ιστό (web) από το εσωτερικό ενός firewall. Ένας proxy server είναι ένας ειδικός HTTP server που, κατά κανόνα εκτελείται σε μία firewall μηχανή. Ο proxy server δέχεται αιτήσεις από το εσωτερικό του firewall τις οποίες και προωθεί στον κατάλληλο απομακρυσμένο server (εγκατεστημένο εκτός του firewall). Στην συνέχεια ενεργεί ως αποδέκτης των απαντήσεων από τον απομακρυσμένο server τις οποίες και προωθεί στον αρχικό client. Η εικόνα 6-1 παρουσιάζει την θέση ενός proxy server σε συνδυασμό με το firewall.



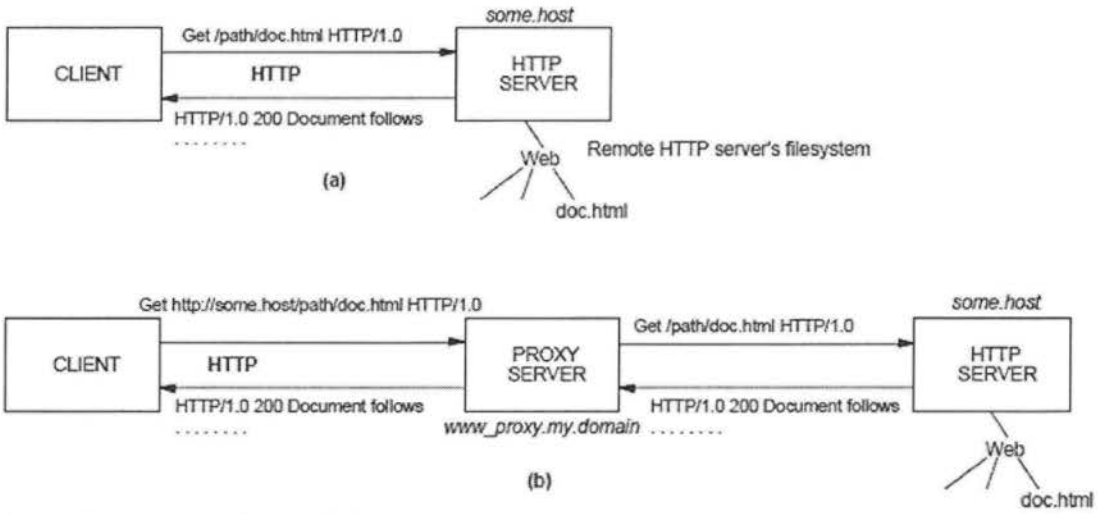
Εικόνα 6-1: Proxy server και firewall

Συχνά χρησιμοποιείται από όλους τους clients στο υποδίκτυο ο ίδιος proxy server. Αυτό επιτρέπει στον proxy server να πραγματοποιήσει caching στα αρχεία που ζητούνται από ένα πλήθος clients. Μέσω του caching επιτυγχάνεται ελάττωση των χρόνων απόκρισης, μετά την πρώτη μεταφορά του αρχείου στον proxy server. Ετσι ο proxy server δεν συμβάλει αποκλειστικά και μόνο στην ασφάλεια του υποδικτύου αλλά βελτιώνει και την ποιότητα των παρεχόμενων υπηρεσιών. Το caching στον proxy server κρίνεται πιο αποδοτικό σε σχέση με το caching στους browsers. Ελαχιστοποιεί τις απαιτήσεις περιφερειακής μνήμης εφόσον μόνο ένα αντίγραφο διατηρείται πλέον στο δίκτυο (στον proxy server και όχι στον κάθε client ξεχωριστά). Το υποσύστημα caching του proxy server μπορεί να χρησιμοποιήσει την τεχνική look-ahead ή άλλον αλγόριθμο πρόβλεψης (predictive algorithm) στηριζόμενος στο μεγαλύτερο πλήθος αιτήσεων το οποίο καλείται να εξυπηρετήσει.

Ο proxy server, όπως αναφέρθηκε παραπάνω, αποτελεί έναν εξελιγμένο HTTP server. Η ενσωμάτωση του μηχανισμού ασφαλείας proxy στο επίπεδο εφαρμογής επιτρέπει στους

χρήστες να διαπεράσουν το firewall χωρίς να δημιουργεί σοβαρά προβλήματα ασφαλείας (security holes) μέσω των οποίων θα ήταν δυνατή η μη-εξουσιοδοτημένη πρόσβαση στο ιδιωτικό δίκτυο. Οι WWW clients ρυθμίζονται ιδιαίτερα εύκολα για την προσαρμογή τους στο περιβάλλον proxy (proxy clients). Επίσης δεν απαιτείται η χρήση εξειδικευμένων FTP, Gopher και WAIS clients για την επικοινωνία μέσα από το firewall. Όλες αυτές οι περιπτώσεις καλύπτονται με την χρήση του WWW client και του proxy server. Ο proxy server χειρίζεται όλα τα σχετικά πρωτόκολλα κατά τρόπο διαφανή. Δεν υποβιβάζεται η λειτουργικότητα των πρωτοκόλλων εφόσον αυτά μπορούν με εύκολο τρόπο να προσαρμοστούν στις μεθόδους του HTTP.

Οι proxy servers επιτρέπουν εκτενές logging σχετικά με την ανταλλασσόμενη πληροφορία. Ετσι είναι δυνατή η καταγραφή στοιχείων όπως: client IP διεύθυνση, ώρα και ημερομηνία, το αιτούμενο URL, κωδικοί κατάστασης καθώς και μέγεθος ανταλλασσόμενων δεδομένων. Επίσης μπορεί να υλοποιηθεί κάποιο φλιτάρισμα στις δοσοληψίες των clients. Ο proxy server μπορεί να πραγματοποιήσει έλεγχο πρόσβασης περιορίζοντας αιτήσεις για συγκεκριμένες μεθόδους, hosts και domains κλπ. Στην εικόνα 6-2 που ακολουθεί παρουσιάζονται ο συνήθης HTTP διάλογος μεταξύ ενός client και του HTTP server (6-2.a) καθώς και οι μεταβολές στον διάλογο αυτό όταν παρεμβληθεί ο proxy ενδιάμεσος (6-2.b).

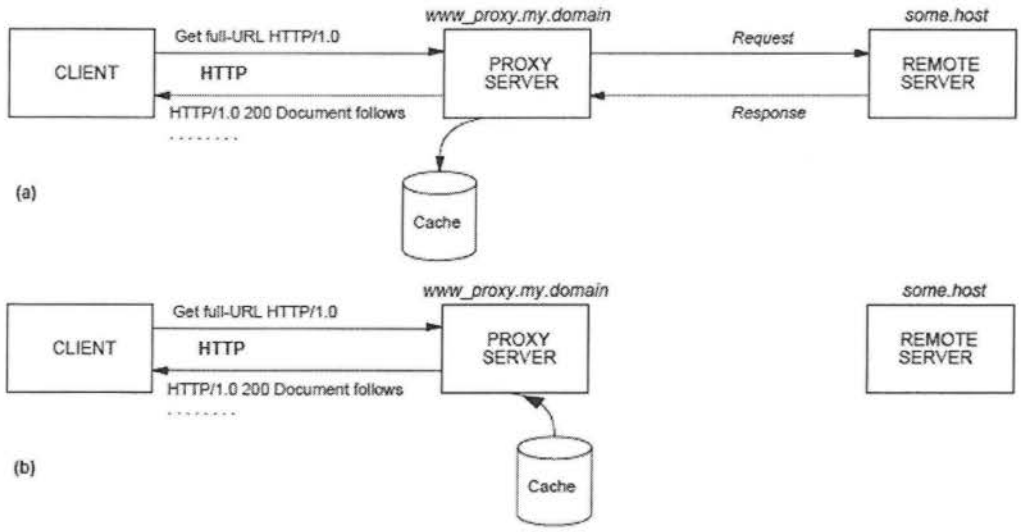


Εικόνα 6-2: HTTP επικοινωνία μεταξύ client και server

Στην περίπτωση της συνήθους, απευθείας HTTP επικοινωνίας (6-2.a) ο client υποβάλλει μία αίτηση στην οποία καθορίζει κάποιο πόρο αναφορικά (relative) με τον server αποδέκτη (στο URL της αίτησης δεν καθορίζονται πρωτόκολλο π.χ. http: ή το hostname). Σε περίπτωση ενδιάμεσου (6-2.b, proxied HTTP transaction) στην αίτηση που υποβάλλεται σε αυτόν ορίζονται σαφώς το hostname καθώς και το πρωτόκολλο άντλησης της πληροφορίας (full URL). Ακολούθως, ο proxy server εμπλέκεται σε ένα διάλογο όμοιο με αυτόν που παρουσιάστηκε στην εικόνα 6-2.a με τον HTTP server στον οποίον είναι διαθέσιμος ο αιτούμενος πόρος. Στην δεύτερη αυτή περίπτωση, ο client πάντα χρησιμοποιεί HTTP για την επικοινωνία του με τον ενδιάμεσο ακόμα και στις περιπτώσεις που πρόθεση του είναι η προσπέλαση με βάση ένα άλλο πρωτόκολλο π.χ. Gopher, FTP. Οι απαντήσεις που αφορούν άλλα πρωτόκολλα (π.χ. Gopher ή FTP directory listings) επιστρέφονται στον client σε μορφή HTML εγγράφου το οποίο συνθέτει ο proxy server.

Ο proxy server, με βάση τα παραπάνω, θα πρέπει να είναι σε θέση να ενεργεί τόσο σαν client όσο και σαν server. Τα πεδία επικεφαλίδας τα οποία ενσωματώνονται στην HTTP αίτηση που υποβάλλει ο client στον proxy server περνούν, χωρίς μεταβολή, στην αίτηση του τελευταίου προς τον απομακρυσμένο server. Ένας πλήρης proxy server θα πρέπει να είναι σε θέση να υποστηρίξει όλα τα πρωτόκολλα του Web, τα σπουδαιότερα από τα οποία είναι: HTTP, FTP, Gopher, WAIS και NNTP.

Ο CERN HTTPD μπορεί να λειτουργήσει και σαν proxy server. Μπορεί να δεχτεί πλήρη URLs και να διεκπεραιώσει αιτήσεις πολλαπλών πρωτοκόλλων. Μπορεί ταυτόχρονα να λειτουργήσει σαν proxy server αλλά και σαν συνήθης HTTPD. Επίσης είναι σε θέση να εφαρμόσει caching στα ανταλλασσόμενα δεδομένα. Στην εικόνα 6-3 το οποίο ακολουθεί παρουσιάζεται η βασική ιδέα του μηχανισμού caching σε proxy servers.



Εικόνα 6-3: Caching σε proxy servers

Υπάρχουν αρκετά προβλήματα τα οποία θα πρέπει να αντιμετωπιστούν με την εισαγωγή του caching στους proxy servers. Το πλέον σημαντικό από αυτά αφορά την εγκυρότητα σε σχέση με τον χρόνο παραμονής ενός εγγράφου-αρχείου στην cache. Ο χρόνος ζωής ενός εγγράφου έχει προβλεφτεί στον σχεδιασμό του HTTP (κατάλληλα πεδία στην επικεφαλίδα του πρωτοκόλλου). Επίσης έχει προβλεφτεί η μέθοδος HEAD μέσω της οποίας είναι δυνατή η επιβεβαίωση της εγκυρότητας του αρχείου ενώ μία GET αίτηση μπορεί να συνοδευτεί από συνθήκη (conditional GET). Το πεδίο της ημερομηνίας λήξεως όμως δεν χρησιμοποιείται ευρέως από τους HTTP servers.

6.2 ΓΙΑΤΙ ΧΡΕΙΑΖΟΜΑΣΤΕ ΤΑ FIREWALL

Ο ρόλος των firewall είναι η προστασία των δικτύων. Ποιο συγκεκριμένα επειδή ο όρος δίκτυο δεν είναι απόλυτα σαφής μπορεί να μην είναι και κατανοητό, για ποιο λόγο ένα δίκτυο χρειάζεται προστασία.

Από πλευράς επιχειρήσεων τα πολύτιμα στοιχεία ενός δικτύου είναι:

- Η εμπιστευτικότητα των δεδομένων που μεταφέρονται
- Η αξιοπιστία μεταφοράς δεδομένων
- Η διαθεσιμότητα του δικτύου

Μερικά δεδομένα είναι πολύτιμα γιατί δεν είναι ευρέως γνωστά. Για παράδειγμα θα ήταν πολύτιμο αν γνωρίζαμε τις αυριανές τιμές του χρηματιστηρίου. Αν όμως είχαν όλοι πρόσβαση σε αυτές τις πληροφορίες τότε πιθανόν να ήταν άνευ αξίας. Πολλές εταιρίες κατέχουν εμπιστευτικά δεδομένα σε αρχεία υπολογιστών. Αυτοί οι υπολογιστές πρέπει να προστατευτούν από επιθέσεις, μη εξουσιοδοτημένη χρήση και από άλλα γεγονότα τα οποία μπορούν να οδηγήσουν σε διαρροές δεδομένων.

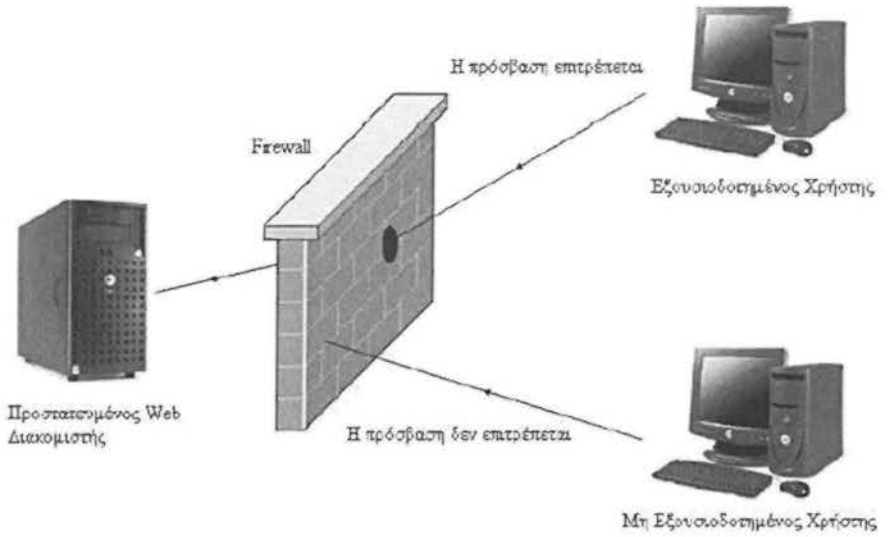
Η αξιοπιστία των δεδομένων είναι ο βαθμός στον οποίο μπορεί να είναι σίγουρος ότι τα δεδομένα θα είναι πλήρη και ακριβή. Η αξιοπιστία των δεδομένων είναι σημαντική γιατί η αξία τους μειώνεται ή χάνεται αν τα περιεχόμενά τους μεταβληθούν ή είναι ανακριβή.

Μερικά δίκτυα υποστηρίζουν επιχειρήσεις ή παρέχουν υπηρεσίες. Έτσι μια δυσλειτουργία τους μπορεί να είναι πολλές φορές καταστροφική. Για παράδειγμα το "EBay" έχει γίνει πολλές φορές θύμα επιθέσεων λόγω του ότι παρέχει On-Line συναλλαγές. Η αναστολή της λειτουργίας του για κάποιο διάστημα μπορεί να αποφέρει σημαντικές απώλειες κερδών.

Πριν την εξάπλωση του Ιντερνετ οι οργανισμοί και οι επιχειρήσεις διατηρούσαν ιδιωτικά δίκτυα υπολογιστών τα οποία δεν παρείχαν απομακρυσμένη πρόσβαση. Έτσι η επίθεση σε έναν υπολογιστή ή γενικότερα σε ένα δίκτυο απαιτούσε φυσική επαφή με το στόχο. Ο επιτιθέμενος έπρεπε να είχε εξουσιοδοτημένη πρόσβαση ή με κάποιο τρόπο να εισβάλει παράνομα στην εταιρία με σκοπό την πρόσβαση σε κάποιο τερματικό. Έτσι οι περισσότερες επιθέσεις γίνονταν από υπαλλήλους. Με τη χρήση των modem ανοίχτηκε μια νέα οδός επιθέσεων και αρκούσε μια τηλεφωνική κλήση από απομακρυσμένη περιοχή αντί για μια παραβίαση φυσικών μέτρων ασφαλείας. Στις μέρες μας όλες οι επιχειρήσεις έχουν δημόσια δίκτυα τα οποία περιέχουν και δεδομένα τα οποία χρειάζονται προστασία. Ένας επιτιθέμενος ο οποίος μπορεί να αποκτήσει παραπάνω προνόμια πρόσβασης μπορεί να προκαλέσει από απώλεια δεδομένων μέχρι την κατάρρευση του δικτύου.

Το firewall ενός αυτοκινήτου είναι σχεδιασμένο για να εμποδίζει την εξάπλωση της φωτιάς από το χώρο της μηχανής στην καμπίνα των επιβατών. Σκοπός του είναι ο περιορισμός. Ένα firewall δικτύου είναι επίσης μια συσκευή περιορισμού. Ένα firewall λειτουργεί με το να διαιρεί εμποδίζει τη μη εξουσιοδοτημένη κίνηση να εισέρχεται ή να εξέρχεται. Αν ρυθμιστεί σωστά μπορεί να εμποδίσει μια επίθεση να φτάσει στο προορισμό της.

Η εικόνα 6-4 δείχνει ένα απλό τυπικό firewall το οποίο επιτρέπει μόνο σε συγκεκριμένους χρήστες να προσπελάσουν τον Web Server. Ένας επιτιθέμενος για να αποκτήσει πρόσβαση στον Web Server πρέπει πρώτα να νικήσει το firewall.



Εικόνα 6-4: Firewall

Ένα firewall κάνει περισσότερα από το να μπλοκάρει απλά τη μη εξουσιοδοτημένη πρόσβαση. Το firewall έχει δύο πρωταρχικούς ρόλους: παρεμπόδιση και ανίχνευση. Η πολιτική ασφαλείας καθορίζει τις λειτουργίες που ένας χρήστης είναι εξουσιοδοτημένος να εκτελεί. Το firewall εμποδίζει τις επιθέσεις. Η ανίχνευση αυτή γίνεται με το να κρατάει το αρχείο (log) για τις προσπάθειες και της πραγματοποιήσεις συνδέσεων με μηχανήματα του δικτύου και να ενημερώσει τους διαχειριστές με ύποπτες ενέργειες. Προφανώς το firewall δεν μπορεί να εμποδίσει μια επίθεση την οποία απέτυχε να εντοπίσει. Άρα η ανίχνευση προηγείται της παρεμπόδισης. Ένα firewall μπορεί να φιλτράρει την κίνηση με διάφορους τρόπους. Οι πιο κοινοί είναι μέσω των:

- IP διευθύνσεων
- Υπηρεσιών

Η κίνηση του δικτύου σηματοδοτείτε από τις IP διευθύνσεις, οι οποίες δείχνουν την αφετηρία και το προορισμό του host. Η κίνηση η οποία απαιτεί πρόσβαση σε μια υπηρεσία καθορίζεται από τον αριθμό του port το οποίο συνδυαζόμενο με την IP καθορίζει την υπηρεσία. Επίσης ένα firewall πρέπει να μπορεί να προστατεύει τον κεντρικό υπολογιστή του δικτύου. Κάτι τέτοιο μπορεί να μοιάζει ασήμαντο αλλά φανταστείτε αν ο επιτιθέμενος αποκτώντας πρόσβαση, χρησιμοποιήσει τον κεντρικό υπολογιστή σαν μεσάζοντα των επιθέσεων του. Αυτό σημαίνει ότι ένα firewall πρέπει να ελέγχει την εισερχόμενη και την εξερχόμενη κίνηση του.

6.3 ΜΕΙΟΝΕΚΤΗΜΑΤΑ FIREWALL

Τα firewall παρουσιάζουν διάφορα μειονεκτήματα στις περισσότερες περιπτώσεις τα οφέλη υπερέχουν των μειονεκτημάτων, παρόλα αυτά πρέπει να γνωρίζουμε τα μειονεκτήματα με σκοπό να τα ελαχιστοποιήσουμε ή να τα υπερπηδήσουμε. Τα μειονεκτήματα πηγάζουν από

το γεγονός ότι το firewall μετατρέπεται σε ένα στενό πέρασμα επηρεάζοντας το δίκτυο με τους τρεις παρακάτω τρόπους:

- Αξιοπιστία
- Απόδοση
- Ευκαμψία

Η αποτυχία λειτουργίας ενός firewall μειώνει την αξιοπιστία του δικτύου. Αυτό αποφεύγει με τη χρήση πολλαπλών firewall ρυθμιζόμενα έτσι ώστε η αποτυχία λειτουργίας του ενός να ενεργοποιήσει το άλλο. Με παρόμοιο τρόπο μπορεί να μειωθεί και η απόδοση του δικτύου. Επειδή όλη η κίνηση πρέπει να περάσει μέσα από ένα firewall η απόδοση του δικτύου επηρεάζεται από την ικανότητα χειρισμού του όγκου πληροφοριών από το firewall και επιπλέον για να χειριστεί ένα νέο τύπο δεδομένων το firewall πρέπει να ρυθμιστεί ξανά.

6.4 ΤΕΧΝΟΛΟΓΙΕΣ FIREWALL

Οι τεχνολογίες οι οποίες χρησιμοποιούνται κατά τη δόμηση των firewall περιλαμβάνουν την προώθηση των πακέτων και το φιλτράρισμα τους, τους εξυπηρετητές εφαρμογών και τέλος πιο σύγχρονες τεχνολογίες όπως τα λεπτομερούς επιθεώρησης και τα υβριδικά firewall. Επιπλέον υπάρχουν κάποιες άλλες σημαντικές τεχνολογίες οι οποίες χρησιμοποιούνται σε συνδυασμό με τα firewall όπως η μετάφραση διευθύνσεων δικτύου (NAT) και η χρήση εικονικών ιδιωτικών δικτύων (VPNs).

6.4.1 ΠΡΟΩΘΗΣΗ ΠΑΚΕΤΩΝ.

Στο κόσμο του Linux είναι πολύ πιθανό να δούμε έναν υπολογιστή να εκτελεί χρέη δρομολογητή. Ένα firewall μπορεί να πραγματοποιηθεί με το συνδυασμό τεχνολογιών δρομολόγησης και άλλων λειτουργιών. Ένας δρομολογητής αποτελείται από δυο ή περισσότερους προσαρμογείς δικτύου οι οποίοι συνδέουν διαφορετικά δίκτυα μεταξύ τους. Κάθε δρομολογητής περιλαμβάνει ένα πακέτο δρομολόγησης με ένα σύνολο κανόνων το οποίο καθορίζει ποια πακέτα θα προωθηθούν και που. Το πότε και που θα προωθηθούν τα πακέτα καθορίζεται από:

- Τον προσαρμογέα δικτύου στον οποίο φτάνει το πακέτο
- Τη διεύθυνση αφετηρίας του πακέτου
- Τη διεύθυνση προορισμού του πακέτου

6.4.2 ΦΙΛΤΡΑΡΙΣΜΑ ΠΑΚΕΤΩΝ

Θεωρήστε το firewall σαν μια συσκευή η οποία διαχειρίζεται την κυκλοφορία. Τα firewall υλικού αποτελούνται από δρομολογητές ή από υπολογιστές οι οποίοι φέρουν το κατάλληλο λογισμικό. Οι δρομολογητές λειτουργούν σε επίπεδο δικτύου και μπορούν να φιλτράρουν IP πακέτα βασιζόμενα στις τιμές τους περιεχομένου της επικεφαλίδας του πακέτου όπως η διεύθυνση αφετηρίας και προορισμού. Οι δρομολογητές μπορούν να παραμετροποιηθούν ώστε να επιτρέπουν τη πρόσβαση μόνο σε συγκεκριμένα πακέτα, να επιτρέπουν την πραγματοποίηση συνδέσεων μόνο από συγκεκριμένους υπολογιστές και να μπλοκάρουν την πρόσβαση των μη εξουσιοδοτημένων μηχανημάτων. Αυτή η διαδικασία συχνά αναφέρετε σαν φιλτράρισμα πακέτων (Packet Filtering).

6.4.3 ΜΕΤΑΦΡΑΣΗ ΔΙΕΥΘΥΝΣΕΩΝ ΔΙΚΤΥΟΥ (NAT)

Η μετάφραση διευθύνσεων δικτύου σχεδιάστηκε για να επιτρέψει σε πολλαπλούς host να μοιράζονται μια IP διεύθυνση. Το NAT είναι μια απλή λειτουργία φιλτραρίσματος πακέτων η οποία πραγματοποιείται από δρομολογητές ή firewall κατά την οποία η διεύθυνση προορισμού ή αφετηρίας μεταβάλλεται.

Με χρήση DTAT (Destination) μεταβάλλεται η διεύθυνση προορισμού ενώ με χρήση SNAT (source) μεταβάλλεται η διεύθυνση αφετηρίας. Με τη χρήση NAT επιτυγχάνεται η οικονομία διευθύνσεων IP. Για παράδειγμα ο πελάτης καλώντας την ίδια IP διεύθυνση μπορεί να συνδεθεί μέσο δρομολογητή με πολλούς διακομιστές ανάλογα με την υπηρεσία για την οποία έχει κάνει αίτηση.

6.5 ΑΡΧΙΤΕΚΤΟΝΙΚΕΣ FIREWALL

Ο σχεδιασμός ενός firewall περιλαμβάνει δύο βασικές διεργασίες

- Σχεδιασμός κανόνων
- Καθορισμός της τοποθέτησης του firewall ή των firewalls

Η τοποθέτηση του firewall σε ένα δίκτυο ονομάζεται αρχιτεκτονική. Η αρχιτεκτονική του firewall είναι πολύ στενά συνδεδεμένη με την αρχιτεκτονική του δικτύου. Για αυτό και ο σχεδιασμός του firewall συνδέεται με το σχεδιασμό του δικτύου. Ο σχεδιασμός του firewall είναι μια διεργασία του σχεδιασμού δικτύων. Αυτός ο τομέας περιγράφει μερικές κοινές αρχιτεκτονικές firewall.

Οι τρόποι υλοποίησης ενός firewall αν και είναι εξαρτώμενοι από τις εκάστοτε ανάγκες, διαδικτυακές εφαρμογές και την τοπολογία του εσωτερικού δικτύου μπορούν να περιγραφούν μέσω των παρακάτω τεσσάρων μμεθόδων. Οι άλλες είναι συνδυασμοί των βασικών αυτών αρχιτεκτονικών.

6.5.1 ROUTER FIREWALL

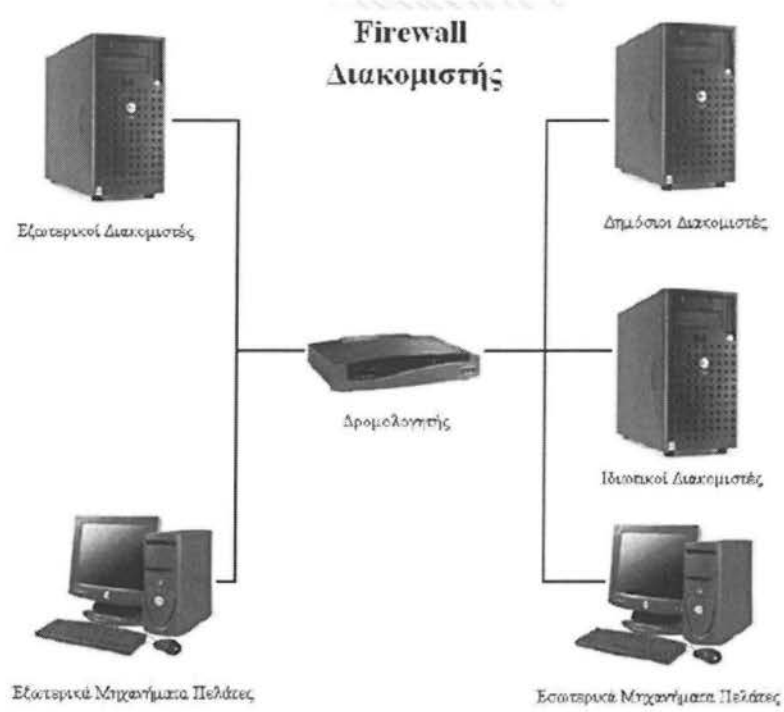
Η εικόνα 6-5 μας δείχνει την αρχιτεκτονική η οποία ονομάζεται router firewall. Στην πραγματικότητα αυτή η αρχιτεκτονική δεν περιλαμβάνει firewall. Αντί για αυτό ένας δρομολογητής αναλαμβάνει την επικοινωνία του εξωτερικού δικτύου με το εσωτερικό. Το ότι ο δρομολογητής παρέχει λειτουργίες προώθησης πακέτων είναι μια απλή μορφή φιλτραρίσματος πακέτων η οποία μπορεί να χρησιμοποιηθεί για να προστατέψει ένα δίκτυο. Το μέγεθος προστασίας είναι μικρό διότι η προώθηση πακέτων επιθεωρεί μόνο την διεύθυνση IP του πακέτου. Η άμυνα που παρέχει αυτή η αρχιτεκτονική δεν είναι επαρκής επειδή έχει μόνο ένα επίπεδο ασφαλείας. Ο δρομολογητής δεν μπορεί να προγραμματιστεί για να μπλοκάρει ή να δέχεται πακέτα βασιζόμενος στο port. Έτσι κάθε υπηρεσία που παρέχεται από το εσωτερικό δίκτυο είναι διαθέσιμη και ως προς τρίτους. Για αυτό δεν είναι δυνατό να παρέχουμε ιδιωτικές υπηρεσίες. Παρά τις αδυναμίες της αρχιτεκτονικής αυτής μπορούμε να το επιλέξουμε λόγο του ελάχιστου κόστους και της μέγιστης απόδοσης. Η προώθηση των πακέτων αν και παρέχει μικρή ασφάλεια είναι η ποιο γρήγορη από της άλλες τεχνολογίες firewall.

Μερικά παραδείγματα φίλτρου της κυκλοφορίας είναι τα ακόλουθα :

- Απαγόρευση όλων των συνδέσεων από συστήματα του εξωτερικού δικτύου (INTERNET) εκτός αυτών που υλοποιούν SMTP συνδέσεις (email).
- Περιορισμός όλων των συνδέσεων από και προς τα «ευαίσθητα συστήματα» που χρειάζονται ιδιαίτερη προστασία.
- Ελευθερία συνδέσεων που υλοποιούν e-mail και υπηρεσίες FTP, απαγόρευση των συνδέσεων που αφορούν υπηρεσίες όπως TFTP, X-Window system, RPC, και «r» υπηρεσίες (rlogin, rsh, rcp, κλπ).

Τα κυριότερα μειονεκτήματα της αρχιτεκτονικής αυτής είναι τα ακόλουθα:

- Υλοποιείται με μία συσκευή, ή οποία επιπλέον εκτελεί και λειτουργίες δρομολόγησης. Σε περίπτωση βλάβης, δυσλειτουργίας ή «κατάληψης» από εισβολέα το εσωτερικό δίκτυο μένει απροστάτευτο.
- Περιορίζει ή επιτρέπει υπηρεσίες στο σύνολό τους και δεν προστατεύει από ενέργειες που εκτελούνται διαμέσου μιας υπηρεσίας.
- Σημαντική δυσκολία στον προγραμματισμό των ειδικών δρομολογητών και αδυναμία εκτέλεσης λειτουργιών κρυπτογράφησης.



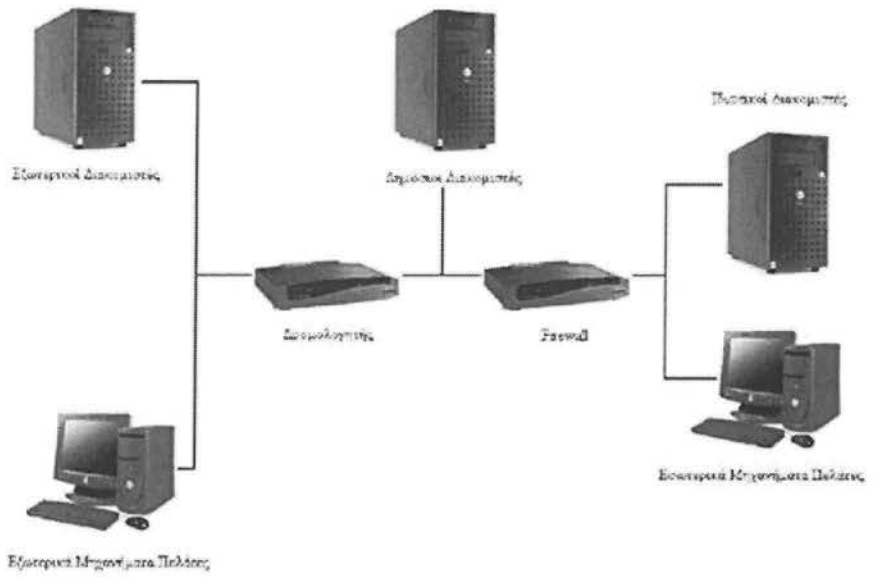
Εικόνα 6-5: Αρχιτεκτονική Router Firewall

6.5.2 SINGLE HOST FIREWALL

Αυτός ο τομέας εξηγεί τις αρχιτεκτονικές firewall οι οποίες πραγματοποιούνται μόνο με ένα firewall φιλτραρίσματος πακέτων ή proxy. Τα firewall φιλτραρίσματος πακέτων είναι πιο δημοφιλή από τα proxy γιατί μπορούν να στεγάσουν μεγαλύτερη ποικιλία πρωτοκόλλων. Εντούτοις τα proxy firewalls είναι ικανά να πραγματοποιήσουν πιο εξειδικευμένο φιλτράρισμα από αυτά των φιλτραρίσματος πακέτων. Έχοντας ένα φιλτράρισμα πακέτων ή proxy firewall διαιρούμε το δίκτυο σε δυο υποδίκτυα:

- Το εσωτερικό ιδιωτικό δίκτυο
- Το περιμετρικό δίκτυο, γνωστό σαν DMZ(Demilitarized Zone)

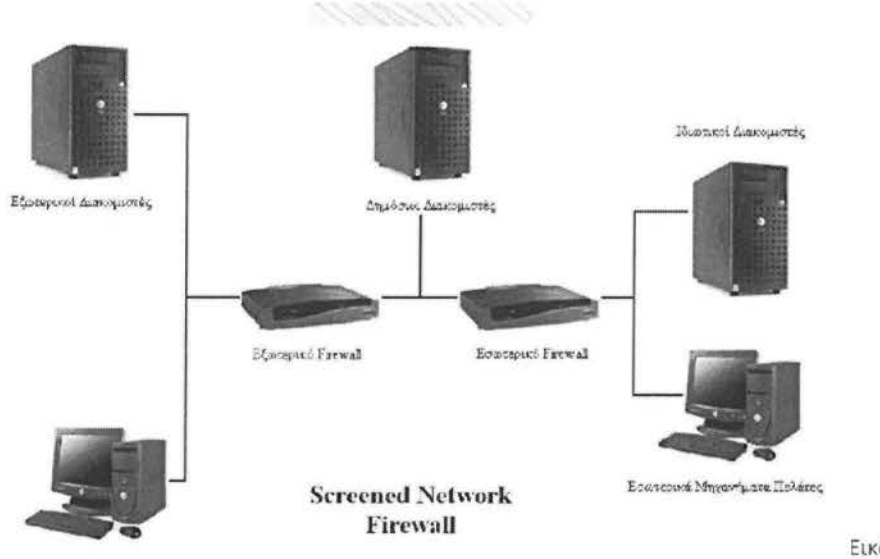
Η εικόνα 6-6 απεικονίζει μια απλή αρχιτεκτονική single host firewall γνωστή σαν exposed host firewall. Το firewall προωθεί πακέτο από και προς το εξωτερικό δίκτυο αλλά και φιλτράρει με βάση κάποιους κανόνες. Το αδύνατο σημείο της αρχιτεκτονικής αυτής, είναι οι εκτεθειμένοι hosts. Αυτή η αρχιτεκτονική τοποθετεί τους δημόσιους διακομιστές σε μια ευαίσθητη περιοχή και έτσι είναι εκτεθειμένοι σε εξωτερικές επιθέσεις. Ένα single host firewall περιέχει μόνο περιέχει μόνο δύο δίκτυα (δύο διεπαφές) και έτσι οι δημόσιοι διακομιστές μπορεί να είναι στο περιμετρικό δίκτυο ή στο εσωτερικό δίκτυο.



Εικόνα 6-6: Αρχιτεκτονική Single Host Firewall

6.5.3 MULTI-HOST FIREWALL

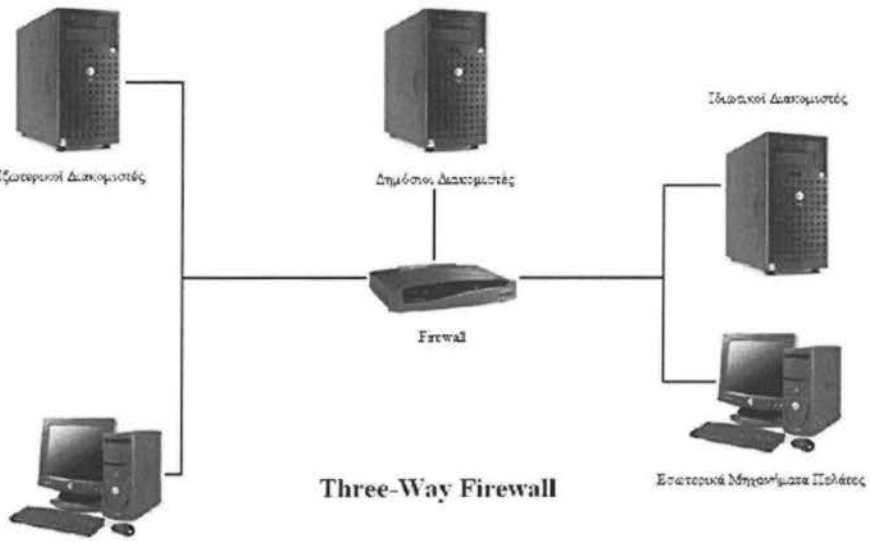
Με τα multi-host firewall μπορούμε να νικήσουμε πολλούς περιορισμούς από τους single host firewall. Στην αρχιτεκτονική αυτή περιλαμβάνονται δύο firewall. Το ένα firewall βρίσκεται στην περίμετρο στο εξωτερικό δίκτυο ενώ το άλλο βρίσκεται στη περίμετρο του εσωτερικού ιδιωτικού δικτύου. Ανάμεσα στο δύο firewall είναι η περιοχή DMZ όπου εκεί βρίσκονται και οι δημόσιοι διακομιστές μας.



Εικόνα 6-7: Αρχιτεκτονική Screened Network Firewall

Το να έχουμε ένα δεύτερο firewall έχει πολλά πλεονεκτήματα. Μπορούμε να εφαρμόσουμε διαφορετικές πολιτικές ασφαλείας στα δύο firewall. Το εξωτερικό firewall γνωστό και ως firewall gateway προστατεύει τους δημόσιους διακομιστές ενώ το εσωτερικό firewall

μπορεί να έχει πολιτικές για την ασφάλεια του εσωτερικού μας δικτύου. Κάθε host λοιπόν προστατεύεται από ένα firewall. Αν θέλουμε να αποφύγουμε την χρησιμοποιήσει του δεύτερου firewall μπορούμε να προσομοιώσουμε την αρχιτεκτονική αυτή με τη χρήση ενός firewall το οποίο έχει τρείς δικτυακές διεπαφές. Το αποτέλεσμα φαίνεται παρακάτω γνωστό σαν three way firewall. Όπως με τους screened network firewalls τοποθετούμε τους δημόσιους διακομιστές και τους ιδιωτικούς Host πίσω από το firewall επιτυγχάνοντας έτσι μεγάλο βαθμό ασφαλείας.



Εικόνα 6-8: Αρχιτεκτονική Three-Way Firewall

6.6 ΣΧΕΔΙΑΣΜΟΣ FIREWALL

Αρχικό κριτήριο για την επιλογή ενός firewall είναι το κόστος. Πολλοί οργανισμοί αποφεύγουν την αγορά πολύ ακριβών firewall και πάνε σε μεσαίες λύσεις. Αρχικά γίνεται ανάλυση για τις απαιτήσεις του δικτύου μας και ύστερα καταφεύγουμε σε κάποιες λύσεις που τις ικανοποιούν. Πολλοί συχνά όμως μας ξεφεύγει ότι μπορούμε με ελεύθερο ανοιχτό λογισμικό να δημιουργήσουμε πολύ αξιόπιστα firewall σε ένα παλιό Η/Υ.

Αρχικά πρέπει να αναγνωρίσουμε την αρχιτεκτονική και τις τεχνολογίες που θα χρησιμοποιήσουμε. Ύστερα πρέπει να καθορίσουμε τις πολιτικές που θα χρησιμοποιήσουμε. Αρχικά αναγνωρίσουμε ποιες υπηρεσίες πρέπει να επιτρέπονται και ποιοι host θέλουν εξουσιοδότηση και σε ποιες υπηρεσίες. Πρέπει να αναγνωρίσουμε τα χαρακτηριστικά κάθε υπηρεσίας και προετοιμασία των εγγράφων για το firewall μας.

Αφού σχεδιαστούν και αναλυθούν οι απαιτήσεις επόμενο στάδιο είναι η υλοποίηση της αρχιτεκτονικής του δικτύου και του firewall. Βήμα βήμα υλοποιούνται και εξετάζονται όλα τα κομμάτια της ανάλυσης σύμφωνα με τα έγγραφα που έχουν δημιουργηθεί.

Πολύ μεγάλο ρόλο σε μια τέτοια ανάλυση παίζει ο project manager που καθορίζει το χρονοδιάγραμμα με το οποίο θα υλοποιηθεί το έργο. Συγχρονίζει τα πάντα έτσι ώστε όλα να γίνουν στην ώρα τους. Από τεχνικούς που θα στηθεί ένα rack, τους προγραμματιστές που αναπτύσσουν προγράμματα εφαρμογών, εκείνους που ασχολούνται με τα δίκτυα αλλά και την ασφάλεια δικτύων.

Επόμενο στάδιο είναι η δοκιμή του δικτύου μας και να ελέγξουμε ότι όλες οι πολιτικές ασφαλείας λειτουργούν σύμφωνα με τις προδιαγραφές που έχουν αναλυθεί και σχεδιαστεί. Όλες οι δοκιμές πρέπει να συγκριθούν με τα αναμενόμενα αποτελέσματα των εγγράφων και κάθε λάθος πρέπει να διορθωθεί. όλες οι εργασίες που γίνονται: σχεδιασμός, ανάλυση, υλοποίηση γίνονται σύμφωνα με κάποια μοντέλα. Τα μοντέλα αυτά συνήθως αποτρέπουν την ανάγκη να ξανασχεδιαστεί ένα δίκτυο σε περίπτωση λάθους πριν την ολοκλήρωση του.

Μετά τους ελέγχους περνάμε στη παραγωγή, όπου το δίκτυο μας είναι online. Απομένει επομένως η διαχείριση και συντήρηση του. Τα firewall πρέπει να κρατάνε logs και ο διαχειριστής να επεμβαίνει άμεσα όταν χρειάζεται. Η διαχείριση όμως πρέπει να γίνεται ακόμα και στο ποιο μικρό firewall και συνήθως είναι ακριβή.

6.6.1 ΕΠΙΛΟΓΗ ΛΟΓΙΣΜΙΚΟΥ ΚΑΙ ΥΛΙΚΟΥ

Κύριο κριτήριο δεν είναι η τεχνολογία, αλλά ένα προϊόν που περιέχει πολλά χαρακτηριστικά:

- Κόστος
- NAT, VPN, Logs κτλ
- Τεχνική υποστήριξη
- Ευκολία χρήσης
- Σταθερότητα
- Απόδοσης

6.7 ΟΔΗΓΟΣ ΡΥΘΜΙΣΗΣ FIREWALL

Ένας οργανισμός θα πρέπει να συμμορφώνεται με την γενικευμένη πολιτική τείχους προστασίας που αναφέρεται παρακάτω.

- Το firewall θα πρέπει να ρυθμιστεί ώστε να αρνηθεί όλες τις υπηρεσίες που δεν επιτρέπονται.
- Οι κανόνες θα πρέπει να ρυθμιστούν αναλυτικά με τις διευθύνσεις IP αποστολέα/παραλήπτη, πόρτα και ποια πλευρά ξεκίνησε τη σύνδεση.
- Όλα τα firewall θα πρέπει να χρησιμοποιούν stateful επιθεώρηση.
- Οι λεπτομέρειες του εσωτερικού δικτύου του οργανισμού δεν θα πρέπει να φαίνονται έξω από το firewall.

- Τα ιδιωτικά δίκτυα πρέπει να είναι αόρατα.
- Κανένα πρωτόκολλο δρομολόγηση δεν πρέπει να χρησιμοποιείται μεταξύ το Διαδίκτυο και το εσωτερικό δίκτυο.
- Τα firewall δεν πρέπει να τρέχουν κανένα πρωτόκολλο δρομολόγησης.
- Όλες οι προσπάθειες σύνδεσης που δεν πληρούν καθορισμένη πολιτική πρέπει να μπλοκαριστούν .
- Εκτός από τον διαχειριστή κανένας άλλος χρήστης δεν θα πρέπει να έχει πρόσβαση στον firewall.
- Εξωτερικά πακέτα με εσωτερική διεύθυνση IP θα πρέπει να αναγνωρίζονται και να μπλοκάρονται.
- Εσωτερικά πακέτα με εξωτερική διεύθυνση IP θα πρέπει να αναγνωρίζονται και να μπλοκάρονται.
- Το firewall θα πρέπει να λειτουργεί σαν προωθητής για τα εξωτερικά πακέτα του Ιντερνετ και όχι σαν δρομολογητής. Το firewall δεν θα πρέπει να δρομολογεί καμιά κίνηση μεταξύ της εξωτερικής διεπαφής και τη εσωτερικής διεπαφής του Intranet.

6.8 ΔΙΑΧΕΙΡΙΣΗ DMZ

Οι τύποι τις κίνησης που είναι απαραίτητοι για τη διαχείριση των συστημάτων στη DMZ ζώνη ενός οργανισμού είναι:

Ping/Traceroute: Η εντολή Ping και Traceroute είναι χρήσιμα εργαλεία για να δούμε αν ένας χρήστης σε ένα δίκτυο είναι online και λειτουργεί σωστά στο δίκτυο. Ωστόσο ένα ICMP πακέτο μπορεί να χρησιμοποιηθεί για διάφορες επιθέσεις. Οι υπηρεσίες ping και traceroute θα πρέπει να χρησιμοποιηθούν όπου χρειάζεται. Να επιτρέπονται δηλαδή μόνο ICMP echo-request μηνύματα και ICMP echo-reply μηνύματα. Τα ping πακέτα θα πρέπει να μπλοκάρονται αν έρχονται από το Ιντερνετ προς την DMZ ζώνη αλλά και στους δρομολογητές, switches και firewalls. Αν το ICMP είναι απαραίτητο για να διαχειριστούμε το firewall, ένας κανόνας θα πρέπει να δημιουργηθεί για να περιορίσει τη πρόσβαση μόνο στις IPs που είναι στο δίκτυο των διαχειριστών.

SNMP: Το SNMP είναι απαραίτητο για να διατηρήσει την απόδοση των συστημάτων στο DMZ. Το SNMP πρωτόκολλο μπορεί να χρησιμοποιηθεί για να στείλει αναφορά σε ένα Network Management System (NMS) ότι υπάρχει πρόβλημα που χρειάζεται την προσοχή μας. Το πρωτόκολλο SNMP θα πρέπει να ενεργοποιηθεί στα συστήματα που το χρειάζονται, αλλά μόνο με "Read-Only" community συμβολοσειρά. Η "Read-Write" δυνατότητα θα πρέπει να απαγορεύεται αυστηρά σε όλα τα συστήματα, hosts, δρομολογητές, switches και firewalls

Παρακάτω θα δούμε μερικές εφαρμογές που δεν πρέπει να τρέχουν σε κανένα διακομιστή της DMZ ζώνης. Αυτές οι υπηρεσίες συνήθως μπλοκάρονται.

- **Finger:** Όπου εμφανίζει πληροφορίες για κάποιο χρήστη σε κάποιο σύστημα που τρέχει την υπηρεσία finger.

- **Rlogin, rexec, και rsh:** Υπηρεσίες που τρέχουν στις πόρτες 513, 512 και 514 μπορούν να επιτρέψουν μη εξουσιοδοτημένη πρόσβαση σε χρήστες αν δεν έχουν ρυθμιστεί σωστά.
- **X-window, OpenWindows,** πόρτες 6000+ και 2000 μπορεί να επιτρέψει στους εισβολείς να δουν όλη τη κίνηση του πληκτρολογίου και της οθόνης και ακόμη να πάρουν τον έλεγχο.
- **Remote Procedure Call (RPC):** πόρτα 111, υπηρεσίες όπως NIS και NFS μπορούν να χρησιμοποιηθούν για να καταγράψουν κωδικούς και να διαβάσουν και να γράψουν σε αρχεία.
- **Tftp:** πόρτα 69, μπορεί να χρησιμοποιηθεί για την ανάγνωση αρχείων αν το σύστημα δεν είναι ρυθμισμένο σωστά.

Διαδραστικά πρωτόκολλα όπως τα: **Telnet, FTP, SSH,SCP.** Τα πρωτόκολλα telnet, ssh και scp χρησιμοποιούνται για τον απομακρυσμένο έλεγχο του συστήματος. Πάντα προτείνεται το SSH πρωτόκολλο ή το SCP αφού η επικοινωνία με αυτά τα πρωτόκολλα είναι κρυπτογραφημένη. Πρωτόκολλα που δίνουν δικαιώματα διαχείρισης θα πρέπει να επιτρέπονται μόνο από το δίκτυο των διαχειριστών.

Πρωτόκολλα διαχείρισης: **SNMP, system backup**. Τα μη διαδραστικά πρωτόκολλα διαχείρισης θα πρέπει να αντιμετωπίζονται με το ίδιο τρόπο.

Πρωτόκολλα εφαρμογών: **HTTP, HTTPS, POP, SQL-Net.**

- Μόνο τα πρωτόκολλα που χρειάζονται πρέπει να επιτρέπονται.
- Διαφορετικά πρωτόκολλα εφαρμογών θα πρέπει να χρησιμοποιούνται σε κάθε μεριά του DMZ ώστε οι επίθετες να μην χρησιμοποιούν το DMZ σαν gateway.
- Τα ιδιόκτητα πρωτόκολλα πρέπει να αξιολογηθούν και να εγκριθούν πριν από τη χρήση τους.

Telnet: πόρτα 23.

- Θα πρέπει να απενεργοποιείται σε hosts που έχουν εξωτερική συνδεσιμότητα.
- Όλα τα συστήματα στη DMZ ζώνη που χρειάζονται telnet θα πρέπει να χρησιμοποιούν το SSH.
- Αν το telnet επιτρέπεται από τη πολιτική ασφαλείας, τότε κάθε χρήστης πρέπει να έχει ισχυρούς κωδικούς πρόσβασης και ο διαχειριστής δεν μπορεί να συνδεθεί στο σύστημα σαν **root**.

SMTP: Πόρτα 25 –email servers.

- Εσωτερικά ονόματα και IPs επομένως οι επικεφαλίδες από εξερχόμενα email δεν πρέπει να περιέχουν καθόλου ονόματα ή διευθύνσεις αλλά μόνο το επίσημη εξωτερική διεύθυνση του αποστολέα. Αυτό ισχύει και για μηνύματα σφαλμάτων που στέλνονται πίσω σε έναν εξωτερικό χρήστη.
- Εξωτερικά συστήματα πρέπει να είναι ενημερωμένα για όλα τα λογισμικά που τρέχουν κυρίως των υπηρεσιών που τρέχουν δημόσια.

- Οι διεργασίες του sendmail πρέπει να γίνονται chroot έτσι ώστε να μην τρέχουν από το root χρήστη.

Πρωτόκολλα δρομολόγησης: RIP, IGRP, EIGRP, OSPF, BGP.

- Πρωτόκολλα δρομολόγησης δεν θα πρέπει να τρέχουν στην DMZ ζώνη.
- Αν χρησιμοποιηθεί κάποιο πρωτόκολλο δρομολόγησης τότε θα πρέπει να χρησιμοποιηθεί ένα πρωτόκολλο που υποστηρίζει πιστοποίηση.

DNS: Πόρτα 53.

- Τα εσωτερικά ονόματα πρέπει να παραμένουν ιδιωτικά και να μην φαίνονται στο δημόσιο δίκτυο.
- Ερωτήματα σε μη υπαρκτά ονόματα θα πρέπει να καταγράφονται και να απορρίπτονται.
- Οι εγγραφές στους δημόσιους DNS διακομιστές θα πρέπει περιορίζονται σε διακομιστές που είναι εξωτερικά προσβάσιμη.
- Κανένα ερώτημα δεν θα πρέπει να προωθείται σε εσωτερικούς DNS διακομιστές από εξωτερικούς.
- Οι εσωτερικοί DNS διακομιστές δεν θα πρέπει να διαχειρίζονται εξωτερικούς διευθύνσεις.
- Μεταφορές ζωνών δεν πρέπει να μεταφέρονται σε εξωτερικούς διακομιστές.

Το UDP πρωτόκολλο είναι δύσκολο να φιλτραριστεί με stateful φιλτράρισμα διότι δεν είναι connection-oriented πρωτόκολλο όπως το TCP. Επομένως είναι δύσκολο να καθοριστεί ποιος ξεκίνησε τη σύνδεση.

7 ΤΕΧΝΟΛΟΓΙΑ SERVER VIRTUALIZATION

7.1 ΙΣΤΟΡΙΚΗ ΑΝΑΔΡΟΜΗ

Η ιδεατοποίηση ηλεκτρονικών υπολογιστών είναι μία έννοια που αναπτύχθηκε για πρώτη φορά στη δεκαετία του 1960 για τον διαμερισμό των μεγάλων mainframe υπολογιστών. Σήμερα, οι υπολογιστές που βασίζονται σε αρχιτεκτονική x86 αντιμετωπίζουν τα ίδια προβλήματα «ακαμψίας» και περιορισμένης αξιοποίησης που αντιμετώπιζαν και τα mainframes στην δεκαετία του 1960.

7.1.1 MAINFRAME VIRTUALIZATION

Το Virtualization υλοποιήθηκε πριν από 30 χρόνια αρχικά από την IBM σαν μία μέθοδος για τον λογικό διαμερισμό των mainframe υπολογιστών σε ξεχωριστούς ιδεατούς υπολογιστές (virtual machines). Αυτά τα διαμερίσματα (partitions) επέτρεπαν στα mainframes το "multitask": να εκτελούν πολλαπλές εφαρμογές και διεργασίες παράλληλα. Δεδομένου ότι τα mainframes ήταν ακριβοί πόροι εκείνη την εποχή, σχεδιάστηκαν έτσι ώστε να διαμερίζονται, προκειμένου να αντισταθμίζουν πλήρως (fully leverage) την οικονομική επένδυση.

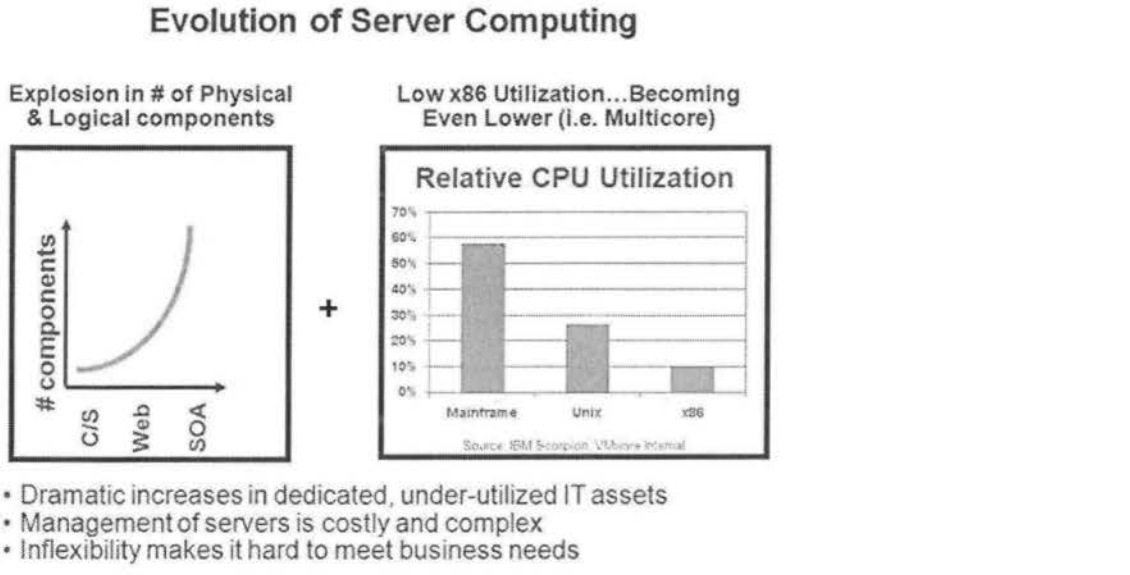
7.1.2 Η ΑΝΑΓΚΗ ΓΙΑ X86 VIRTUALIZATION

Το Virtualization πρακτικά εγκαταλείφθηκε στη διάρκεια των δεκαετιών 1980 και 1990 όταν οι client-server εφαρμογές και οι χαμηλού κόστους x86 εξυπηρετητές και σταθμοί εργασίας, καθιέρωσαν το μοντέλο της κατανεμημένης επεξεργασίας (distributed computing). Αντί της από κοινού χρήσης πόρων κεντρικά, σύμφωνα με το mainframe μοντέλο, οι οργανισμοί χρησιμοποίησαν το χαμηλό κόστος των κατανεμημένων συστημάτων για να δημιουργήσουν «νησίδες» επεξεργαστικής ισχύος. Η ευρεία υιοθέτηση των Microsoft Windows και η ανάδειξη του Linux σαν server operating systems στην δεκαετία του 1990, καθιέρωσαν τους x86 servers ως πρότυπο για την βιομηχανία (industry standard). Η ανάπτυξη στην γρήγορη και εύκολη διάθεση (deployment) των x86 servers και desktops έχει εισάγει νέες προκλήσεις στην IT υποδομή και τις λειτουργίες. Σε αυτές τις προκλήσεις συγκαταλέγονται τα παρακάτω:

- **Περιορισμένη Αξιοποίηση Υποδομής:** Η συνήθης ανάπτυξη των x86 servers επιτυγχάνει μέση χρήση της τάξης του μόνο 10 με 15 τοις εκατό επί της συνολικής χωρητικότητας, σύμφωνα με το International Data Corporation (IDC), μια εταιρεία έρευνας αγορών. Οι οργανισμοί συνήθως εκτελούν μία εφαρμογή ανά εξυπηρετητή προκειμένου να αποφύγουν τον κίνδυνο τα ευπαθή σημεία μίας εφαρμογής να επηρεάσουν την διαθεσιμότητα μιας άλλης εφαρμογής στον ίδιο server.

- **Αυξημένο Κόστος Φυσικής Υποδομής:** Το λειτουργικό κόστος για την υποστήριξη της αυξανόμενης φυσικής υποδομής μεγαλώνει σταθερά. Το μεγαλύτερο μέρος της υποδομής υπολογιστών πρέπει να παραμένει σε λειτουργία συνεχώς, οδηγώντας σε κατανάλωση ισχύος, ανάγκες ψύξης και κόστος εγκαταστάσεων που δεν μεταβάλλονται σύμφωνα με το βαθμό χρήσης.
- **Αυξανόμενο Κόστος Διαχείρισης του IT:** Καθώς τα περιβάλλοντα των υπολογιστών γίνονται περισσότερο σύνθετα, έχει αυξηθεί το επίπεδο εξειδικευμένης εκπαίδευσης και εμπειρίας που απαιτείται από το προσωπικό διαχείρισης της υποδομής, καθώς επίσης το σχετιζόμενο κόστος. Οι οργανισμοί δαπανούν δυσανάλογο χρόνο και πόρους σε χειρονακτικές εργασίες που σχετίζονται με την συντήρηση των εξυπηρετητών, και επομένως απαιτείται περισσότερο προσωπικό για να εκτελέσει αυτές τις εργασίες.
- **Ανεπαρκής Μετάπτωση και Προστασία από Καταστροφές:** Οι οργανισμοί επηρεάζονται αυξανόμενα από την διακοπή λειτουργίας (downtime) κρίσιμων εφαρμογών υπηρεσιών και την αδυναμία πρόσβασης κρίσιμων τελικών χρηστών σε σταθμούς εργασίας. Η απειλή των επιθέσεων στην ασφάλεια, η απειλή των φυσικών καταστροφών, των επιδημιών και της τρομοκρατίας έχει αυξήσει τη σημαντικότητα του σχεδιασμού για τη διατήρηση της ομαλής λειτουργίας του οργανισμού (business continuity planning) τόσο για τους σταθμούς εργασίας όσο και για τους εξυπηρετητές.
- **Έντονη Συντήρηση των Σταθμών Εργασίας τελικών χρηστών:** Η διαχείριση και η ασφάλιση των σταθμών εργασίας των οργανισμών παρουσιάζει πολυάριθμες προκλήσεις. Ο έλεγχος ενός περιβάλλοντος με διεσπαρμένους σταθμούς εργασίας (desktops) και η επιβολή πολιτικής διαχείρισης, πρόσβασης και ασφάλειας χωρίς να εξασθενεί η δυνατότητα των χρηστών να δουλεύουν με αποδοτικό τρόπο είναι πολύπλοκη και δαπανηρή. Πολυάριθμες διορθώσεις (patches) και αναβαθμίσεις πρέπει να εφαρμόζονται συνεχώς στα περιβάλλοντα υπολογιστών γραφείου προκειμένου να εξαλείψουν τα ευπαθή σημεία στην ασφάλεια.

Η εικόνα 7-1 δείχνει την εκρηκτική αύξηση στον πληθυσμό των φυσικών συστημάτων x86 αρχιτεκτονικής τα τελευταία χρόνια, σε συνδυασμό με την ιδιαίτερα περιορισμένη αξιοποίησή τους.



Εικόνα 7-1: Συσχέτιση αύξησης αριθμού εξυπηρετητών & βαθμού αξιοποίησής τους

7.2 ΠΡΟΒΛΗΜΑΤΑ ΣΤΟ ΠΑΡΕΛΘΟΝ

7.2.1 ΠΡΟΚΛΗΣΕΙΣ & ΕΜΠΟΔΙΑ ΣΤΟ X86 VIRTUALIZATION

Αντίθετα με τα mainframes, τα x86 συστήματα δεν σχεδιάστηκαν για να υποστηρίζουν πλήρως το Virtualization, και η κοινωνία της πληροφορικής είχε να ξεπεράσει ισχυρές προκλήσεις για να δημιουργήσει ιδεατά μηχανήματα (virtual machines) βασισμένα σε x86 υπολογιστές.

Η βασική λειτουργία των περισσότερων Κεντρικών Μονάδων Επεξεργασίας (CPUs), τόσο στα mainframes όσο και στα PCs, είναι να εκτελούν μια ακολουθία αποθηκευμένων οδηγιών (π.χ., ένα πακέτο λογισμικού). Στους x86 επεξεργαστές, υπάρχουν 17 συγκεκριμένες οδηγίες που δημιουργούν προβλήματα όταν γίνονται ιδεατές (virtualize), προκαλώντας το λειτουργικό σύστημα να εμφανίσει μια προειδοποίηση, να τερματίσει την εφαρμογή, ή απλά να καταρρεύσει. Σαν αποτέλεσμα, αυτές οι 17 οδηγίες αποτελούσαν ένα σημαντικό εμπόδιο στην αρχική υλοποίηση του Virtualization σε x86 υπολογιστές.

Για τον χειρισμό των προβληματικών οδηγιών στην αρχιτεκτονική των x86 υπολογιστών, οι κατασκευαστές λογισμικού Virtualization ανέπτυξαν μια τεχνική προσαρμογής για το Virtualization, η οποία «παγιδεύει» αυτές τις οδηγίες κατά την δημιουργία τους και τις μετατρέπει σε ασφαλείς οδηγίες που μπορούν να γίνουν «virtualize», ενώ παράλληλα επιτρέπει σε όλες τις άλλες οδηγίες να εκτελούνται χωρίς καμία παρέμβαση. Το αποτέλεσμα είναι ένα υψηλής απόδοσης virtual machine που συνδυάζεται αρμονικά με το hardware του εξυπηρετητή (host) και διατηρεί πλήρη συμβατότητα λογισμικού.

7.3 ΟΡΙΣΜΟΣ SERVER VIRTUALIZATION

7.3.1 ΤΙ ΕΙΝΑΙ ΤΟ SERVER VIRTUALIZATION;

Το Server Virtualization είναι ένα πλαίσιο, μεθοδολογία ή τεχνική που επιτυγχάνει τον διαμερισμό των φυσικών πόρων ενός υπολογιστή σε πολλαπλά περιβάλλοντα εκτέλεσης, εφαρμόζοντας μία ή περισσότερες τεχνολογίες όπως διαμερισμό σε επίπεδο υλικού ή σε επίπεδο λογισμικού, διαμερισμό σε επίπεδο χρόνου, μερική ή ολική προσομοίωση μηχανής, εξομοίωση, ποιότητα υπηρεσιών, και άλλες.

Είναι η μέθοδος εκτέλεσης πολλαπλών ανεξάρτητων ιδεατών λειτουργικών συστημάτων σε έναν φυσικό υπολογιστή. Είναι η απόκρυψη των φυσικών υπολογιστικών πόρων, συμπεριλαμβανομένων του αριθμού και της ταυτότητας των μεμονωμένων φυσικών εξυπηρετητών, επεξεργαστών και λειτουργικών συστημάτων από τους χρήστες του «ιδεατού» εξυπηρετητή.

Το Virtualization, είναι ο διαμερισμός ενός φυσικού συστήματος σε πολλαπλά απομονωμένα μεταξύ τους εικονικά περιβάλλοντα. Τα εικονικά αυτά περιβάλλοντα συνήθως ονομάζονται virtual private servers, αλλά μπορεί κανείς να τα συναντήσει και με το όνομα partitions, guests, instances, containers ή emulations ή virtual machines.

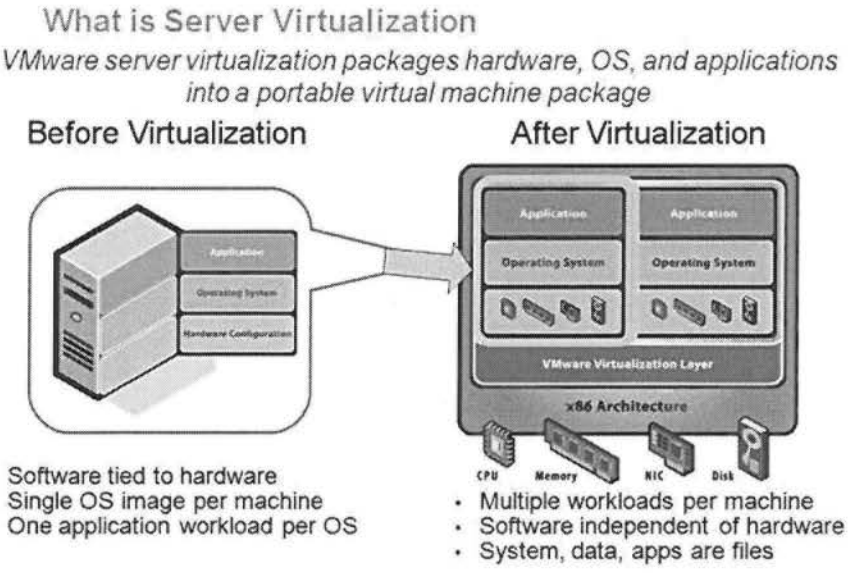
Είναι ένα αφαιρετικό ενδιάμεσο στρώμα που επιτρέπει σε πολλαπλά ιδεατά μηχανήματα, με ετερογενή λειτουργικά συστήματα να λειτουργούν το καθένα ξεχωριστά μέσα σε ένα απομονωμένο περιβάλλον, το ένα δίπλα στο άλλο, πάνω στο ίδιο φυσικό μηχάνημα.

Το Virtualization είναι μια δοκιμασμένη τεχνολογία λογισμικού που μετατρέπει ραγδαία το τοπίο στο IT και αλλάζει θεμελιωδώς τον τρόπο με τον οποίο χρησιμοποιούμε τους υπολογιστές. Οι σημερινοί πολύ ισχυροί x86 υπολογιστές σχεδιάστηκαν αρχικά για να

«τρέχουν» ένα μόνο λειτουργικό σύστημα και μία μόνο εφαρμογή. Το Virtualization καταργεί αυτή τη σύμβαση, κάνοντας εφικτό να εκτελούνται πολλαπλά λειτουργικά συστήματα και πολλαπλές εφαρμογές στον ίδιο υπολογιστή την ίδια χρονική στιγμή, αυξάνοντας έτσι την αξιοποίηση και την προσαρμοστικότητα των φυσικών πόρων.

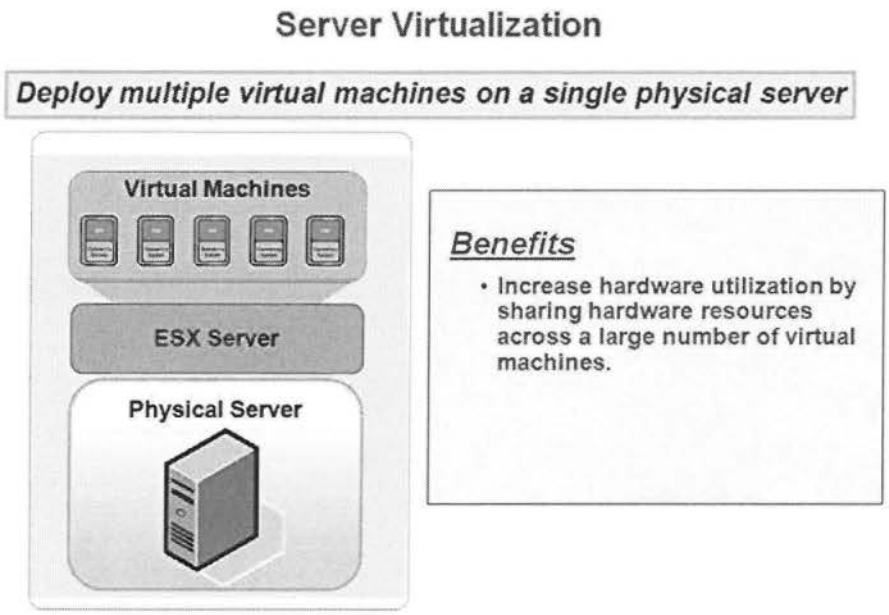
Το Virtualization είναι μια τεχνοτροπία από την οποία μπορούν να ωφεληθούν όλοι όσοι χρησιμοποιούν υπολογιστή, από τους επαγγελματίες της πληροφορικής και τους οπαδούς των Macintosh μέχρι τις εμπορικές επιχειρήσεις και τους κυβερνητικούς οργανισμούς. Χρησιμοποιώντας το Virtualization γίνεται εξοικονόμηση χρόνου, χρημάτων και ενέργειας ενώ παράλληλα δύναται να επιτευχθούν περισσότερα πράγματα με τους ήδη διαθέσιμους ηλεκτρονικούς υπολογιστές.

Η εικόνα 7-2 παρουσιάζει μια γραφική απεικόνιση ενός λειτουργικού συστήματος και των εφαρμογών του μέσα σε ένα φυσικό μηχάνημα, σε αντιπαραβολή με το ίδιο λειτουργικό σύστημα και τις εφαρμογές του μέσα σε ένα ιδεατό μηχάνημα.



Εικόνα 7-2: Μορφή εξυπηρετητή πριν και μετά το Virtualization

Η εικόνα 7-3 απεικονίζει την παρουσία πολλαπλών ιδεατών συστημάτων (virtual machines) μέσα σε ένα φυσικό σύστημα (physical server). Στη συγκεκριμένη εικόνα, το λογισμικό Virtualization που παρεμβάλλεται ανάμεσα στα ιδεατά μηχανήματα και τον φυσικό εξυπηρετητή είναι το «ESX Server» της εταιρείας VMware.

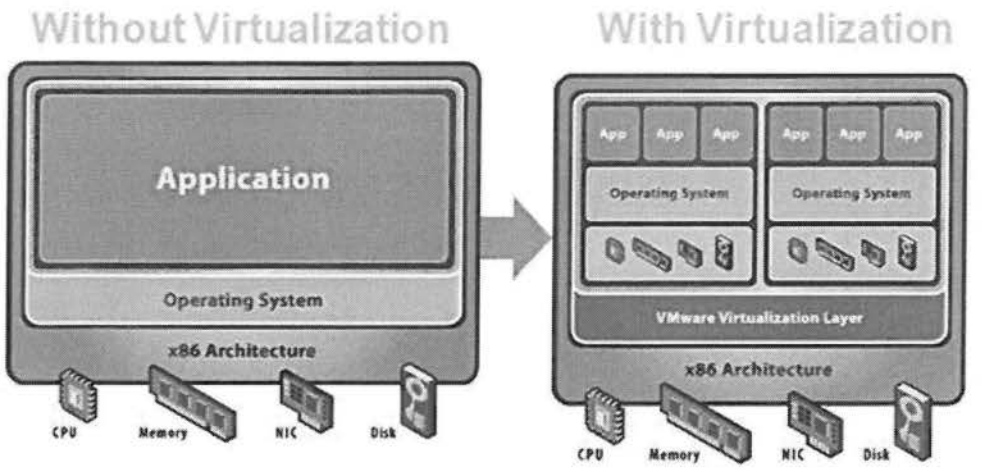


Εικόνα 7-3: Απεικόνιση του Server Virtualization

7.3.2 ΠΩΣ ΛΕΙΤΟΥΡΓΕΙ ΤΟ SERVER VIRTUALIZATION;

Στην ουσία, το Virtualization μας επιτρέπει να μετατρέψουμε το hardware σε software. Μπορούμε να χρησιμοποιήσουμε λογισμικό, όπως το VMware ESX Server, για να μετατρέψουμε ή αλλιώς να κάνουμε «virtualize» τους φυσικούς πόρους ενός x86-based υπολογιστή, συμπεριλαμβανομένων των ΚΜΕ (CPU), Μνήμη (RAM), σκληρό δίσκο (hard disk) και ελεγκτή δικτύου (network controller), προκειμένου να δημιουργήσουμε ένα πλήρως λειτουργικό ιδεατό μηχάνημα (virtual machine) που μπορεί να «τρέχει» το δικό του λειτουργικό σύστημα και τις δικές του εφαρμογές ακριβώς όπως ένας «πραγματικός» υπολογιστής.

Πολλαπλά virtual machines μπορούν να μοιράζονται τους φυσικούς πόρους χωρίς να επηρεάζουν το ένα το άλλο έτσι ώστε να μπορούμε με ασφάλεια να τρέξουμε πολλαπλά λειτουργικά συστήματα και εφαρμογές παράλληλα σε έναν υπολογιστή, μοιράζοντάς τον ουσιαστικά σε πολλούς ιδεατούς υπολογιστές (virtual machines), όπως φαίνεται και στην εικόνα 7-4.



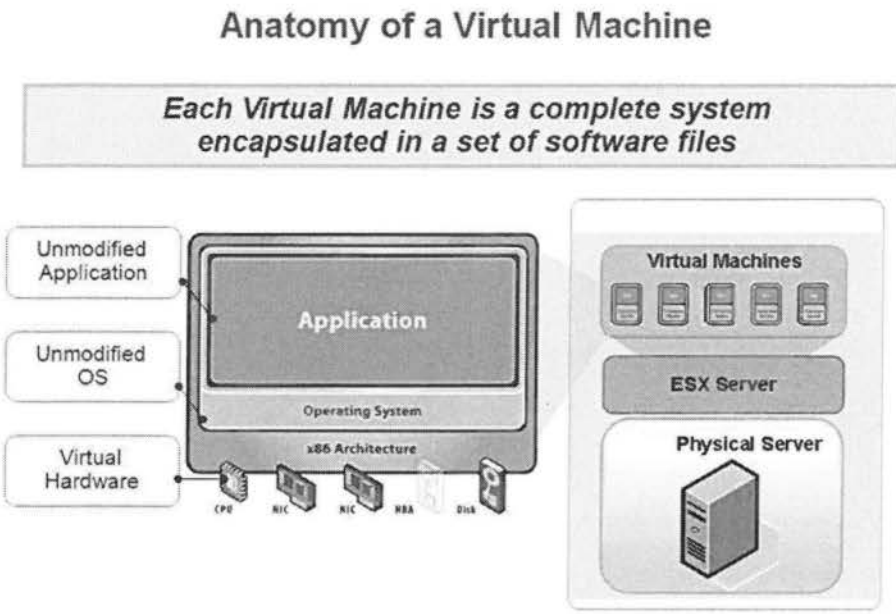
Εικόνα 7-4: Απεικόνιση λειτουργικού συστήματος μέσα σε φυσικό και μέσα σε ιδεατό εξυπηρετητή

7.4 ΟΡΙΣΜΟΣ VIRTUAL MACHINE

7.4.1 ΤΙ ΕΙΝΑΙ ΕΝΑ VIRTUAL MACHINE;

Ένα Ιδεατό Μηχάνημα (Virtual Machine) είναι όπως ένα Φυσικό Μηχάνημα (Physical Machine), αλλά αντί για ηλεκτρονικά στοιχεία, αποτελείται από ένα σύνολο αρχείων λογισμικού. Κάθε virtual machine αντιπροσωπεύει ένα ολοκληρωμένο σύστημα με επεξεργαστές, μνήμη, υποδομή για δικτυακή επικοινωνία, αποθηκευτικό χώρο, και BIOS, όπως δείχνει και η εικόνα 7-5. Ένα ιδεατό μηχάνημα τρέχει ένα ξεχωριστό λειτουργικό σύστημα και αντίστοιχες εφαρμογές, χωρίς καμία τροποποίηση, όπως ένα φυσικός εξυπηρετητής (physical server).

Η διαδικασία διάθεσης/δημιουργίας ενός νέου server (server provisioning) είναι παρόμοια με την αντιγραφή ενός αρχείου. Το server migration γίνεται παρόμοιο με το data migration, δηλαδή οι τεχνικές διαχείρισης δεδομένων μπορούν να χρησιμοποιηθούν για την διαχείριση του server.



Εικόνα 7-5: Ανατομία ιδεατού μηχανήματος

Το «Ιδεατό Μηχάνημα» («Virtual Machine» ή «VM») είναι ένα περιβάλλον ή λειτουργικό σύστημα, που δεν είναι φυσικά υπαρκτό αλλά δημιουργείται μέσα σε ένα άλλο περιβάλλον. Στα πλαίσια αυτά, ένα VM ονομάζεται «guest» ενώ το περιβάλλον μέσα στο οποίο εκτελείται λέγεται «host». Τα virtual machines δημιουργούνται συνήθως για να εκτελέσουν ένα σύνολο εντολών (instruction set) διαφορετικό από αυτό του περιβάλλοντος μέσα στο οποίο φιλοξενούνται (host). Ένα host περιβάλλον μπορεί συνήθως να εκτελεί πολλά virtual machines ταυτόχρονα. Καθώς τα VMs διαχωρίζονται από τους φυσικούς πόρους που χρησιμοποιούν, το host περιβάλλον έχει συχνά τη δυνατότητα να αναθέτει δυναμικά αυτούς τους πόρους ανάμεσα στα VMs.

Η φράση «Virtual Machine» χρησιμοποιείται συχνά για να αναφερθούμε στο Java runtime περιβάλλον της Sun Microsystems, το Java virtual machine (JVM), μέσα στο οποίο μεταφράζονται Java εντολές. Το JVM είναι ένα virtual machine στα πλαίσια του ότι εκτελεί κώδικα που έχει συνταχθεί ειδικά για αυτό, γνωστός ως bytecode, και για την εκτέλεση αποσπά μέρος των πόρων για αυτό τον κώδικα. Η γλώσσα προγραμματισμού Java δεν βασίζεται σε σύνολα οδηγιών εξειδικευμένα για κάθε πλατφόρμα (platform-specific instruction sets), όπως APIs εξειδικευμένα για κάποιο λειτουργικό σύστημα, για να παρουσιάσει κάποιο αποτέλεσμα ή για να έχει πρόσβαση σε πόρους όπως αρχεία. Αντίθετα, το JVM δημιουργεί ιδεατούς πόρους με την bytecode πρόσβαση. Αυτές οι ενέργειες μεταφέρονται στη συνέχεια στους πραγματικούς πόρους του συστήματος, για περαιτέρω επεξεργασία.

Ένας χρήστης που αλληλεπιδρά με έναν ιδεατό εξυπηρετητή μπορεί να δει τον εξυπηρετητή σαν ένα φυσικό μηχάνημα υπό την έννοια ότι ο χρήστης έχει πρόσβαση στους πόρους του μηχανήματος όπως στον σκληρό δίσκο, την μνήμη, τον επεξεργαστή και τις δικτυακές συνδέσεις. Στην πραγματικότητα, όλοι αυτοί οι πόροι του εξυπηρετητή είναι ιδεατοί. Για παράδειγμα, αντί να προσπελαύνει έναν πραγματικό σκληρό δίσκο, ο χρήστης

προσπελαύνει μία δομή του host περιβάλλοντος. Αυτή η δομή προσπελαύνει στη συνέχεια το πραγματικό δίσκο για να καταγράψει τα δεδομένα.

Το Ιδεατό Μηχάνημα (Virtual Machine) (Εικόνα 7-6) είναι ένα ισχυρά απομονωμένο πακέτο λογισμικού που μπορεί να τρέχει το δικό του λειτουργικό σύστημα και τις δικές του εφαρμογές σαν να ήταν ένας φυσικός υπολογιστής. Ένα ιδεατό μηχάνημα λειτουργεί ακριβώς όπως ένας φυσικός υπολογιστής και έχει τη δική του ιδεατή Κ.Μ.Ε (CPU), μνήμη (RAM), σκληρό δίσκο (hard disk) και κάρτα δικτύου (NIC) τα οποία είναι βασισμένα σε λογισμικό.

Ένα λειτουργικό σύστημα δεν μπορεί να αντιληφθεί τη διαφορά ανάμεσα σε ένα ιδεατό και ένα φυσικό μηχάνημα, όπως επίσης δεν μπορούν να αντιληφθούν τη διαφορά ούτε οι εφαρμογές, ούτε οι άλλοι υπολογιστές σε ένα δίκτυο. Ακόμα και το ιδεατό μηχάνημα νομίζει ότι είναι ένας «πραγματικός» υπολογιστής. Παρόλα αυτά, ένα ιδεατό μηχάνημα αποτελείται εξολοκλήρου από software και δεν περιέχει υλικά μέρη/τμήματα. Αυτό έχει σαν αποτέλεσμα, τα ιδεατά μηχανήματα να προσφέρουν μία σειρά από χαρακτηριστικά πλεονεκτήματα έναντι των φυσικών συστημάτων.

Ένα λειτουργικό σύστημα δεν μπορεί να αναγνωρίσει τη διαφορά ανάμεσα σε ένα ιδεατό μηχάνημα και σε ένα φυσικό μηχάνημα, ούτε μπορούν να αναγνωρίσουν την διαφορά οι εφαρμογές ή οι άλλοι υπολογιστές σε ένα δίκτυο. Ακόμα και το virtual machine νομίζει ότι είναι ένας «πραγματικός» υπολογιστής. Ωστόσο, ένα ιδεατό μηχάνημα αποτελείται εξολοκλήρου από λογισμικό και δεν περιέχει κανένα υλικό στοιχείο. Σαν αποτέλεσμα, τα virtual machines προσφέρουν μια σειρά από σαφή πλεονεκτήματα έναντι του φυσικού hardware.

A VMware virtual machine

Εικόνα 7-6: Ιδεατό μηχάνημα

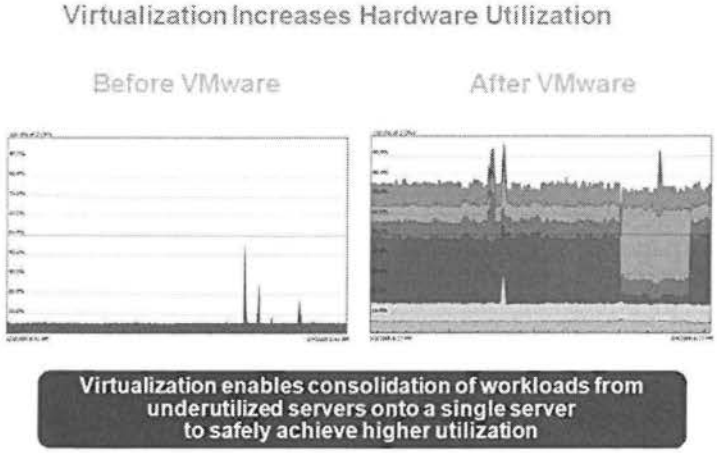
7.5 ΠΛΕΟΝΕΚΤΗΜΑΤΑ ΤΟΥ SERVER VIRTUALIZATION

Η τεχνολογία Server Virtualization παρουσιάζει μια σειρά από πλεονεκτήματα που μπορούν να ωφελήσουν σημαντικά ένα τμήμα IT και συνολικότερα έναν οργανισμό. Ακολούθως παρατίθεται μια λίστα από αντιπροσωπευτικούς λόγους για τους οποίους μπορούμε να αξιοποιήσουμε το Server Virtualization:

- Οι Ιδεατές Μηχανές (Virtual Machines) μπορούν να χρησιμοποιηθούν για την συνένωση του φορτίου εργασίας πολλαπλών υποαπασχολούμενων εξυπηρετητών σε λιγότερα μηχανήματα, ίσως και σε ένα μηχάνημα (server consolidation). Τα σχετιζόμενα οφέλη είναι η εξοικονόμηση σε υλικό εξοπλισμό (hardware), σε περιβαλλοντικό κόστος, σε διοίκηση και διαχείριση της υποδομής εξυπηρετητών.
- Η ανάγκη για εκτέλεση παλαιότερων εφαρμογών εξυπηρετείται απόλυτα από τη χρήση ιδεατών μηχανών. Μια παλαιότερη εφαρμογή ίσως να μην μπορεί να εκτελεστεί σε νεώτερο υλικό και/ή λειτουργικό σύστημα. Ακόμα και αν μπορεί να τρέξει εκεί, πιθανόν να υποαπασχολεί τον εξυπηρετητή, έτσι ώστε να έχει νόημα, όπως αναφέρθηκε στο προηγούμενο σημείο, η συστέγαση πολλαπλών εφαρμογών. Αυτό ίσως να μην είναι εφικτό χωρίς τη χρήση του Virtualization καθώς τέτοιες εφαρμογές συνήθως δεν έχουν γραφεί για να συνυπάρχουν μέσα σε ένα περιβάλλον εκτέλεσης (για παράδειγμα εφαρμογές με hard-coded System V IPC keys).
- Τα virtual machines μπορούν να χρησιμοποιηθούν για να παρέχουν ασφαλή, απομονωμένα περιβάλλοντα για την λειτουργία εφαρμογών που δεν μπορούμε να εμπιστευτούμε. Το Virtualization αποτελεί ένα σημαντικό πλαίσιο για την δημιουργία ασφαλών υπολογιστικών πλατφορμών.
- Τα virtual machines μπορούν να χρησιμοποιηθούν για την δημιουργία λειτουργικών συστημάτων ή περιβάλλοντος εκτέλεσης με περιορισμούς στην χρήση των φυσικών πόρων, εφόσον υπάρχουν οι κατάλληλοι δρομολογητές, και με εγγυημένη πρόσβαση στους πόρους. Ο διαμερισμός (partitioning) συνοδεύεται συνήθως από ποιότητα υπηρεσιών για τη δημιουργία λειτουργικών συστημάτων με εγγυημένη ποιότητα υπηρεσίας.
- Τα ιδεατά ì μηχανήματα μπορούν να παρέχουν την ψευδαίσθηση ότι υπάρχει διαθέσιμο υλικό, όπως συσκευές SCSI, πολλαπλοί επεξεργαστές κτλ., που δεν υπάρχει στην πραγματικότητα. Το Virtualization μπορεί επίσης να χρησιμοποιηθεί για την εξομοίωση δικτύων με ανεξάρτητους υπολογιστές.
- Οι ιδεατές μηχανές μπορεί να χρησιμοποιηθούν για την εκτέλεση πολλαπλών λειτουργικών συστημάτων ταυτόχρονα: διαφορετικές εκδόσεις ή ακόμη και εντελώς διαφορετικά συστήματα, που μπορεί να είναι σε αναμονή. Κάποια τέτοια συστήματα ίσως να είναι δύσκολο ή αδύνατον να τρέξουν σε νέο πραγματικό hardware.
- Τα virtual machines επιτρέπουν με μεγάλη ευκολία το debugging και το performance monitoring. Για παράδειγμα, τέτοια εργαλεία μπορούν να τοποθετηθούν στο virtual machine monitor. Η εξουδετέρωση σφαλμάτων (debugging) μπορεί να γίνει σε λειτουργικά συστήματα χωρίς να διακοπεί η παραγωγική τους λειτουργία, η ακόμα μπορούν να δημιουργηθούν πιο σύνθετα σενάρια εξουδετέρωσης σφαλμάτων.
- Τα virtual machines εκτελούνται απομονωμένα με αποτέλεσμα να μην επηρεάζουν το περιβάλλον τους (άλλες ιδεατές μηχανές που εκτελούνται ταυτόχρονα στο ίδιο φυσικό σύστημα), οποιαδήποτε σφάλματα και αν εμφανίζονται μέσα στο κάθε virtual machine. Δύναται να δημιουργούμε λάθη σκόπιμα στο λογισμικό μια ιδεατής μηχανής για να μελετήσουμε τις επακόλουθες συνέπειες.

- Τα εικονικά μηχανήματα διευκολύνουν την μετακίνηση του λογισμικού από ένα υλικό σε ένα άλλο, πιθανότατα νεώτερης γενιάς, ευνοώντας έτσι την κινητικότητα των εφαρμογών και των συστημάτων.
- Μπορεί κανείς να μεταχειριστεί τις σουίτες εφαρμογών ως συσκευές (appliances) με το να τις αποθηκεύει σε «πακέτα» και να τις τρέχει μέσα σε ένα virtual machine.
- Οι εικονικές μηχανές αποτελούν θαυμάσια εργαλεία για έρευνα και πειράματα σε ακαδημαϊκό επίπεδο. Είναι ιδιαίτερα ασφαλές να δουλεύεις με αυτά καθώς παρέχουν απομόνωση. Ενθυλακώνουν ολόκληρη την κατάσταση ενός υπό εκτέλεση συστήματος: μπορεί κανείς να αποθηκεύσει την κατάστασή του, να την εξετάσει, να την τροποποιήσει, να την επαναφορτώσει, και ούτω καθ' εξής.
- Το Virtualization μπορεί να επιτρέψει σε υπάρχοντα λειτουργικά συστήματα να εκτελεστούν σε πολυεπεξεργαστές διαμοιραζόμενης μνήμης (shared memory multiprocessors).
- Οι ιδεατές μηχανές μπορούν να χρησιμοποιηθούν για τη δημιουργία αυθαίρετων/τυχαίων σεναρίων δοκιμών (test scenarios), και μπορούν να οδηγήσουν σε εντυπωσιακή και πολύ αποτελεσματική εγγύηση ποιότητας.
- Το Virtualization μπορεί να κάνει ευκολότερες και καλύτερα διαχειρίσιμες, λειτουργίες όπως η αλλαγή συστημάτων (system migration), η δημιουργία αντιγράφων ασφαλείας (backup), και η ανάκαμψη από σφάλματα (recovery).
- Το Virtualization μπορεί να είναι ένας αποτελεσματικό τρόπος για binary compatibility (συμβατότητα). Υπάρχουν πολλοί άλλοι λόγοι για τους οποίους θα επέλεγε κανείς να χρησιμοποιήσει τεχνολογία Virtualization.

Η Εικόνα 7-7 δείχνει την αύξηση στην αξιοποίηση των φυσικών συστημάτων με τη χρήση του Virtualization.



Εικόνα 7-7: Σύγκριση βαθμού αξιοποίησης Η/Υ με και χωρίς Virtualization

7.5.1 VIRTUALIZATION

1. **Ενοποίηση Εξυπηρετητών (Server Consolidation) & Βελτιστοποίηση Υποδομής (Infrastructure Optimization):**Το Virtualization κάνει εφικτή την σημαντικά υψηλότερη αξιοποίηση πόρων συγκεντρώνοντας πόρους κοινής υποδομής (common infrastructure resources) και καταργώντας το παραδοσιακό μοντέλο "μία εφαρμογή σε έναν εξυπηρετητή".

2. **Μείωση Κόστους Φυσικής Υποδομής**: Με το Virtualization, μπορούμε να μειώσουμε τον αριθμό των servers και το σχετιζόμενο IT hardware στο κέντρο δεδομένων. Αυτό οδηγεί σε μείωση στις απαιτήσεις σε ακίνητη περιουσία (real estate), ισχύ και ψύξη, οδηγώντας σε σημαντικά χαμηλότερο IT κόστος.

3. **Βελτιωμένη Ευελιξία Λειτουργιών & Ανταπόκριση**: Το Virtualization προσφέρει έναν νέο τρόπο διαχείρισης της IT υποδομής και μπορεί να βοηθήσει τους IT διαχειριστές να δαπανούν λιγότερο χρόνο σε επαναλαμβανόμενες διαδικασίες όπως το provisioning (προμήθειες), η παραμετροποίηση, η παρακολούθηση και η συντήρηση.

4. **Αυξημένη Διαθεσιμότητα Εφαρμογών & Βελτιωμένο Business Continuity**: Εξαλείφεται το προγραμματισμένο downtime και γίνεται εφικτή η γρήγορη ανάκαμψη από απρόσμενες διακοπές λειτουργίας με την δυνατότητα για ασφαλές backup και μετακίνηση ολόκληρων virtual environments, χωρίς καμία διακοπή στην υπηρεσία.

5. **Βελτιωμένη Διαχειρισιμότητα και Ασφάλεια των Σταθμών Εργασίας (Desktops)**: Παρέχεται η δυνατότητα για δημιουργία, διαχείριση και παρακολούθηση ασφαλούς περιβάλλοντος με desktops το οποίο οι χρήστες μπορούν να προσπελάσουν τοπικά ή από απομακρυσμένη τοποθεσία, με ή χωρίς δικτυακή σύνδεση, από σχεδόν οποιοδήποτε standard desktop, laptop ή tablet PC.

7.5.2 ΠΛΕΟΝΕΚΤΗΜΑΤΑ ΤΩΝ VIRTUAL MACHINES

Γενικά, τα virtual machines διαθέτουν τέσσερα σημαντικά χαρακτηριστικά που ωφελούν τον χρήστη:

- Απομόνωση: Τα virtual machines είναι απομονωμένα το ένα από το άλλο αν ήταν φυσικά διαχωρισμένα
- Ενθυλάκωση: Τα virtual machines ενθυλακώνουν ένα ολόκληρο υπολογιστικό περιβάλλον
- Συμβατότητα: Τα virtual machines είναι συμβατά με όλους τους προτυποποιημένους (standard) x86 υπολογιστές
- Ανεξαρτησία από το Hardware: Τα virtual machines τρέχουν ανεξάρτητα από το υποκείμενο hardware

7.6 ΛΟΓΙΣΜΙΚΟ SERVER VIRTUALIZATION

Μεγάλοι κατασκευαστές αναπτύσσουν λογισμικό Virtualization. Μερικοί παρατίθενται στην ακόλουθη λίστα:

- *VMware (http://www.vmware.com)*
- *Microsoft (http://www.microsoft.com)*
- *Citrix Systems (http://www.citrix.com)*
- *XenSource (http://www.xensource.com) Εξαγοράσθηκε από την Citrix τον Αύγουστο του 2007.*
- *Virtual Iron (http://www.virtualiron.com)*
- *Parallels (http://www.parallels.com) Αναφέρεται και ως SWsoft*
- *Sun Microsystems (http://www.sun.com/)*
- *InnoTek (http://www.virtualbox.org) Αγοράσθηκε από την Sun Microsystems Citrix τον Φεβρουάριο του 2008.*
- *Amazon EC2 (http://aws.amazon.com/ec2)*

Οι πιο σημαντικοί από τους παραπάνω κατασκευαστές στο χώρο του Virtualization, σύμφωνα και με το μερίδιό τους στην αγορά, είναι η Microsoft με το λογισμικό πακέτο «Microsoft Virtual Server» και η VMware με την πλατφόρμα Virtualization «ESX Server». Ωστόσο, παγκόσμιος ηγέτης στο χώρο του x86 Virtualization τα τελευταία 10 χρόνια παραμένει η VMware.

7.6.1 MICROSOFT VIRTUAL SERVER

Ο Microsoft Virtual Server είναι μια λύση Virtualization που επιτρέπει τη δημιουργία ιδεατών μηχανών πάνω σε λειτουργικά συστήματα Windows XP, Windows Vista και Windows Server 2003. Αναπτύχθηκε αρχικά από την Connectix, και εξαγοράσθηκε από την Microsoft πριν τη διάθεσή του στην αγορά. Ο Microsoft Virtual Server παρέχει Virtualization σύμφωνα με το μοντέλο Virtual Machine. Ωστόσο, είναι εφαρμογή που παρέχει Virtualization, και όχι πλατφόρμα (λειτουργικό σύστημα) που παρέχει Virtualization. Χρειάζεται κάποιο λειτουργικό σύστημα για να λειτουργήσει. Το λειτουργικό σύστημα που απαιτείται είναι ένα από αυτά που παρέχει η Microsoft. Το μειονέκτημα είναι ότι ο Microsoft Virtual Server υπόκειται στους περιορισμούς του λειτουργικού συστήματος που τον φιλοξενεί, που στην περίπτωση αυτή είναι ένα λειτουργικό σύστημα γενικής χρήσης. Τα Microsoft Windows δεν έχουν δημιουργηθεί για να παρέχουν αποκλειστικά λειτουργίες Virtualization με αποτέλεσμα ο Microsoft Virtual Server να έχει περιορισμένες δυνατότητες στην διαχείριση πόρων και κατ' επέκταση η απόδοση των Virtual Machines που εξυπηρετεί να μην είναι η καλύτερη δυνατή.

7.6.2 VMWARE ESX SERVER

Η VMware είναι ένας μεγάλος κατασκευαστής λογισμικού Virtualization που δημιουργήθηκε το 1998. Ασχολείται αποκλειστικά με λογισμικό Virtualization και πρόσφατα εξαγοράστηκε από την EMC, έναν από τους μεγαλύτερους κατασκευαστές συστημάτων αποθήκευσης δεδομένων παγκοσμίως. Η VMware εφεύρε το Virtualization για πλατφόρμα x86 στην δεκαετία του 1990 προκειμένου να επιληφθεί της περιορισμένης αξιοποίησης και άλλων θεμάτων, ξεπερνώντας πολλές προκλήσεις σε αυτή της την πορεία. Σήμερα, η VMware έχει σημειώσει τόση επιτυχία που δημιουργεί έντονη τάση για Virtualization σε όλους τους x86 υπολογιστές. Η VMware ανοίγει νέο δρόμο σε αυτή την τεχνική και είναι σήμερα ο αδιαμφισβήτητος παγκόσμιος ηγέτης στην τεχνολογία x86 Virtualization.

7.6.2.1 Η ΠΡΟΣΕΓΓΙΣΗ ΤΗΣ VMWARE ΣΤΟ VIRTUALIZATION

Η προσέγγιση της VMware στο Virtualization εισάγει ένα λεπτό/ελαφρύ στρώμα λογισμικού είτε απευθείας στο ηλεκτρομηχανικό μέρος του υπολογιστή είτε σε ένα λειτουργικό σύστημα που θα φιλοξενήσει τα virtual machines. Αυτό το στρώμα λογισμικού περιέχει έναν επόπτη για τα virtual machines ή αλλιώς "hypervisor", ο οποίος αναθέτει φυσικούς πόρους δυναμικά και με διάφανο τρόπο έτσι ώστε πολλαπλά λειτουργικά συστήματα να μπορούν να εκτελούνται ταυτόχρονα σε έναν φυσικό υπολογιστή χωρίς να το γνωρίζουν. Ωστόσο, το Virtualization ενός μεμονωμένου φυσικού υπολογιστή είναι απλά η αρχή. Η VMware προσφέρει μια εύρωστη πλατφόρμα Virtualization που μπορεί να επεκταθεί κατά μήκος εκατοντάδων διασυνδεδεμένων φυσικών υπολογιστών και συστημάτων αποθηκευτικού χώρου για να δημιουργήσουν μια ολόκληρη ιδεατή υποδομή (virtual infrastructure).

7.6.2.2 Η ΛΥΣΗ ΤΗΣ VMWARE: ΠΛΗΡΕΣ VIRTUALIZATION ΤΟΥ X86 HARDWARE

Το 1999, η VMware εισήγαγε το Virtualization για x86 συστήματα σαν ένα μέσο για να αντιμετωπίσει αποδοτικά πολλές από τις προκλήσεις και να μετασχηματίσει τα x86 συστήματα σε γενικής χρήσης, κοινού υλικού εξοπλισμού υποδομή που προσφέρει πλήρη απομόνωση, φορητότητα και επιλογή λειτουργικού συστήματος για περιβάλλοντα εφαρμογών.

Η πρώτη γενιά VMware Virtualization παρείχε διαμερισμό εξυπηρετητών (server partitioning) μέσω ενός hypervisor ή αλλιώς hosted αρχιτεκτονικής. Η δεύτερη γενιά VMware Virtualization προσέθεσε διαχειρισιμότητα, capacity planning, P2V (Physical to Virtual) και άλλα εργαλεία για την συγχώνευση (consolidation) των παραγωγικών servers.

Η τρίτη γενιά φέρνει μια αλματώδη πρόοδο στο Virtualization διαθέτοντας δυνατότητες υποδομής συστημάτων για ολόκληρες φάρμες από ετερογενείς industry standard servers και storage, ανεξάρτητα από το υποκείμενο hardware ή τις εφαρμογές (application/OS). Η γενιά αυτή κάνει δυνατή τη δυναμική συγκέντρωση ετερογενών συστημάτων σε σύνολα πόρων που μπορεί κανείς να διαχειριστεί κεντρικά, τα οποία βελτιστοποιούνται συνεχώς και παρέχουν υψηλή διαθεσιμότητα σε οποιαδήποτε εφαρμογή ή λειτουργικό σύστημα.

Τα τελευταία χρόνια, το Virtualization έχει περάσει από μια τεχνολογία έρευνας και ανάπτυξης (Test/Dev), σε μια τεχνολογία συγχώνευσης παραγωγικών servers (production server consolidation), και τώρα συγκεντρώνει μαζί της ένα ρεύμα που τείνει να την μετατρέψει σε πρότυπο βιομηχανίας για την πληροφορική (industry standard way of computing). Όσοι υιοθέτησαν από νωρίς την τεχνολογία Virtualization χρησιμοποιούσαν τον hypervisor για βασικό partitioning. Καθώς η τεχνολογία ωρίμασε και παρείχε τα μέσα για συγκέντρωση πολλαπλών «virtualized» κόμβων και για κεντρική διαχείριση, οι χρήστες πέρασαν την τεχνολογία αυτή σε παραγωγικά περιβάλλοντα.

Καθώς η αγορά και η τεχνολογία ωρίμασε ακόμα περισσότερο, το Virtualization άρχισε να πηγαίνει πιο μακριά από την αρχική του χρήση για server consolidation και live migration των virtual machines. Διασφαλίζοντας την διαθεσιμότητα και την συνεχή λειτουργία βοήθησε τις εταιρείες να επιτύχουν καλύτερα επίπεδα υπηρεσιών (service levels). Χρησιμοποιώντας το Virtualization για business continuity και DR, οι εταιρείες κατάφεραν να πετύχουν καλύτερα RTOs και RPOs σε κλάσμα του κόστους. Οι δυνατότητες διαχείρισης και αυτοματοποίησης που παρέχει το Virtualization οδήγησαν τις εταιρείες στον κάνουν το Virtualization προεπιλογή (default) στα datacenter.

Μπορεί κανείς να σκεφθεί την πλατφόρμα VMware Virtualization απλά σαν ένα πολλαπλασιαστή πόρων. Κάνει διαθέσιμα τα στοιχεία ενός x86 server σε πολλούς εξυπηρετητές εφαρμογών, αντί σε μόνο έναν που είναι η συνήθης περίπτωση. Η πλατφόρμα Virtualization επιτρέπει να χωρίσουμε αυτούς τους πόρους με ακρίβεια και με έξυπνο τρόπο ανάμεσα σε αυτούς τους εξυπηρετητές για να εξασφαλίσουμε ότι διατηρούνται πάντα εγγυημένα επίπεδα υπηρεσιών.

Η Virtualization πλατφόρμα χειρίζεται την εκτέλεση των επεξεργαστικών αναγκών των virtual machine αναθέτοντας κάθε virtual machine σε διαθέσιμους επεξεργαστές του host συστήματος χρησιμοποιώντας intelligent process scheduling και load balancing (ισομερή κατανομή φορτίου) κατά μήκος όλων των διαθέσιμων επεξεργαστών. Για κάθε virtual machine, μπορεί να ορισθεί ελάχιστο και μέγιστο ποσό της CPU που μπορεί ένα virtual machine να χρησιμοποιήσει, εγγυώντας ένα ποσοστό του πόρου CPU, ανεξαρτήτως ανταγωνισμού. Υπάρχει επίσης η δυνατότητα να ανατεθούν CPU shares για να καθορισθεί η σχετική προτεραιότητα μεταξύ των virtual machines.

Η πλατφόρμα Virtualization μεγιστοποιεί την χρησιμοποίηση του επεξεργαστή (processor utilization) στους εξυπηρετητές επιτρέποντας να τρέχουν συνήθως 8 virtual machines ανά επεξεργαστή, και αξιοποιεί το hyperthreading και τις δυνατότητες των dual-core ή quad-core επεξεργαστών. Συνήθως οι εταιρείες τρέχουν 20 με 30 παράλληλα virtual machines σε 4-way servers. Ο VMware ESX Server υποστηρίζει μέχρι και 80 virtual CPUs σε παράλληλη χρήση.

Επίσης, υπάρχει η δυνατότητα για υπερκατανάλωση της μνήμης στους servers. Αυτό σημαίνει ότι το άθροισμα της μνήμης που έχει ανατεθεί σε όλα τα VMs που βρίσκονται σε λειτουργία μπορεί να ξεπερνά την φυσική μνήμη που είναι εγκατεστημένη στον host κατά ποσοστό περίπου 2:1. Αυτό επιτρέπει να κερδίσουμε περισσότερα από την επένδυσή μας σε ακριβή μνήμη για servers.

Κατά την παραμετροποίηση ενός virtual machine, μπορεί να καθορισθεί το ποσό μνήμης του host που μπορεί να χρησιμοποιεί, μέχρι το μέγιστο των 128 GB. Επιπλέον, μπορούν να καθορισθούν memory allocations σε κλάσματα της διαθέσιμης μνήμης του host χρησιμοποιώντας μια μέθοδο ίσου καταμερισμού (fair share). Επίσης, δύναται να ορισθεί ένα ελάχιστο ποσό μνήμης που ένα virtual machine έχει πάντα διαθέσιμο, έτσι ώστε τα απασχολημένα virtual machines έχουν δυνατότητα να δανείζονται μνήμη από τα virtual machines που βρίσκονται σε κατάσταση ηρεμίας.

Η πλατφόρμα Virtualization επιτρέπει την κοινή χρήση των ακριβών στοιχείων των storage networks από πολλούς εξυπηρετητές, ενώ διατηρείται η προστασία από/ανοχή σε αστοχίες υλικού. Αντί να αφιερώνονται δύο οπτικές κάρτες και ένα μέρος του storage switch σε κάθε server όπως απαιτείται στους συμβατικούς servers, η πλατφόρμα Virtualization μοιράζει αυτούς τους HBAs και τα storage switches σε πολλά virtual machines, διατηρώντας το την προστασία από αστοχίες (fault tolerance) ενώ παράλληλα μειώνοντας το κόστος της πρόσβασης στο storage ανά server. Υπάρχει η δυνατότητα να ορισθεί το ακριβές τμήμα από storage I/O bandwidth που διατίθεται σε κάθε virtual machine.

Οι δικτυακοί πόροι πολλαπλασιάζονται με παρόμοιο τρόπο. Αν έχουμε επενδύσει σε ομάδες καρτών δικτύου (teamed NICs) για ανοχή σε αστοχίες υλικού (hardware fault tolerance), κάθε virtual machine που τρέχει σε αυτό τον host μπορεί να μοιράζεται αυτά τα οφέλη υψηλής διαθεσιμότητας. Τα virtual machines μπορούν να παραμετροποιηθούν με ιδεατές NICs και ιδεατά switches και μπορούμε να διαλέξουμε ποιές NICs του host ή teams χρησιμοποιούνται από τα virtual machines. Η πλατφόρμα Virtualization επιτρέπει να ορίσουμε ακριβώς το δικτυακό bandwidth που διατίθεται στο κάθε virtual machine και παρέχει network traffic shaping για να προσδιορίσουμε το average και peak bandwidth και το maximum burst size.

7.6.2.3 VMWARE INFRASTRUCTURE DISTRIBUTED SERVICES

Τα VMware VMotion, VMware Storage VMotion, VMware DRS, and VMware HA είναι κατανεμημένες υπηρεσίες που επιτρέπουν την αποδοτική και αυτόματη διαχείριση πόρων και την υψηλή διαθεσιμότητα των virtual machines. Είναι λύση που παρέχει η εταιρία VMware ενώ αντίστοιχες λύσεις παρέχουν και οι υπόλοιπες εταιρίες προϊόντων Virtualization.

7.6.2.4 ΙΔΕΑΤΗ ΜΕΤΑΦΟΡΑ (VMWARE VMOTION)

Τα virtual machines τρέχουν σε έναν ESX Server και καταναλώνουν πόρους από αυτόν. Το VMotion επιτρέπει την μετακίνηση (migration) των virtual machines που βρίσκονται σε λειτουργία, από έναν φυσικό server σε έναν άλλο χωρίς διακοπή υπηρεσιών. Αυτό επιτρέπει να μετακινούμε virtual machines από έναν ισχυρά φορτωμένο server σε έναν λιγότερο φορτωμένο. Το αποτέλεσμα είναι η πιο αποδοτική ανάθεση πόρων. Με το VMotion, οι πόροι μπορεί να επανατεθούν δυναμικά στα virtual machines κατά μήκος των φυσικών servers. Το Storage VMotion επιτρέπει την μετακίνηση των virtual machines από ένα Datastore σε ένα άλλο χωρίς τη διακοπή υπηρεσιών. Αυτό επιτρέπει στους storage administrators να μεταφέρουν (off-load) virtual machines από ένα storage array σε ένα

άλλο προκειμένου να εκτελέσουν εργασίες συντήρησης, επαναπαραμετροποίησης των LUNs, και αναβάθμισης των VMFS volumes. Οι διαχειριστές μπορούν να βελτιστοποιήσουν το περιβάλλον αποθήκευσης δεδομένων για βελτιωμένη απόδοση και σε με ένα μόνο βήμα ενεργειών να μετακινήσουν virtual machines.

7.6.2.5 ΔΥΝΑΜΙΚΗ ΚΑΤΑΝΟΜΗ ΦΟΡΤΙΟΥ (VMWARE DRS)

Το VMware DRS (Dynamic Resource Scheduling) συνεισφέρει στην δυνατότητα ελέγχου και διαχείρισης των πόρων στο virtual datacenter. Ένα cluster (συστοιχία) μπορεί να θεωρηθεί ως συγκέντρωση των υπολογιστικών πόρων και των πόρων μνήμης των υποκείμενων φυσικών hosts που έχουν τοποθετηθεί όλοι μαζί σε ένα μεμονωμένο σύνολο. Τα virtual machines μπορούν να ανατεθούν σε αυτό το σύνολο. Το DRS παρακολουθεί το workload (φορτίο εργασίας) των εν λειτουργία virtual machines και τη χρησιμοποίηση πόρων (resource utilization) των hosts προκειμένου αυτοί να εκχωρούνται ανάλογα με τη χρήση. Χρησιμοποιώντας το VMotion και έναν έξυπνο δρομολογητή πόρων (resource scheduler), το VMware DRS αυτοματοποιεί την εργασία ανάθεσης των virtual machines σε servers μέσα στο cluster για να χρησιμοποιήσουν τους πόρους της ΚΜΕ και της Μνήμης αυτού του server. Το DRS κάνει τους υπολογισμούς και αυτοματοποιεί την αντιστοίχιση.

Όταν ένας νέος φυσικός εξυπηρετητής γίνεται διαθέσιμος, το DRS αυτόματα ανακατανείμει τα virtual machines χρησιμοποιώντας το VMotion για να εξισορροπήσει τα workloads. Εάν ένας φυσικός server πρέπει να τεθεί εκτός λειτουργίας για οποιονδήποτε λόγο, το DRS αυτόματα αναθέτει τα virtual machines του σε άλλους εξυπηρετητές. Όταν το DPM (Dynamic Power Management) είναι ενεργό, το σύστημα συγκρίνει το cluster- και host-level capacity με τις ανάγκες των virtual machines που τρέχουν στο cluster. Εάν ένας host βρεθεί να έχει αρκετή περίσσεια χωρητικότητας πόρων για να απορροφήσει τα virtual machines ενός άλλου host, τα virtual machines μετακινούνται (γίνονται migrate) και ο μη χρησιμοποιούμενος host τοποθετείται σε κατάσταση αναμονής. Με αυτό τον τρόπο, το DPM βελτιστοποιεί την κατανάλωση ισχύος του cluster. Το DRS μπορεί να παραμετροποιηθεί ώστε να εκτελεί αυτόματα ενέργειες ισομερούς κατανομής φορτίου και διαχείρισης ενέργειας, ή να παρέχει προτάσεις τις οποίες ο διαχειριστής του datacenter μπορεί να αξιολογήσει και να ενεργήσει για κάθε μία ξεχωριστά.

7.6.2.6 ΥΨΗΛΗ ΔΙΑΘΕΣΙΜΟΤΗΤΑ (VMWARE HA)

Το VMware HA (High Availability) προσφέρει μία απλή και χαμηλού κόστους εναλλακτική λύση υψηλής διαθεσιμότητας από ότι το clustering σε επίπεδο εφαρμογής. Επιτρέπει την γρήγορη επανεκκίνηση των virtual machines σε έναν διαφορετικό φυσικό server μέσα σε ένα cluster αυτόματα εάν ο εξυπηρετητής που τα φιλοξενούσε παρουσίαση αστοχία. Όλες οι εφαρμογές μέσα στα virtual machines απολαμβάνουν το όφελος της υψηλής διαθεσιμότητας και όχι μόνο μία (όπως συμβαίνει στο application clustering).

Το HA παρακολουθεί όλους τους φυσικούς hosts σε ένα cluster και ανιχνεύει αστοχίες των host. Ένας πράκτορας (agent) που τοποθετείται σε κάθε φυσικό host διατηρεί μία στοιχειώδη επικοινωνία (heartbeat) με τους άλλους hosts στο resource pool, και η απώλεια του heartbeat ενεργοποιεί την διαδικασία επανεκκίνησης όλων των επηρεαζόμενων virtual

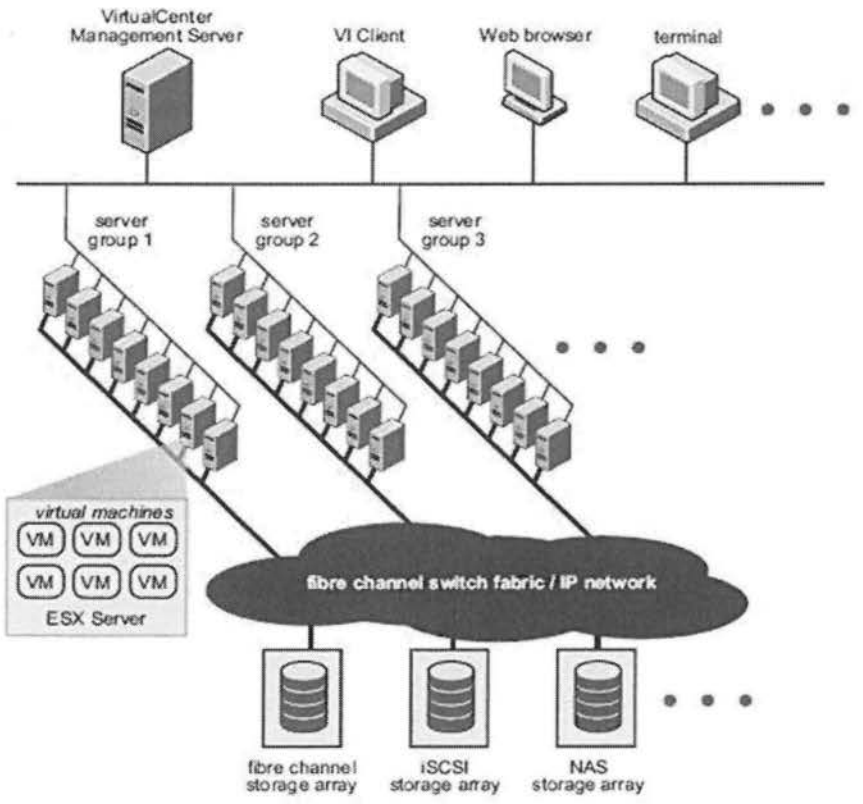
machines σε άλλους hosts. Το HA διασφαλίζει ότι υπάρχουν διαθέσιμοι επαρκείς πόροι στο cluster κάθε χρονική στιγμή για να επανεκκινήσουν τα virtual machines σε διαφορετικούς φυσικούς hosts στην περίπτωση αστοχίας κάποιου/ων hosts.

Το VMware Infrastructure είναι μια πλήρης σουίτα infrastructure Virtualization που παρέχει εκτενείς δυνατότητες Virtualization, διαχείρισης, βελτιστοποίησης πόρων, διαθεσιμότητας εφαρμογών, και αυτοματοποίησης λειτουργιών σε ένα ολοκληρωμένη πακέτο. Το VMware Infrastructure κάνει «virtualize» και συγκεντρώνει τους υποκείμενους φυσικούς πόρους κατά μήκος πολλαπλών συστημάτων και παρέχει σύνολα ιδεατών πόρων στο datacenter σε ένα ιδεατό περιβάλλον.

Επιπροσθέτως, το VMware Infrastructure παρέχει ένα σύνολο κατανεμημένων υπηρεσιών που επιτρέπουν επιλεκτική, καθοδηγούμενη από πολιτικές ανάθεσης πόρων, υψηλή διαθεσιμότητα, και συγκεντρωτικό backup ολόκληρου του virtual datacenter. Αυτές οι κατανεμημένες υπηρεσίες επιτρέπουν σε έναν IT οργανισμό να δημιουργήσει και να τηρεί τα production Service Level Agreements που έχει θέσει με τους πελάτες του, με οικονομικά αποδοτικό τρόπο.

7.6.2.7 ΦΥΣΙΚΗ ΤΟΠΟΛΟΓΙΑ ΕΝΟΣ VMWARE VIRTUAL DATACENTER

Όπως δείχνει η εικόνα 7-8, ένα τυπικό VMware Infrastructure datacenter αποτελείται από φυσικά δομικά στοιχεία όπως x86 computing servers, storage networks και arrays, IP networks, έναν management server, και desktop clients.



Εικόνα 7-8:Φυσική τοπολογία ιδεατού μηχανογραφικού κέντρου

7.6.2.7.1 COMPUTING SERVERS

Οι computing servers είναι industry standard x86 servers που τρέχουν VMware ESX Server απευθείας πάνω στο υλικό. Το ESX Server λογισμικό διαθέτει πόρους στα virtual machines και τα «τρέχει». Κάθε computing server αναφέρεται ως ένας standalone host στο virtual περιβάλλον. Ένας αριθμός από παρόμοια παραμετροποιημένους x86 servers μπορεί να ομαδοποιηθεί με συνδέσεις στο ίδιο δίκτυο και στα ίδια υποσυστήματα χωρητικότητας για να παρέχει ένα συγκεντρωμένο σύνολο πόρων στο virtual περιβάλλον, και ονομάζεται cluster.

7.6.2.7.2 STORAGE NETWORKS ΚΑΙ ARRAYS

Τα Fiber Channel SAN arrays, τα iSCSI SAN arrays, και τα NAS arrays είναι ευρέως χρησιμοποιούμενες τεχνολογίες storage που υποστηρίζονται από το VMware Infrastructure για να ικανοποιεί διαφορετικές ανάγκες των datacenters σε storage. Μοιράζοντας τα storage arrays ανάμεσα σε ομάδες από servers μέσω των storage area networks, επιτρέπεται η συγκέντρωση πόρων χωρητικότητας και παρέχεται μεγαλύτερη ευελιξία στη διάθεσή τους στα virtual machines.

7.6.2.7.3 IP NETWORKS

Κάθε computing server μπορεί να έχει πολλαπλές κάρτες δικτύου (NICs) για να παρέχει υψηλή χωρητικότητα και αξιόπιστη δικτύωση σε ολόκληρο το datacenter.

7.6.2.7.4 VIRTUALCENTER SERVER

Η εφαρμογή VirtualCenter Server της εταιρείας VMware παρέχει ένα βολικό κεντρικό σημείο ελέγχου του datacenter. Παρέχει πολλές βασικές υπηρεσίες για το datacenter όπως έλεγχο πρόσβασης, performance monitoring, και παραμετροποίηση. Ενοποιεί τους πόρους από τους μεμονωμένους computing servers ώστε να διαμοιραστούν μεταξύ των virtual machines σε ολόκληρο το datacenter. Το επιτυγχάνει αυτό διαχειρίζοντας την ανάθεση των virtual machines στους computing servers και την ανάθεση των πόρων στα virtual machines μέσα σε έναν δεδομένο computing server βάσει των πολιτικών (policies) που έχουν τεθεί από τον system administrator.

Οι computing servers θα συνεχίσουν να λειτουργούν ακόμα και στην απίθανη περίπτωση που το VirtualCenter Server δεν είναι προσβάσιμο (για παράδειγμα, το δίκτυο έχει αποκοπεί). Μπορούμε να τους διαχειριστούμε ξεχωριστά και θα συνεχίσουν να τρέχουν τα virtual machines που τους έχουν ανατεθεί βάσει της τελευταίας ανάθεσης πόρων. Αφού ο VirtualCenter Server γίνει και πάλι προσβάσιμος, μπορεί να διαχειριστεί datacenter σαν σύνολο πάλι.

7.6.2.7.5 DESKTOP CLIENTS

Το VMware Infrastructure διαθέτει σειρά από interfaces για την διαχείριση του datacenter και την πρόσβαση στα virtual machines. Οι χρήστες μπορούν να επιλέξουν το interface που

καλύπτει καλύτερα τις ανάγκες τους: VMware Infrastructure Client (VI Client), Web Access μέσω ενός Web browser, ή terminal services (όπως Windows Terminal Services).

7.6.2.7.6 ΑΡΧΙΤΕΚΤΟΝΙΚΗ ΔΙΚΤΥΟΥ

Το VMware Infrastructure είναι ί ία από τις υπάρχουσες λύσεις Virtualization που διαθέτει ένα πλούσιο σύνολο από ιδεατά δικτυακά στοιχεία που κάνουν την δικτύωση των virtual machines στο data center τόσο εύκολη και απλή όσο στο φυσικό περιβάλλον. Επιπλέον, διαθέτει ένα σύνολο νέων δυνατοτήτων που δεν είναι διαθέσιμες στο φυσικό περιβάλλον διότι πολλοί από τους περιορισμούς του φυσικού κόσμου δεν ισχύουν.

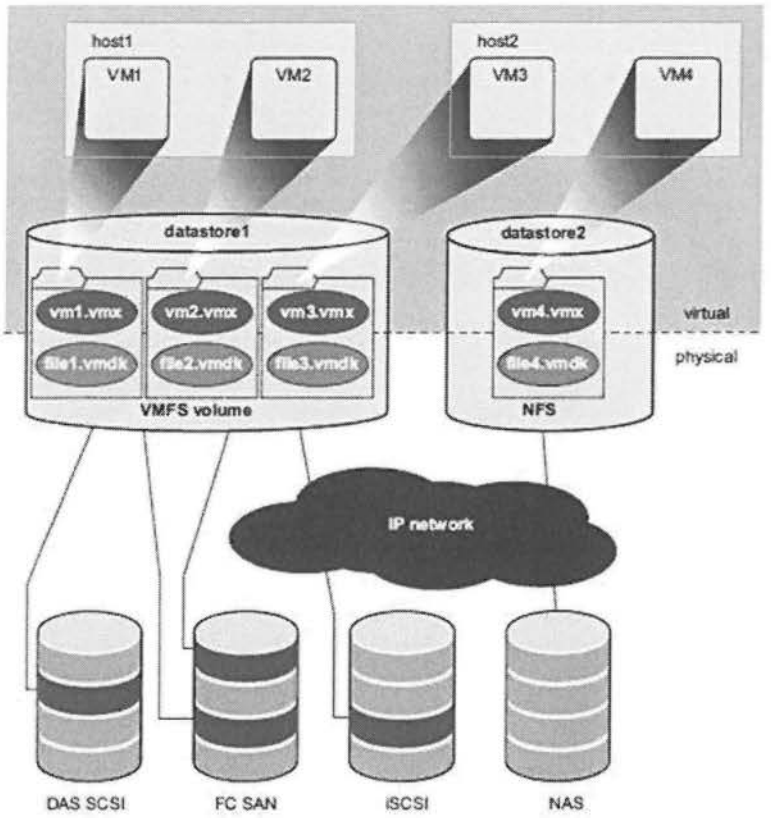
Το virtual περιβάλλον παρέχει παρόμοια δικτυακά στοιχεία με του φυσικού κόσμου. Αυτά είναι οι ιδεατές κάρτες δικτύου (vNIC), τα virtual switches (vSwitch), και τα port groups (ομάδες από πόρτες). Όπως και ένα φυσικό μηχάνημα, ένα ιδεατό μηχάνημα έχει την δική του vNIC. Το λειτουργικό σύστημα και οι εφαρμογές μιλάνε στην vNIC μέσω ενός standard device driver ή ενός VMware βελτιστοποιημένου device driver σαν να ήταν η vNIC μία φυσική NIC.

Στον έξω κόσμο, η vNIC έχει την δική της MAC διεύθυνση και μία ή περισσότερες IP διευθύνσεις, και ανταποκρίνεται στο standard Ethernet πρωτόκολλο ακριβώς όπως θα έκανε μια φυσική NIC. Στην πραγματικότητα, ένας εξωτερικός agent δεν γνωρίζει ότι επικοινωνεί με ένα virtual machine.

Ένα virtual switch λειτουργεί σαν ένα φυσικό switch επιπέδου 2 (ISO/OSI layer 2). Κάθε server έχει τα δικά του virtual switches. Στην μία πλευρά του virtual switch υπάρχουν port groups (ομάδες από πόρτες) που συνδέονται στα virtual machines. Στην άλλη πλευρά υπάρχουν uplink συνδέσεις σε φυσικούς Ethernet προσαρμογείς στον server όπου εδρεύει το virtual switch. Τα virtual machines συνδέονται στον έξω κόσμο μέσω των φυσικών Ethernet adapters που είναι συνδεδεμένοι στα virtual switch uplinks.

7.6.2.7.7 ΑΡΧΙΤΕΚΤΟΝΙΚΗ STORAGE

Η storage αρχιτεκτονική του VMware Infrastructure, που φαίνεται στην εικόνα 7-10, αποτελείται από αφαιρετικά επίπεδα που αποκρύπτουν και διαχειρίζονται την πολυπλοκότητα και τις διαφορές ανάμεσα στα φυσικά storage subsystems.



Εικόνα 7-9: Αρχιτεκτονική αποθηκευτικών μέσων

Για τις εφαρμογές και τα guest operating systems μέσα σε κάθε virtual machine το υποσύστημα χωρητικότητας (storage subsystem) είναι ένας απλός virtual Bus Logic ή LSI SCSI host bus adapter συνδεδεμένος σε έναν ή περισσότερους virtual SCSI disks.

Οι virtual SCSI δίσκοι παρέχονται από Datastore στοιχεία στο datacenter. Ένα *Datastore* είναι σαν ένα storage appliance που παρέχει αποθηκευτικό χώρο για πολλά virtual machines κατά μήκος πολλαπλών φυσικών hosts. Το Datastore παρέχει ένα απλό μοντέλο για την ανάθεση αποθηκευτικού χώρου σε ξεχωριστά virtual machines χωρίς αυτά να εκτίθενται στην πολυπλοκότητα και την ποικιλία των διαθέσιμων τεχνολογιών φυσικού storage, όπως Fibre Channel SAN, iSCSI SAN, direct attached storage (DAS), και NAS.

Ένα virtual machine είναι αποθηκευμένο σε ένα σύνολο αρχείων σε έναν κατάλογο του Datastore. Ένας virtual δίσκος μέσα σε κάθε virtual machine είναι ένα ή περισσότερα αρχεία μέσα στον κατάλογο. Σαν αποτέλεσμα, μπορούμε να χρησιμοποιήσουμε ένα virtual disk (αντιγραφή, μετακίνηση, αντίγραφα ασφαλείας, κτλ.) ακριβώς όπως ένα αρχείο. Νέοι virtual disks μπορούν να προστεθούν ("hot-added") σε ένα virtual machine χωρίς να το κλείσουμε. Σε αυτή την περίπτωση, ένα virtual disk file (.vmdk) δημιουργείται στο VMFS για να παρέχει νέα χωρητικότητα για τον hot-added virtual disk ή ένας ήδη υπάρχον virtual disk file συνδέεται με ένα virtual machine.

Κάθε Datastore είναι από φυσικής πλευράς ένα VMFS volume (ή, για NAS Datastores, ένα NFS volume με VMFS χαρακτηριστικά) πάνω σε μια συσκευή storage. Τα Datastores μπορούν να διατρέχουν/γεφυρώνουν πολλαπλά φυσικά storage subsystems. Ένα μεμονωμένο VMFS volume μπορεί να περιλαμβάνει ένα η περισσότερα LUNs από ένα

τοπικό SCSI disk array σε έναν φυσικό host, μία Fibre Channel SAN φάρμα δίσκων, ή μία φάρμα iSCSI SAN δίσκων. Τα νέα LUNs που προστίθενται σε οποιοδήποτε από τα physical storage subsystems ανιχνεύονται αυτόματα και γίνονται διαθέσιμα σε όλα τα υπάρχοντα ή τα νέα Datastores. Ο αποθηκευτικός χώρος σε ένα προηγούμενα δημιουργημένο VMFS volume (Datastore) μπορεί να επεκταθεί (hot-extended) χωρίς να κλείσουν οι φυσικοί hosts ή τα storage subsystems, προσθέτοντας ένα νέο φυσικό LUN από οποιοδήποτε από τα storage subsystems που είναι ορατά σε αυτό. Αντιστρόφως, εάν ένα από τα LUNs μέσα σε ένα VMFS volume (Datastore) αστοχήσει ή πάψει να είναι διαθέσιμο, μόνο τα virtual machines που αγγίζουν αυτό το LUN επηρεάζονται. Όλα τα υπόλοιπα virtual machines με virtual disks που εδρεύουν σε άλλα LUNs συνεχίζουν να λειτουργούν κανονικά.

Το VMFS είναι ένα clustered file system που χρησιμοποιεί το shared storage για να επιτρέπει σε πολλαπλούς φυσικούς hosts να διαβάζουν και να γράφουν στο ίδιο storage ταυτόχρονα. Το VMFS διαθέτει on-disk locking για να διασφαλίσει ότι το ίδιο machine δεν θα ξεκινήσει από πολλαπλούς servers την ίδια στιγμή. Εάν ένας φυσικός host αστοχήσει, το on-disk lock για κάθε virtual machine αφαιρείται έτσι ώστε τα virtual machines να μπορούν να επανεκκινηθούν σε άλλους φυσικούς hosts.

Το VMFS επίσης χαρακτηρίζεται από enterprise-class crash consistency και μηχανισμούς διόρθωσης, όπως το distributed journaling, ένα crash consistent virtual machine I/O path, και στιγμιότυπα της κατάστασης του συστήματος (machine state snapshots). Αυτοί οι μηχανισμοί μπορεί να βοηθήσουν για quick root-cause και επαναφορά από αστοχίες του virtual machine, του physical host, και του storage subsystem.

8 PFSENSE

Η διανομή pfSense είναι ένα λειτουργικό σύστημα ανοιχτού λογισμικού το οποίο χρησιμοποιείτε για να μετατρέψει έναν Η/Υ σε firewall/router. Η διανομή pfSense είναι μια τροποποιημένη του λειτουργικού συστήματος FreeBSD το οποίο στηρίχτηκε στο m0n0wall project μια ισχυρή αλλά ελαφρύ διανομή firewall. Το pfSense στηρίζεται πάνω στο m0n0wall και επεκτείνει τις δυνατότητες του προσθέτοντας πολλές άλλες δικτυακές υπηρεσίες. Το firewall Που χρησιμοποιεί δεν είναι το iptables που χρησιμοποιούν τα Linux, αλλά στηρίζεται στο PF (Packet Filter) το οποίο μεταφέρθηκε από το OpenBSD στο FreeBSD το 2004.

Το pfSense είναι ένα open source Firewall, πολύ δημοφιλές, αξιόπιστο και με απεριόριστες δυνατότητες, προστατεύοντας χιλιάδες δικτυακές συσκευές. Μπορεί άνετα να τρέξει σε μικρής υπολογιστικής δύναμης "μηχανάκια" με αποτέλεσμα να αποτελεί μία ισχυρή και ευέλικτη πλατφόρμα ασφαλείας καθώς με μηδέν κόστος διαθέτει πλήθος χαρακτηριστικών, βασιζόμενο πάντα σε ένα σύστημα πακέτων / addons.

Το pfsense μπορεί να χρησιμοποιηθεί για τους παρακάτω σκοπούς:

- Firewall
- Router
- Load Balancer
- VPN λύσεις
- Internet filter
- DHCP/DNS server
- Παρακολούθηση χρήσης

Το FreeBSD χρησιμοποιείτε σαν πλατφόρμα από πολλού οργανισμούς όπως η Cisco, Apple, Juniper και NetApp. Το FreeBSD χρησιμοποιείτε ακόμη και σαν webserver για μεγάλους οργανισμούς όπως η Yahoo.

Για να ξεκινήσουμε την εγκατάσταση οι ελάχιστες απαιτήσεις που χρειάζεται το pfsense είναι:

- 100MHz Pentium CPU
- 128MB RAM
- CD-ROM ή Flash USB για εγκατάσταση λογισμικού
- 1GB hard drive

Για ενσωματωμένα συστήματα θα χρειαστούμε 128MB Compact Flash και μια σειριακή πόρτα για console port.

8.1 ΔΥΝΑΤΟΤΗΤΕΣ PFSENSE

Παρακάτω θα εξετάσουμε της δυνατότητες που μας προσφέρει η διανομή pfsense:

- Firewall
  - Φιλτράρισμα με βάση την IP αποστολέα και παραλήπτη, πρωτόκολλο IP, πόρτα προορισμού και αποστολέα για TCP και UDP κίνηση.
  - Ικανό να περιορίσει ταυτόχρονες συνδέσεις βάση κανόνων.
  - Μπορεί να φιλτράρει με βάσει το λειτουργικό σύστημα (pOf – OS fingerprint utility) το οποίο ξεκινά μια σύνδεση. Έτσι μπορούμε να μπλοκάρουμε μηχανές που τρέχουν windows να έχουν πρόσβαση στο Ιντερνετ, ενώ αντίθετα να επιτρέψουμε σε μηχανές Linux/Unix να έχουν πρόσβαση στο Ιντερνετ.
  - Μπορεί να καταγράφει κάθε κίνηση που ταιριάζει με κάθε κανόνα.
  - Εξαιρετική ευέλικτη πολιτική δρομολόγησης διαλέγοντας κάθε φορά την προεπιλεγμένη πύλη με βάση κανόνων (για εξισορρόπηση φορτίου, πολλαπλές WAN γραμμές κτλ)
  - Τα ψευδώνυμα επιτρέπουν την τη ομαδοποίηση και την ονοματοποίηση των IP, δικτύων και πορτών. Αυτό μας βοηθάει να διατηρήσουμε τους κανόνες του firewall ευκολοδιάβαστους και εύκολα κατανοητούς, ειδικά σε περιβάλλοντα που έχουμε πολλές δημόσιες IPs και πολλούς server.

- Το pfSense προσφέρει τρεις λειτουργίες για VPN συνδέσεις:
  - IPsec (το οποίο είναι και default πρωτόκολλο IPv6)
  - OpenVPN
  - PPTP

- Δεσμευμένη Δικτυακή Πύλη (Captive Portal)
  - Η λειτουργία αυτή σου επιτρέπει να εξαναγκάσεις τον έλεγχο ταυτοποίησης ή την ανακατεύθυνση για να έχεις πρόσβαση στο δίκτυο. Αυτό χρησιμοποιείτε κυρίως σε δίκτυα hot spot, αλλά χρησιμοποιείτε και σε δίκτυα επιχειρήσεων για ένα επιπλέον στρώμα ασφαλείας σε φιλοξενούμενα ασύρματα τερματικά για πρόσβαση στο Ιντερνετ.

- Εξισορρόπηση φόρτου
  - Εξερχόμενη Εξισορρόπηση φορτίου η οποία χρησιμοποιείτε με πολλαπλές WAN συνδέσεις για να παρέχουν εξισορρόπηση φόρτου αλλά και failover δυνατότητες. Η κίνηση κατευθύνεται στην επιθυμητή προεπιλεγμένη πύλη με βάση κάποιους κανόνες.
  - Εισερχόμενη εξισορρόπηση φόρτου χρησιμοποιείτε για να διανείμει το φόρτο μεταξύ πολλαπλών server. Χρησιμοποιείτε κυρίως σε web server, mail severs κα.

- Υποβολή εκθέσεων και παρακολούθηση με RRD γραφήματα τα οποία περιέχουν
  - Χρησιμοποίηση της CPU

- Συνολική απόδοση
- Κατάσταση firewall
- Απόδοση για κάθε διασύνδεση
- Τα πακέτα ανά δευτερόλεπτο για όλες τις διασυνδέσεις
- WAN πύλη διεπαφής (-ες) του χρόνου απόκρισης ping

- Πληροφορίες σε πραγματικό χρόνο
  - Οι πληροφορίες είναι σημαντικές αλλά μερικές φορές είναι απαραίτητα να δούμε πληροφορίες σε πραγματικό χρόνο
  - Τα γραφήματα SVG μας δείχνουν πληροφορίες σε πραγματικό χρόνο για την απόδοση κάθε διεπαφής
  - Η πρώτη σελίδα περιέχει AJAX όργανα μέτρησης που μας εμφανίζουν σε πραγματικό χρόνο την χρησιμοποίηση της CPU, μνήμης, swap και δίσκου.

- NAT

Άλλες δυνατότητες του pfsense είναι ο πλεονασμός (redundancy) και η υψηλή διαθεσιμότητα (high availability). Δύο ή περισσότερα firewall μπορούν να χρησιμοποιηθούν και να ρυθμιστούν σαν ομάδα failover. Αν μια διεπαφή αποτύχει στο πρωτεύων firewall ή το πρωτεύων firewall πέσει το δευτερεύων γίνεται ενεργό.

Ένα από τα μεγαλύτερα πλεονεκτήματα που έχει το pfsense από άλλες διανομές firewall είναι η ικανότητά του να διαμορφώσει επακριβώς κυκλοφορία. Πολλοί εμπορικοί router/firewall προσφέρουν QoS (Quality of Service) υπηρεσίες ή οποίες όμως είναι δύσκολα κατανοητά στο στους περισσότερους χρήστες. Το pfsense έχει έναν οδηγό όπου σε βοηθά να αντιμετώπισης τα προβλήματα VOIP που μπορεί να έχεις. Απαντώντας τις ερωτήσεις του οδηγού, το pfsense ρυθμίζει του κανόνες κυκλοφορίας όπου διαιρεί τη κίνηση σε ουρές. Το ίδιο κάνει και για άλλου τύπου κυκλοφορία όπως για παράδειγμα peer-to-peer εφαρμογές.

Το pfsense χρησιμοποιείτε σε κάθε είδους και μέγεθος δικτύου και είναι κατάλληλο για ένα δίκτυο που περιέχει λίγους Η/Υ ή χιλιάδες. Παρακάτω θα δούμε τις ποιο κοινές υλοποιήσεις.

- Firewall περιμέτρου: Είναι η ποιο κοινή υλοποίηση του pfsense όπου μια γραμμή Ιντερνετ συνδέσεις είναι τοποθετημένη στη WAN διεπαφή του και το εσωτερικό δίκτυο μας είναι τοποθετημένο στη μεριά των LAN διεπαφών.
- LAN ή WAN router: Η δεύτερη ποιο κοινή υλοποίηση (αν και για δρομολογητή προτιμούνται άλλες διανομές όπως Vyatta) όπου μπορούμε να χρησιμοποιήσουμε το pfsense για το διαχωρισμό του δικτύου μας σε VLANs με το 802.1Q πρωτόκολλο.
- Wireless Access Point
- Συσκευή VPN: Πολλοί χρησιμοποιούν το pfsense για να προσθέσουν VPN δυνατότητες χωρίς να προκαλέσουν μεταβολές στην είδη υπάρχων δικτυακή τοπολογία.

Το pfSense είναι ένα ισχυρό και σταθερό project με πολύ προχωρημένες δυνατότητες. Έχει σημειωθεί από χρήστες που το χρησιμοποιούνε ότι λειτουργεί άριστα με εκατοντάδες υπολογιστικά συστήματα να λειτουργούν πίσω από αυτό.

9 VYATTA

Το Vyatta είναι το πρώτο εμπορικά υποστηριζόμενο ανοιχτού λογισμικού router, firewall και οι λύσεις VPN που προσφέρει είναι μια πραγματική εναλλακτική λύση από εκείνα του κλειστού κώδικα ιδιόκτητα προϊόντα δρομολόγησης. Το λογισμικό Vyatta και οι συσκευές που το τρέχουν συνδυάζουν τα χαρακτηριστικά, τις επιδόσεις και την αξιοπιστία ενός ακριβού εμπορικού δρομολογητή ενώ παράλληλα προσφέρει εξοικονόμηση κόστους, ευελιξία και την ασφάλεια των ανοιχτών λογισμικών για να παρέχει μια ιδανική λύση για SMB, Υποκαταστήματα, επιχειρήσεων και παρόχους υπηρεσιών.

Το λογισμικό Vyatta μπορεί να εγκατασταθεί σε έναν παλιό Η/Υ και είναι μια ολοκληρωμένη λύση για router/firewall/VPN όπου δεν χρειάζεται καθόλου προγραμματισμός ούτε μεταγλώττιση του κώδικα. Απλά κατεβάζουμε το ISO image από το website www.vyatta.org. Το Vyatta είναι ένα λειτουργικό σύστημα που βασίζεται στη Debian διανομή Linux και παρέχει δικτυακές εφαρμογές όπως το Quagga, OpenVPN και πολλά άλλα.

Το Vyatta προορίζεται σαν αντικατάσταση του Cisco IOS 1800 μέσω του της σειράς ASR1000 και συσκευές ASA 5500 με μεγάλη έμφαση στο κόστος και την ευελιξία ενός ανοιχτού λογισμικού το οποίο βασίζεται στο Linux.

Παρακάτω αναφέρονται μερικές εφαρμογές που χρησιμοποιείτε το λογισμικό Vyatta:

- LAN/WAN δρομολόγηση για μικρές έως μεγάλες επιχειρήσεις
- Firewall για μικρές έως μεγάλες επιχειρήσεις
- Πάροχοι υπηρεσιών: POPs και CPE
- Ασφαλείς site-to-site IPsec VPN
- Εφεδρικά δίκτυα
- Gigabit Ethernet
- Κατάτμηση LAN για δίκτυα που μεγαλώνουν

Μερικές από τις δυνατότητες που περιέχει είναι οι εξής:

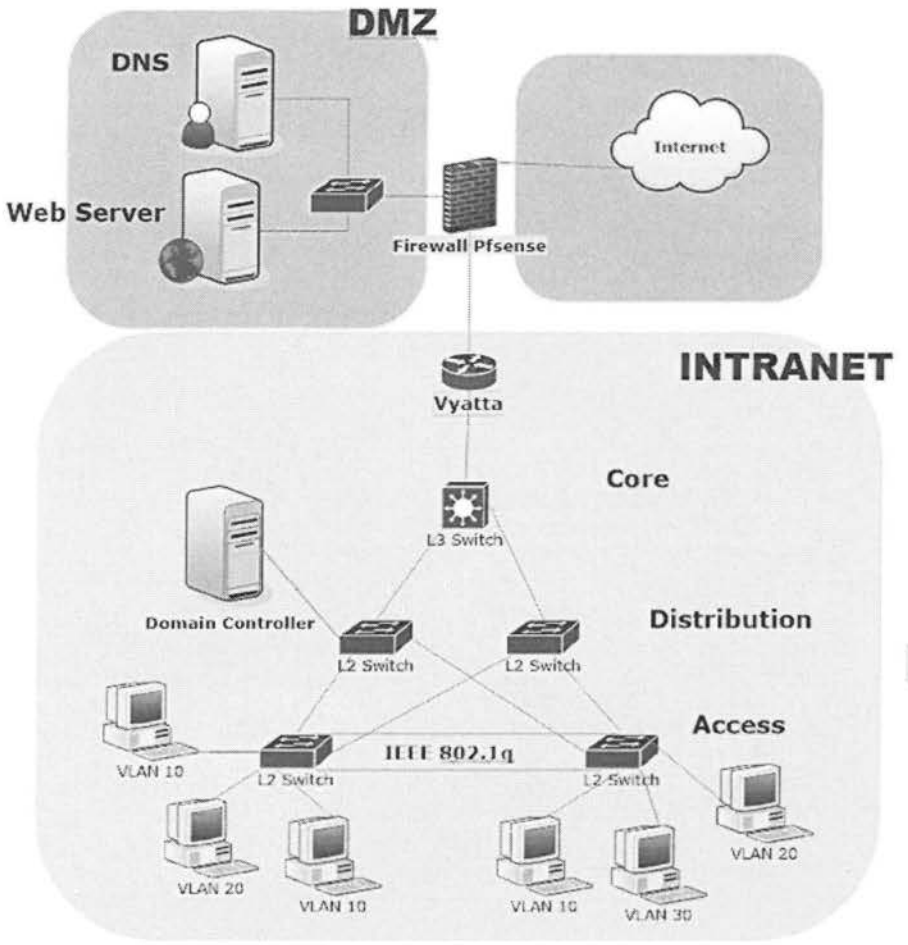
- IPv4/IPv6 δρομολόγηση με RIPv2/RIPng, OSPFv2/OSPFv3 και BGP
- DHCP/DNS server
- 802.1q VLANs
- WAN υποστήριξη T1/E1 και E3
- Firewall
- Site-to-site IPsec VPN
- NAT
- RADIUS πιστοποίηση
- VRRP
- Syslog

- SNMP
- CLI και web GUI
- Telnet και SSH

Το Vyatta έχει σχεδιαστεί για να τρέχει σε x86 αρχιτεκτονική. Ένα βασικό σύστημα θα πρέπει να έχει τουλάχιστον 512MB RAM, 1GB δίσκο και 1GHz CPU. Μια πλατφόρμα με x86 αρχιτεκτονική είναι αρκετά γρήγορη για να εκτελέσει δικτυακές λειτουργίες.

Οι x86 επεξεργαστές που παράγονται από την Intel, AMD και άλλες εταιρίες είναι από τις ποιο γρήγορες στο πλανήτη. Σε περιβάλλοντα δοκιμών που έγιναν, μια συσκευή που έτρεχε το λογισμικό Vyatta, είχε πολύ καλύτερες επιδόσεις από ένα δρομολογητή 2821 της Cisco. Είναι ένα λογισμικό λοιπόν που αξίζει τη προσοχή από οικιακό χρήστη μέχρι και έναν μεγάλο οργανισμό για πολύ αξιόπιστη open source λύση.

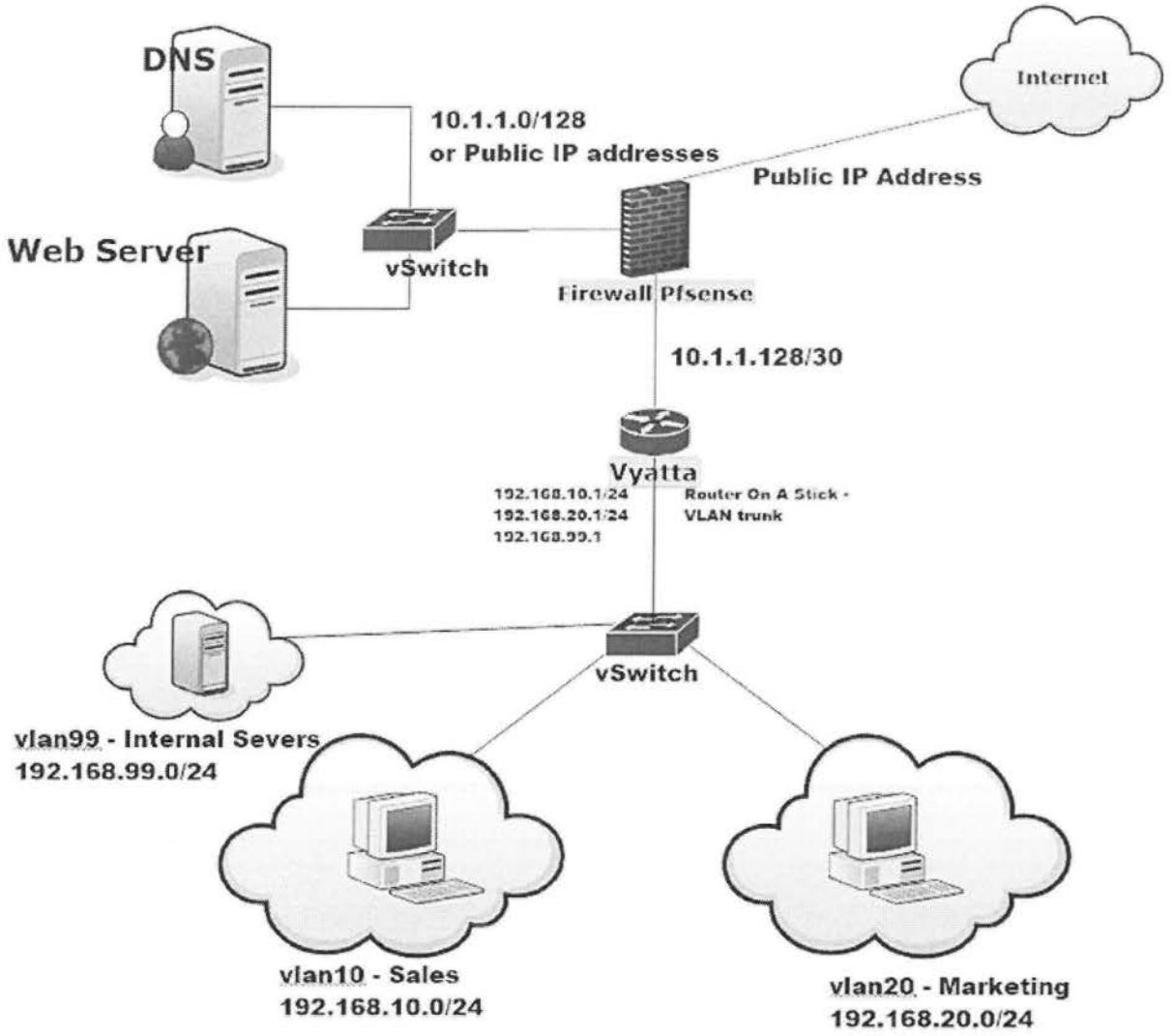
Μια βασική τοπολογία για έναν οργανισμό θα ήταν η εξής (Εικόνα 10-1):



Εικόνα 10-1: Τοπολογία για οργανισμό με full redundancy στα switches

Είναι μια βασική τοπολογία, όπου στο εσωτερικό δίκτυο (intranet) χρησιμοποιείτε το μοντέλο σχεδιασμού της Cisco (Access – Distribution - Core). Δεν θα μπούμε στη διαδικασία να αναλύσουμε το μοντέλο αυτό παρά μόνο περιληπτικά. Στο access layer συνδέονται τα τερματικά μας. Στο distribution layer εφαρμόζονται οι πολιτικές μας ενώ στο core layer γίνεται η μεγάλη μεταφορά δεδομένων του δικτύου μας αλλά εφαρμόζονται και πολιτικές επίσης. Το Firewall περιέχει τρεις δικτυακές διεπαφές, όπου χωρίζονται έτσι τρία δίκτυα: Το εσωτερικό (intranet), DMZ και το εξωτερικό (internet).

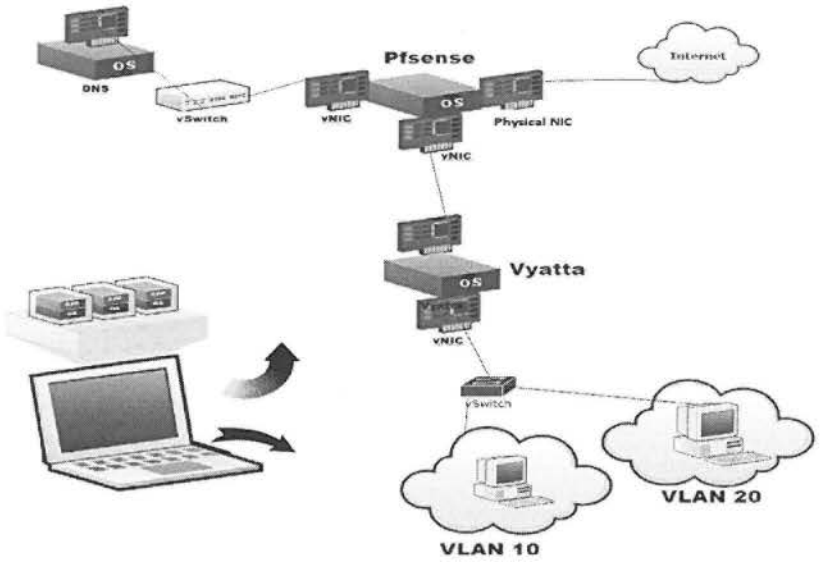
Στην εργασία αυτή όμως όλη η υλοποίηση θα γίνει σε εικονικό περιβάλλον σε ESX server της VMware. Επομένως η δικτυακή τοπολογία που θα υλοποιήσουμε για το firewall, DNS και δρομολογητή είναι η παρακάτω (Εικόνα 10-2):



Εικόνα 10-2: Τοπολογία firewall – Router και DNS

Στην παρακάτω εικόνα (Εικόνα 10-3) βλέπουμε πως από το laptop που τρέχει τον ESX server χωρίζεται το κάθε κομμάτι. Στον ESX server που τρέχει στο laptop μας δημιουργούμε τρεις τουλάχιστον εικονικές μηχανές:

- Για firewall το pfsense
- Για δρομολογητή το Vyatta
- Και έναν DNS server

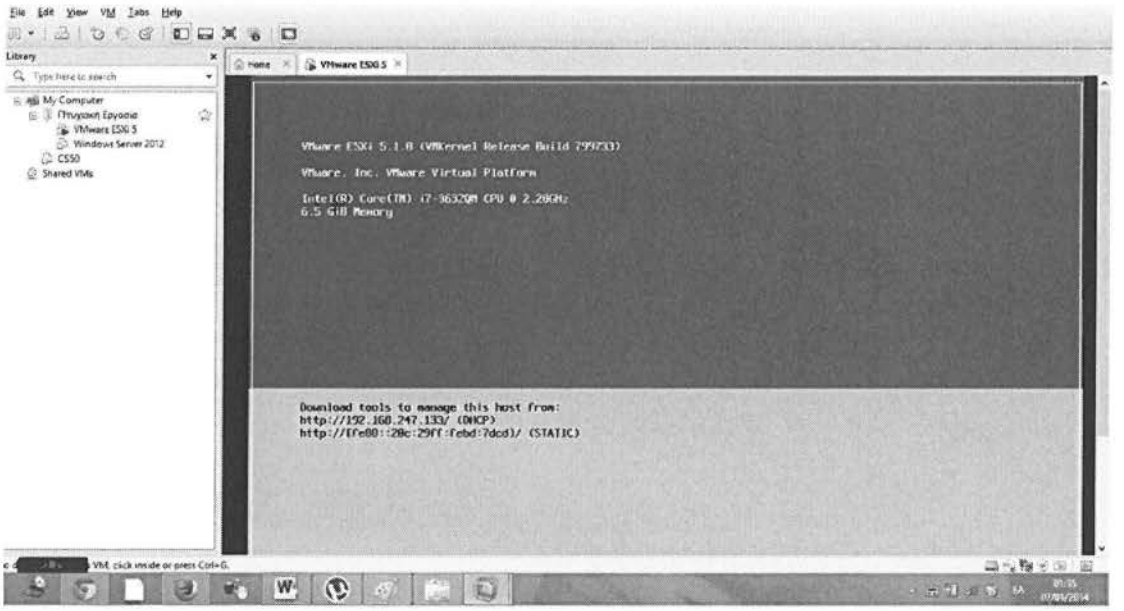


Εικόνα 10-3: Ο ESX server στο laptop μας

Εφόσον όμως δεν διαθέτουμε αρκετές κάρτες δικτύου, όλο το intranet και η DMZ ζώνη θα είναι επίσης εικονικοί. Επομένως δημιουργούμε και τις αντίστοιχες εικονικές κάρτες δικτύου (vnic). Όπου χρειαζόμαστε switch δημιουργούμε και τα αντίστοιχα vSwitches. Η μόνη φυσική κάρτα δικτύου που έχει το laptop τη χρησιμοποιούμε σαν την WAN διεπαφή του pfSense για να επικοινωνήσει με τον έξω «φυσικό» κόσμο.

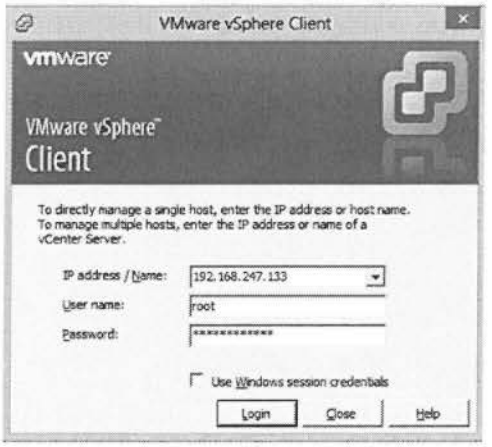
10.1 ESX

Στην εικόνα 10-4 βλέπουμε τον ESXi server να τρέχει στο laptop μας.



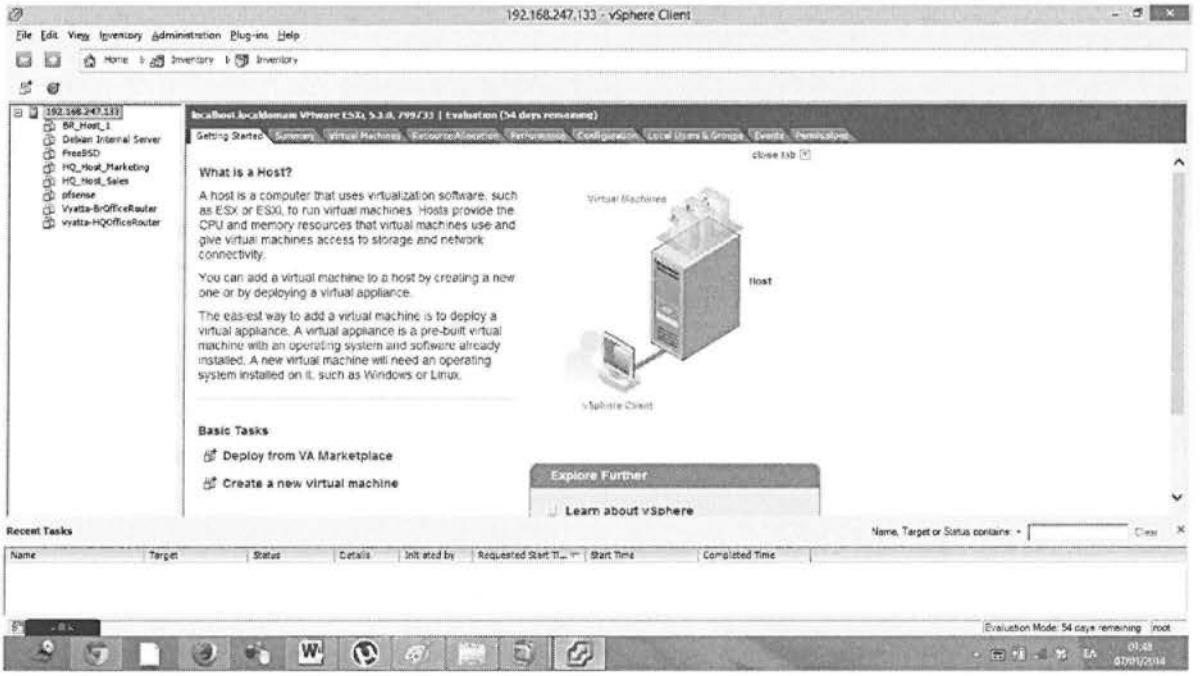
Εικόνα 10-4: Ο ESX server έτοιμος για διαχείριση

Για την διαχείριση του συνδεόμαστε σε αυτόν με ένα client πρόγραμμα (VMware vSphere Client) χρησιμοποιώντας την IP που έχει ο server μας. Στην συγκεκριμένη περίπτωση, ο Server έχει πάρει την IP 192.168.247.133 μέσω DHCP συνδεόμαστε λοιπόν στο server στην IP αυτή (Εικόνα 10-5).



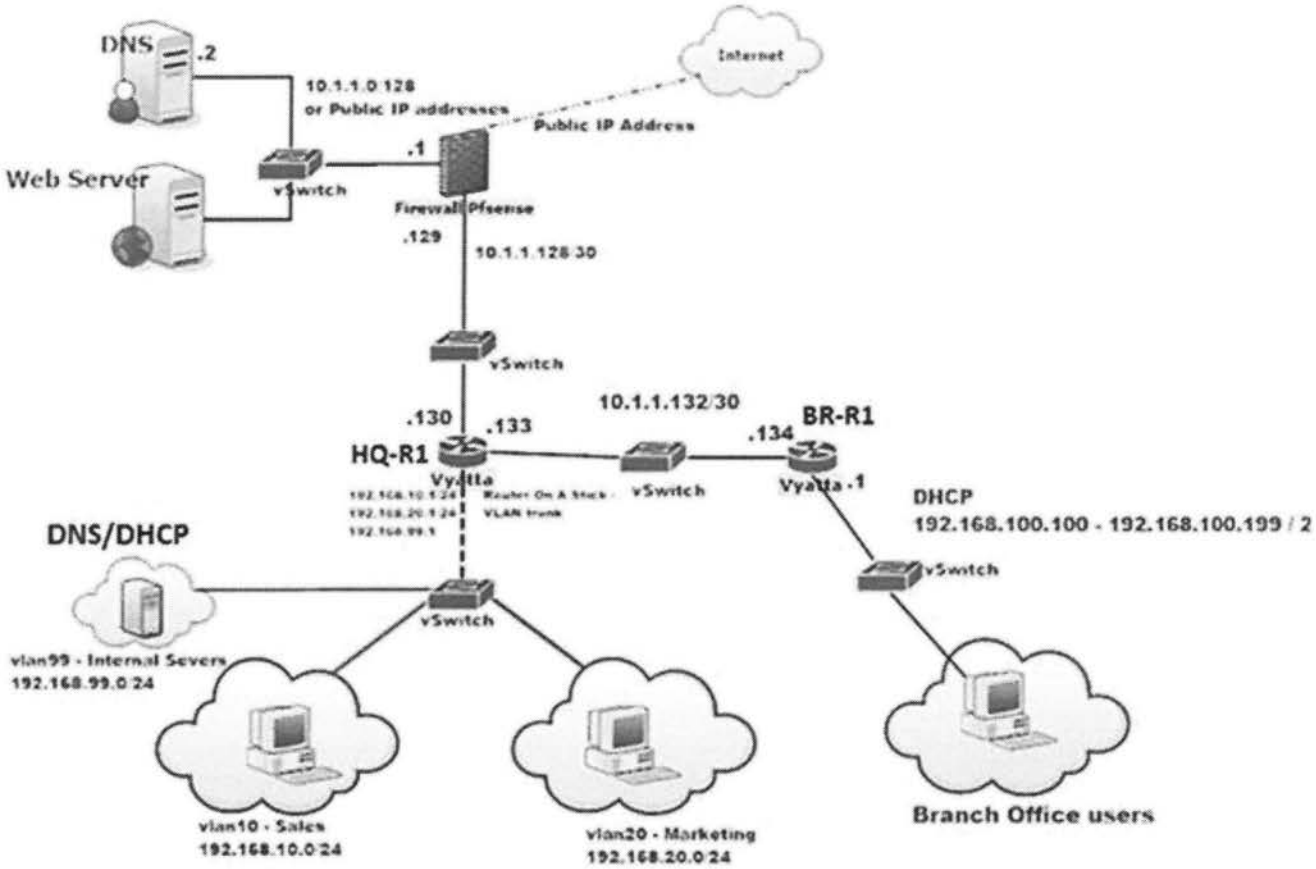
Εικόνα 10-5: VMware vSphere Client

Όταν συνδεθούμε ερχόμαστε στην παρακάτω εικόνα (Εικόνα 10-6):



Εικόνα 10-6: Περιβάλλον διαχείρισης του ESX server

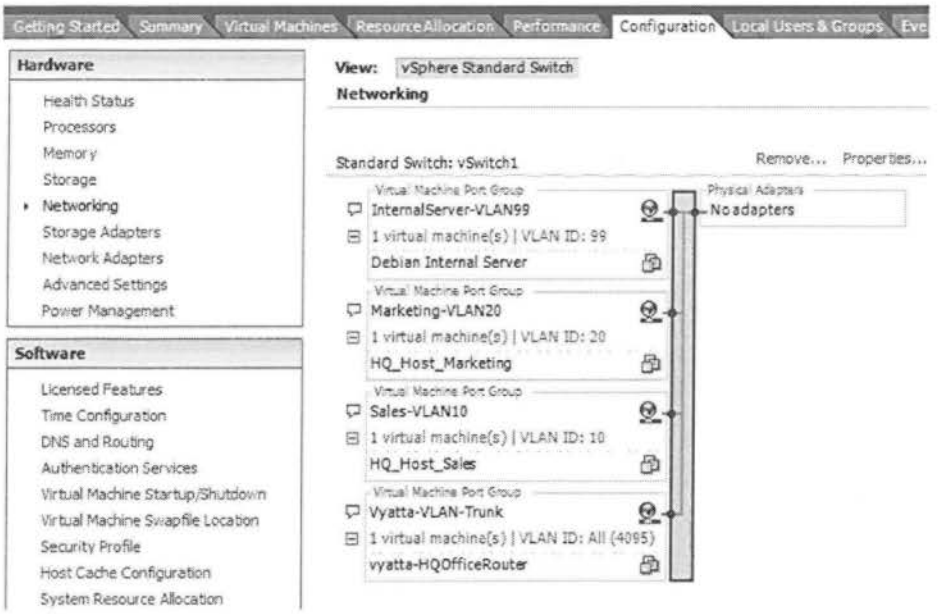
Από δω πλέον μπορούμε να διαχειριστούμε πλήρως τον ESX server. Οι λειτουργίες και οι δυνατότητες εξηγήθηκαν στο κεφάλαιο 6. Στο αριστερό τμήμα φαίνονται όλες οι εικονικές μηχανές που έχουμε δημιουργήσει και εγκαταστήσει. Ολόκληρη η τοπολογία που θα υλοποιήσουμε στον ESX server φαίνεται στην εικόνα 10-7.



Εικόνα 10-7: Τελική τοπολογία της υλοποίησης μας

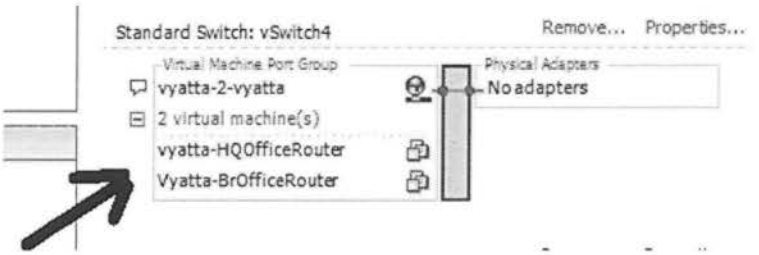
Ολόκληρο το δίκτυο τρέχει το πρωτόκολλο RIP σαν εσωτερικό πρωτόκολλο δρομολόγησης. Το HQ δίκτυο, έχει σπάσει σε τρία VLANs (Internal Servers, Sales και Marketing) και μπορεί και επικοινωνεί πλήρως με το BR δίκτυο. Μπορούμε να φανταστούμε τους δύο δρομολογητές να μην ενώνονται μέσω Ethernet, αλλά η σύνδεση τους να είναι μια dedicated leased γραμμή (E1) ή ακόμα ένα NBMA (Non Broadcast Multi Access) δίκτυο όπως το frame relay. Το πρωτόκολλο RIP θα δούλευε εξίσου το ίδιο και στις δύο περιπτώσεις. Το μόνο που θα άλλαζε είναι οι ρυθμίσεις του δρομολογητή για τις WAN αυτές διεπαφές.

Στην παρακάτω εικόνα (Εικόνα 10-8) βλέπουμε πως στον ESX έχει ρυθμιστεί το vSwitch που ενώνει μια διεπαφή του δρομολογητή HQ-R1 με τα υποδίκτυα VLAN10, VLAN20 και VLAN99 και πως συνδέονται οι εικονικές μηχανές που έχουμε εγκαταστήσει σε αυτόν. Μπορούμε να φανταστούμε τα Virtual Machine Port Group σαν Ethernet πόρτες σε switches. Έτσι λοιπόν σε κάθε Port Group συνδέουμε και τις αντίστοιχες εικονικές μηχανές που έχουμε δημιουργήσει. Στο Sales-VLAN10 για παράδειγμα έχουμε "συνδέσει" την εικονική μηχανή με όνομα HQ_Host_Sales.



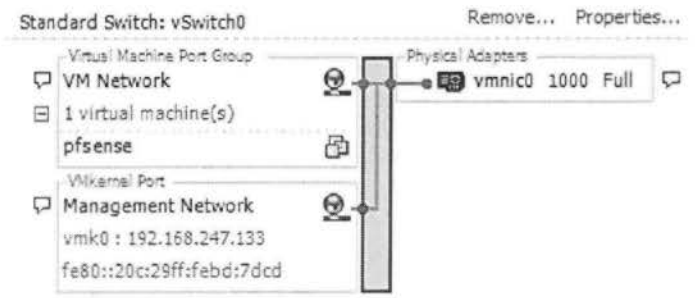
Εικόνα 10-8: Ρυθμίσεις vSwitch για HQ δίκτυο

Ομοίως στην εικόνα 10-9 βλέπουμε το switch που συνδέονται οι δύο δρομολογητές HQ-R1 και BR-R1 (στην εικόνα έχουν τα ονόματα vyatta-HQOfficeRouter και Vyatta-BrOfficeRouter αντίστοιχα). Οι δύο αυτοί δρομολογητές χρειάζονται ένα vSwitch με ένα Port Group και στο vSwitch αυτό συνδέουμε από μία διεπαφή για κάθε δρομολογητή.



Εικόνα 10-9: Ρυθμίσεις vSwitch για ένωση των δύο δρομολογητών

Τέλος στην εικόνα 10-10 βλέπουμε πως το pfsense συνδέεται με την φυσική κάρτα δικτύου για να έχουμε επικοινωνία με τον έξω κόσμο:



Εικόνα 10-10: Ρυθμίσεις vSwitch για pfsense για επικοινωνία με την φυσική κάρτα δικτύου μας

Εκτός από τις παραπάνω ρυθμίσεις έχουν γίνει και άλλες οι οποίες όμως δεν θα αναλυθούν περισσότερο. Ο κλάδος του virtualization είναι μεγάλος και απαιτεί περισσότερο από μια

αναφορά στην πτυχιακή αυτή. Από μεριά της πτυχιακής εργασίας αυτής είναι ότι έχει ρυθμιστεί και τροποποιηθεί για την τοπολογία της εικόνας 10-7.

10.2 VYATTA - VLANS ΚΑΙ RIP

Στον δρομολογητή BR-R1 μπορούμε να δούμε τις IPs των διεπαφών που έχουμε ρυθμίσει σύμφωνα με την εικόνα 10-7. Θυμίζουμε ότι οι κάρτες δικτύου για κάθε εικονική μηχανή είναι εικονικές εκτός από μία του firewall, και έχουν φτιαχτεί και ρυθμιστεί από το vSphere Client. Στην εικόνα 10-11 βλέπουμε ότι και στις δύο διεπαφές του BR-R1 (eth0 και eth1) έχουνε δοθεί οι IPs και ότι και στο φυσικό επίπεδο και στο επίπεδο ζεύξης, έχουμε σήμα (S/L – u/u, up/up).

```
vyatta@BR-R1:~$ show interffa

  Invalid command: show [interffa]

vyatta@BR-R1:~$ show interfaces
Codes: S - State, L - Link, u - Up, D - Down, A - Admin Down
Interface        IP Address                        S/L  Description
---------        ----------                        ---  -----------
eth0             10.1.1.134/30                     u/u
eth1             192.168.100.1/24                  u/u
lo               127.0.0.1/8                       u/u
                 ::1/128
vyatta@BR-R1:~$
```

Εικόνα 10-11: Προβολή των IPs στις διεπαφές στο δρομολογητή BR-R1

Για να ρυθμίσουμε μια IP σε μια συγκεκριμένη διεπαφή χρησιμοποιούμε την εντολή που φαίνεται στην εικόνα 10-12

```
vyatta@R1# set interfaces ethernet eth2 address 192.168.1.100/24
[edit]
vyatta@R1#
```

Εικόνα 10-12: Ρύθμιση IP σε μια Ethernet διεπαφή

Στην εικόνα 10-13 βλέπουμε τις IPs της κάθε διεπαφής του δρομολογητή HQ-R1.

```
vyatta@HQ-R1:~$ show interfaces
Codes: S - State, L - Link, u - Up, D - Down, A - Admin Down
Interface        IP Address                        S/L  Description
---------        ----------                        ---  -----------
eth0             10.1.1.130/30                     u/u  Connection-To-pfSense
eth1             -                                 u/u  VLAN-Trunk
eth1.10          192.168.10.1/24                   u/u  VLAN-10
eth1.20          192.168.20.1/24                   u/u  VLAN-20
eth1.99          192.168.99.1/24                   u/u  VLAN-99
eth2             10.1.1.133/30                     u/u
lo               127.0.0.1/8                       u/u
                 ::1/128
vyatta@HQ-R1:~$ _
```

Εικόνα 10-13: Οι IPs σε κάθε διεπαφή του δρομολογητή HQ-R1

Παρατηρούμε πως η διεπαφή eth1 έχει «σπάσει» σε τρείς υπό-διεπαφές, μία διεπαφή για κάθε μας VLAN. Ο σχεδιασμός αυτός ονομάζεται router-on-a-stick. Με τον τρόπο αυτό μπορούμε να απομονώσουμε όλα τα τερματικά που βρίσκονται στο eth1 interface του

δρομολογητή στο δικό του υποδίκτυο, που συνεπάγεται στο δικό του broadcast domain. Τα VLANs χρησιμοποιούνται ευρέως σήμερα σε κάθε οργανισμό και έχουν πολλά πλεονεκτήματα. Κυριότερα πλεονεκτήματα είναι η ασφάλεια και μείωση του φόρτου του δικτύου. Φανταστείτε να είχαμε 200 χρήστες κάτω από την eth1 διεπαφή και να μην χρησιμοποιούσαμε VLANs. Το εύρος ζώνης θα έπεφτε πολύ στο δίκτυο και ο φόρτος θα ήταν πολύ υψηλός αφού όλοι οι 200 χρήστες θα ανήκουν σε ένα broadcast domain. Με τα VLAN δημιουργούμε πολλά broadcast domain κατά συνέπεια μειώνεται πολύ ο φόρτος και η καθυστέρηση. Επίσης η τεχνολογία των VLANs επιτρέπει σε έναν οργανισμό να κατηγοριοποιεί κάθε τερματικό ανάλογα με το τμήμα που βρίσκεται. Στο παράδειγμα μας έχουμε τρία VLANs (Sales, Marketing και Internal servers). Έτσι κάθε τμήμα είναι στο δικό του υποδίκτυο και μπορούν να εφαρμοστούν διαφορετικές πολιτικές ασφαλείας και διαχείρισης για το κάθε τμήμα όπως θα δούμε παρακάτω. Τέλος αξίζει να σημειωθεί ότι κάθε τερματικό μπορεί να επικοινωνεί με ένα άλλο τερματικό που βρίσκεται στο ίδιο υποδίκτυο μέσω του switch (είναι στο ίδιο υποδίκτυο άρα μόνο η mac αρκεί του παραλήπτη όπως είδαμε στο κεφάλαιο 4 ενότητα 4). Για να επικοινωνήσει όμως με ένα τερματικό που ανήκει σε ένα άλλο VLAN, το τερματικό πρέπει να στείλει το πακέτο του στην προεπιλεγμένη πύλη, που είναι μία από τις υπό-διεπαφές της eth1 του δρομολογητή HQ-R1. Ο δρομολογητής με τη σειρά του θα προωθήσει το πακέτο στην υποδιεπαφή του VLAN που χρειάζεται. Αυτό το ρόλο κάνει η eth1 του δρομολογητή και η διαδικασία αυτή χαρακτηρίζεται ως trunking (πρωτόκολλο 802.1q).

Οι δύο δρομολογητές (HQ-R1 και BR-R1) και το firewall έχουν ρυθμιστεί και χρησιμοποιούν το πρωτόκολλο δρομολόγησης RIP. Θα μπορούσαμε να ρυθμίσουμε το OSPF σαν πρωτόκολλο δρομολόγησης αλλά για μια τόσο απλή τοπολογία δεν είναι αναγκαίο. Ο πίνακας δρομολόγησης για κάθε δρομολογητή, φαίνεται παρακάτω:

```
vyatta@BR--R1:~$ show ip route
Codes: K - kernel route, C - connected, S - static, R - RIP, O - OSPF,
       I - ISIS, B - BGP, > - selected route, * - FIB route

S>* 0.0.0.0/0 [1/0] via 10.1.1.133, eth0
R>* 10.1.1.0/25 [120/3] via 10.1.1.133, eth0, 00:59:39
R>* 10.1.1.128/30 [120/2] via 10.1.1.133, eth0, 02:19:38
C>* 10.1.1.132/30 is directly connected, eth0
C>* 127.0.0.0/8 is directly connected, lo
R>* 192.168.10.0/24 [120/2] via 10.1.1.133, eth0, 02:19:38
R>* 192.168.20.0/24 [120/2] via 10.1.1.133, eth0, 02:19:38
R>* 192.168.99.0/24 [120/2] via 10.1.1.133, eth0, 02:19:38
C>* 192.168.100.0/24 is directly connected, eth1
R>* 192.168.247.0/24 [120/3] via 10.1.1.133, eth0, 00:59:39
vyatta@BR--R1:~$
```

Στην παραπάνω εικόνα ο δρομολογητής BR-R1 έχει μάθει από τον γειτονικό δρομολογητή (HQ-R1) για όλα τα δίκτυα του εσωτερικού δικτύου μας που δεν είναι συνδεδεμένα στον δρομολογητή BR-R1: 10.1.1.128/30, 192.168.10.0/24, 192.168.20.0/24, 192.168.99.0/24.

Στην παρακάτω εικόνα (Εικόνα 10-14) βλέπουμε μερικές πληροφορίες για το πρωτόκολλο RIP που τρέχει στον δρομολογητή μας. Συγκεκριμένα βλέπουμε ότι ο δρομολογητής στέλνει RIPv2 μηνύματα και δέχεται και τις δύο εκδόσεις. Μπορούμε να δούμε τους μετρητές του πρωτοκόλλου, σε ποια δίκτυα διαφημίζει (Διεπαφές eth0 και eth1, δίκτυα 10.1.1.128/30 και 10.1.1.132/30) και ποιοι είναι οι γειτονικοί δρομολογητές που ανταλλάσσει RIP μηνύματα (ο 10.1.1.134 – BR-R1 και ο 10.1.1.129 –firewall).

```
vyatta@HQ-R1:~$ show ip rip status
Routing Protocol is "rip"
  Sending updates every 30 seconds with +/-50%, next due in 4 seconds
  Timeout after 180 seconds, garbage collect after 120 seconds
  Outgoing update filter list for all interface is not set
  Incoming update filter list for all interface is not set
  Default redistribution metric is 1
  Redistributing: connected
  Default version control: send version 2, receive any version
    Interface        Send  Recv   Key-chain
    eth0             2     1 2
    eth2             2     1 2
  Routing for Networks:
    10.1.1.128/30
    10.1.1.132/30
  Routing Information Sources:
    Gateway          BadPackets BadRoutes  Distance Last Update
    10.1.1.134               0         0       120   00:00:19
    10.1.1.129               0         0       120   00:00:24
  Distance: (default is 120)
vyatta@HQ-R1:~$ _
```

Εικόνα 10-14: Το πρωτόκολλο RIP τρέχει στον δρομολογητή HQ-R1

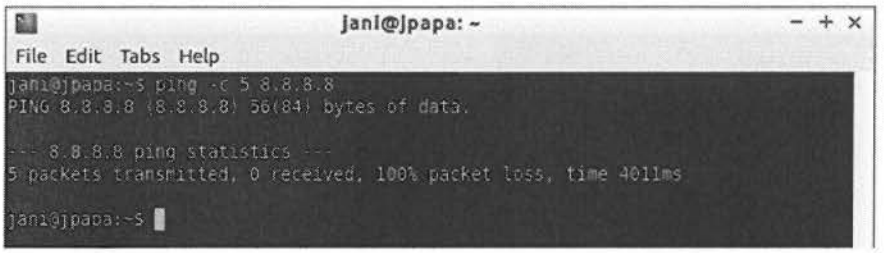
Από τα παραπάνω λοιπόν συμπεραίνουμε ότι στο εσωτερικό δίκτυο, όλοι μπορούν να επικοινωνούν μεταξύ τους εφόσον δεν υπάρχει κάποιος περιορισμός ασφαλείας που δεν μας επιτρέπει να επικοινωνήσουμε. Στην περίπτωση μας έχουν γίνει οι απαραίτητες ρυθμίσεις στο firewall για να επιτρέπει σε όλα τα δίκτυα να επικοινωνούν με το DMZ τμήμα του δικτύου και η πρόσβαση στο διαδίκτυο να επιτρέπεται μόνο στο υποδίκτυο 192.168.10.0/24 (Sales). Στις παρακάτω εικόνες γίνεται επαλήθευση σε όσα αναφέρθηκαν:

```
laertis@lpapa: ~
File  Edit  Tabs  Help
laertis@lpapa:~$ ifconfig
eth0      Link encap:Ethernet  HWaddr 00:0c:29:94:9e:69
          inet addr:192.168.10.10  Bcast:192.168.10.255  Mask:255.255.255.0
          inet6 addr: fe80::20c:29ff:fe94:9e69/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:30 errors:0 dropped:0 overruns:0 frame:0
          TX packets:69 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:2788 (2.7 KB)  TX bytes:10413 (10.4 KB)
          Interrupt:18 Base address:0x2000

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:1124 errors:0 dropped:0 overruns:0 frame:0
          TX packets:1124 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:89192 (89.1 KB)  TX bytes:89192 (89.1 KB)

laertis@lpapa:~$ ping 10.1.1.1
PING 10.1.1.1 (10.1.1.1) 56(84) bytes of data.
64 bytes from 10.1.1.1: icmp_seq=1 ttl=63 time=1.78 ms
64 bytes from 10.1.1.1: icmp_seq=2 ttl=63 time=2.30 ms
64 bytes from 10.1.1.1: icmp_seq=3 ttl=63 time=2.79 ms
64 bytes from 10.1.1.1: icmp_seq=4 ttl=63 time=3.49 ms
^C
--- 10.1.1.1 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3008ms
rtt min/avg/max/mdev = 1.782/2.567/3.492/0.643 ms
```

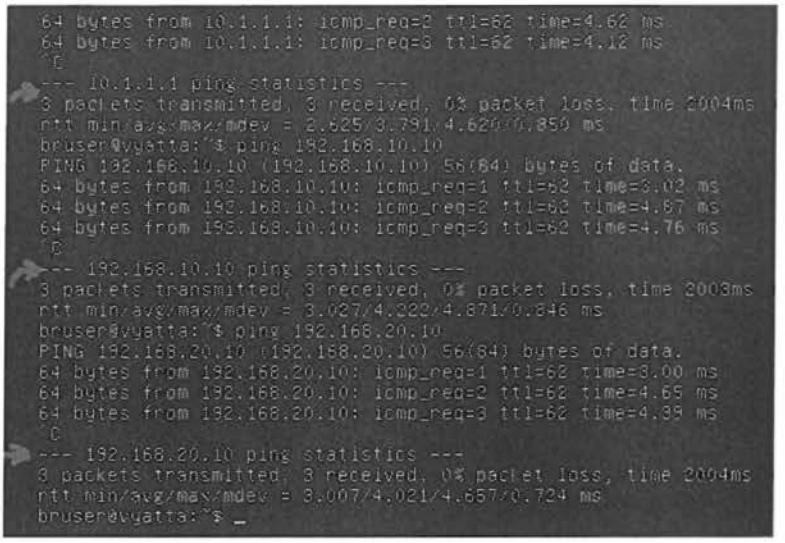
Εικόνα 10-15: Ο χρήστης στο SALES VLAN επικοινωνεί με το DMZ.

Εικόνα 10-16: Ο χρήστης στο Marketing VLAN επικοινωνεί με DMZ

Εικόνα 10-17: Ο χρήστης στο SALES VLAN επικοινωνεί με διακομιστή στο Ιντερνετ (δημόσιος DNS της Google)



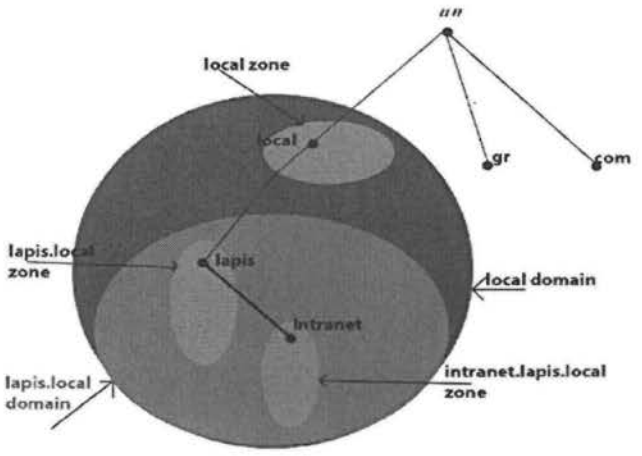
Εικόνα 10-18: Ο χρήστης στο Marketing VLAN δεν επικοινωνεί με δημόσιο DNS της Google (Κόβεται από firewall)



Εικόνα 10-19: Επικοινωνία BRuser με DMZ και INTRANET

10.3 DNS ΚΑΙ DHCP

Θα εγκαταστήσουμε δύο διακομιστές DNS: έναν στο εσωτερικό δίκτυο (Intranet) και έναν DNS διακομιστή όπου θα βρίσκεται στην DMZ ζώνη. Η αρχιτεκτονική αυτή είναι η καλύτερη, διότι ο δημόσιος διακομιστής μας δεν θέλουμε να περιέχει εγγραφές για κανένα στοιχείο του εσωτερικού μας δικτύου, κυρίως για λόγους ασφαλείας αλλά και για να μην επιβαρύνουμε άσκοπα το firewall με κίνηση από το εσωτερικό δίκτυο στο DMZ. Έτσι οι χρήστες που θέλουν DNS υπηρεσίες θα απευθύνονται στο εσωτερικό ιδιωτικό DNS server. Ο δημόσιος DNS χρησιμοποιείτε για να προοσφέρει DNS πληροφορίες στους δημόσιους διακομιστές μας αλλά και σε εξωτερικούς χρήστες που θέλουν να επικοινωννήσουν με τους δημόσιους διακομιστές μας (Authorative DNS server). Σε καμία περίπτωση δεν πρέπει ο δημόσιος διακομιστής μας να περιέχει πληροφορίες για το εσωτερικό μας αλλά και ούτε να χρησιμοποιηθεί σαν recursive dns server (για λόγους ασφαλείας).



Εικόνα 10-20: Local Domain και lapis.local domain

Στην εικόνα 10-20 βλέπουμε την ιεραρχική δομή του domain μας. Το domain local δεν υπάρχει σαν top level domain αλλά το βάλαμε για τα πλαίσια της πτυχιακής εργασίας αυτής. Το local θα μπορούσε να είναι gr, com.

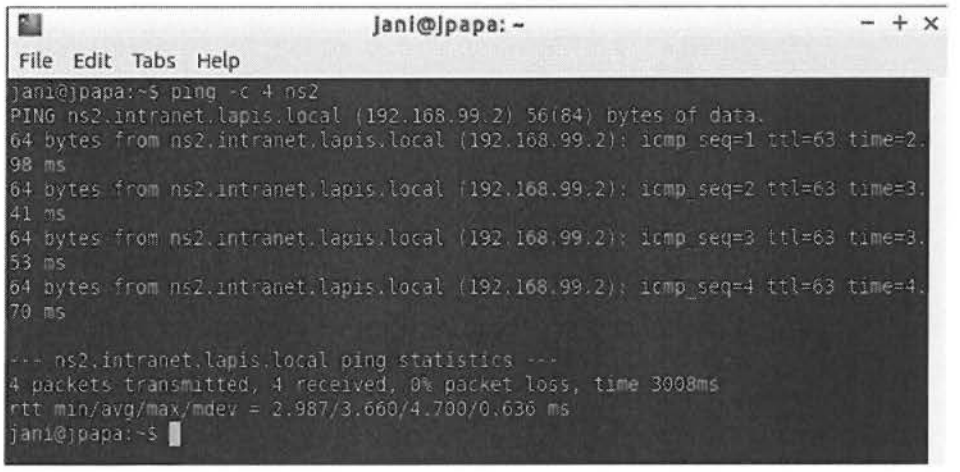
Στην εικόνα 10-20 φαίνεται επίσης ότι έχουμε ορίσει ένα domain lapis.local. Αυτό είναι και το θεωρητικά δημόσιο domain μας και εκεί βρίσκονται όλοι οι δημόσιοι διακομιστές μας στη DMZ ζώνη. Το εσωτερικό μας domain είναι το intranet.lapis.local και αυτό είναι το εσωτερικό μας ιδιωτικό δίκτυο.

Στο VLAN 99 έχουμε το υποδίκτυο που περιέχει εσωτερικούς διακομιστές. Σ' αυτό το υποδίκτου έχουμε εγκαταστήσει ένα δικομιστή Linux ο οποίος χρησιμοποιείται σαν DNS και DHCP server για να δώσει IPs στους χρήστες στα τοπικά δίκτυα (192.168.10.0/24 –Sales, 192.168.20.0/24 –Marketing, 192.168.99.0/24 –Internal Servers και 192.168.100.0/24 – Branch Office users) αλλα και να προσφέρει DNS υπηρεσίες.

Κάθε χρήστης σε κάθε υποδίκτυο στο τοπικό μας δίκτυο λαμβάνει μια IP από τον DHCP server και παράλληλα η IP αυτή μαζί με το hostname του χρήστη ενημερώνεται αυτόματα στη βάση δεδομένων του DNS (RR). Την ενημέρωση αυτή την κάνει το DHCP και η δυναμική ενημέρωση στις εγγραφές του DNS ονομάζεται DDNS (Dynamic DNS). Θυμίζουμε ότι τα DHCP μηνύματα είναι broadcat πλαίσια και κόβονται από τον δρομολογητή κατά προεπιλογή. Για να προωθήσουμε τα πλαίσια αυτά στο DHCP/DNS server μας (192.168.99.2) χρειάζεται να γίνουν οι απαραίτητες ρυθμίσεις στους δρομολογητές έτσι ώστε να μην κόβονται τα broadcast αυτά μηνύματα. Η ρύθμιση στοους δρομολογητές cisco ονομάζεται ip helper ενώ στα Vyatta το κάνεις με την εντολή:

SET SYSTEM DHCP-RELAY [OPTIONS]

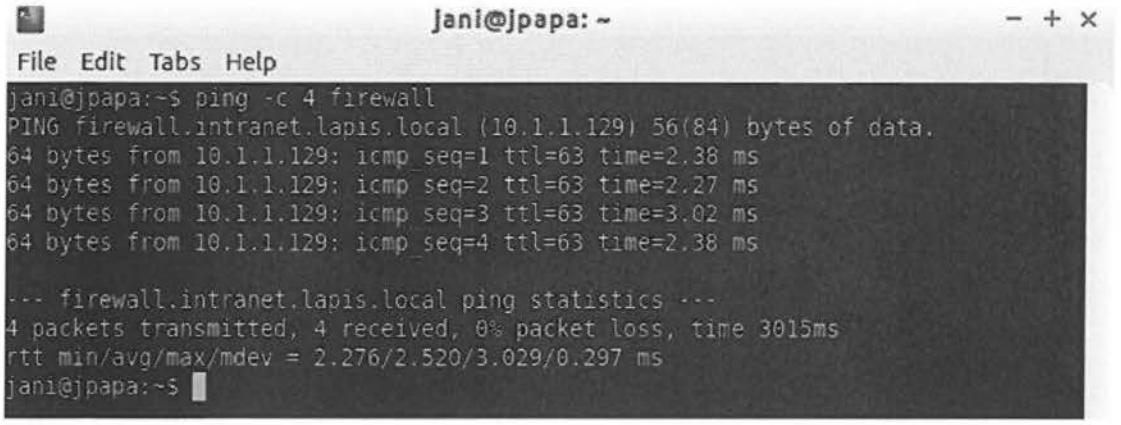
Ύστερα από την εγκατάσταση και τη ρύθμιση του DNS/DHCP server επαληθεύουμε τις ρυθμίσεις μας.



Εικόνα 10-21: Ο χρήστης jani στο υποδίκτυο Marketing μπορεί και επικοινωνεί με το DNS μέσω του ονόματος ns2

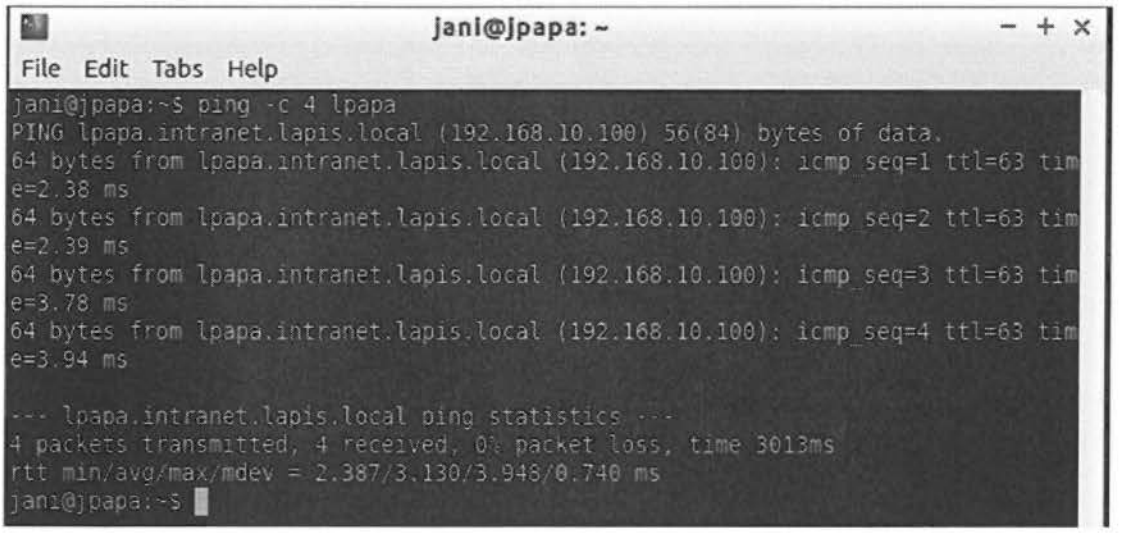
Στις παρακάτω εικόνες (10-22, 10-23, 10-24) ο χρήστης jpapa μπορεί επίσης να επικοινωνήσει με το firewall, HQ-R1, τον χρήστη lpapa που βρίσκεται στο Sales VLAN αλλά

και να πάρει την IP του διακομιστή www.google.com (Ping δεν γίνεται διότι μας κόβει το firewall).



Εικόνα 10-22: Ο χρήστης κάνει resolve το όνομα firewall στην IP 10.1.1.129

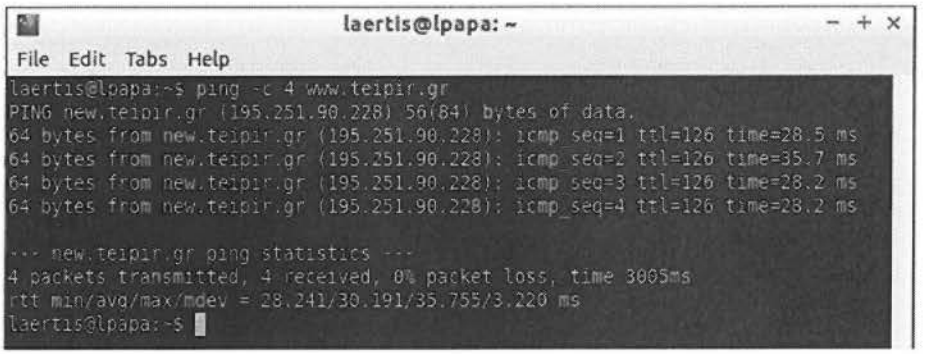
Εικόνα 10-23: Ο χρήστης κάνει resolve το όνομα HQ-R1 στην IP 10.1.1.130



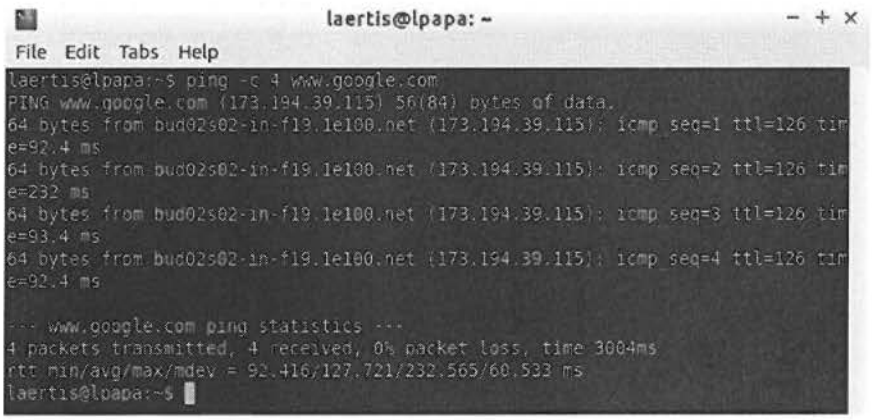
Εικόνα 10-24: Ο χρήστης κάνει resolve το όνομα του χρήστη lpapa στην IP 192.168.10.100

Εικόνα 10-25: Ο χρήστης κάνει resolve το όνομα www.google.com στην IP 173.194.39.116

Στο Sales VLAN όλοι οι χρήστες επιτρέπεται να βγαίνουν στο Internet. Παρακάτω (Εικόνα 10-26 και Εικόνα 10-27) επαληθεύουμε βλέποντας ότι μπορούμε να επικοινωνήσουμε με τους διακομιστές των www.google.com και www.teipir.gr

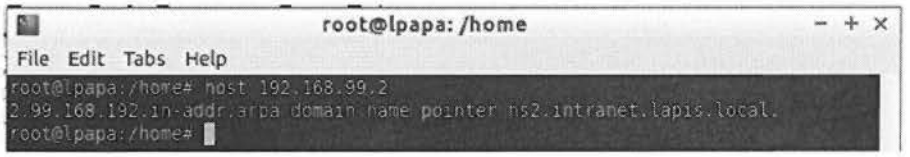


Εικόνα 10-26: Επικοινωνία από Sales VLAN με το διακομιστή www.teipir.gr



Εικόνα 10-27: Επικοινωνία από χρήστη στο Sales VLAN με το διακομιστή www.google.com

Στην εικόνα 10-28 και 10-29 βλέπουμε την ορθή λειτουργία του Reverse DNS για την IP 192.168.99.2 (DNS) και 192.168.100.100 (Bruser)

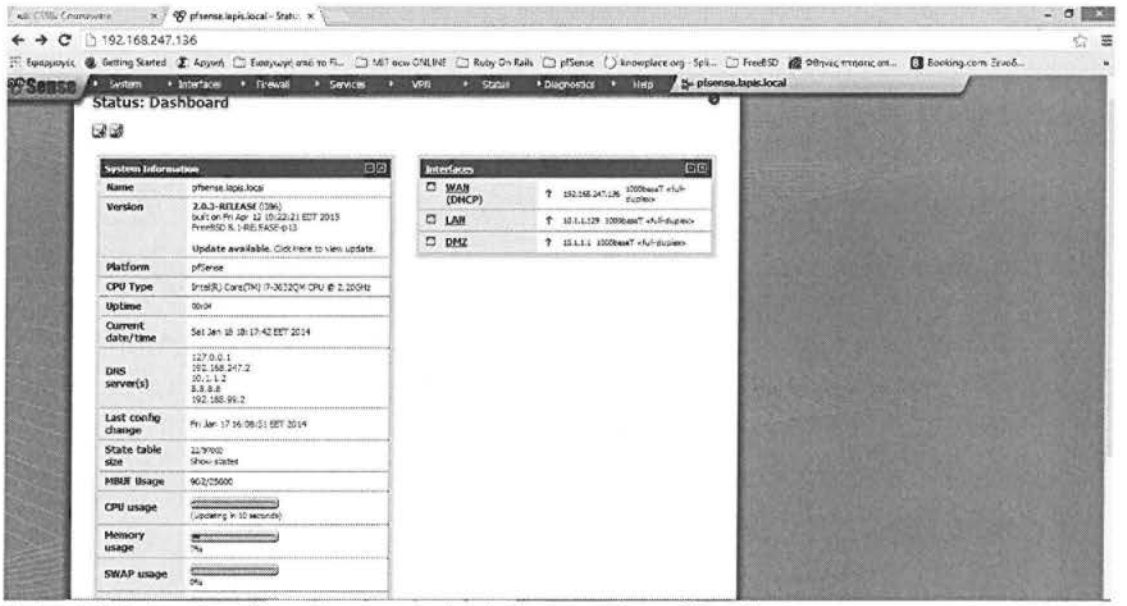


Εικόνα 10-28:Reverse DNS για την IP 192.168.99.2

Εικόνα 10-29 Reverse DNS για την IP του BRuser 192.168.100.100

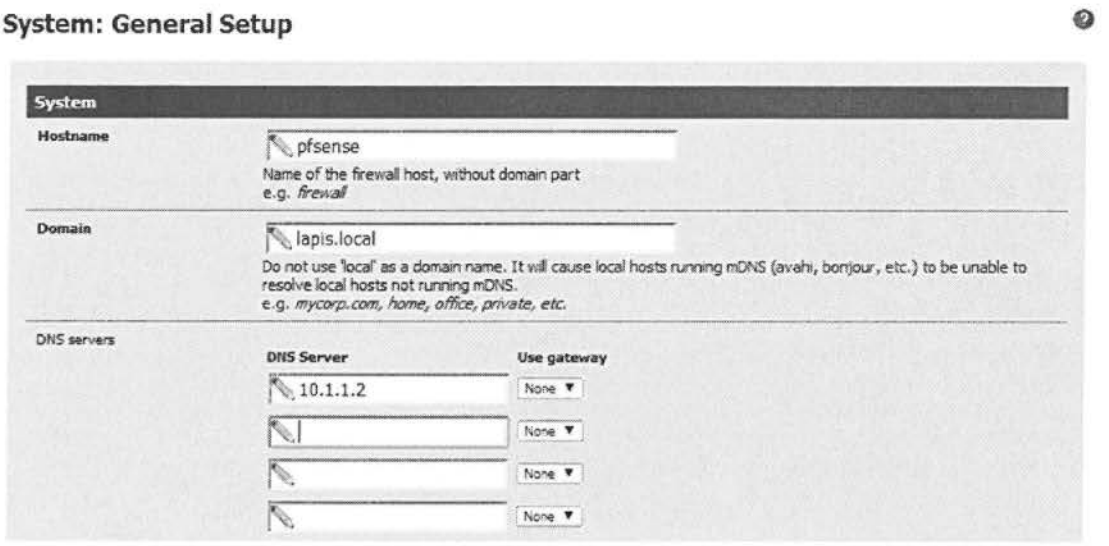
10.4 PFSENSE FIREWALL

Μετά την εγκατάσταση του pfsense μπορούμε να συνδεθούμε στο firewall από έναν web browser σε μια από τις IPs των διεπαφών του όπως φαίνεται στην εικόνα 10-30.



Εικόνα 10-30: Αρχική σελίδα μετά την επιτυχής σύνδεση στο pfsense firewall.

Από δω πλέον μπορούμε να διαχειριστούμε πλήρως το firewall μας από το μενού που βρίσκεται πάνω. Σημειώνεται ότι έχουν γίνει οι βασικές ρυθμίσεις στις διεπαφές για να έχουν τις IPs της τοπολογίας της εικόνα 10-7. Οι ρυθμίσεις για της διεπαφές γίνονται από το μενού Interfaces. Επίσης από το μενού System – general setup ορίζουμε και το hostname, domain name και τους DNS servers που θα έχει το firewall (Εικόνα 10-31).

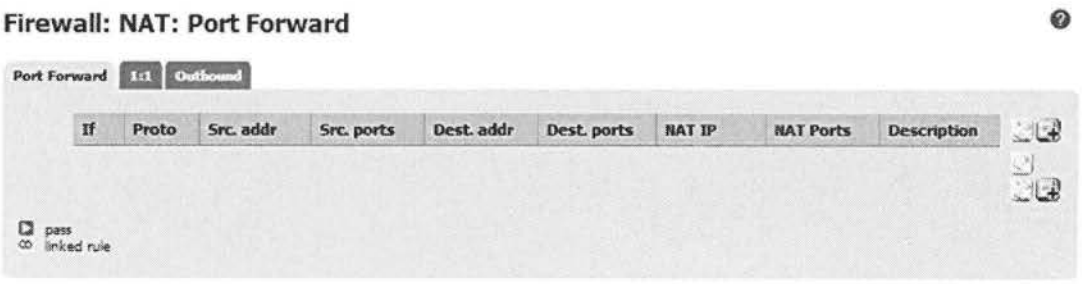


Εικόνα 10-31: Ρυθμίσεις Hostname, Domain και DNS server στο pfsense

Τέλος μια άλλη ρύθμιση που είναι απαραίτητη να γίνει είναι να ορίσουμε ποιες IPs θα μπορούν να γίνονται NAT στη WAN διεπαφή. Από default μόνο το εσωτερικό δίκτυο Local του pfsense Μπορεί να γίνει NAT (10.1.1.0/24). Εμείς όμως έχουμε σπάσει το δίκτυο μας σε πολλά υποδίκτυα συμπεριλαμβανόμενου και των δικτύων τρίτης κλάσης 192.168.10.0, 192.168.20.0, 192.168.99.0, 192.168.100.0.

Επομένως για όποια από αυτά τα δίκτυα θέλουμε να βγαίνουν στο Internet θα πρέπει να φτιάξουμε και τους αντίστοιχους κανόνες να γίνονται NAT στο firewall είτε σε μια Public IP που έχουμε είτε στην IP της WAN διεπαφής του Firewall (PAT). Εμείς ρυθμίζουμε το firewall για τη δεύτερη λύση μιας και δεν έχουμε Public IP. Επίσης για public IPs για την εργασία αυτή προσομοιώνουμε τη διεύθυνση 192.168.247.0/24.

Από το μενού Firewall -> NAT μπορούμε να διαχειριστούμε πλήρως της δυνατότητες του NAT (Εικόνα 10-32) όπως port forwarding, 1:1 NAT αν θέλουμε να αντιστοιχήσουμε 1 προς 1 διευθύνσεις, δηλαδή μια private σε μια public IP και τέλος να ρυθμίσουμε ποιες εσωτερικές διευθύνσεις να μεταφράζονται στη WAN διεύθυνση (PAT).

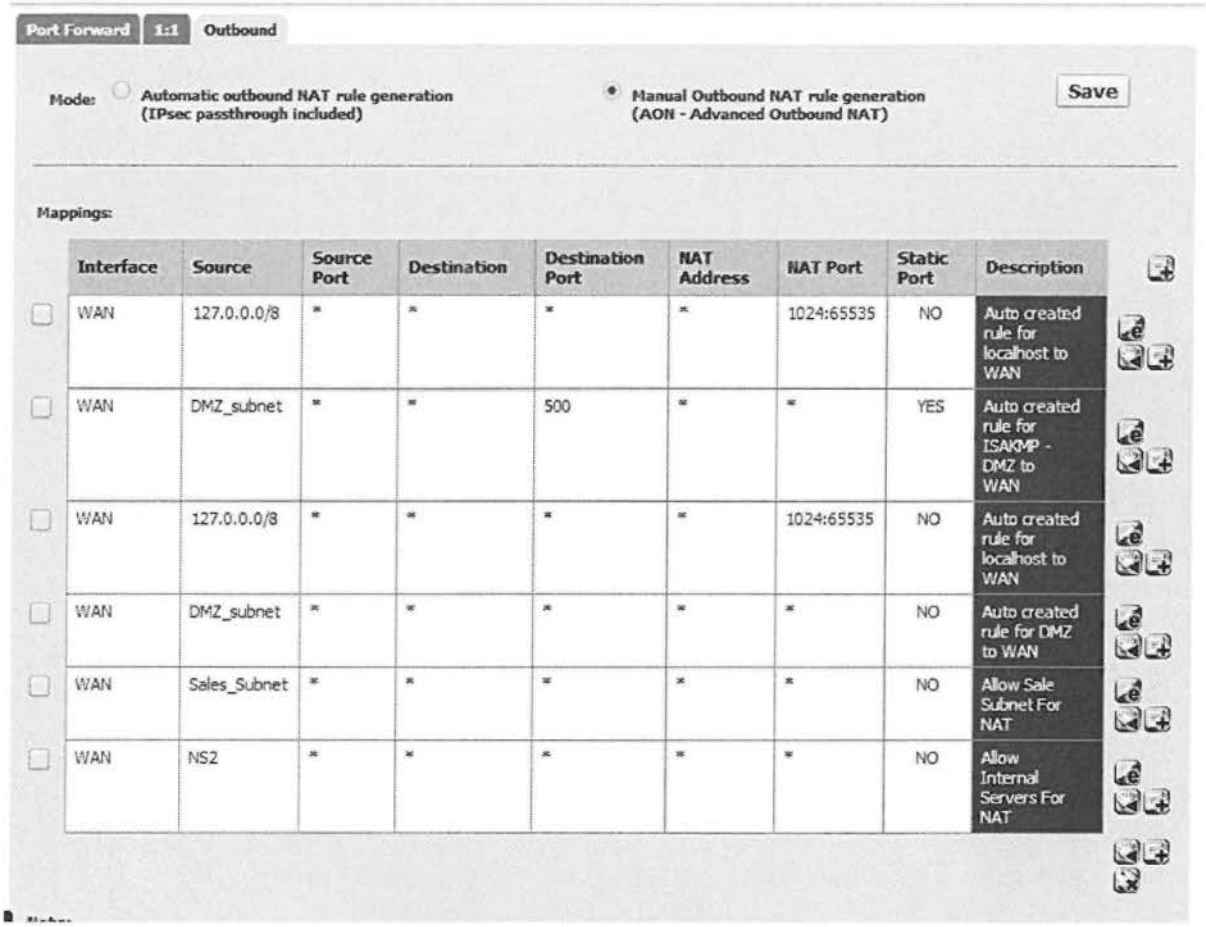


Εικόνα 10-32: NAT ρυθμίσεις στο pfsense

Στην καρτέλα outbound γίνονται οι ρυθμίσεις για μεταφράσεις PAT (Port Address Translation). Στην καρτέλα αυτή επιλέγουμε την επιλογή Manual Outbound NAT Rule Generation και ορίζουμε ποιες διευθύνσεις επιτρέπονται για NAT (Εικόνα 10-33).

Οι πρώτοι 6 κανόνες δημιουργούνται αυτόματα από το pfsense με το που επιλέξουμε την επιλογή Manual Outbound NAT Rule Generation. Αυτοί είναι οι default κανόνες που αναφέραμε προηγουμένως για το δίκτυο 10.1.1.0/24. Το pfsense έχει αναγνωρίσει ότι έχουμε σε 2 διεπαφές τα υποδίκτυα 10.1.1.0/25 και 10.1.1.128/30 και έχει φτιάξει τους κανόνες για να επιτρέπονται σε αυτές τις διευθύνσεις μεταφράζονται. Αν δεν θέλουμε να μεταφράζονται μπορούμε πολύ απλά να τις σβήσουμε. Εμείς θα τροποποιήσουμε τους κανόνες να γίνονται μεταφράσεις μόνο στο DMZ υποδίκτυο (10.1.1.0/25), στο υποδίκτυο Sales (192.168.10.0/24) κα μόνο στο εσωτερικό DNS server 192.168.99.2. Επομένως οι κανόνες φαίνονται στην εικόνα 10-34.

Εικόνα 10-33: NAT default outbound configuration



Εικόνα 10-34: Κανόνες NAT για τη τοπολογία μας

Παρατηρούμε ότι επιτρέπουμε στους source DMZ-subnet, Sales_subnet και NS2 να μπορούν να γίνονται μετάφραση σε οποιαδήποτε IP που έχει η WAN διεπαφή. Αν ρυθμίζαμε να έχει συγκεκριμένη IP η IP αυτή θα φαινόταν στη στήλη NAT Address.

Στη στήλη source παρατηρούμε ότι δεν έχουμε IPs αλλά ονόματα. Αυτά στο pfsense λέγονται aliases και μας δίνουν μεγάλη διαχειρισιμότητα στο firewall. Με τα aliases, μπορούμε να ονομάσουμε υποδίκτυα, Hosts, Πόρτες και URLS και να τα χρησιμοποιούμε στο pfsense όπου χρειάζεται να βάλουμε κάποιο από τα παραπάνω. Αυτό μας δίνει μεγάλη ευελιξία, διότι φανταστείτε να είχαμε το υποδίκτυο 10.1.1.0/25 (DMZ υποδίκτυο), ορισμένο σε 10 διαφορετικές ρυθμίσεις (φιλτράρισμα, NAT, VPN, Poxy κτλ). Αν θέλαμε να κάνουμε μια αλλαγή στις IPs του DMZ θα έπρεπε να αλλάξουμε και όλες τις IPs σε κάθε μας ρύθμιση. Με τα aliases απλά αλλάζουμε τις IPs στο alias και όλες οι άλλες ρυθμίσεις παραμένουν ίδιες χωρίς να χρειαστεί να ξανά ρυθμίσουμε κάθε κομμάτι χωριστά. Ακόμα και το όνομα να αλλάξουμε στο alias θα αλλάξει και το όνομα του alias σε κάθε μας ρύθμιση.

Για να ρυθμίσουμε τα aliases πάμε από το μενού firewall -> Aliases όπως φαίνεται στην εικόνα 10-35. Βλέπουμε ότι έχουμε ρυθμίσει κάθε όνομα για κάθε υποδίκτυο, για όλο μας το εσωτερικό υποδίκτυο (INRTANET) ή για κάποιο host ξεχωριστά (NS2, DMZ_DNS κτλ).

Firewall: Aliases

Name	Values	Description
BranchOfficeVLAN	192.168.100.0/24	branch office VLAN
DMZ_DNS	10.1.1.2	DMZ server01 - DNS
DMZ_subnet	10.1.1.0/26	DMZ
DMZ_WEB_FTP	10.1.1.2	DMZ server 02 - WEB-FTP
External_DNS	8.8.8.8	External DNS for forwarder
Interal_admin_subnet	192.168.0.0/24	Internal admin subnet
Internal_servers_subnet	192.168.99.0/24	Internal server subnet
INTRANET	10.1.1.128/30, 10.1.1.132/30, 192.168.0.0/24, 192.168.10.0/24, 192.168.20.0/24, 192.168.99.0/24, 192.168.100.0/24	INTRANET addresses
Marketing_Subnet	192.168.20.0/24	Marketing Subnet Description
ME	192.168.226.0/24, 192.168.247.0/24	My Subnet
NS2	192.168.99.2	Internal DNS server
Sales_Subnet	192.168.10.0/24	Sales Subnet Description

Εικόνα 10-35: Aliases στο pfsense

10.4.1 ΦΙΛΤΡΑΡΙΣΜΑ ΠΑΚΕΤΩΝ

Το πρώτο βήμα που πρέπει να κάνει κάποιος με το που εγκαταστήσει ένα firewall είναι να ρυθμίσει τους κανόνες για το φιλτράρισμα πακέτων. Οι κανόνες αυτοί θα πρέπει να ρυθμιστούν με προσοχή όπως είδαμε στο κεφάλαιο 6.

Οι κανόνες ρυθμίζονται για κάθε interface του firewall. Επομένως θα πρέπει να ρυθμιστούν κανόνες σε κάθε διεπαφή όπου χρειάζεται. Γενικά η συμπεριφορά στο pfsense είναι να μπλοκάρει οποιαδήποτε κίνηση εκτός και αν κάποιος κανόνας το επιτρέψει. Οι κανόνες διαβάζονται από πάνω προς τα κάτω. Αν βρεθεί κάποιος κανόνας που να ταιριάζει με το πακέτο που διέρχεται η εξέρχεται από τη διαπαφή, τότε σταματά ο έλεγχος και εκτελεί το κανόνα για το πακέτο αυτό (ή μπλοκάρει το πακέτο ή το επιτρέπει να περάσει).

Στην εικόνα της τοπολογίας μας (Εικόνα 10-7) αρχικά ορίζουμε κάποιους κανόνες στη διεπαφή WAN όπως φαίνεται στην εικόνα 10-37. Αριστερά από κάθε κανόνα το κόκκινο εικονίδιο σημαίνει μπλοκάρισμα το πράσινο εικονίδιο σημαίνει ότι επιτρέπεται ο κανόνας. Από πάνω προς τα κάτω μετράμε τους κανόνες από 1 έως Ν.

Αναλυτικότερα έχουμε:

- **Κανόνας 1:** Να μπλοκάρεται οποιοσδήποτε που προσπαθεί να επικοινωνήσει με το εσωτερικό μας δίκτυο.
- **Κανόνας 2 και 3:** Να επιτρέπεται η κίνηση που απευθύνεται στο εξωτερικό δημόσιο DNS server μας και **δεν** έχει διεύθυνση αποστολέα IP από το DMZ δίκτυο αλλά ούτε IP από το εσωτερικό μας δίκτυο. Η διεύθυνση αποστολέα ελέγχεται για να αποτραπούν επιθέσεις IP spoofing.

Floating | WAN | LAN | DMZ

	ID	Proto	Source	Port	Destination	Port	Gateway	Queue	Schedule	Description
☐ ✖		*	*	*	INTRANET	*	*	none		Block access to internal network
☐ ▶		TCP/UDP	! DMZ_subnet	*	DMZ_DNS	53 (DNS)	*	none		Allow Traffic to Public DNS
☐ ▶		TCP/UDP	! INTRANET	*	DMZ_DNS	53 (DNS)	*	none		Allow Traffic to Public DNS
☐ ▶		TCP	! INTRANET	*	DMZ_WEB_FTP	80 (HTTP)	*	none		Allow Traffic to Public HTTP
☐ ▶		TCP	! DMZ_subnet	*	DMZ_WEB_FTP	80 (HTTP)	*	none		Allow Traffic to Public HTTP
☐ ▶		TCP	! DMZ_subnet	*	DMZ_WEB_FTP	21 (FTP)	*	none		Allow Traffic to Public FTP
☐ ▶		TCP	! INTRANET	*	DMZ_WEB_FTP	21 (FTP)	*	none		Allow Traffic to Public FTP
☐ ▶		TCP	Admin_subnet	*	DMZ_subnet	22 (SSH)	*	none		Allow Traffic to Public FTP
☐ ▶		*	Sales_Subnet	*	*	*	*	none		Allow Only Sales Network
☐ ▶		TCP	ME	*	192.168.247.136	80 (HTTP)	*	none		Allow ME to pfsense WAN HTTP ONLY
☐ ▶		*	*	*	*	*	*	none		Allow everything

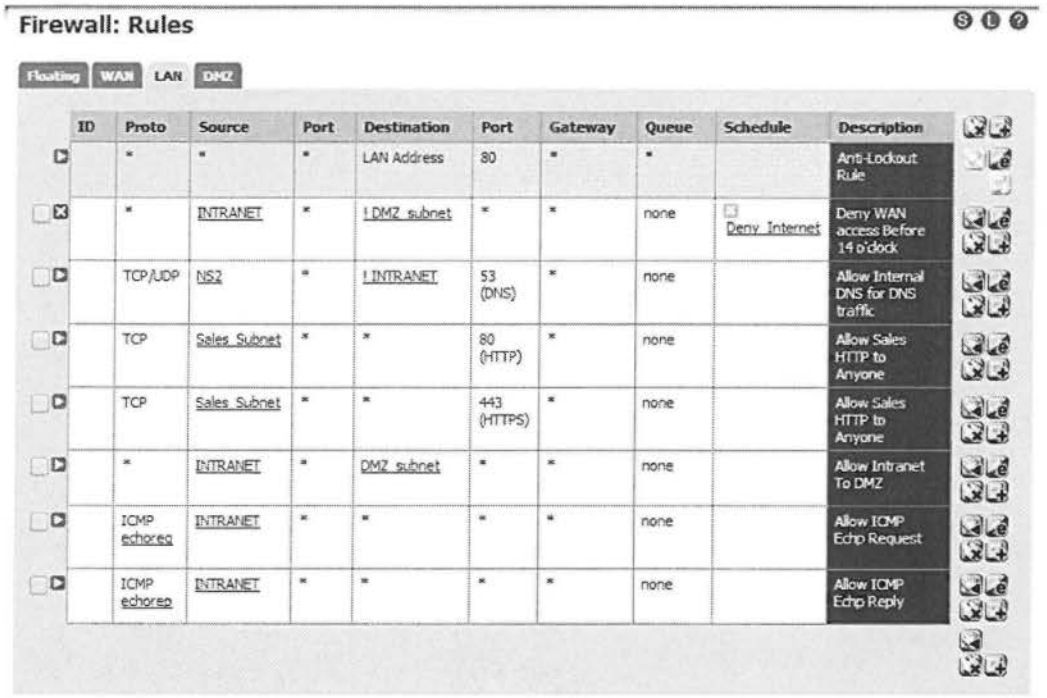
Εικόνα 10-36: Κανόνες φιλτραρίσματος για τη WAN διεπαφή στο pfsense

- **Κανόνας 4 και 5:** Να επιτρέπεται κίνηση στον εξωτερικό δημόσιο Web server μας και οι IPs αποστολέα να μην είναι καμία από τις IPs του εσωτερικού ή του DMZ υποδικτύου μας.
- **Κανόνας 6 και 7:** Όμοια με τους κανόνες 4,5 και 2,3 απλά για ftp κίνηση.
- **Κανόνας 8:** Να επιτρέπεται SSH κίνηση στο DMZ υποδίκτυο μας, αλλά μόνο για τις IPs αποστολέα που ανήκουν στο Admin_Subnet.
- **Κανόνας 9:** Επιτρέπει να περνάει η κίνηση στο Sales VLAN.
- **Κανόνας 10:** Προσωρινός κανόνας που μου επιτρέπει να συνδέομαι στη WAN διεπαφή του pfsense από το στην πόρτα 80 (HTTP).
- **Κανόνας 11:** Είναι απενεργοποιημένος και τον είχαμε στην αρχή δοκιμαστικά και επιτρέπει όλη τη ροή της κίνησης.

Στην εικόνα 10-37 βλέπουμε τους κανόνες που έχουμε δημιουργήσει για τη διεπαφή LAN. Ο πρώτος κανόνας έχει δημιουργηθεί αυτόματα από το pfsense και υπάρχει για να μην κλειδωθούμε έξω από το firewall. Επιτρέπει στο εσωτερικό δίκτυο LAN να συνδέονται στο pfsense μέσω HTTP πρωτόκολλο. Αναλυτικότερα:

- **Κανόνας 2:** Είναι προγραμματισμένος κανόνας που λειτουργεί από τις 7:00 – 14:00 και κόβει όλα τα πακέτα.
- **Κανόνας 3:** Επιτρέπει στο NS2 (εσωτερικό DNS) να περάσει μόνο για DNS πρωτόκολλο κυκλοφορία (για τη διαδικασία της ανάλυσης).
- **Κανόνας 4 και 5:** Επιτρέπει στο Sales VLAN (192.168.10.0/24) να περάσει HTTP και HTTPS κυκλοφορία.
- **Κανόνας 6:** Επιτρέπει σε όλο το εσωτερικό μας δίκτυο να επικοινωνήσει με το DMZ τμήμα μας.
- **Κανόνας 7:** Επιτρέπει πακέτα ICMP echo request.

- **Κανόνας 8:** Επιτρέπει πακέτα ICMP echo reply.



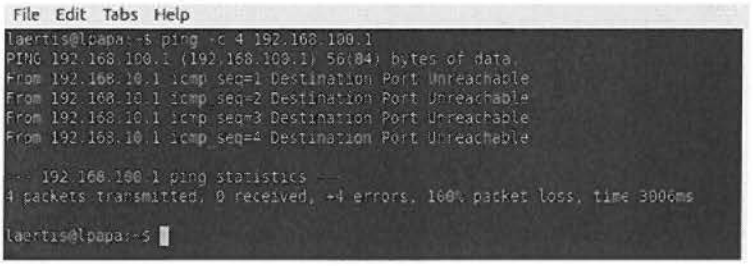
Εικόνα 10-37: Κανόνες φιλτραρίσματος για LAN διεπαφή

Υπενθυμίζουμε ότι φιλτράρισμα πακέτων μπορούμε να κάνουμε και με έναν δρομολογητή. Στο vyatta για παράδειγμα μπορούμε να προσθέσουμε επίσης κανόνες που να αποτρέπουν κάποια πακέτα. Για παράδειγμα στο δρομολογητή HQ-R1 μπορούμε να το ρυθμίσουμε έτσι ώστε το υποδίκτυο Sales VLAN (192.168.10.0) να μην μπορεί να επικοινωνήσει με το υποδίκτυο στο Branch Office (192.168.100.0/24). Το φιλτράρισμα στους δρομολογητές έχει σαν αποτέλεσμα να μειώνεται η κίνηση στο firewall συνεπώς να κερδίζουμε περισσότερο εύρος ζώνης. Στην εικόνα 10-38 βλέπουμε τη ρύθμιση στο δρομολογητή HQ-R1 ενώ στις εικόνες 10-39 και 10-40 βλέπουμε πως από το Sales VLAN κόβονται τα πακέτα ενώ από το marketing VLAN περνάνε στο δίκτυο 192.168.100.0/24 (Branch Office).

```
vyatta@HQ-R1#
[edit]
vyatta@HQ-R1# show firewall
 name FWTEST {
     rule 1 {
         action reject
         destination {
             address 192.168.100.0/24
         }
         source {
             address 192.168.10.0/24
         }
     }
     rule 2 {
         action accept
         destination {
             address 0.0.0.0/0
         }
         source {
             address 0.0.0.0/0
         }
     }
 }
[edit]
vyatta@HQ-R1# _
```

Εικόνα 10-38: Φιλτράρισμα πακέτων για το υποδίκτυο Sales στο δρομολογητή HQ-R1

Εικόνα 10-39: Επιτυχής επικοινωνίας μεταξύ Marketing και Branch Office



Εικόνα 10-40: Ανεπιτυχής επιτυχία μεταξύ Sales και Branch Office

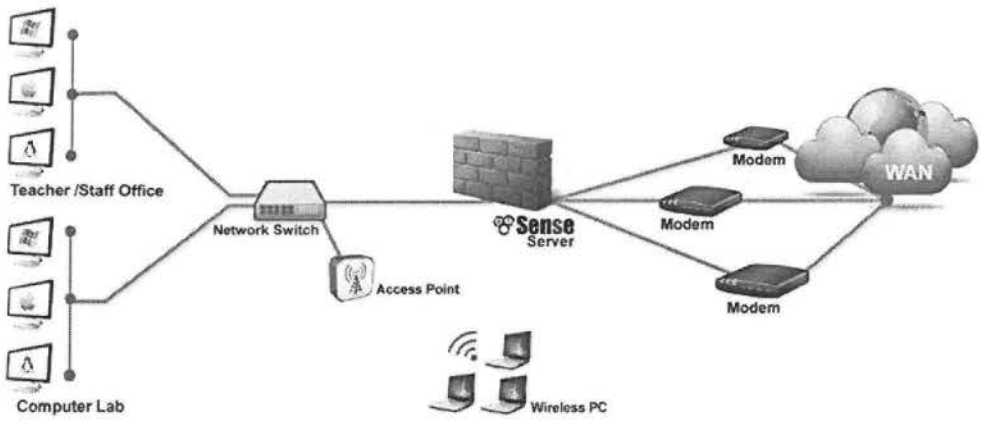
10.4.2 DUAL WAN – LOAD BALANCE

Πολλοί οργανισμοί χρησιμοποιούν δύο ή και περισσότερες WAN συνδέσεις (συνδέονται σε δύο ή περισσότερους παρόχους) και ένας δρομολογητής χρησιμοποιείται για να εξισορρόπηση του φορτίου στις συνδέσεις αυτές. Με τη χρήση του δημοφιλούς open source router/firewall pfsense μπορούμε δημιουργήσουμε ένα πολύ ισχυρό με πολλές δυνατότητες δρομολογητή.

Η χρησιμοποίηση ενός dual wan δρομολογητή έχει κυρίως δύο πλεονεκτήματα:

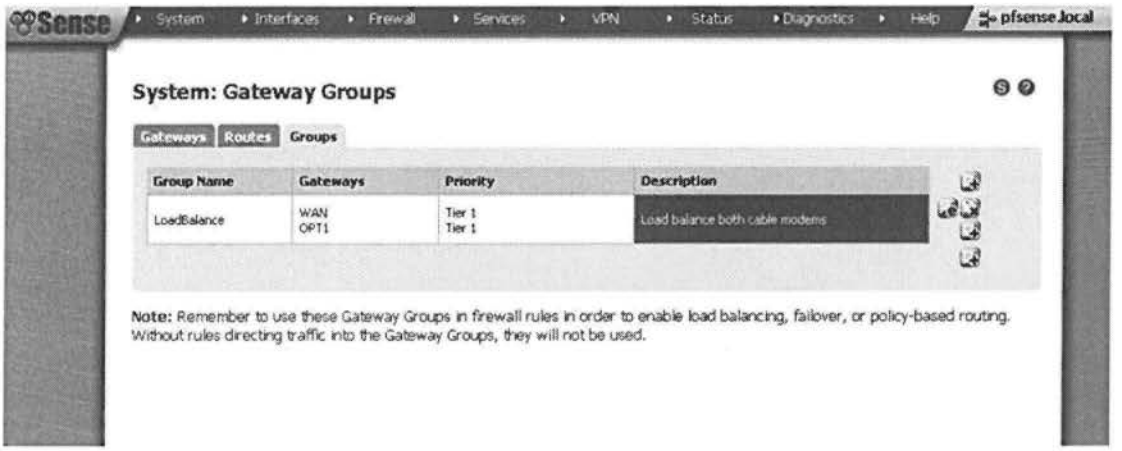
- **Αύξηση του εύρους ζώνης του Διαδικτύου**: Αν έχουμε πολλαπλές διαδικτυακές συνδέσεις, μπορούμε να εξισορροπήσουμε το φορτίο για να παρέχουμε περισσότερο εύρος ζώνης στου Η/Υ στο δίκτυο μας.
- **Παρέχει μια backup σύνδεση για το Διαδίκτυο**: Μία άλλη δημοφιλής εφαρμογή είναι να παρέχει πλεόνασμα (redundancy) σε περίπτωση που μια σύνδεση στο Ιντερνετ βγαίνει offline. Σε αυτή την περίπτωση, είναι σημαντικό να περιέχουμε τουλάχιστον δύο διαφορετικές ISP γραμμές. Για παράδειγμα 2 DSL συνδέσεις.

Για την υλοποίηση θα χρειαστούμε N + 1 κάρτες δικτύου που θα πρέπει να έχει ο Η/Υ που τρέχει το pfsense (όπου N οι WAN συνδέσεις μας). Ν συνδέσεις για τους ISPs και μια για το εσωτερικό τοπικό μας δίκτυο. Στην εικόνα 10-41 βλέπουμε ένα παράδειγμα με το pfsense να λειτουργεί και σαν load balancer σε τρείς διαφορετικές WAN συνδέσεις.



Εικόνα 10-41: pfSense load balance

Στην εικόνα 10-42 έχουμε φτιάξει ένα group με τις δύο wan συνδέσεις (καρτέλα system->routing).



Εικόνα 10-42: Load Balancer – Group Interfaces

Στην καρτέλα status-> gateways βλέπουμε ότι οι δύο διεπαφές λειτουργούν σωστά και ότι η κατάσταση και στις δύο διεπαφές είναι Online(εικόνα 10-43).

Εικόνα 10-43: Έλεγχος κατάστασης των wan διεπαφών στο pfsense

Η τελευταία ρύθμιση που πρέπει να κάνουμε είναι στους κανόνες του firewall και συγκεκριμένα στη διεπαφή LAN (καρτέλα firewall -> Rules). Συγκεκριμένα πρέπει για κάθε

κανόνα να βάλουμε για gateway το group των διεπαφών που δημιουργήσαμε νωρίτερα (Εικόνα 10-44).

Εικόνα 10-44: Ρύθμιση Gateway για κάθε κανόνα

Οι ρυθμίσεις για το gateway είναι στο advanced features για κάθε κανόνα (firewall->rules [new rule || edit rule])

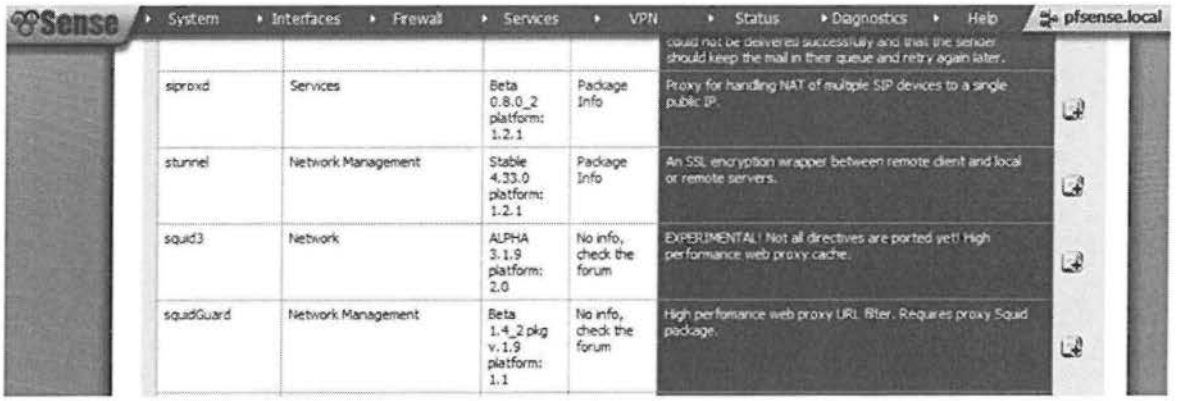
Το pfsense χρησιμοποιεί ένα round robin αλγόριθμο για να καθορίσει ποια διεπαφή θα χρησιμοποιήσει για την έξοδο της δικτυακής κίνησης. Στην σελίδα system->advanced και έπειτα στην καρτέλα miscellaneous -> load balancing τσεκάροντας την επιλογή use sticky connections, όπου οι επιτυχές συνδέσεις θα στέλνονται στην ίδια έξοδο που σημαίνει στην ίδια IP. Αυτή η ρύθμιση μπορεί να χρειαστεί διότι μερικές SSL συνδέσεις συμπεριφέρονται περίεργα όταν αλλάζει η IP διεύθυνση αποστολέα.

10.4.3 ΦΙΛΤΡΑΡΙΣΜΑ URL

Το URL φιλτράρισμα είναι μια μέθοδος για να μπλοκάρουμε προσβάσεις σε συγκεκριμένες ιστοσελίδες με βάση το όνομα τους (όχι το περιεχόμενο). Υπάρχουν πολλές εμπορικές λύσεις για φιλτράρισμα URL ή περιεχομένου. Παρακάτω θα δούμε πώς με το pfsense και συγκεκριμένα με το πακέτο SquidGuard όπου είναι ένα πολύ χρήσιμο plugin για το δημοφιλές Squid proxy server οπού μπορεί να χρησιμοποιηθεί για να μπλοκάρει ή να ανακατευθύνει web αιτήματα.

Το SquidGuard έχει πολλές δυνατότητες όπου μπορούν να παραμετροποιηθούν για να καλύψουν πλήρως τις ανάγκες μας. Επίσης είναι πολύ γρήγορο και δεν θα δούμε καθυστέρηση στην ταχύτητα του Internet. Το SquidGuard είναι πολύ ευέλικτο καθιστώντας το πολύ εύκολο να προσαρμοστεί σε διαφορετικές εφαρμογές. Αν θέλουμε βασικό φιλτράρισμα URL στο οικιακό μας δίκτυο ή να δημιουργήσουμε σύνθετους κανόνες για ένα μεγάλο δημόσιο δίκτυο το SquidGuard είναι η open source λύση που χρειαζόμαστε.

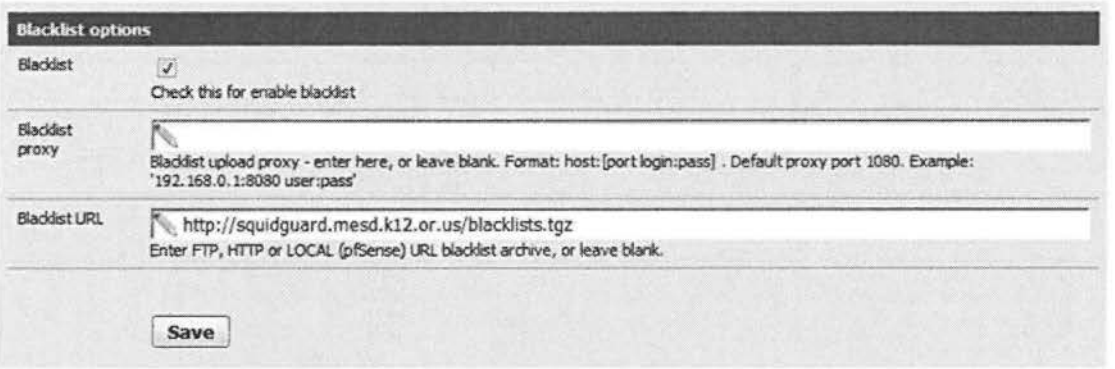
Η εγκατάσταση του SquidGuard είναι πολύ εύκολη και γίνεται μέσα από το διαχειριστή πακέτων του pfsense (System -> Packages, εικόνα 10-45).



Εικόνα 10-45: Διαχειριστής πακέτων στο pfsense

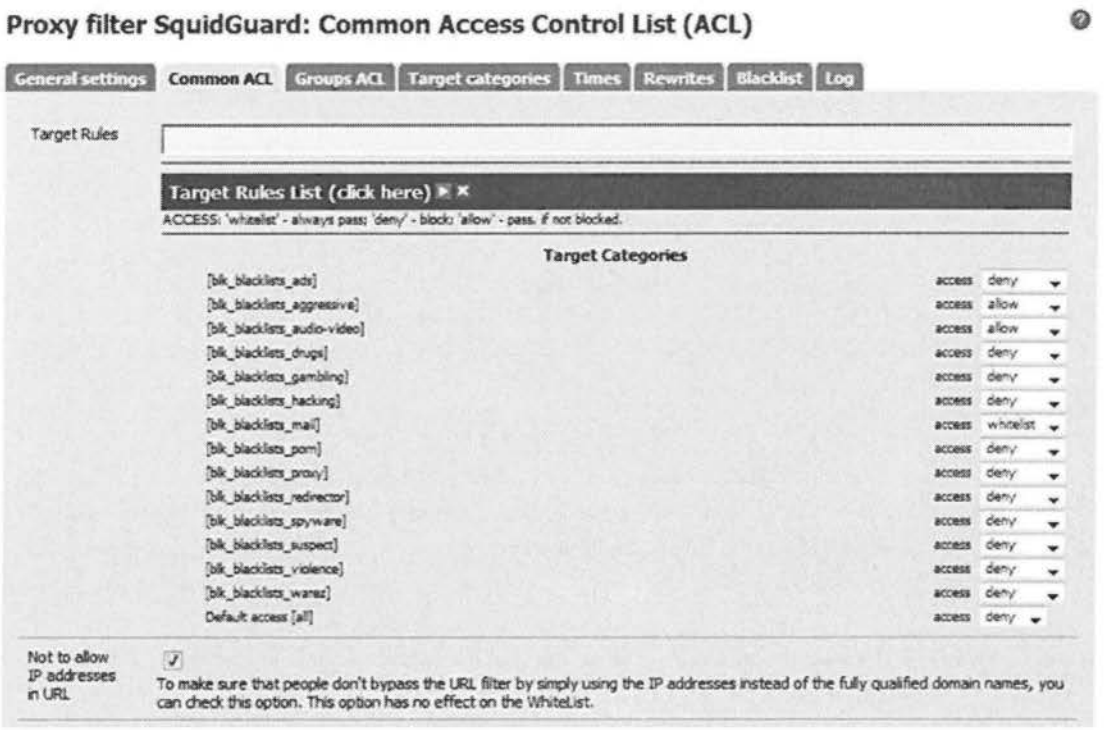
Πατώντας το κουμπί + μπορούμε να εγκαταστήσουμε το πακέτο που θέλουμε. Όταν τελειώσει η εγκατάσταση το νέο μενού είναι κάτω από την καρτέλα Services -> proxy filter.

Κάτω από τις γενικές ρυθμίσεις (General Settings) του SquidGuard μπορούμε να ενεργοποιήσουμε το blacklist και να βάλουμε κάποια δικιά μας blacklist ή κάποια άλλη δημόσια (εικόνα 10-46).



Εικόνα 10-46: SquidGuard BlackList

Μόλις ανεβάσουμε το blacklist στο pfsense μας, επόμενο βήμα είναι ρυθμίσουμε ποιες κατηγορίες θα πρέπει να επιτρέπονται, μπλοκάρονται ή να είναι λευκές (white). Ο ποιο εύκολος τρόπος για τη ρύθμιση αυτή είναι να χρησιμοποιήσουμε τη καρτέλα ACL (Εικόνα 10-47).



Εικόνα 10-47: SquidGuard Access List

Οι ρυθμίσεις στην καρτέλα Common ACL επιδρούν σε όλους τους χρήστες αν θέλουμε διαγορετικές ρυθμίσεις για κάθε δίκτυο θα πρέπει να γίνουν από την καρτέλα Groups ACL.

Για κάθε κατηγορία έχουμε τρείς διαφορετικές ενέργειες:

- **Allow:** Επιτρέπει την πρόσβαση στη συγκεκριμένη κατηγορία εκτός και αν μπλοκάρεται από άλλο κανόνα.
- **Deny:** Μπλοκάρει όλες τις προσβάσεις στις ιστοσελίδες στη συγκεκριμένη κατηγορία.
- **Whitelist:** Πάντα επιτρέπει τη πρόσβαση στις ιστοσελίδες στη συγκεκριμένη κατηγορία.

Για να προσθέσουμε μια καινούργια κατηγορία μπορούμα να πάμε στην καρτέλα target categories και να προσθέοσυμε τις ιστοσελίδες που θέλουμε να μπλοκάρουμε ή να επιτρέψουμε την πρόσβαση. Μολις δημιουργήσουμε και σώσουμε τη λίστα θα πρέπει να πάμε πάλι στη καρτέλα ACL και να δώσουμε μία από τις τρείς ενέργειες (Allow, Deny, Whitelist).

Μια άλλη σημαντική δυνατότητα που έχει το SquidGuard είναι να δημιουργεί φίλτρα χρησιμοποιώντας regular expressions. Για να δημιουργήσουμε ένα φίλτρο που χρησιμοποιεί regular expressions πάμε στη καρτέλα target categories και είτε δημιουργούμε μια καινούργια κατηγορία ή κάνουμε επεξεργασία μια υπάρχουσα. Στο πεδίο Regular Expression γράφουμε το κώδικα μας. Στη συνέχεια πάμε στη καρτέλα ACL και επιλέγουμε όποια από τις τρεις ενέργειες θέλουμε. Παρακάτω βλέπουμε μερικά παραδείγματα regular expressions.

Μπλοκάρισμα Download με βάση την κατάληξη του αρχείου.

(.\/.*\.(zip|rar|exe|msi|mpeg|avi))*

Μπλοκάρισμα συγκεκριμένων domain ανώτατου επιπέδου.

(.gov|.xxx|.mil|.net)

Μπλοκάρισμα αναζητήσεων για «proxy bypass» στο Google και Yahoo

(.(google|yahoo).*(search_query|keywords|search|query|q|p)=.*(\+|\%20)*(proxy|bypass).*(\-|\+|\%20).*(proxy|bypass).*)*

Τέλος το SquidGuard μας δίνει την δυνατότητα να εφαρμόσουμε τα φίλτρα URL με βάση το χρόνο (schedule). Τα schedules είναι χρήσιμα για την εφαρμογή κανόνων σε διαφορετικές χρονικές στιγμές κατά τη διάρκεια της ημέρας, ή μόνο σε ορισμένες ημέρες της εβδομάδας.

Για παράδειγμα μπορούμε να χρησιμοποιήσουμε κάποιους αυστηρούς κανόνες φιλτραρίσματος κατά τις εργάσιμες ώρες και αυτόματα να απενεργοποιήσουμε τους κανόνες μετά τις 17:00. Για τη δημιουργεία κανόνων με βάση το χρόνο πατάμε στη καρτέλα times και στη συνέχεια πατάμε το κουμπί +. Σε οποιοδίποτε Group ACL μετά μπορούμε να ενεργοποιήσουμε το schedule που δημιουργήσαμε.

Εμπορικές λύσεις για φιλτράρισμα web μπορεί να είναι πολύ ακριβές και δύσκολο να ρυθμιστούν. Το pfsense και το πακέτο SquidGuard είναι απολύτος ελεύθερα και πολύ ισχυρά όπως είδαμε (ειδικά με τα regular expression). Το SquidGuard προσφέρει και άλλες δυνατότητες που δεν καλύπτονται στην εργασία αυτή απλά έγινε μόνο μια αναφορά με ένα παράδειγμα για να δούμε πόσες λύσεις μπορεί να μας προσφέρει το pfsense.

10.4.4 TRANSPARENT PROXY SERVER

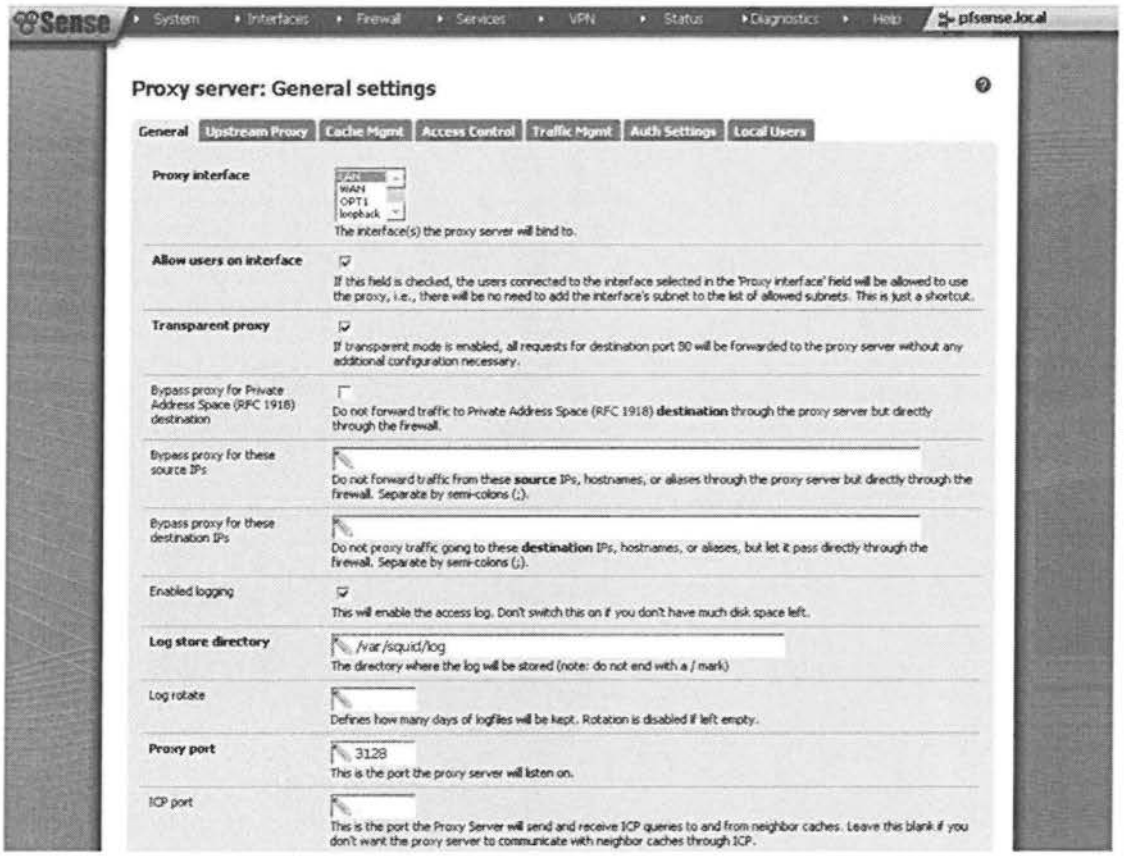
Όπως είδαμε στο κεφάλαιο 6 οι proxy server ενεργούν σαν διακομιστές μεσολάβησης για πελάτες σε ένα δίκτυο που ζητάνε πόρους από έναν εξωτερικό διακομιστή. Ο ποιο ευρέως χρησιμοποιούμενος proxy διακομιστής είναι ο web proxy. Οι διακομιστές proxy βελτιώνουν πολύ την ταχύτητα του Ιντερνετ με τη διαδικασία του caching ή το φιλτράρισμα της κίνησης.

Η διαφορά του παραδοσιακού proxy server με τον transparent proxy είναι ότι οι transparent proxies δρομολογούν τη κίνηση των χρηστών απευθείας στο proxy server αυτόματα, σε αντίθεση με τους παραδοσιακούς proxy server που χρειάζονται επιπλέον ρυθμίσεις στα συστήματα των χρηστών (συνήθως στον web browser).

Το πρώτο πράγμα που κάνουμε είναι να εγκαταστήσουμε το πακέτο από το διαχειριστή πακέτων του pfsense (Squid) στην σελίδα System->packages. Η διαδικασία είναι απλή και θέλει μόλις λίγα λεπτά να ολοκληρωθεί.

Μόλις ολοκληρωθεί η εργασία κάτω από την καρτέλα Services δημιουργείται ένα καινούργιο μενού με το όνομα Proxy Server. Κάνουμε κλικ στο μενού αυτό για να πάμε στις

ρυθμίσεις. Η πρώτη επιλογή που κάνουμε είναι να θέσουμε τη διεπαφή του proxy που είναι η τοπική μας επαφή και τσεκάρουμε την επιλογή "Enable transparent proxy" Εικόνα 10-48. Στη συνέχεια αποθηκεύουμε τις ρυθμίσεις για να ξεκινήσει η proxy υπηρεσία στο pfsense.



Εικόνα 10-48: General proxy server configuration

Σε αυτό το στάδιο έχουμε ένα trasparent proxy server που τρέχει στο pfsense. Δεν χρειάζονται καθόλου αλλαγές στις ρυθμίσεις των χρηστών μας. Κάθε χρήστης που ζητάει μια ιστοσελίδα θα πηγαίνει μέσω του proxy που μόλις φτιάξαμε.

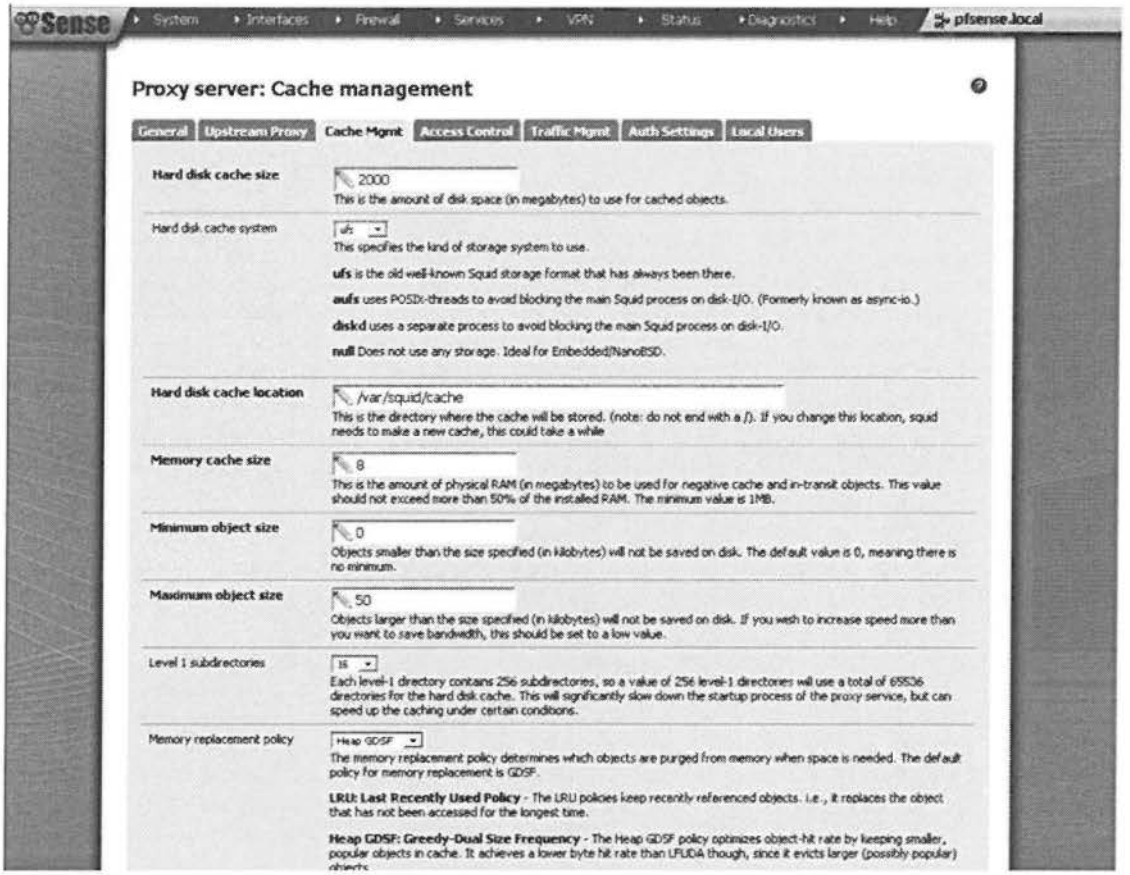
Στη καρτέλα traffic management μπορούμε να ορίσουμε και να περιορίσουμε το εύρος ζώνης που θα χρησιμοποιείται. Μπορούμε να ορίσουμε ένα μέγιστο download και upload μέγεθος όπου θα περιορίσει κινήσεις πάνω από αυτά τα συγκεκριμένα μεγέθοι.

Στην καρτέλα cache υπάρχουν διάφορες επιλογές όπου μπορούμε να τροποποιήσουμε ώστε να βελτιώσουμε την επίδοση στο περιβάλλον μας. Αν βέβαια το pfsense τρέχει σε έναν υπολογιστή που δεν έχει πολύ χώρο HDD και RAM θα πρέπει να είμαστε προσεκτικοί να μην υπερβούμε πολύ τις τιμές. Παρακάτω θα δούμε μερικές επιλογές (Εικόνα 10-49):

- **Hard disk cache size:** Μέγιστος χώρος στο δίσκο όπου το squiz θα χρησιμοποιεί για τη προσωρινή αποθήκευση αντικειμένων. Αν έχουμε μεγάλο δίσκο μπορούμε να αυξήσουμε αυτή την επιλογή ώστε να αποθηκεύουμε περισσότερα αντικείμενα.
- **Memory cache size:** Αν το pfsense σύστημα μας έχει πολύ μνήμη RAM καλό θα ήταν να αυξήσουμε την επιλογή αυτή. Αντικείμενα που δεν μπορούν να

αποθηκευτούν στη μνήμη RAM καταλείγουν να γίνονται SWAP στο σκληρό δίσκο HDD.

- **Maximum object size:** Η προεπιλεγμένη ρύθμιση είναι 4k που είναι πολύ μικρή καλό θα ήταν να το αλλάξουμε σε 50.
- **Επεξεργασία του /boot/loader.conf.local:** Αυτή η ρύθμιση θα πρέπει να γίνει μέσω SSH. Χρησιμοποιώντας ένα επεξεργαστή κειμένου (vi) προσθέτουμε: ***kern.ipc.nmbclusters="32768"*** στο αρχείο και το αποθηκεύουμε. Κάνουμε επανεκίνηση το pfsense. Αυτή η επιλογή αυξάνει το συνολικό ποσό της μνήμης που χρησιμοποιείται για socket buffers σε 32M.



Εικόνα 10-49: Squid cache management

Το squid έχει το δικό του σύστημα για να διαγράφει παλιά αντικείμενα. Σε περίπτωση που θέλουμε να διαγράψουμε χειροκίνητα τη cache του proxy μας θα πρέπει να συνδεθούμε με SSH στο pfsense και να τρέξουμε τις παρακάτω εντολές:

squid -k shutdown *// Τερματισμός υπηρεσίας*

*rm -fr /var/squid/cache/** *// διαγραφή περιεχομένων στο κατάλογο /var/squid/cache/*

squid –z

/usr/local/sbin/squid -D

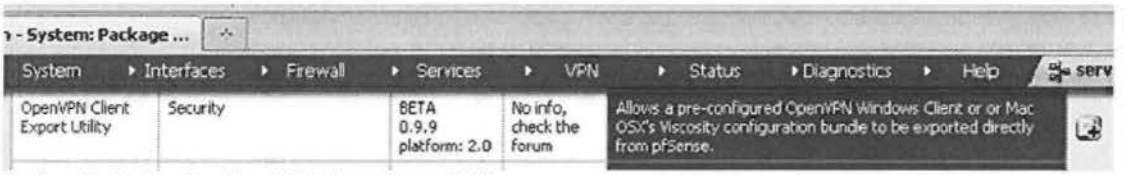
Τέλος με το Lightsquid μπορούμε να παίρνουμε αναφορές όπου παίρνουμε πληροφορίες σχετικά με ποιες ιστοσελίδες επισκέφτηκαν από τους χρήστες και ποια IP και ποια ώρα την επισκέφτηκε. Το lightweight μπορεί να εγκατασταθεί από το διαχειριστή πακέτων. Μετά

την εγκατάσταση δημιουργείται ένα καινούργιο μενού κάτω από το Status (Status -> Proxy Report).

10.4.5 OPENVPN ΑΠΟΜΑΚΡΥΣΜΕΝΗ ΠΡΟΣΒΑΣΗ

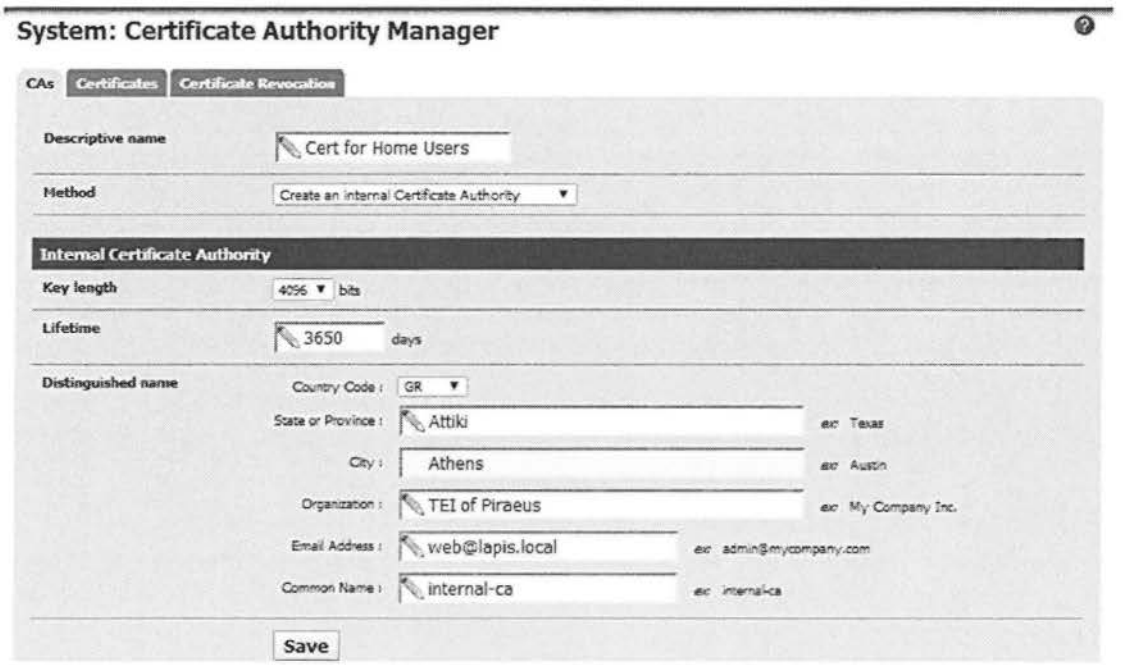
Παρακάτω θα δούμε πως να ενεργοποιήσουμε την δυνατότητα απομακρυσμένης πρόσβασης μέσω ενός VPN δικτύου (TLS και πιστοποίηση χρηστών) στο pfsense. Το OpenVPN είναι ένα VPN λογισμικό όπου ενώνει δύο απομακρυσμένα δίκτυα χρησιμοποιώντας τα πρωτόκολλα SSL/TLS.

Αρχικά από το μενού System -> Packages κάνουμε εγκατάσταση το πακέτο OpenVPN Client Export Utility το οποίο μας εξάγει τις ρυθμίσεις για το κάθε χρήστη που θα δημιουργούμε (Εικόνα 10-51). Το εγκαθιστούμε κλικάροντας στο + κουμπι δεξιά.



Εικόνα 10-50: Πακέτο OpenVPN Client Export Utility

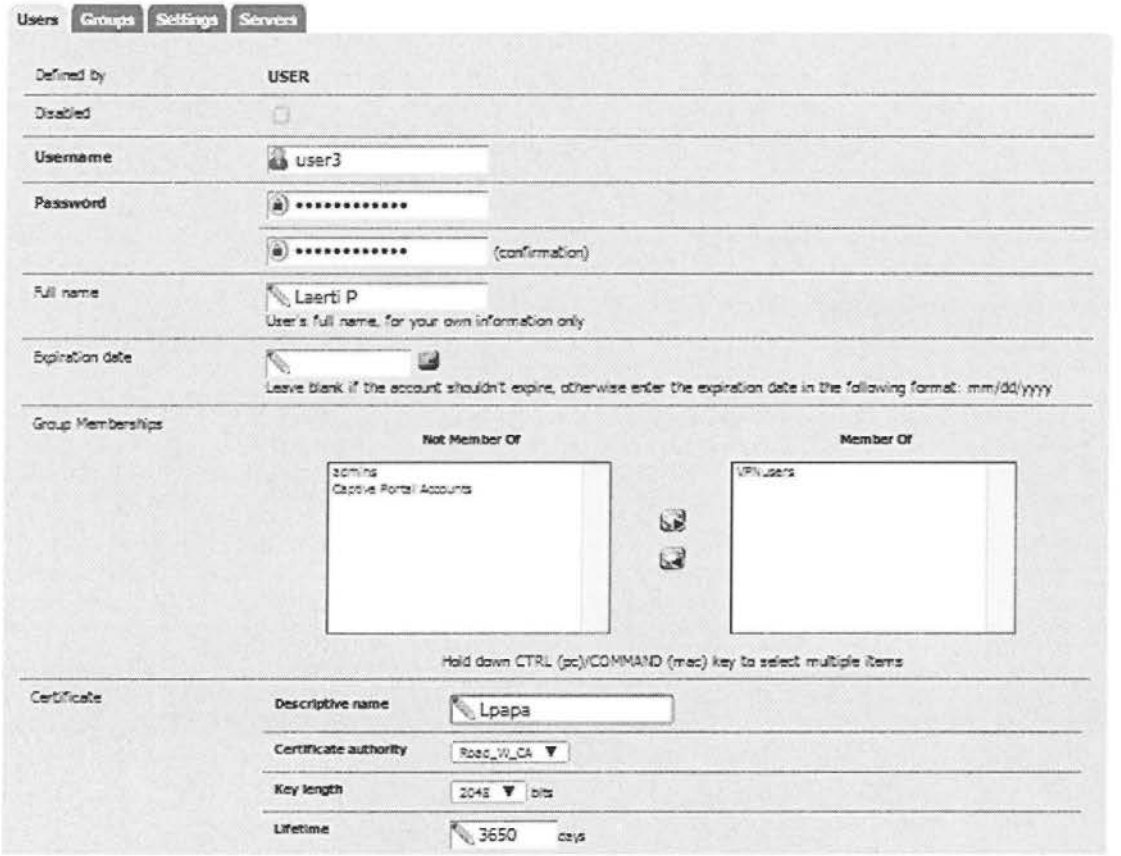
Από το μενού System -> Cert Manager πηγαίνουμε για να δημιουργήσουμε τα πιστοποιητικά. Στη καρτέλα CAs πατάμε το κουμπί σύν (+) για να δημιουργήσουμε ένα νέο πιστοποιητικό. Συμπληρώνουμε τα πεδία και πατάμε save (Εικόνα 10-51). Είναι σημαντικό το Common Name να μην περιέχει κενά.



Εικόνα 10-51: Δημιουργία ενός καινούργιου πιστοποιητικού

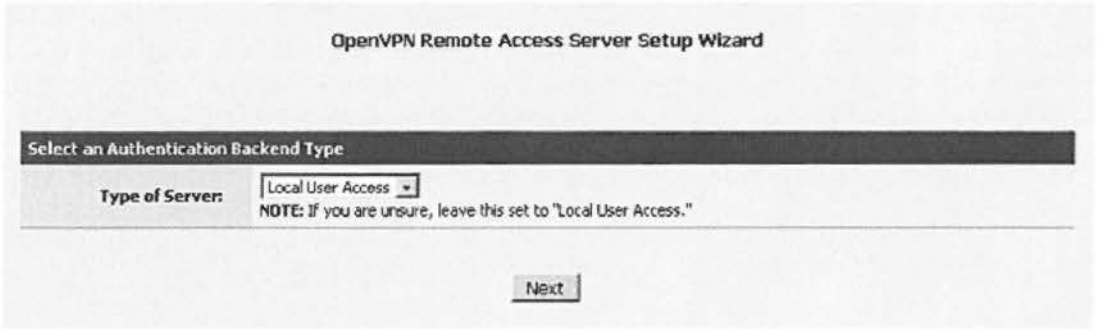
Από το μενού System -> User Manager δημιουργούμε ένα καινούργιο Group για παράδειγμα VPNusers και στη συνέχεια δημιουργούμε τους λογαριασμούς χρηστών που

θέλουμε. Προσέχουμε όταν δημιουργούμε έναν χρήστη στο πεδίο Certificate να επιλέξουμε το πιστοποιητικό δημιουργήσαμε προηγουμένος (Εικόνα 10-52).



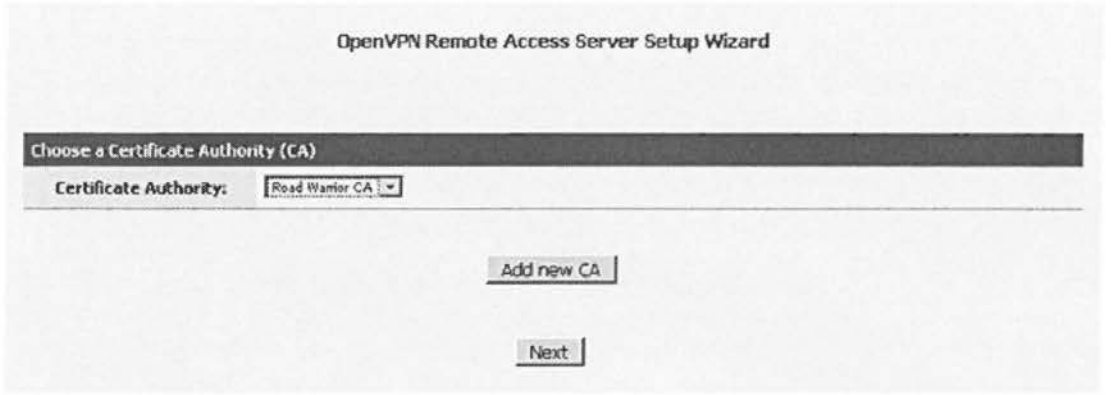
Εικόνα 10-52: Δημιουργία χρήστη και ορισμός certificate για το χρήστη αυτό

Στη συνέχεια δημιουργούμε την VPN υπηρεσία στο pfsense. Από το μενού VPN -> OpenVPN κάνουμε κλικ στη καρτέλα wizard και απιλέγουμε Local User Access για Type Of Server (Εικόνα 10-53).

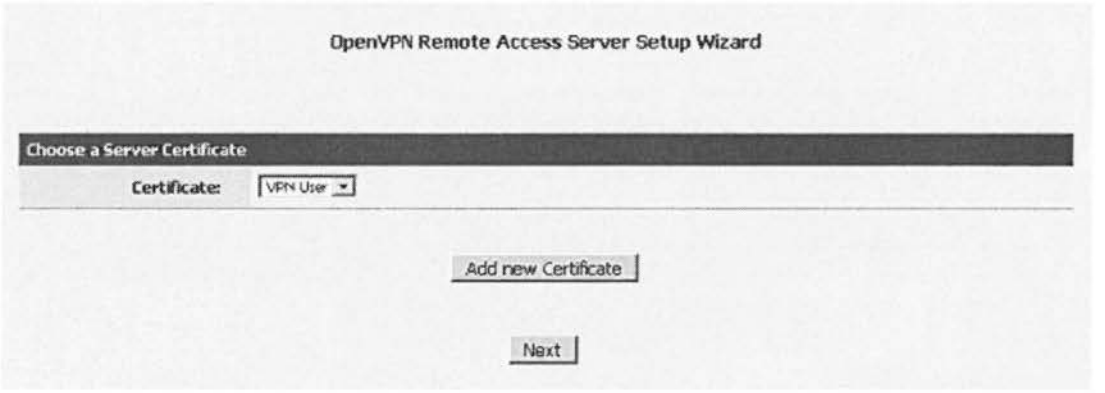


Εικόνα 10-53: Type Of Server

Στη συνέχεια επιλέγουμε το πιστοποιητικό που δημιουργήσαμε (Εικόνα 10-54). Πατάμε Next και για πιστοποιητικό διακομιστή επιλέγουμε Add new Certificate (Εικόνα 10-55).



Εικόνα 10-54: Επιλογή πιστοποιητικού



Εικόνα 10-55: Επιλογή Add new Certificate για server

Δημιουργούμε ένα καινούργιο πιστοποιητικό για το server μας (Εικόνα 10-57), ρυθμίζουμε τις ρυθμίσεις για την κρυπτογράφηση (Εικόνα 10-58), ρυθμίσεις για το tunel (Εικόνα 10-59) και πατάμε next και στη συνέχεια finish. Το pfsense έιναι έτοιμο πλεον να δεχτεί OpenVPN συνδέσεις από clients για πρόσβαση στο δίκτυο που ρυθμίσαμε στην εικόνα 10-58. Για να συνδεθούν οι απομακρυσμένοι χρήστες, πρέπει να κατεβάσουν το OpenVPN λογισμικό από την ιστοσελίδα http://openvpn.net/ να κάνουν την εγκατάσταση και να βάλουν το αρχείο ρυθμίσεων στο φάκελο /%openVPNroot%/config/, όπου openVPNroot είναι ο κύριος κατάλογος του OpenVPN (για Windows c:\program files\openVPN).

Το αρχείο των ριθμήσεων για κάθε χρήστη μπορούμε να το πάρουμε από το pfsense από το μενού: VPN -> OpenVPN και στην καρτέλα Client Export κατεβάζουμε το archive για το λειτουργικό σύστημα του κάθε χρήστη (Εικόνα 10-56).

Client Install Packages		
User	**Certificate Name**	**Export**
user1	VPN_USER_User1	- Standard Configurations: Archive Config Only - Inline Configurations: Android OpenVPN Connect (iOS/Android) Others - Windows Installers: 2.3-x86 2.3-x64 - Mac OSX: Viscosity Bundle
user2	VPN_USER_User2	- Standard Configurations: Archive Config Only - Inline Configurations: Android OpenVPN Connect (iOS/Android) Others - Windows Installers: 2.3-x86 2.3-x64 - Mac OSX: Viscosity Bundle

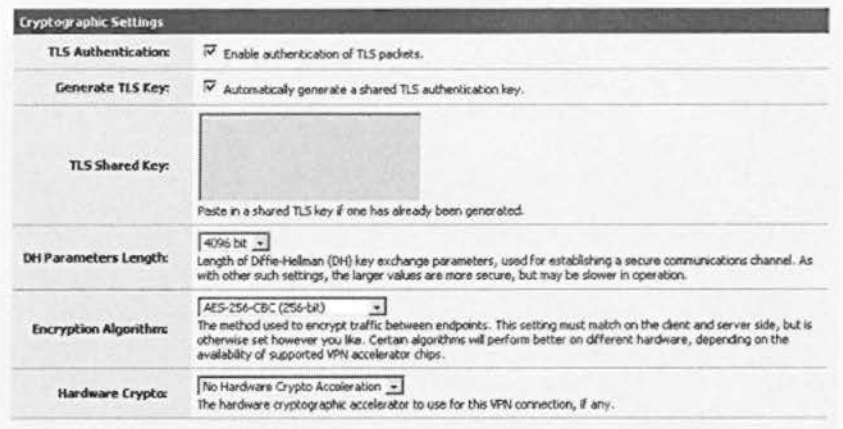
Εικόνα 10-56: Εικόνα 10-56 OpenVPN Ρυθμίσεις για κάθε OpenVPN χρήστη που φτιάξαμε

OpenVPN Remote Access Server Setup Wizard

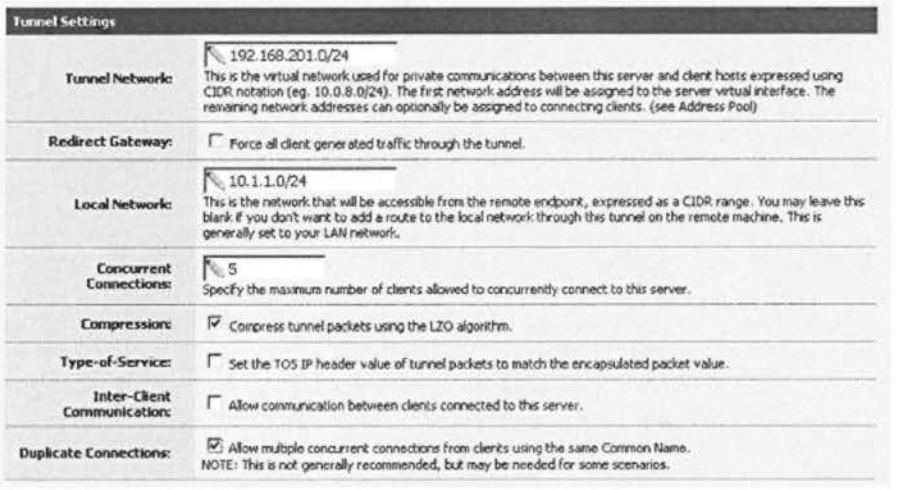
Create a New Server Certificate	
Descriptive name:	Road Warrior Server Ce A name for your reference, to identify this certificate. This is also known as the certificate's "Common Name."
Key length:	4096 bits Size of the key which will be generated. The larger the key, the more security is offers, but larger keys are generally slower to use.
Lifetime:	3650 Lifetime in days. This is commonly set to 3650 (Approximately 10 years.)
Country Code:	US Two-letter ISO country code (e.g. US, AU, CA)
State or Province:	State Full State of Province name, not abbreviated (e.g. Kentucky, Indiana, Ontario).
City:	City City or other Locality name (e.g. Louisville, Indianapolis, Toronto).
Organization:	Organization Organization name, often the Company or Group name.
E-mail:	admin@test.com E-mail address for the Certificate contact. Often the e-mail of the person generating the certificate (i.e. You.)

Create new Certificate

Εικόνα 10-57: Ρυθμίσεις Server Certificate



Εικόνα 10-58: Ρυθμίσεις κρυπτογράφησης

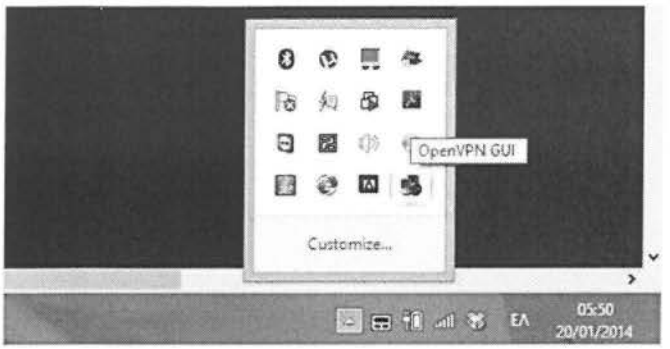


Εικόνα 10-59: Tunnel ρυθμίσεις

Ύστερα από τη ρύθμιση του OpenVPN server μας, το pfsense δημιουργεί αυτόματα στους κανόνες φιλτραρίσματος, έναν κανόνα που να επιτρέπει OpenVPN συνδέσεις στην WAN διεπαφή (UDP πόρτα 1194) και επίσης έναν κανόνα που να επιτρέπει κάθε κίνηση για όλους τους VPN χρήστες.

10.4.5.1 ΕΠΙΚΟΙΝΩΝΙΑ ΧΩΡΙΣ VPN

Στην εικόνα 10-60 φαίνεται ένα τερματικό που έχει εγκατεστημένο το OpenVPN client. Αρχικά δεν έχουμε κάνει σύνδεση και θα δούμε πως η επικοινωνία επιτρέπεται μόνο στη DMZ ζώνη και όχι στο εσωτερικό (Ενότητα 10.4.1). Στην εικόνα 10-61 φαίνεται ο πίνακας δρομολόγησης για το συγκεκριμένο χρήστη.



Εικόνα 10-60: Χρήστης που τρέχει το OpenVPN λογισμικό.

```
IPv4 Route Table
===========================================================================
Active Routes:
Network Destination        Netmask          Gateway       Interface  Metric
          0.0.0.0          0.0.0.0      192.168.1.1    192.168.1.10     25
         10.1.1.0    255.255.255.0  192.168.247.136    192.168.247.1     21
         10.1.1.0  255.255.255.128  192.168.247.136    192.168.247.1     21
        127.0.0.0        255.0.0.0         On-link         127.0.0.1    306
        127.0.0.1  255.255.255.255         On-link         127.0.0.1    306
  127.255.255.255  255.255.255.255         On-link         127.0.0.1    306
      192.168.1.0    255.255.255.0         On-link      192.168.1.10    281
     192.168.1.10  255.255.255.255         On-link      192.168.1.10    281
    192.168.1.255  255.255.255.255         On-link      192.168.1.10    281
     192.168.10.0    255.255.255.0  192.168.247.136    192.168.247.1     21
     192.168.20.0    255.255.255.0  192.168.247.136    192.168.247.1     21
     192.168.99.0    255.255.255.0  192.168.247.136    192.168.247.1     21
    192.168.100.0    255.255.255.0  192.168.247.136    192.168.247.1     21
    192.168.226.0    255.255.255.0         On-link    192.168.226.1    276
    192.168.226.1  255.255.255.255         On-link    192.168.226.1    276
  192.168.226.255  255.255.255.255         On-link    192.168.226.1    276
    192.168.247.0    255.255.255.0         On-link    192.168.247.1    276
    192.168.247.1  255.255.255.255         On-link    192.168.247.1    276
  192.168.247.255  255.255.255.255         On-link    192.168.247.1    276
        224.0.0.0        240.0.0.0         On-link         127.0.0.1    306
        224.0.0.0        240.0.0.0         On-link      192.168.1.10    281
        224.0.0.0        240.0.0.0         On-link    192.168.226.1    276
        224.0.0.0        240.0.0.0         On-link    192.168.247.1    276
  255.255.255.255  255.255.255.255         On-link         127.0.0.1    306
  255.255.255.255  255.255.255.255         On-link      192.168.1.10    281
  255.255.255.255  255.255.255.255         On-link    192.168.226.1    276
  255.255.255.255  255.255.255.255         On-link    192.168.247.1    276
===========================================================================
```

Εικόνα 10-61: Πίνακας δρομολόγησης για το χρήστη

Παρατηρούμε όμως ότι όλα τα ICMP πακέτα μπλοκάρονται και ότι ο χρήστης έχει πρόσβαση μόνο στη DMZ ζώνη του δικτύου μας (HTTP και FTP) αφού οι υπόλοιπες κινήσεις φιλτράρονται από το firewall. Οι παρακάτω εικόνες επαληθεύουν όλα τα παραπάνω.

```
C:\Windows\system32>ping 10.1.1.2

Pinging 10.1.1.2 with 32 bytes of data:
Request timed out.
Request timed out.
Request timed out.
Request timed out.

Ping statistics for 10.1.1.2:
    Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),
```

Εικόνα 10-62: Αποτυχία η εντολή ping στο web server

```
C:\Windows\system32>ping 192.168.99.2

Pinging 192.168.99.2 with 32 bytes of data:
Request timed out.
Request timed out.
Request timed out.
Request timed out.

Ping statistics for 192.168.99.2:
    Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),
```

Εικόνα 10-63: Αποτυχία επικοινωνίας με εσωτερικό DNS.

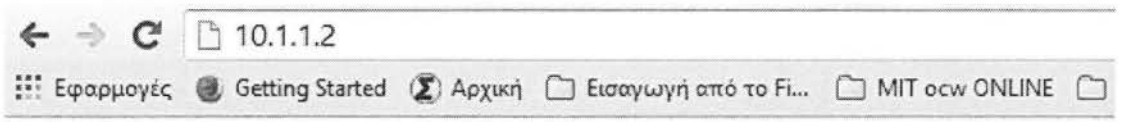
```
C:\Windows\system32>
C:\Windows\system32>ping 192.168.10.1

Pinging 192.168.10.1 with 32 bytes of data:
Request timed out.
Request timed out.
Request timed out.
Request timed out.

Ping statistics for 192.168.10.1:
    Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),

C:\Windows\system32>_
```

Εικόνα 10-64: Αποτυχία επικοινωνίας στο Sales VLAN .

```
C:\Windows\system32>ftp 10.1.1.2
Connected to 10.1.1.2.
220---------- Welcome to Pure-FTPd [privsep] ----------
220-You are user number 1 of 50 allowed.
220-Local time is now 21:35. Server port: 21.
220-IPv6 connections are also welcome on this server.
220 You will be disconnected after 15 minutes of inactivity.
User (10.1.1.2:(none)): _
```

Εικόνα 10-65: Επιτυχείς σύνδεση σε δημόσιο FTP διακομιστή

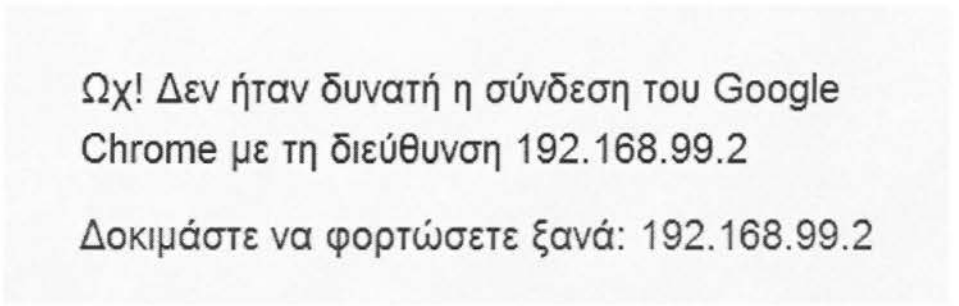
Στην εικόνα 10-66 βλέπουμε ότι μπορούμε να συνδεθούμε στο δημόσιο Web server.



It works!

Εικόνα 10-66: Επιτυχής σύνδεση στο δημόσιο webserver

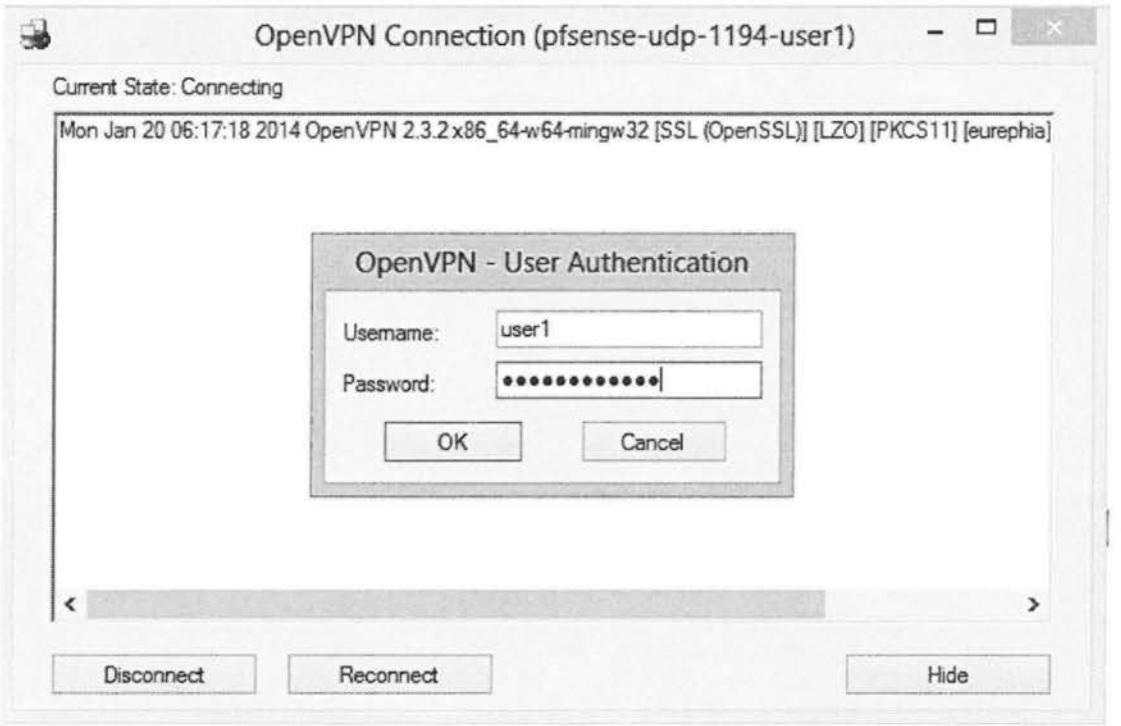
Αλλά όχι στο εσωτερικό webserver.



Εικόνα 10-67: Αποτυχία σύνδεσης με εσωτερικό webserver από τον εξωτερικό δίκτυο

10.4.5.2 ΕΠΙΚΟΙΝΩΝΙΑ ΜΕ VPN

Στην εικόνα 10-68 βλέπουμε το χρήστη user1 να κάνει login μέσω του OpenVPN στο εσωτερικό ιδιωτικό μας δίκτυο. Ύστερα από την επιτυχής σύνδεση δοκιμάζουμε για την επικοινωνία του χρήστη με το υπόλοιπο εσωτερικό δίκτυο.



Εικόνα 10-68: Σύνδεση του χρήστη user1 στο VPN δίκτυο μας

Στην εικόνα 10-69 κάνουμε ping το intranet. Όπως βλέπουμε γίνεται resolve το όνομα στην IP του, επομένως επικοινωνούμε με το εσωτερικό DNS.

```
C:\Windows\system32>ping intranet

Pinging NS2.intranet.lapis.local [192.168.99.2] with 32 bytes of data:
Reply from 192.168.99.2: bytes=32 time=4ms TTL=62
Reply from 192.168.99.2: bytes=32 time=5ms TTL=62
Reply from 192.168.99.2: bytes=32 time=5ms TTL=62
Reply from 192.168.99.2: bytes=32 time=4ms TTL=62

Ping statistics for 192.168.99.2:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 4ms, Maximum = 5ms, Average = 4ms
```

Εικόνα 10-69: Ping το hostname intranet μετά από VPN σύνδεση

Ομοίως στην εικόνα 10-70 κάνουμε Ping το hostname jpapa και έχουμε πλήρη επικοινωνία.

```
C:\Windows\system32>ping jpapa

Pinging jpapa.intranet.lapis.local [192.168.20.20] with 32 bytes of data:
Reply from 192.168.20.20: bytes=32 time=4ms TTL=62
Reply from 192.168.20.20: bytes=32 time=8ms TTL=62
Reply from 192.168.20.20: bytes=32 time=4ms TTL=62
Reply from 192.168.20.20: bytes=32 time=4ms TTL=62

Ping statistics for 192.168.20.20:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 4ms, Maximum = 8ms, Average = 5ms
```

Εικόνα 10-70: Ping το hostname jpapa μετά από VPN σύνδεση

Στην εικόνα 10-71 φαίνεται να έχουμε πρόσβαση στο εσωτερικό web server από το browser του χρήστη.

Εικόνα 10-71: Πρόσβαση στο εσωτερικό web server

ΠΑΡΑΡΤΗΜΑ Α – ΑΡΧΕΙΑ ΡΥΘΜΙΣΕΩΝ ISC DHCP SERVER

/etc/dhcp/dhcpd.conf

```
ddns-updates on;                        #enable global ddns
ddns-update-style interim;
ignore client-updates;
ddns-domainname "intranet.lapis.local.";
ddns-rev-domainname "in-addr.arpa.";

authoritative;                          #primary/official dhcp forl local
key "rndc-key"{
        algorithm hmac-md5;
        secret "SRIoA0q1eEAMZy/wmLaFQw==";
};

log-facility local7;                    #allow debugging
default-lease-time 600;
max-lease-time 7200;
option domain-name "intranet.lapis.local";
option domain-name-servers 192.168.99.2;

        subnet 192.168.10.0 netmask 255.255.255.0{
                pool{
                        range 192.168.10.10 192.168.10.150;
                }
                option routers                  192.168.10.1;
                option domain-name-servers      192.168.99.2;
                option broadcast-address        192.168.10.255;
                option subnet-mask              255.255.255.0;
        }

        subnet 192.168.99.0 netmask 255.255.255.0{

                pool{
                        range 192.168.99.100 192.168.99.150;
                }

                option routers                  192.168.10.1;
                option domain-name-servers      192.168.99.2;
                option broadcast-address        192.168.10.255;
                option subnet-mask              255.255.255.0;
        }

        subnet 192.168.20.0 netmask 255.255.255.0{
                pool{
                        range 192.168.20.20 192.168.20.150;
```

```
        }
        option routers                  192.168.20.1;
        option domain-name-servers      192.168.99.2;
        option broadcast-address        192.168.20.255;
        option subnet-mask              255.255.255.0;
}

subnet 192.168.100.0 netmask 255.255.255.0{
        pool{
                range 192.168.100.100 192.168.100.199;
        }
        option routers                  192.168.100.1;
        option domain-name-servers      192.168.99.2;
        option broadcast-address        192.168.100.255;
        option subnet-mask              255.255.255.0;
}
```

ΠΑΡΑΡΤΗΜΑ Β – ΑΡΧΕΙΑ ΡΥΘΜΙΣΕΩΝ ΔΡΟΜΟΛΟΓΗΤΗ: HR-R1/BR-R1

HR-R1: /opt/vyatta/etc/config/config.boot

```
firewall {
    all-ping enable
    broadcast-ping disable
    ipv6-receive-redirects disable
    ipv6-src-route disable
    ip-src-route disable
    log-martians enable

    name FWTEST {
        default-action drop
        rule 1 {
            action drop
            destination {
                address 192.168.100.0/24
            }
            source {
                address 192.168.10.0/24
            }
        }

        rule 2 {
            action accept
            destination {
                address 0.0.0.0/0
            }
            source {
                address 0.0.0.0/0
            }
        }
    }
    receive-redirects disable
    send-redirects enable
    source-validation disable
    syn-cookies enable
}

interfaces {
    ethernet eth0 {
        address 10.1.1.130/30
        description Connection-To-pfSense
        duplex auto
        hw-id 00:0c:29:22:cc:44
        smp_affinity auto
        speed auto
    }

    ethernet eth1 {
```

```
        description VLAN-Trunk
        duplex auto
        hw-id 00:0c:29:22:cc:4e
        smp_affinity auto
        speed auto
        vif 10 {
            address 192.168.10.1/24
            description "VLAN10 Sales"
            firewall {
                in {
                    name FWTEST
                }
            }
        }
        vif 20 {
            address 192.168.20.1/24
            description "VLAN10 Marketing"
        }
        vif 99 {
            address 192.168.99.1/24
            description "VLAN99 Internal Servers"
        }
        vif 100 {
            address 192.168.0.1/24
            description "VLAN100 ADMIN NET"
        }
    }

    ethernet eth2 {
        address 10.1.1.133/30
        description "Connection to BR-R1"
        duplex auto
        hw-id 00:0c:29:22:cc:58
        smp_affinity auto
        speed auto
    }

    loopback lo {
    }
}

protocols {
    rip {
        default-information {
        }
        network 10.1.1.128/30
        network 10.1.1.132/30
        passive-interface eth1
        redistribute {
            connected {
            }
```

```
        }
    }

    static {
        route 0.0.0.0/0 {
            next-hop 10.1.1.129 {
            }
        }
    }
}

service {
    dhcp-relay {
        interface eth2
        interface eth1.10
        interface eth1.20
        interface eth1.99
        relay-options {
            hop-count 10
            max-size 576
            relay-agents-packets discard
        }
        server 192.168.99.2
    }
    ssh {
        allow-root
        port 22
    }
}

system {
    config-management {
        commit-revisions 20
    }
    console {
        device ttyS0 {
            speed 9600
        }
    }


    domain-name intranet.lapis.local
    domain-search {
    }

    host-name HQ-R1
    login {
        user vyatta {
            authentication {
                encrypted-password $1$WKamHjWV$fVSCZU34Fr54OcTtS5oCk/
            }
```

```
        level admin
      }
    }

    name-server 192.168.99.2
    ntp {
      server 0.vyatta.pool.ntp.org {
      }
      server 1.vyatta.pool.ntp.org {
      }
      server 2.vyatta.pool.ntp.org {
      }
    }

    package {
      auto-sync 1
      repository community {
        components main
        distribution stable
        password ""
        url http://packages.vyatta.com/vyatta
        username ""
      }
    }

    syslog {
      global {
        facility all {
          level notice
        }
        facility protocols {
          level debug
        }
      }
    }
    time-zone GMT
}
```

BR-R1: /opt/vyatta/etc/config/config.boot

```
interfaces {
    ethernet eth0 {
      address 10.1.1.134/30
      duplex auto
      hw-id 00:0c:29:01:f1:9e
      smp_affinity auto
      speed auto
    }
    ethernet eth1 {
      address 192.168.100.1/24
      duplex auto
```

```
        hw-id 00:0c:29:01:f1:a8
        smp_affinity auto
        speed auto
    }
    loopback lo {
    }
}

protocols {
    rip {
        network 10.1.1.132/30
        redistribute {
            connected {
            }
        }
    }
    static {
        route 0.0.0.0/0 {
            next-hop 10.1.1.133 {
            }
        }
    }
}

service {
    dhcp-relay {
        interface eth0
        interface eth1
        relay-options {
            hop-count 10
            max-size 576
            relay-agents-packets discard
        }
        server 192.168.99.2
    }
    ssh {
        allow-root
        port 22
    }
}

system {
    config-management {
        commit-revisions 20
    }
    console {
        device ttyS0 {
            speed 9600
        }
    }
    domain-name intranet.lapis.local
```

```
    host-name BR-R1
    login {
        user hqadmin {
            authentication {
                encrypted-password $1$c4BuCJWc$yauxTvWJP.N3SVjuiuLwm.
                plaintext-password ""
            }
            group backup
            level admin
        }
        user vyatta {
            authentication {
                encrypted-password $1$LDlFkIjf$zciu4ejkmLXRLITw1WBXd1
            }
            level admin
        }
    }

    name-server 192.168.99.2
    ntp {
        server 0.vyatta.pool.ntp.org {
        }
        server 1.vyatta.pool.ntp.org {
        }
        server 2.vyatta.pool.ntp.org {
        }
    }

    package {
        auto-sync 1
        repository community {
            components main
            distribution stable
            password ""
            url http://packages.vyatta.com/vyatta
            username ""
        }
    }

    syslog {
        global {
            facility all {
                level notice
            }
            facility protocols {
                level debug
            }
        }
    }
    time-zone GMT
}
```

ΠΑΡΑΡΤΗΜΑ Γ – ΑΡΧΕΙΑ ΡΥΘΜΙΣΕΩΝ NAMED DNS SERVER

/etc/bind/named.conf

```
options {
        directory "/var/cache/bind";
        forwarders {
                8.8.8.8;
                8.8.4.4;
        };
        allow-query{
                192.168.10.0/24;
                192.168.20.0/24;
                192.168.99.0/24;
                192.168.100.0/24;
                10.1.1.128/30;
                10.1.1.132/30;
        };
        auth-nxdomain no;    # conform to RFC1035
        listen-on-v6 { any; };
};

//key: /usr/sbin/confgen -a
key "rndc-key" {
        algorithm hmac-md5;
        secret "SRIoA0q1eEAMZy/wmLaFQw==";
};

// prime the server with knowledge of the root servers
zone "." {
        type hint;
        file "/etc/bind/db.root";
};

zone "intranet.lapis.local" IN {
        type master;
        file "/var/lib/bind/intranet.lapis.local.hosts";
        allow-update { key rndc-key; };
};

zone "10.168.192.in-addr.arpa" IN {
        type master;
        file "/var/lib/bind/10.168.192.rev";
        allow-update { key rndc-key; };
};

zone "20.168.192.in-addr.arpa" IN {
```

```
        type master;
        file "/var/lib/bind/20.168.192.rev";
        allow-update { key rndc-key; };
};

zone "99.168.192.in-addr.arpa" IN {
        type master;
        file "/var/lib/bind/99.168.192.rev";
        allow-update { key rndc-key; };
};

zone "100.168.192.in-addr.arpa" IN {
        type master;
        file "var/lib/bind/100.168.192.rev";
        allow-update { key rndc-key; };
};

zone "facebook.com" IN {
        type master;
        file "/var/lib/bind/block-zones";
};

// be authoritative for the localhost forward and reverse zones, and for
// broadcast zones as per RFC 1912
zone "localhost" {
        type master;
        file "/etc/bind/db.local";
};
zone "127.in-addr.arpa" {
        type master;
        file "/etc/bind/db.127";
};
zone "0.in-addr.arpa" {
        type master;
        file "/etc/bind/db.0";
};
zone "255.in-addr.arpa" {
        type master;
        file "/etc/bind/db.255";
};
```

Αρχείο /var/lib/bind/intranet.lapis.local.hosts

```
$ORIGIN .
$TTL 907200             ; 1 week 3 days 12 hours
intranet.lapis.local    IN SOA ns2.intranet.lapis.local. webmaster.intranet.lapis.local.
(
                                37              ; serial
                                10800           ; refresh (3 hours)
                                3600            ; retry (1 hour)
                                604800          ; expire (1 week)
                                38400           ; minimum (10 hours 40 minutes)
                                )

                        NS      ns2.intranet.lapis.local.

$ORIGIN intranet.lapis.local.
; A records
BR-R1-134               IN                      A       10.1.1.134
firewall                IN                      A       10.1.1.129
HQ-R1-130               IN                      A       10.1.1.130
HQ-R1-133               IN                      A       10.1.1.133
localhost               IN                      A       127.0.0.1
ns2                     IN              A       192.168.99.2

;
; Aliases
;
server01                IN              CNAME   ns2
server02                IN              CNAME   NS2
HQ-R1                   IN              CNAME   HQ-R1-130
www                     IN              CNAME   NS2
BR-R1                   IN              CNAME   BR-R1-134
intranet                IN              CNAME   NS2

;
; DDNS RECORDS. Δημιουργούνται αυτόματα από DHCP
;

$TTL 300        ; 5 minutes
jpapa                   IN                      A       192.168.20.20
                        IN                      TXT
        "0016dfc14d8b7d5b757c07002454a9376d"
$TTL 300        ; 5 minutes
lpapa                   IN                      A       192.168.10.102
                        IN                      TXT
        "0036997dab3659be025b8dfb8ab474c364"
```

Αρχείο /var/lib/bind/10.168.192.rev

```
$ORIGIN .
$TTL 907200     ; 1 week 3 days 12 hours

10.168.192.in-addr.arpa          IN SOA  ns2.intranet.lapis.local.
webmaster.intranet.lapis.local. (
                                 9        ; serial
                                 10800    ; refresh (3 hours)
                                 3600     ; retry (1 hour)
                                 604800   ; expire (1 week)
                                 38400    ; minimum (10 hours 40 minutes)
                                 )
                         NS      ns2.intranet.lapis.local.

$ORIGIN 10.168.192.in-addr.arpa.

;
;
;

$TTL 300        ; 5 minutes
100             IN              PTR     lpapa.intranet.lapis.local.
102             IN              PTR     lpapa.intranet.lapis.local.
```

Αρχείο /var/lib/bind/20.168.192.rev

```
$ORIGIN .
$TTL 907200     ; 1 week 3 days 12 hours
20.168.192.in-addr.arpa          IN SOA  ns2.intranet.lapis.local.
webmaster.intranet.lapis.local. (
                                 4        ; serial
                                 10800    ; refresh (3 hours)
                                 3600     ; retry (1 hour)
                                 604800   ; expire (1 week)
                                 38400    ; minimum (10 hours 40 minutes)
                                 )
                         NS      ns2.intranet.lapis.local.

$ORIGIN 20.168.192.in-addr.arpa.
$TTL 300        ; 5 minutes
20                      IN              PTR     jpapa.intranet.lapis.local.
```

Αρχείο /var/lib/bind/99.168.192.rev

```
$TTL 907200             ;1 week 3 days 12 hours
@       IN SOA ns2.intranet.lapis.local. webmaster.intranet.lapis.local. (
        1               ;serial
        10800           ;  refresh
        3600            ;retry
        604800          ;expire
        38400           ;minimum
        )

@       IN      NS      ns2.intranet.lapis.local.
2       IN      PTR     ns2.intranet.lapis.local.
```

Αρχείο /var/lib/bind/100.168.192.rev

```
$ORIGIN .
$TTL 907200     ; 1 week 3 days 12 hours

10.168.192.in-addr.arpa         IN SOA  ns2.intranet.lapis.local.
webmaster.intranet.lapis.local. (
                                9       ; serial
                                10800     ; refresh (3 hours)
                                3600      ; retry (1 hour)
                                604800    ; expire (1 week)
                                38400     ; minimum (10 hours 40 minutes)
                                )
                        NS      ns2.intranet.lapis.local.

$ORIGIN 100.168.192.in-addr.arpa.

;
;
;

$TTL 300        ; 5 minutes
101             IN              PTR     bruser.intranet.lapis.local.
```

Αρχείο /var/lib/bind/block-zones

```
$TTL    24h
@       IN      SOA     ns2.intranet.lapis.local. webmaster.intranet.lapis.local.  (
        2003052800
        86400
        300
```

`

```
        604800
        3600
)
@       IN      NS      ns2.intranet.lapis.local.
@       IN      NS      127.0.0.1
.facebook.com IN        A       127.0.0.1
```

ΒΙΒΛΙΟΓΡΑΦΙΑ

Aitchison, R. (n.d.). *Pro DNS and BIND 10.* Apress.

Anthony Bruno, S. J. (2009). *Official Cert Guide CCDA 640-864.* Cisco Press.

Barrie Dempster, J. E.-L. (2006). *Configuring IPCop Firewalls.* Packt Publishing.

Benvenuti, C. (December 2005). *Understanding Linux Network Internals.* O'Reilly.

Christopher M. Buechler, J. P. (2009). *pfSense: The Definitive Guide.*

Ciampa, M. (2012). *Security+ Guide to network security fundamentals.* Information Security Professionals.

Consortium, I. S. (2001). *BIND 9 Administrator Reference Manual.* Internet Software Consortium.

Davies, G. (2004). *Designing and Developing Scalable IP Networks.* John Willey & Sons, Ltd.

Eric Cole, R. K. (n.d.). *Network Security Bible.* Wiley Publishing.

Forbes Guthrie, S. L. (2011). *VMware vSphere design 2nd Edition.* Cybex.

Gheorghe, L. (2006). *Linux Firewalls and QoS.* Packt Publishing.

Held, G. (2003). *Ethernet Networks: Design, Implementation, Operation, Management.* John Wiley & Sons, Ltd.

Holden, G. (2004). *Guide to firewalls and network security: Intrusion detection and VPNs.* Information Security Professionals.

Hong, B. J. (March 27, 2008).Bu*ilding a Server with FreeBSD 7.* No Starch.

Kozierok, C. M. (September 20, 2005). *The TCP/IP Guide.*

Lammle, T. (2011). *Cisco Certified Network Associate.* Cybex.

Lehey, G. (February 2003). *The Complete FreeBSD.* O'Reilly.

Libor Dostálek, A. K. (2006). *DNS in Action, A detailed and practical guide to DNS implementation, configuration, and administration.* Packt Publishing.

Liu, C. (October 2002). *DNS & BIND Cookbook.* O'Reilly.

Lowe, S. (2011). *Mastering VMware vSphere 5.* Cybex.

Lucas, M. (2002). *Absolute BSD—The Ultimate Guide to FreeBSD.* NO STARCH PRESS.

Mancill, T. (2002). *Linux Routers.* Prentice Hall PTR.

Neil Matthew, R. S. (2008). *Beginning Linux Programming 4th Edition.* Willy Publishing, Inc.

Paco Hope, Y. K. (March 2005). *Mastering FreeBSD and OpenBSD Security.* O'Reilly.

Paul Albitz, C. L. (May 2006). *DNS and BIND, 5th Edition* . O'Reilly.

Sosinsky, B. (2009). *Networking Bible.* Willey Publishing, Inc.

Stephen Figgins, R. L. (July 2005). *Linux in a Nutshell, 5th Edition* . O'Reilly.

Steve Suehring, R. Z. (September 14, 2005). *Linux Firewalls, Third Edition* . Sams Publishing.

Stewart, J. M. (2011). *Network Security, Firewalls, and VPNs.* Jones & Barlett Learning.

Tony Bautts, T. D. (February 2005). *Linux Network Administrator's Guide, 3rd Edition.* O'Reilly.