ΠΑΝΕΠΙΣΤΗΜΙΟ ΜΑΚΕΔΟΝΙΑΣ

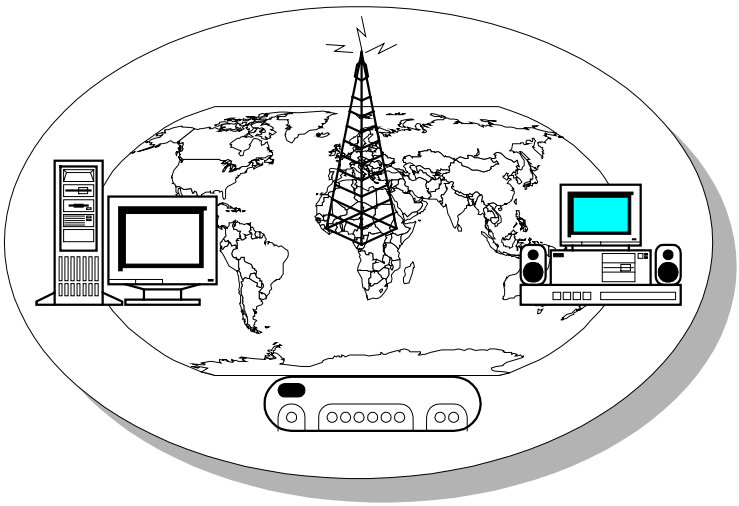
ΔΙΑΤΜΗΜΑΤΙΚΟ ΠΡΟΓΡΑΜΜΑ ΜΕΤΑΠΤΥΧΙΑΚΩΝ ΣΠΟΥΔΩΝ ΣΤΑ ΠΛΗΡΟΦΟΡΙΑΚΑ ΣΥΣΤΗΜΑΤΑ

Μάθημα: **Τεχνολογίες Τηλεπικοινωνιών και Δικτύων**

Τίτλος: **"*Architectures, Types and Management for Web Proxies*"**

Υπεύθυνος Καθηγητής: **Αναστάσιος Οικονομίδης**

Εισηγητής: **Δημήτριος Φωλίνας**



Θεσσαλονίκη, Ιανουάριος 2000

Περιεχόμενα

Σελίδα

Περιεχόμενα (*συνέχεια*) *Σελίδα*

Abstract: As organizations and companies take advantage of the opportunities presented when Internet access is available, system managers are faced with higher bandwidth demand, increased security headaches and a heavier system management load. Proxy servers can take advantage of the promise of the Internet without the associated pain and troubles. They provide a variety of essential functions like web-caching, packet filtering, access control, logging and monitoring. A proxy server can improve the network's performance and immunity, reduce network's response time and Internet access cost and provide a centralized network administration. This paper gives an overview of proxies. It focuses on general properties, services, basic types and different architectures, advantages and disadvantages of every kind of them. It also examines the critical area of management and how to optimise proxies in order to get full services, maximum performance and benefits. Finally, the "National Proxy Cached Schema" of EDET is analyzed as an example.

Περίληψη: Οι εταιρίες και οι διάφοροι οργανισμοί εκμεταλλευόμενοι τις δυνατότητες που πρόσφεραν οι τεχνολογίες των τηλεπικοινωνιών και της πληροφορικής γενικότερα, δημιούργησαν πρώτα εσωτερικά δίκτυα που αργότερα συνδέθηκαν με το Διαδίκτυο. Με την είσοδό τους σ' αυτό οι διαχειριστές των συστημάτων βρέθηκαν μπροστά σε ένα περιβάλλον που τους έδινε τεράστιες δυνατότητες και ευκαιρίες, αλλά τους δημιουργούσε και ορισμένα προβλήματα όπως η απαίτηση για μεγαλύτερο διαθέσιμο εύρος, η έλλειψη ασφαλείας και συγκεντρωτικού ελέγχου των εσωτερικών δικτύων. Ο πληρεξούσιος εξυπηρετητής (proxy server) είναι ένας εξυπηρετητής που ενεργεί σαν ο ενδιάμεσος μεταξύ των χρηστών ενός εσωτερικού δικτύου και του Διαδικτύου, εξασφαλίζοντας για το περιβάλλον αυτό ασφάλεια, συγκεντρωτικό έλεγχο, διαχείριση και υπηρεσίες επαναποθήκευσης και φιλτραρίσματος.

Η εργασία αυτή θα επικεντρωθεί στις βασικές λειτουργίες, στις δυνατότητες που μπορεί να παρέχει ο πληρεξούσιος εξυπηρετητής, στην ανάλυση των τύπων και αρχιτεκτονικών του, τα πλεονεκτήματα και μειονεκτήματά τους και στο σχεδιασμό, την υλοποίηση και τη διαχείριση ενός ή πολλών πληρεξούσιων εξυπηρετητών, ώστε να λαμβάνουμε τα βέλτιστα αποτελέσματα. Τέλος θα δοθεί ένα παράδειγμα, το σύστημα των πληρεξούσιων εξυπηρετητών του ΕΔΕΤ των ελληνικών ΑΕΙ και ΤΕΙ και θα εξετασθεί η δυνατότητα σύνδεσης σ' αυτό ή εγκατάστασης αυτόνομων στο Πανεπιστήμιο Μακεδονίας.

Α. ΕΙΣΑΓΩΓΗ

Η παγκοσμιοποιημένη οικονομία αναπτύχθηκε με βάση τη τεχνολογική δυνατότητα της άμεσης μεταφοράς δεδομένων-πληροφοριών, σε κάθε σημείο του πλανήτη. Οι εταιρίες εκμεταλλευόμενες τις τεχνολογίες της πληροφορικής και των τηλεπικοινωνιών και τη τρομακτική άνθηση του Διαδικτύου, δημιούργησαν εταιρικά δίκτυα, τα οποία αργότερα συνδέθηκαν με το Διαδίκτυο και εγκατέστησαν εξυπηρετητές (servers), όπου περιέχονταν πληροφορίες για την εταιρία, τα προϊόντα της και παρείχαν ακόμα και τη δυνατότητα των εμπορικών συναλλαγών των πελατών, μ' αυτήν.

Ταυτόχρονα με την είσοδο στο Διαδίκτυο, εμφανίστηκαν διάφοροι εξωτερικοί κίνδυνοι, όπως άτομα που επιθυμούσαν να "εισβάλλουν" στο εσωτερικό δίκτυο (hackers), με σκοπό να λάβουν γνώση των πληροφοριών -που περιέχονταν στους εξυπηρετητές και γενικά στο δίκτυο των εταιριών ή οργανισμών- ή να μεταβάλλουν το περιεχόμενό τους. Το πρόβλημα αυτό σε συνδυασμό με τη μεγάλη χρονική καθυστέρηση απάντησης σε κάθε αίτηση (request) του πελάτη (client) από τον εξυπηρετητή, την ανυπαρξία λεπτομερούς ελέγχου του περιεχομένου (filtering) και γενικά ελέγχου της κυκλοφορίας της πληροφορίας, δημιούργησαν την ανάγκη του σχεδιασμού και κατασκευής κατάλληλου λογικού και λογισμικού. Έτσι σχεδιάστηκαν οι πληρεξούσιοι εξυπηρετητές (proxy servers) και τα φράγματα (firewalls).

Η διαφορά του φράγματος και του πληρεξούσιου εξυπηρετητή είναι δύσκολο να προσδιοριστεί με ακρίβεια, αλλά μπορούμε να πούμε ότι: ο όρος φράγμα, δεν αναφέρεται σε ένα συγκεκριμένο τμήμα υλικού ή λογισμικού, αλλά μπορεί να θεωρηθεί σαν ένας συνδυασμός αυτών των δύο που τοποθετείται μεταξύ του εσωτερικού δικτύου και του Διαδικτύου, παρακολουθεί και ελέγχει την εισερχόμενη και εξερχόμενη "πληροφορία" και ανάλογα επιτρέπει ή όχι τη κίνησή της μέσω αυτού.

Οι πληρεξούσιοι εξυπηρετητές από την άλλη, μπορούν να θεωρηθούν γενικά ως οι υπηρεσίες (services), που εκτελούνται-"τρέχουν" πάνω σε ένα φράγμα και παρέχουν όχι μόνο ασφάλεια αλλά και ένα καλύτερο και πιο λεπτομερή έλεγχο στη κυκλοφορία του συστήματος και πολλές άλλες λειτουργίες.

Στην αρχή της δεκαετίας που διανύουμε, που συμπίπτει χρονικά με την αρχή της διάδοσης του παγκόσμιου ιστού (World Wide Web), χρησιμοποιήθηκε ο όρος "θύρα" (gateway) που παραδοσιακά απέδιδε τις συσκευές εκείνες που προωθούν τα πακέτα μεταξύ των δικτύων.

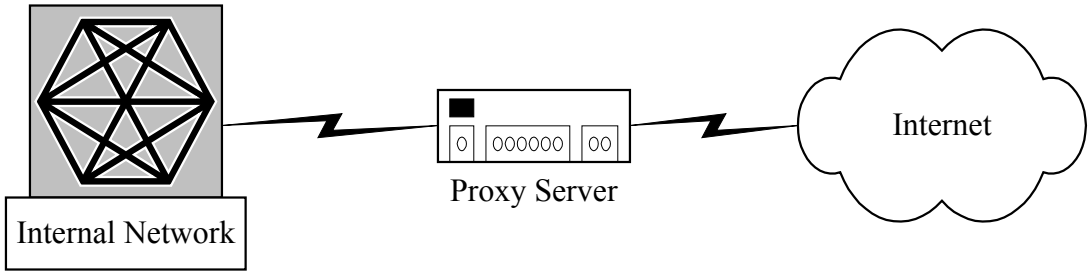
Το 1993, επιλέχθηκε ο όρος proxy-server ώστε να γίνει καλύτερη η διάκριση μεταξύ των θυρών (firewall/proxies gateways) που επιτρέπουν την κυκλοφορία που σχετίζεται μεταξύ του παγκόσμιου ιστού και των εσωτερικών δικτύων και των θυρών (information gateways) που αποτελούν την είσοδο συστημάτων στο δίκτυο γενικότερα. Ο όρος θύρες firewall/proxies αντικαταστάθηκε από τον όρο proxy-server για να τονίσει το γεγονός ότι ενεργούν εκ μέρους του πελάτη ενώ οι θύρες (information gateways) βρίσκονται και ενεργούν από τη μεριά του εξυπηρετητή (αυτοί ονομάζονται και αντίστροφοι πληρεξούσιοι reverse proxies).

Η εργασία αυτή θα επικεντρωθεί στο πληρεξούσιο εξυπηρετητή. Αφού αναφέρουμε το τρόπο λειτουργίας του, τα πλεονεκτήματα και μειονεκτήματά του, τις βασικές λειτουργίες που εκτελεί, τις δυνατότητες που μπορεί να παρέχει, θα προχωρήσουμε στην ανάλυση των τύπων και αρχιτεκτονικών, στο σχεδιασμό, την υλοποίηση και τη διαχείριση ενός ή πολλών πληρεξούσιων εξυπηρετητών, ώστε να λαμβάνουμε τα βέλτιστα αποτελέσματα. Τέλος θα δοθεί ένα παράδειγμα, το σύστημα των πληρεξούσιων εξυπηρετητών του ΕΔΕΤ των ελληνικών ΑΕΙ και ΤΕΙ και θα εξετασθεί η δυνατότητα σύνδεσης σ' αυτό ή εγκατάστασης αυτόνομων στο Πανεπιστήμιο Μακεδονίας.

Β. ΑΝΑΛΥΣΗ

1. Ορισμός

Όπως αναφέραμε, σε ένα περιβάλλον εσωτερικού δικτύου που χρησιμοποιεί το Διαδίκτυο, ο πληρεξούσιος εξυπηρετητής είναι ένας εξυπηρετητής που ενεργεί σαν ο ενδιάμεσος μεταξύ των χρηστών του δικτύου και του Διαδικτύου, εξασφαλίζοντας για το περιβάλλον αυτό ασφάλεια, συγκεντρωτικό έλεγχο, διαχείριση και υπηρεσίες επαναποθήκευσης και φιλτραρίσματος. Για να δείξουμε που τοποθετείται ο πληρεξούσιος εξυπηρετητής έχουμε το παρακάτω σχήμα:



Ο πληρεξούσιος εξυπηρετητής ενεργεί ως πελάτης και ως εξυπηρετητής: ως εξυπηρετητής που εξυπηρετεί τους συνδεόμενους σ' αυτόν πελάτες και ως πελάτης που εξυπηρετείται από τους συνδεόμενους σ' αυτόν εξυπηρετητές. Έτσι όλοι οι πελάτες έχουν πλήρη πρόσβαση στο Διαδίκτυο εύκολα, γρήγορα και με ασφάλεια.

Ειδικότερα, ο πληρεξούσιος εξυπηρετητής είναι ο εξυπηρετητής, που δέχεται τις αιτήσεις για την παρουσίαση σελίδων του World Wide Web από το πρόγραμμα πλοήγησης (browser) ενός χρήστη και αναλαμβάνει να προσκομίσει τις ζητούμενες σελίδες. Έστω για παράδειγμα ένας πελάτης ενός δικτύου που επιθυμεί να κατεβάσει μία Web σελίδα.Το πρόγραμμα πλοήγησης (π.χ. Netscape Navigator, MS Explorer κλπ.) αντί να επικοινωνήσει απευθείας με τον εξυπηρετητή που επιθυμεί ο χρήστης, ζητά από το πληρεξούσιο εξυπηρετητή να του προσκομίσει τη σελίδα.

Εάν η αίτηση είναι σύμφωνη με τις απαιτήσεις φιλτραρίσματος ο πληρεξούσιος εξυπηρετητής ενεργώντας και ως cache εξυπηρετητής αναζητεί πρώτα στη δική του κρυφή μνήμη την επαναποθηκευμένη σελίδα. Εάν βρίσκεται εκεί, εξυπηρετεί ο ίδιος τον πελάτη στέλνοντας την απάντηση σ' αυτόν χωρίς να απαιτείται η επιπλέον προώθηση της αίτησης του πελάτη στο Διαδίκτυο.

Στη περίπτωση που η σελίδα δεν είναι επαναποθηκευμένη, ο πληρεξούσιος εξυπηρετητής ενεργώντας ως πελάτης εκ μέρους του χρήστη, χρησιμοποιώντας τη δική του IP διεύθυνση, προωθεί την αίτηση σε άλλους εξυπηρετητές. Αυτοί μπορεί να είναι άλλοι πληρεξούσιοι εξυπηρετητές ή οι πραγματικοί εξυπηρετητές (origin servers), δηλαδή αυτοί που περιέχουν τη Web σελίδα, το HTML αρχείο κλπ. Όταν η απάντηση φθάνει στο πληρεξούσιο εξυπηρετητή, ο τελευταίος στέλνει τις σελίδες που ζητήθηκαν στο πρόγραμμα πλοήγησης του χρήστη, το οποίο του τις παρουσιάζει.

Για το χρήστη ο πληρεξούσιος εξυπηρετητής δεν είναι ορατός. Όλες οι αιτήσεις και οι απαντήσεις δηλαδή, φαίνονται να γίνονται κατευθείαν προς και από τον εξυπηρετητή του Διαδικτύου.

Ένα από τα βασικά πλεονεκτήματά του, είναι το γεγονός ότι η επαναποθήκευση που εκτελείται, μπορεί να εξυπηρετήσει όλους τους χρήστες του δικτύου. Έτσι εάν μία ή περισσότερες διευθύνσεις ζητούνται συχνά, οι σελίδες τους μπορούν να επαναποθηκεύονται, μειώνοντας έτσι κατά πολύ το χρόνο απάντησης προς τους πελάτες.

Οι λειτουργίες των πληρεξούσιων εξυπηρετητών, (φιλτράρισμα, επαναποθήκευση, έλεγχος, ασφάλεια) και φραγμάτων, μπορούν να αποτελούν μέρος ενός μόνο προγράμματος και να εκτελούνται σε ένα μόνο υπολογιστικό σύστημα ή διαφορετικά προγράμματα να εκτελούνται σε διαφορετικά υπολογιστικά συστήματα. Για παράδειγμα, ένας πληρεξούσιος εξυπηρετητής μπορεί να βρίσκεται στο ίδιο σύστημα Η/Υ μαζί με ένα εξυπηρετητή φράγματος ή να βρίσκεται σε διαφορετικό σύστημα και να προωθεί τις αιτήσεις μέσω του φράγματος.

2. Γενικές ιδιότητες

Οι γενικές ιδιότητες των πληρεξούσιων εξυπηρετητών είναι:

♦ Πέρα από τις λειτουργίες φιλτραρίσματος, το τελικό αποτέλεσμα δεν επηρεάζεται. Οι χρήστες θα λαμβάνουν την ίδια απάντηση είτε η σύνδεση γίνεται απευθείας στον εξυπηρετητή του Διαδικτύου, είτε μέσω του πληρεξούσιου εξυπηρετητή.

♦ Ο έλεγχος γίνεται από το χρήστη, ο χρήστης δηλαδή αποφασίζει εάν θα χρησιμοποιήσει ή όχι το πληρεξούσιο εξυπηρετητή.

♦ Ο απομακρυσμένος εξυπηρετητής δεν επηρεάζεται από οποιουσδήποτε ενδιάμεσους εξυπηρετητές και συνήθως δεν αντιλαμβάνεται την ύπαρξή τους.

3. Οι βασικές λειτουργίες

Οι πληρεξούσιοι εξυπηρετητές μπορούν να χρησιμοποιηθούν για να εκτελέσουν τις εξής βασικές λειτουργίες:

∗ Να επιτρέψουν ή να απαγορεύσουν τη πρόσβαση του πελάτη στο Διαδίκτυο, με βάση την IP του διεύθυνση.

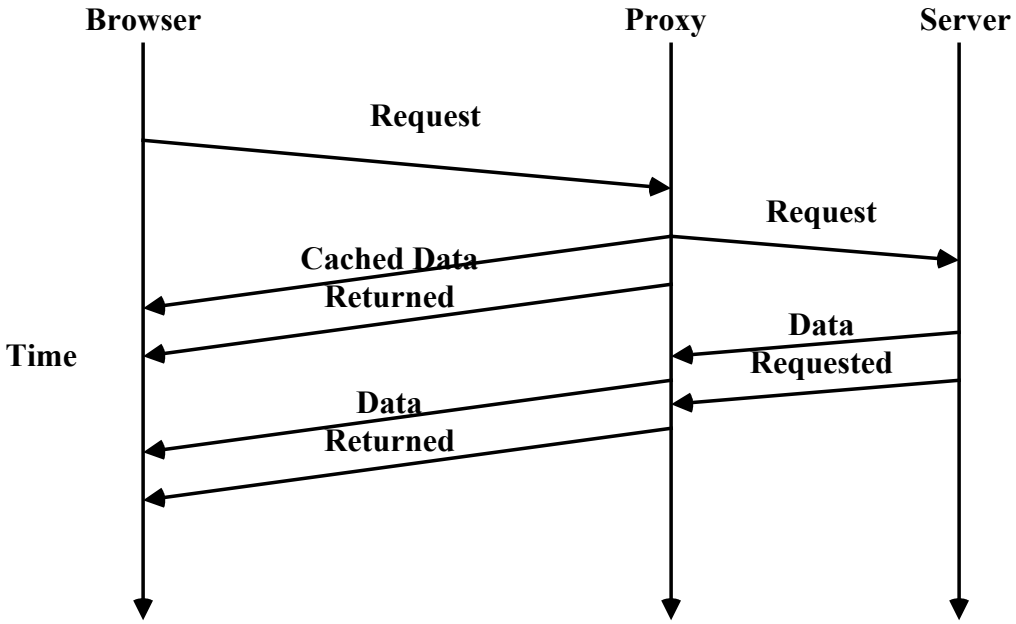
∗ Να περιορίσουν επιλεκτικά τη πρόσβαση στο Διαδίκτυο ή σε άλλα υποδίκτυα, στο επίπεδο πρωτοκόλλου, κάνοντας χρήση της URL αυτών και να καθορίσουν ποια πρωτόκολλα επιτρέπεται να χρησιμοποιήσουν οι πελάτες με βάση τις IP τους διευθύνσεις.

∗ Να επαναποθηκεύουν τις σελίδες που λαμβάνουν από τους εξυπηρετητές του Διαδικτύου διότι τις περισσότερες φορές ένας proxy είναι και ένας cache εξυπηρετητής. Ο τελευταίος αποθηκεύει τις αιτήσεις των προγραμμάτων πλοήγησης και τις αντίστοιχες απαντήσεις των εξυπηρετητών του Διαδικτύου ώστε να διαχειριστεί νέες αιτήσεις.

Για παράδειγμα, αν έχουμε ένα σύνολο χρηστών που αξιοποιούν τη λειτουργία ενός proxy-cache εξυπηρετητή, τότε αν κάποιος απ' αυτούς ζητήσει μια συγκεκριμένη σελίδα από ένα εξυπηρετητή που βρίσκεται στην Ιαπωνία ο proxy-cache θα φέρει τη σελίδα και αφενός θα την παραδώσει στο πρόγραμμα πλοήγησης του χρήστη, αφετέρου θα την αποθηκεύσει για μελλοντική χρήση. Αν τώρα ένας άλλος χρήστης -ή και ο ίδιος- ζητήσει τη ίδια σελίδα τότε ο proxy-cache εξυπηρετητής θα του προσκομίσει το αντίγραφο που έχει κρατήσει και δε θα αναζητήσει τη σελίδα στην Ιαπωνία.

Επομένως, αν την ίδια σελίδα θέλουν να δούν 50 άτομα, με τη χρήση του proxy-cache, μόνο ο πρώτος θα χρειαστεί να περιμένει να έρθει η σελίδα από τον αρχικό εξυπηρετητή, ενώ

οι υπόλοιποι 49 θα δουν τη σελίδα να έρχεται ταχύτερα, αφού θα τους διατεθεί από τον proxy-cache εξυπηρετητή. Επειδή, η σύνδεση και μεταφορά δεδομένων από την Ιαπωνία δεν χρειάζεται πλέον, μειώνεται η σπατάλη του εύρους ζώνης και ο χρήστης βλέπει τις σελίδες ταχύτερα. Αν θέλαμε σχηματικά να δείξουμε τη λειτουργία της επαναποθήκευσης θα είχαμε:



Η επαναποθήκευση είναι αποδοτικότερη όταν γίνεται στο πληρεξούσιο εξυπηρετητή γιατί μόνο μία σελίδα είναι επαναποθηκευμένη για εσωτερική χρήση. Επίσης κάνει δυνατή τη πρόσβαση ακόμα και όταν οι εξυπηρετητές του Διαδικτύου ή ολόκληρα άλλα δίκτυα βγουν εκτός λειτουργίας.

∗ Να κωδικοποιήσουν τη συνδιάλεξη με τον εξυπηρετητή του Διαδικτύου, ώστε να εξασφαλιστεί ότι κάποιοι δεν θα μπορέσουν να δουν το περιεχόμενο των πακέτων και να ελέγξουν αν τα εισερχόμενα πακέτα περιέχουν ιούς ή αν έχουν πιθανό τροποποιηθεί.

∗ Να παρέχουν πρόσβαση στο Διαδίκτυο στις εταιρίες χρησιμοποιώντας ιδιωτικά εταιρικά δίκτυα αρκεί να είναι ορατοί και στο εσωτερικό δίκτυο και στο Διαδίκτυο μέσω κυρίως μέσω δύο διαφορετικών καρτών δικτύου και τέλος

∗ να μετατρέπουν τα δεδομένα σε μορφή HTML που είναι αναγνώσιμη από το σύστημα πλοήγησης του χρήστη-πελάτη.

4. Επικοινωνία δια μέσου των πληρεξούσιων εξυπηρετητών

Για να κατανοήσουμε πως επικοινωνεί ο πληρεξούσιος εξυπηρετητής με το πελάτη και τους άλλους εξυπηρετητές του Διαδικτύου, θα πρέπει να αναλύσουμε την επικοινωνία μέσω ενός κανονικού εξυπηρετητή. Αυτή γίνεται ως εξής: κάθε πελάτης έχει τη δική του IP διεύθυνση και έχει τη δυνατότητα της απευθείας σύνδεσης με τους εξυπηρετητές του Διαδικτύου. Όταν μία κανονική αίτηση, για παράδειγμα μία αίτηση HTTP εκτελείται από το σύστημα πλοήγησης ενός πελάτη, ο HTTP εξυπηρετητής δε λαμβάνει όλη τη URL και συγκεκριμένα δε λαμβάνει το αναγνωριστικό του πρωτοκόλλου "http:" και το όνομα του εξυπηρετητή (host name). Έτσι αν ο πελάτης πληκτρολογήσει:

http://microsoft.com/products/ProxyDetails.html

το πρόγραμμα πλοήγησης το μετατρέπει σε:

GET /products/ProxyDetails.html.

Το πρόγραμμα πλοήγησης συνδέει το πελάτη με τον εξυπηρετητή της Microsoft, εισάγει την εντολή και αναμένει την απάντηση. Η αίτηση προσδιορίζει το ζητούμενο αρχείο *ProxyDetails.html* και η απάντηση που λαμβάνει είναι ή το αρχείο αυτό ή ένα μήνυμα λάθους.

Στη περίπτωση του πληρεξούσιου εξυπηρετητή, είναι αυτός που λαμβάνει τις αιτήσεις των συστημάτων πλοήγησης και ενεργεί με τη σειρά του σαν πελάτης προς τους απομακρυσμένους εξυπηρετητές για να λάβει απ' αυτούς τα ζητούμενα αρχεία. Εδώ όμως το πρόγραμμα πλοήγησης αποστέλνει όλη τη URL διεύθυνση στον πληρεξούσιο εξυπηρετητή. Έτσι αυτός έχει όλη την απαραίτητη πληροφορία που χρειάζεται για να κάνει την αίτηση στον απομακρυσμένο εξυπηρετητή που καθορίζεται από την αποστέλλουσα URL χρησιμοποιώντας το πρωτόκολλο που προσδιορίζεται από αυτήν. Όταν ο πελάτης δηλαδή πληκτρολογήσει:

http://microsoft.com/products/ProxyDetails.html,

το σύστημα πλοήγησης το μετατρέπει σε: *GET http://microsoft.com /products/ProxyDetails.html*, στη συνέχεια το σύστημα πλοήγησης συδέεται με το πληρεξούσιο εξυπηρετητή και αυτός με τη σειρά του παρέχει τη σύνδεση με αυτόν του Διαδικτύου. Ο πληρεξούσιος εξυπηρετητής μετατρέπει την αίτηση σε:

GET /products/ProxyDetails.html

και συνδέεται με τον εξυπηρετητή της Microsoft. Αυτός δέχεται την εντολή και επιστρέφει την απάντηση στον πληρεξούσιο και αυτός τέλος στον πελάτη.

Τα ίδια βήματα θα είχαν ακολουθηθεί σε περιπτώσεις που επιθυμούμε να χρησιμοποιήσουμε άλλα πρωτόκολλα (FTP, Gopher, WAIS).

5. Πλεονεκτήματα-μειονεκτήματα

α. Τα πλεονεκτήματα από τη χρήση των πληρεξούσιων εξυπηρετητών είναι:

- **Απόδοση**: με τη χρησιμοποίηση των πληρεξούσιων εξυπηρετητών για την πρόσβαση σε συχνά ζητούμενα έγγραφα, μειώνεται ο χρόνος απόκρισης και τα έγγραφα εμφανίζονται πιο γρήγορα στην οθόνη μας σε σύγκριση με την ανάκτησή τους από τους εξυπηρετητές που τα περιέχουν.
- **Προστασία**: καθιστά τα εσωτερικά δίκτυα απρόσβλητα από εξωτερικούς κινδύνους που προέρχονται από τη δικτύωσή τους.
- **Κόστος**: μικρότερη εξωτερική κυκλοφορία του δικτύου ισοδυναμεί με μικρότερο κόστος πρόσβασης στο Διαδίκτυο.
- **Συμφόρηση**: μικρότερη κυκλοφορία στο Διαδίκτυο σημαίνει χαμηλότερα ποσοστά συμφόρησης καθώς λιγότερες αιτήσεις κυκλοφορούν μέσα σ' αυτό.
- **Μέλλον**: η χρησιμοποίηση των πληρεξούσιων εξυπηρετητών μπορεί να οδηγήσει τη κυκλοφορία του παγκόσμιου ιστού να φθάσει σε ακόμα πιο αστρονομικά επίπεδα και να αμβλύνει το πρόβλημα διευθυνσοποίησης του Διαδικτύου που παρουσιάζεται με την εξάντληση των IP διευθύνσεων.

β. Τα μειονεκτήματα από τη χρήση των πληρεξούσιων εξυπηρετητών είναι:

- **Διαφήμιση**: η χρήση των πληρεξούσιων εξυπηρετητών έχει αρνητικά αποτελέσματα στα εμπορικά sites του Διαδικτύου. Αυτό συμβαίνει διότι οι διαφημιστές δεν έχουν τη δυνατότητα να γνωρίζουν τον αριθμό των επισκεπτών (μπορεί να είναι από ένας ή χιλιάδες), παρά μόνο αν ζητήσουν τις πληροφορίες αυτές από τους διαχειριστές των πληρεξούσιων εξυπηρετητών. Μία προτεινόμενη λύση θα ήταν να επιτρέπεται στον εξυπηρετητή να κρατά στη κρυφή του μνήμη μία σελίδα του παγκόσμιου ιστού μόνο αν είναι υποχρεωμένος ανά τακτά χρονικά διαστήματα να ενημερώνει τους εξυπηρετητές του Διαδικτύου για τον αριθμό των επισκεπτών και άλλες πληροφορίες χρήσιμες για τους διαφημιστές. Μερικοί υποστηρίζουν ότι η χρήση των πληρεξούσιων εξυπηρετητών παραβιάζει το νόμο της πνευματικής ιδιοκτησίας, διότι το αντίγραφο ενός εγγράφου που βρίσκεται στη κρυφή μνήμη του εξυπηρετητή στην πραγματικότητα δεν έχει την εξουσιοδότηση του ιδιοκτήτη.
- **Ενημέρωση**: ένα από τα σημαντικότερα μειονεκτήματα της επαναποθήκευσης των πληρεξούσιων είναι ο κίνδυνος ανάκτησης από τη κρυφή μνήμη του εξυπηρετητή, εγγράφων που έχουν τροποποιηθεί ή διαγραφεί στους πραγματικούς εξυπηρετητές. Το πρόβλημα αυτό,

απαιτεί ιδιαίτερες ρυθμίσεις στα πρωτόκολλα, ώστε να προσδιορίζεται η συχνότητα με την οποία θα ενημερώνονται τα έγγραφα (π.χ. μέθοδος HEAD όπου οι πληροφορίες που λαμβάνονται από ένα έγγραφο προέρχονται από την επικεφαλίδα του και όχι από το ίδιο έγγραφο).

- **Αποτελεσματικότητα**: ο πληρεξούσιος εξυπηρετητής δεν είναι πανάκεια. Είναι ουσιαστικά αδύνατος ο έλεγχος και το φιλτράρισμα όλων των εγγράφων και σελίδων του παγκόσμιου ιστού. Επίσης δεν είναι δυνατός ο έλεγχος στα E-mail's, στα προσαρτημένα σ' αυτά αρχεία και τις συνομιλίες (chat).

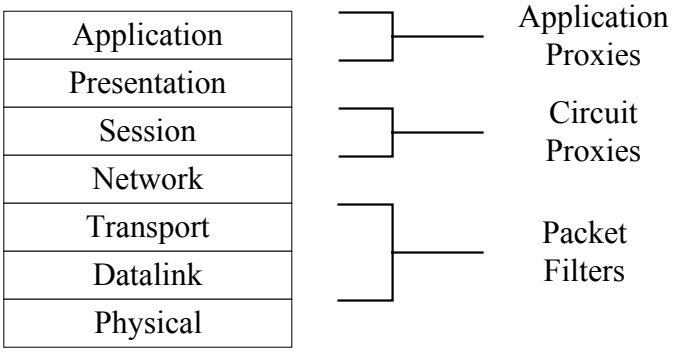
Γ. ΤΥΠΟΙ & ΑΡΧΙΤΕΚΤΟΝΙΚΕΣ

1. Βασικοί Τύποι και αντίστοιχες Αρχιτεκτονικές

Όπως αναφέραμε στην εισαγωγή οι πληρεξούσιοι εξυπηρετητές, είναι οι υπηρεσίες που εκτελούνται "τρέχουν" πάνω σε ένα φράγμα. Βασικά υπάρχουν τρεις τύποι στη τεχνολογία των φραγμάτων:

1. *Φιλτράρισμα πακέτων (Filter packets).*
2. *Πληρεξούσιοι εξυπηρετητές στο επίπεδο εφαρμογών (Application-level proxies).*
3. *Πληρεξούσιοι εξυπηρετητές στο επίπεδο συνόδου (Circuit-level proxies).*

Η διαφορά μεταξύ των ανωτέρω είναι ότι προστατεύουν τα δεδομένα σε διαφορετικό επίπεδο του OSI μοντέλου. Έτσι όπως φαίνεται στο παρακάτω σχήμα το φίλτρο των πακέτων εκτελείται στο επίπεδο μεταφοράς (Transport level), ενώ οι πληρεξούσιοι εξυπηρετητές λειτουργούν οι μεν πρώτοι στο επίπεδο εφαρμογής (Application level) οι δεύτεροι στο επίπεδο συνόδου (Session level).

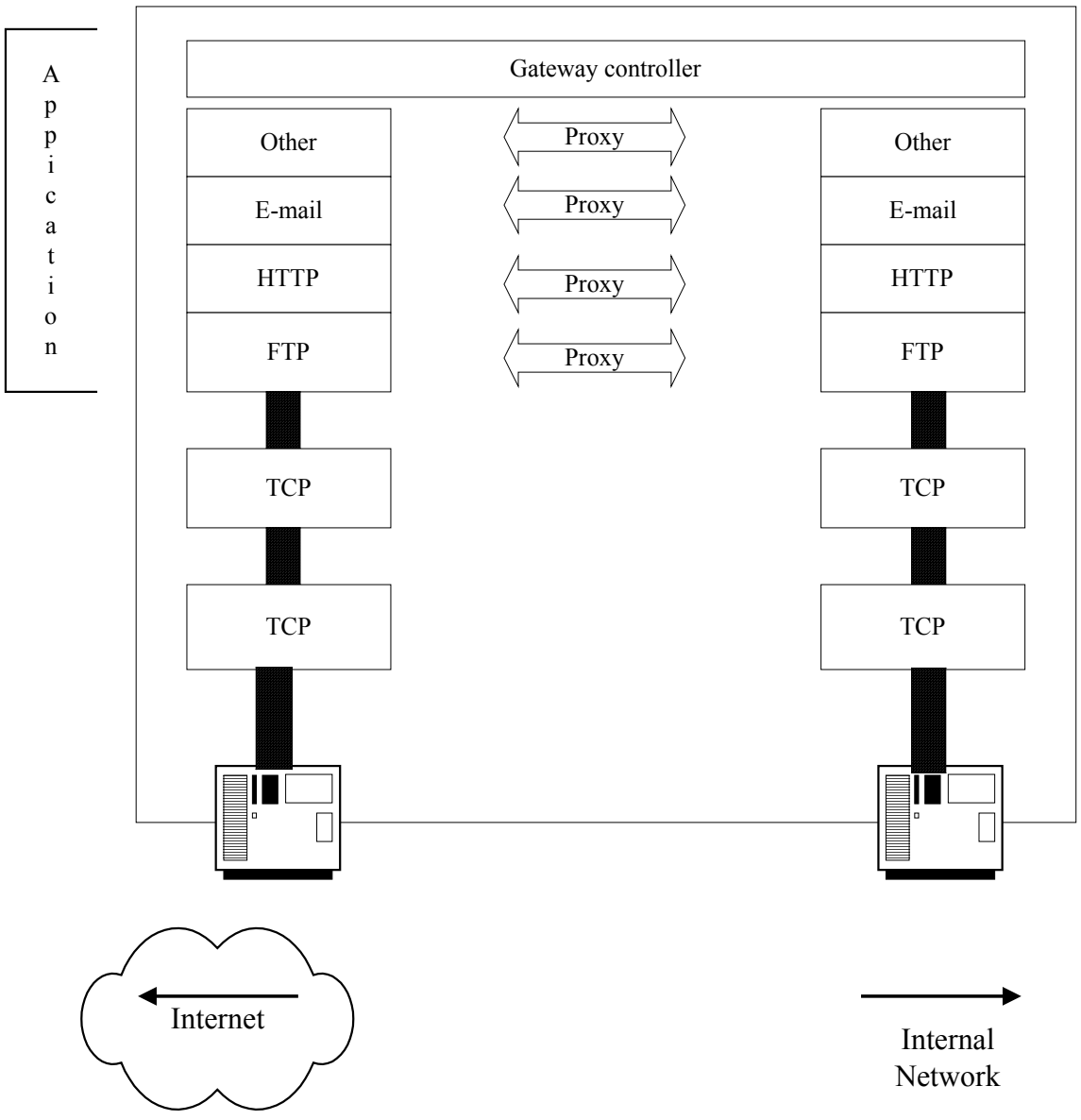


1.1. Packet Filters

Στη πρώτη περίπτωση (packet filters) χρησιμοποιούνται δρομολογητές για το φιλτράρισμα της πληροφορίας που κινείται από και προς το δίκτυο. Οι δρομολογητές ελέγχουν κάθε πακέτο αν ανήκει στο πίνακα που περιέχει τις επιτρεπόμενες διευθύνσεις και ανάλογα επιτρέπουν ή όχι το πακέτο. Αποτελούν μία εύκολη και φθηνή λύση, αλλά δεν παρέχουν επιπρόσθετες λειτουργίες. Οι άλλες δύο κατηγορίες αποτελούν τις πιο συνηθισμένες λύσεις.

1.2. Application-Level

Η πρώτη μορφή πληρεξούσιων εξυπηρετητών είναι αυτοί που "τρέχουν" σε ένα φράγμα στο επίπεδο εφαρμογής (application-level proxy-servers) και παρέχουν εργασίες ελέγχου και παρακολούθησης της κυκλοφορίας του συστήματος. Οι εξυπηρετητές σχετίζονται με ένα ή περισσότερα πρωτόκολλα και εκτελούν τις υπηρεσίες τους για κάθε τύπο εφαρμογής στο Διαδίκτυο (όπου απαιτείται έλεγχος). Για κάθε διαφορετικό τύπο εφαρμογής, μπορούμε να έχουμε και τις αντίστοιχες proxy υπηρεσίες του. Αυτή η λειτουργία φαίνεται στο σχήμα όπου ένας HTTP proxy ελέγχει τις Web υπηρεσίες, ενώ ένας FTP proxy ελέγχει τις υπηρεσίες μεταφοράς αρχείων κλπ.



Firewall/proxy server

Για παράδειγμα, έστω ένας χρήστης που επιθυμεί να έχει πρόσβαση σε κάποιον Web εξυπηρετητή και να κάνει κάποια αίτηση. Η HTTP proxy υπηρεσία που "τρέχει" σε ένα φράγμα, ανακόπτει τα πακέτα του χρήστη, τα ανασυσκευάζει και στέλνει τα ανασυσκευασμένα αυτά πακέτα στον εξυπηρετητή του Διαδικτύου. Τα πακέτα "κουβαλούν" μόνο την IP διεύθυνση του πληρεξούσιου και όχι την IP διεύθυνση του χρήστη. Όλα τα πακέτα φαίνεται να έχουν την ίδια IP διεύθυνση (αυτή του πληρεξούσιου) και όλες οι διευθύνσεις των χρηστών αποκρύπτονται. Όταν ο Web εξυπηρετητής στέλνει τις απαντήσεις, ο πληρεξούσιος τις λαμβάνει και τις δρομολογεί στον εσωτερικό χρήστη.

Σ' αυτή την περίπτωση ο εξυπηρετητής στον οποίο "τρέχουν" οι proxy υπηρεσίες έχει δύο ή περισσότερες κάρτες δικτυακής επικοινωνίας NICS (Network Interface Cards). Από αυτές η μία είναι συνδεδεμένη με το εσωτερικό δίκτυο και η άλλη (-ες) συνδέεται με το Διαδίκτυο. Αυτή η περίπτωση ονομάζεται dual-homed (ή multi-homed αντίστοιχα).

Ένας τέτοιος εξυπηρετητής επιτρέπει ή όχι τις λειτουργίες δρομολόγησης μεταξύ καρτών NICS ανάλογα με τις απαιτήσεις μας. Τα πακέτα δηλαδή, μεταδίδονται μέσω των δικτύων εφόσον οι εξυπηρετητές το επιτρέψουν. Εάν για μία συγκεκριμένη εφαρμογή δεν υπάρχει αντίστοιχη proxy υπηρεσία κανένα πακέτο που σχετίζεται με την εφαρμογή δεν επιτρέπεται να περάσει. Αυτή η εγκατάσταση όταν εφαρμόζεται κατάλληλα εμποδίζει κάθε άτομο που επιθυμεί να εισβάλλει στα συστήματα του εσωτερικού δικτύου. Εκτός από τον παράγοντα ασφάλεια η συγκεκριμένη εγκατάσταση προσφέρει επίσης:

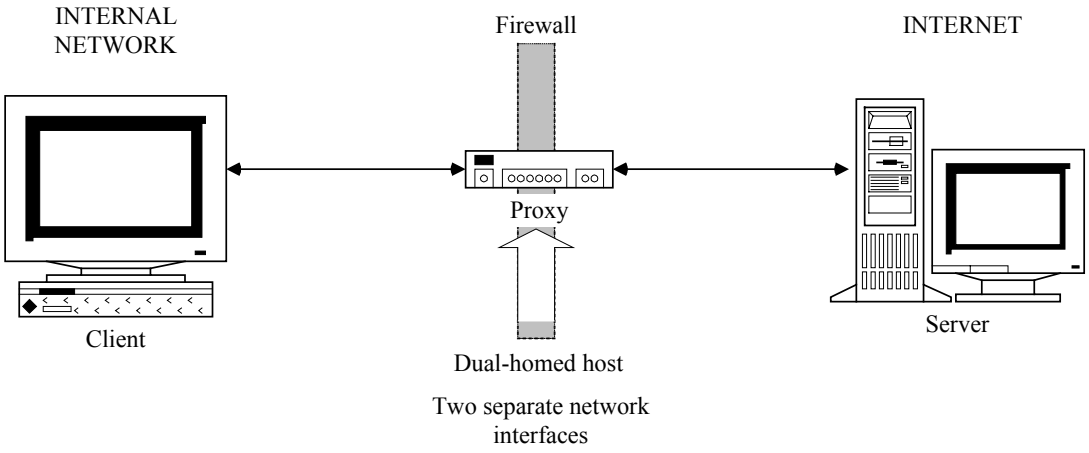
♦ Ελαχιστοποίηση του χρόνου εξυπηρέτησης των πελατών λόγω της ικανότητας επαναποθήκευσης δεδομένων των εξυπηρετητών.

♦ Αύξηση του ελέγχου πρόσβασης με τη χρήση της επικύρωση ταυτοποίησης (authentication).

♦ Καλύτερες διαδικασίες φιλτραρίσματος διότι είναι διαθέσιμες για επεξεργασία όχι μόνο οι πληροφορίες που περιλαμβάνονται στις επικεφαλίδες των πακέτων, όπως συμβαίνει στις κανονικές περιπτώσεις αλλά το σύνολό του και

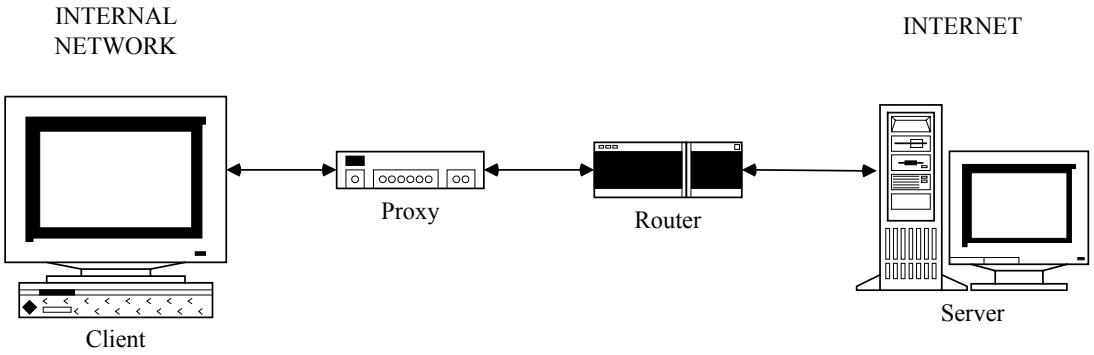
♦ παροχή της ικανότητας εξυπηρέτησης πολλών αιτήσεων ταυτόχρονα.

Το επόμενο σχήμα απεικονίζει τη συγκεκριμένη αυτή εγκατάσταση:



Υπάρχει και μία άλλη χρήση αυτής της εγκατάστασης. Ας υποθέσουμε ότι ένα τέτοιο σύστημα εκτελεί HTTP υπηρεσίες του παγκόσμιου ιστού και είναι προγραμματισμένο να μην επιτρέπει τη δρομολόγηση. Στη περίπτωση αυτή και τα δύο συστήματα (υποδίκτυα) που βρίσκονται "πίσω" από τις κάρτες μπορούν να εκτελούν τις Web υπηρεσίες χωρίς κανένα πρόβλημα, αλλά τα πακέτα δε μπορούν να κυκλοφορήσουν από το ένα σύστημα στο άλλο. Αυτή η εγκατάσταση μπορεί να χρησιμοποιηθεί αν έχουμε πολλά τμήματα μιας εταιρίας που θέλουμε να μοιράζονται τον ίδιο εξυπηρετητή και να μην έχουν τη δυνατότητα επικοινωνίας μεταξύ τους.

Συνήθως οι application-proxy-servers συνδυάζονται με δρομολογητές (routers) για να παρέχουν μεγαλύτερη ασφάλεια. Στο επόμενο σχήμα ο δρομολογητής "προστατεύει" το πληρεξούσιο εξυπηρετητή από το Διαδίκτυο (Αυτή η αρχιτεκτονική ονομάζεται από πολλούς Screening Host).



Συγκεκριμένα ο δρομολογητής :

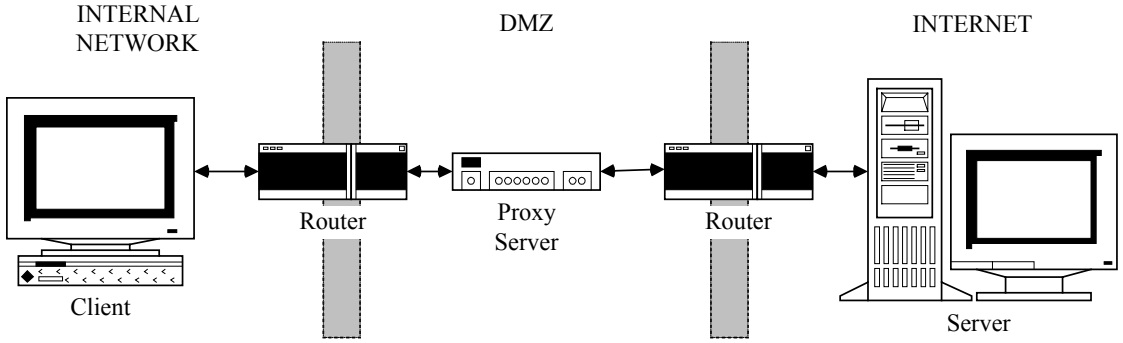
♦Επιτρέπει τις συνδέσεις μεταξύ του εσωτερικού δικτύου και του εξυπηρετητή.

♦Απαγορεύει οποιαδήποτε προσπάθεια απευθείας σύνδεσης από το εσωτερικό δίκτυο στο Διαδίκτυο. Αυτό έχει σαν αποτέλεσμα οποιαδήποτε τέτοια προσπάθεια να γίνεται μόνο μέσω του εξυπηρετητή, δίνοντας μία πλήρη εικόνα της δραστηριότητας που λαμβάνει χώρα μεταξύ του εσωτερικού και του Διαδικτύου.

♦Επιτρέπει συνδέσεις από το εξυπηρετητή προς το Διαδίκτυο ή για συγκεκριμένα πρωτόκολλα όπως FTP και HTTP.

♦Απαγορεύει όλες τις εισερχόμενες συνδέσεις από το Διαδίκτυο στο εξυπηρετητή ή σε οποιοδήποτε άλλο σύστημα του εσωτερικού δικτύου. Εξαίρεση αποτελούν γνωστά πρωτόκολλα ή E-mail, news.

Μία ακόμα πιο ασφαλή αρχιτεκτονική φαίνεται στο παρακάτω σχήμα:



Όπως βλέπουμε από το σχήμα η αρχιτεκτονική αυτή παρέχει τρία επίπεδα ασφαλείας. Ο εξυπηρετητής βρίσκεται ανάμεσα σε δύο δρομολογητές (bastion), δημιουργώντας ένα υποδίκτυο-"ζώνη" στο οποίο συμπεριλαμβάνεται ο εξυπηρετητής και αναφέρεται ως DMZ (DeMilitarized Zone).

Το DMZ είναι το τμήμα του δικτύου που βρίσκεται ανάμεσα στο Διαδίκτυο και το εσωτερικό δίκτυο (πχ Intranet). Είναι περισσότερο εκτεθειμένο στις απειλές που προέρχονται από το Διαδίκτυο σε σχέση με το εσωτερικό δίκτυο, με αποτέλεσμα απαιτούνται για τη ζώνη αυτή ιδιαίτερα μέτρα ασφαλείας. Ταυτόχρονα όμως η ζώνη αυτή προστατεύει το υπόλοιπο εσωτερικό δίκτυο από τις απειλές του Διαδικτύου (η αρχιτεκτονική αυτή ονομάζεται και ως Screened Subnet).

Στην αρχιτεκτονική αυτή ο:

Εσωτερικός δρομολογητής:

♦ Επιτρέπει ορισμένες από τις εξερχόμενες προς το εξυπηρετητή συνδέσεις, από το εσωτερικό δίκτυο.

♦ Εμποδίζει όλες τις υπόλοιπες εξερχόμενες συνδέσεις από το εσωτερικό δίκτυο.

♦ Εμποδίζει όλες εισερχόμενες συνδέσεις που προσπαθούν να εισέλθουν στο εσωτερικό δίκτυο.

Πληρεξούσιος εξυπηρετητής:

♦ Επιτρέπει την εκτέλεση των proxy υπηρεσιών πρωτοκόλλων του Παγκόσμιου Ιστού όπως το FTP, HTTP Gopher κ.ά. ή μέσω ασφαλών SSL (Secure Socket Layer).

Εξωτερικός δρομολογητής:

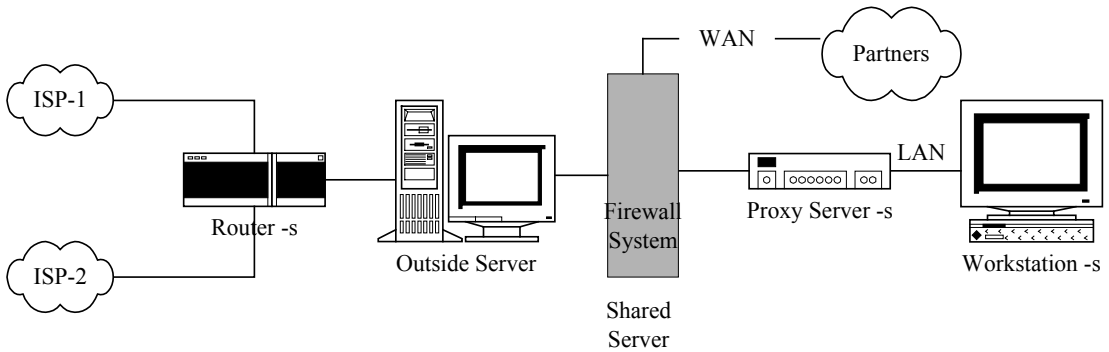
♦ Επιτρέπει μερικές από τις εξερχόμενες συνδέσεις από τον εξυπηρετητή, στο Διαδίκτυο.

♦ Εμποδίζει κάθε άλλη εξερχόμενη σύνδεση από το εσωτερικό δίκτυο.

♦ Εμποδίζει όλες τις εισερχόμενες συνδέσεις από το Διαδίκτυο.

Στη παραπάνω περίπτωση δεν επιτρέπεται το E-mail ή το USENET. Για να υλοποιούνται και αυτές οι υπηρεσίες θα πρέπει ή οι δρομολογητές να είναι προγραμματισμένοι να επιτρέπουν τις εισερχόμενες συνδέσεις E-mail και news να περάσουν στους εσωτερικούς mail και news εξυπηρετητές ή να εγκαταστήσουμε ένα πληρεξούσιο εξυπηρετητή στο DMZ που θα επιτρέπει την εισερχόμενη κυκλοφορία από το Διαδίκτυο στο πληρεξούσιο εξυπηρετητή και από εκεί στους εσωτερικούς εξυπηρετητές των εισερχόμενων mail και news.

Μία ενδιαφέρουσα περίπτωση είναι αν θέλουμε να δημιουργήσουμε μία υπηρεσία στο Διαδίκτυο όπως YAHOO ή SlashDot χρησιμοποιώντας μεγάλο αριθμό δρομολογητών ή φραγμάτων και κάνοντας load balancing τεχνικών που θα αναφερθούν παρακάτω, για τους πληρεξούσιους εξυπηρετητές.



Οι βασικότερες αρχιτεκτονικές που βασίζονται στους πληρεξούσιους εξυπηρετητές επιπέδου εφαρμογής μπορούν να συνοψιστούν στις εξής κατηγορίες:

1. Ένας *γενικευμένος πληρεξούσιος εξυπηρετητής* (generic proxy server) ή περισσότεροι, συνδεδεμένοι παράλληλα μεταξύ τους.

2. *Αλυσίδα πολλών πληρεξούσιων εξυπηρετητών* (proxy chaining) που κατανέμονται ιεραρχικά ή διατμηματικά.

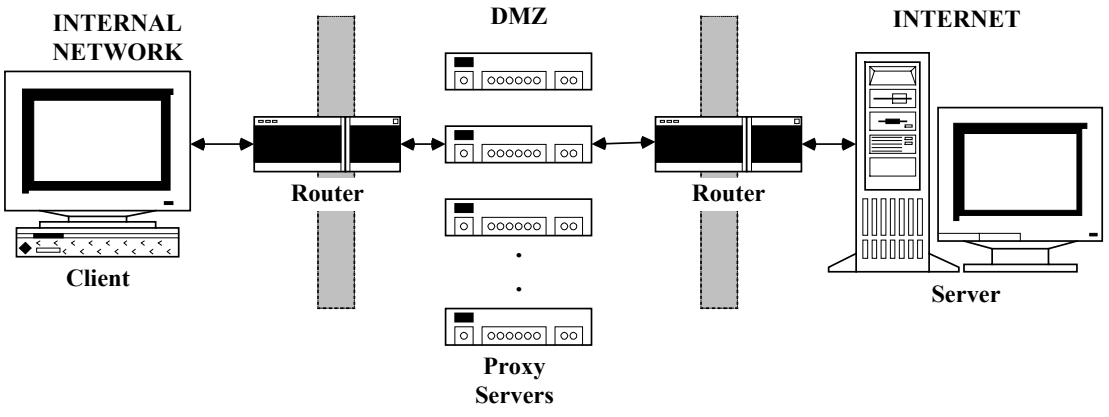
3. *Ειδικές αρχιτεκτονικές* που αποτελούνται από ειδικούς πληρεξούσιους εξυπηρετητές (specialized proxies) και υλοποιούνται για να εξυπηρετήσουν συγκεκριμένους σκοπούς.

4. Αρχιτεκτονικές που βασίζονται σε πληρεξούσιους εξυπηρετητές από τη μεριά του εξυπηρετητή του δικτύου, *αντίστροφοι* (reverse proxy servers).

Ειδικότερα,

1. Γενικευμένοι (Generic) - Πίνακας (Array)

Οι γενικευμένοι πληρεξούσιοι εξυπηρετητές (generic proxies), αποτελούν τη συνηθέστερη μορφή. Διαχειρίζονται τη κίνηση του παγκόσμιου ιστού συμπεριλαμβανόμενων των πρωτοκόλλων HTTP, FTP και Gopher όπως και ασφαλή πρωτόκολλα με τη χρήση του SSL όπως HTTPS και SNEWS. Παρέχουν όλες τις λειτουργίες και δυνατότητες που προαναφέρθηκαν και από τις εγκαταστάσεις που αναφέραμε (χωρίς κανένα, με ένα ή δύο δρομολογητές) συνήθως βρίσκονται μέσα σε μία ζώνη DMZ όπως φαίνεται στο παρακάτω σχήμα:



Δέχονται τις αιτήσεις, όπως είδαμε, από το εσωτερικό του φράγματος και τις προωθούν στο Διαδίκτυο, στέλνοντας τις απαντήσεις στους πελάτες.

Στη πράξη -όπως φαίνεται και στο σχήμα- ποτέ ένας μόνος του γενικευμένος πληρεξούσιος εξυπηρετητής δεν μπορεί να αντιμετωπίσει ένα μεγάλο αριθμό αιτήσεων. Γι' αυτό το λόγο έχουμε την εγκατάσταση πολλών παράλληλων εξυπηρετητών που αποτελούν μία ομάδα ή καλύτερα ένα πίνακα "array", όπου τα στοιχεία του (μέλη), συνεργάζονται μεταξύ τους.

Αυτή η αρχιτεκτονική βασίζεται στη τοπολογία της κατανεμημένης επαναποθήκευσης (Distributed caching). Σύμφωνα μ' αυτήν, τα περιεχόμενα της κρυφής μνήμης κατανέμονται σε ένα αριθμό εξυπηρετητών που συδέονται μεταξύ τους δημιουργώντας όπως αναφέραμε ένα πίνακα και ένα αποτελεσματικό σύστημα επαναποθήκευσης για την κατανομή του φορτώματος.

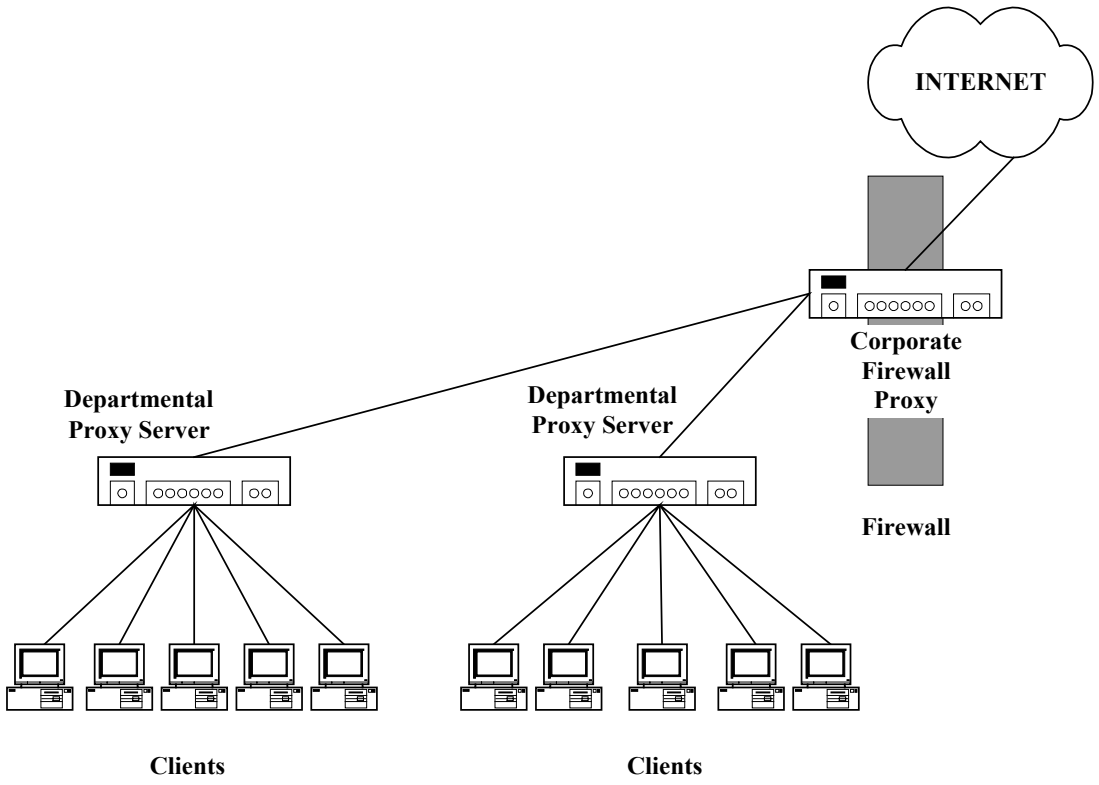
Μία επέκταση αυτής της αρχιτεκτονικής που παρουσιάζει μεγάλο ενδιαφέρον θα ήταν να επαναποθηκεύεται σε κάθε εξυπηρετητή -εκτός από τις σελίδες- και ορισμένες άλλες πληροφορίες που αφορούν αυτές, όπως μία λίστα των εξυπηρετητών που την κατέχουν, ώστε γνωρίζοντας την τοπολογία του συστήματός μας να εγκαταστήσουμε ένα μηχανισμό που θα αποφασίζει ποιος είναι ο κοντινότερος εξυπηρετητής από τον οποίο θα ανακτήσουμε τη σελίδα που επιθυμούμε.

Το βασικότερο πρόβλημα που θα αντιμετωπίσουμε στη περίπτωση πολλών πληρεξούσιων εξυπηρετητών γενικά, είναι με ποιο τρόπο (τεχνική) θα γίνει η κατανομή των αιτήσεων ανάμεσα στους εξυπηρετητές (load balancing). Οι τεχνικές αυτές θα αναφερθούν στο επόμενο κεφάλαιο.

Τέλος, οι κυριότερες online υπηρεσίες όπως η Compuserve και America Online, χρησιμοποιούν την αρχιτεκτονική αυτή έχοντας εγκαταστήσει ένα πίνακα "array" από πληρεξούσιους εξυπηρετητές.

2. Διατμηματικοί (Departmental) - Αλυσίδα (Daisy chained)

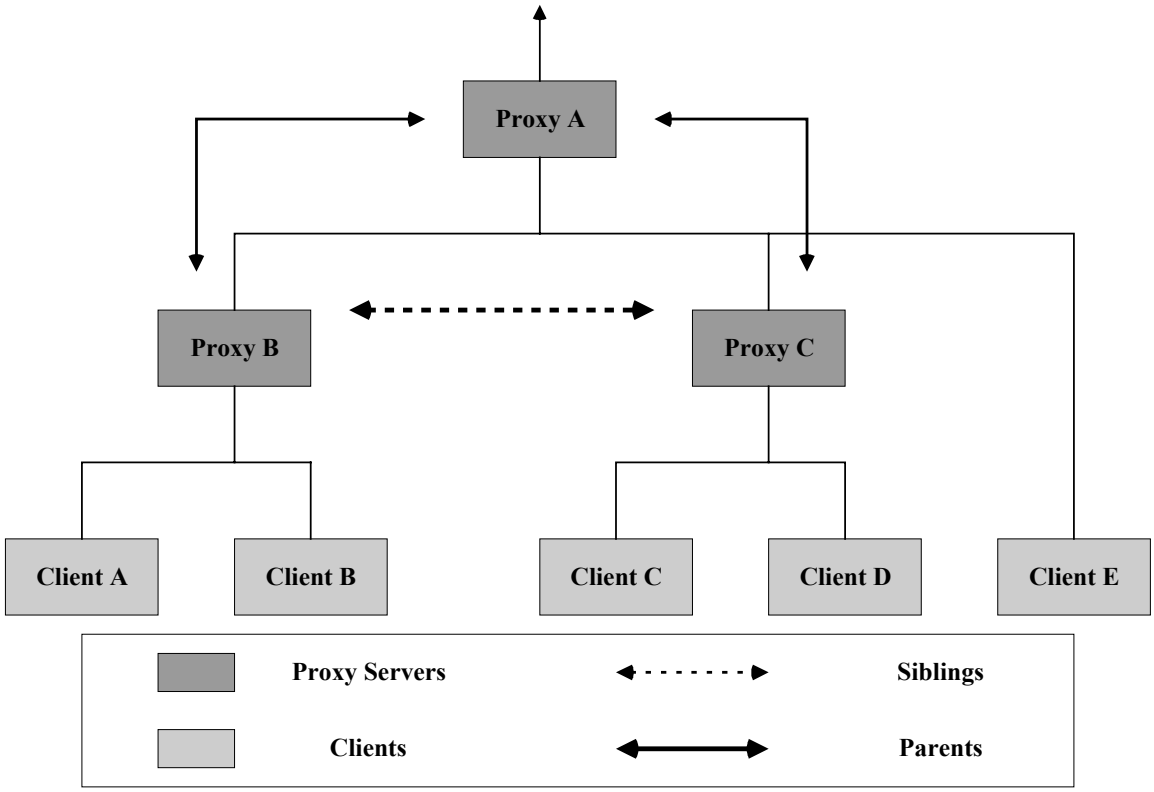
Οι πελάτες μπορούν επίσης να αιτούνται έγγραφα μέσω ενός διατμηματικού πληρεξούσιου εξυπηρετητή (departmental proxy server), ο οποίος με τη σειρά του είναι συνδεδεμένος (daisy-chained) με το πληρεξούσιο εξυπηρετητή που "τρέχει" πάνω σε ένα φράγμα. Daisy-chained σημαίνει ουσιαστικά την ανακατεύθυνση του διατμηματικού πληρεξούσιου εξυπηρετητή για να εκτελέσει τις αιτήσεις που του έχουν αναθέσει σε άλλον, στη περίπτωσή μας σ' αυτόν που βρίσκεται πάνω στο φράγμα. Μία κλασσική περίπτωση περιγράφεται στο επόμενο σχήμα:



Αυτή η αλυσίδα από πληρεξούσιους εξυπηρετητές επιτρέπει αυτούς που βρίσκονται σε χαμηλότερο επίπεδο (και κατά συνέπεια πιο κοντά στους πελάτες) να επωφεληθούν από την επαναποθήκευση αυτών που βρίσκονται σε υψηλότερο επίπεδο. Δηλαδή, αν ο εξυπηρετητής που "τρέχει" πάνω στο φράγμα έχει ήδη ανακτήσει κάποιο έγγραφο για να εξυπηρετήσει κάποιον εξυπηρετητή που βρίσκεται σε χαμηλότερο επίπεδο, τότε κάθε άλλος που βρίσκεται σ' αυτό το επίπεδο (άλλος διατμηματικός πληρεξούσιος εξυπηρετητής) μπορεί να κατεβάσει το ίδιο έγγραφο από τη κρυφή μνήμη του πρώτου.

Αυτή η αρχιτεκτονική ελαφρύνει το φόρτωμα στους κεντρικούς πληρεξούσιους εξυπηρετητές διότι με την εγκατάσταση διατμηματικών εξυπηρετητών, οι τελευταίοι εξυπηρετούν τους πελάτες απευθείας από τη δική τους κρυφή μνήμη.

Μόνο το τμήμα εκείνο των αιτήσεων που δε μπορεί να εξυπηρετηθεί από αυτούς θα προωθηθεί στο κεντρικό εξυπηρετητή. Ο τελευταίος λαμβάνει όλες τις αιτήσεις από όλους τους διατμηματικούς εξυπηρετητές με αποτέλεσμα να παρέχει μεγαλύτερη πιθανότητα για cache-hit. Αυτή η αρχιτεκτονική βασίζεται στη τοπολογία της ιεραρχικής επαναποθήκευσης (Hierarchical caching). Σύμφωνα μ' αυτήν, η οργάνωση των τοπικών περιοχών κρυφής μνήμης γίνεται σε ιεραρχικά σχήματα που συνδέονται μεταξύ τους ώστε να αλληλοεξυπηρετούν τις αιτήσεις των χρηστών. Για παράδειγμα σχηματικά θα μπορούσαμε να έχουμε την εξής διάταξη:



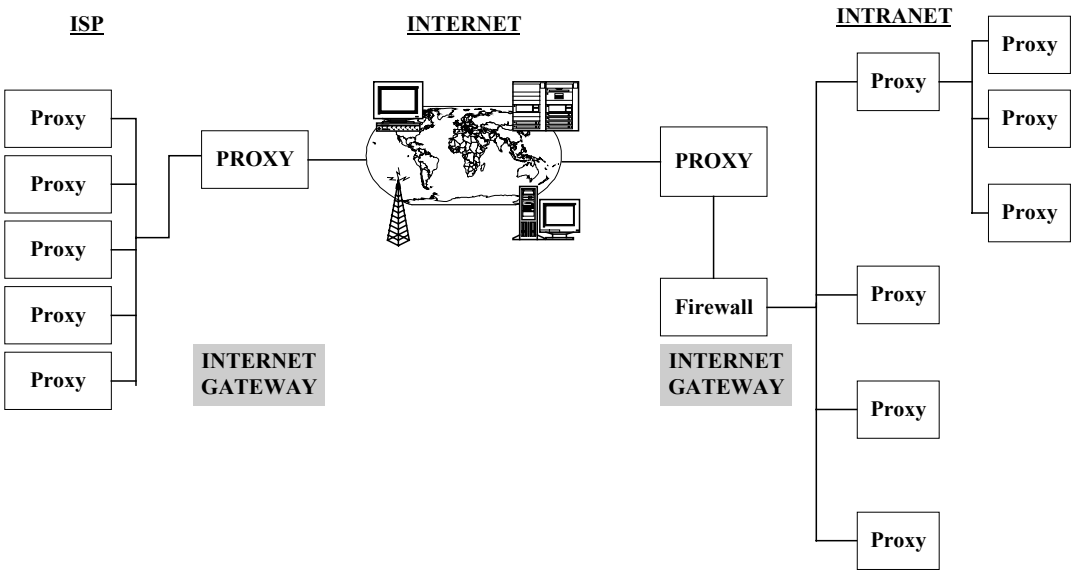
Οι σχέσεις που μπορεί να υπάρχουν μεταξύ συστημάτων επαναποθήκευσης μπορεί να είναι είτε sibling (ίδιου επιπέδου), είτε parent (σε υψηλότερο επίπεδο). Η ροή των αιτήσεων γίνεται πάντα από το χαμηλότερο προς το υψηλότερο επίπεδο. Εάν η κρυφή μνήμη ενός συστήματος δεν περιέχει τη ζητούμενη πληροφορία, τη ζητάει από τα συστήματα που βρίσκονται στο ίδιο επίπεδο. Εάν βρεθεί, στέλνεται στο πελάτη αλλιώς προωθείται στο υψηλότερο επίπεδο ή στο πραγματικό εξυπηρετητή του Διαδικτύου.

Στο παραπάνω σχήμα, οι πελάτες A - D έχουν δύο επίπεδα ιεραρχίας, ενώ ο πελάτης E ένα. Αυτό σημαίνει ότι στέλνοντας μία αίτηση, αυτή προωθείται απευθείας στον εξυπηρετητή A και οι εξυπηρετητές B και C δεν απασχολούνται. Αν κάποιος από τους πελάτες A - D στέλνει μία αίτηση, αυτή προωθείται στον αντίστοιχο εξυπηρετητή και αν υπάρξει "cache miss", προωθείται στο sibling ή parent εξυπηρετητή.

Οι διατμηματικοί πληρεξούσιοι εξυπηρετητές είναι και αυτοί γενικευμένοι (generic) εξυπηρετητές, παρόμοιοι με τους κεντρικούς μόνο που έχουν μικρότερη βάση χρήσης δηλαδή ένα τμήμα ενός οργανισμού ή επιχείρησης. Το λογισμικό τους είναι παρόμοιο μ' αυτό των κεντρικών με μικρές διαφοροποιήσεις. Μερικά τμήματα μπορεί να έχουν πιο περιορισμένο έλεγχο πρόσβασης από άλλα ή ακόμα η πρόσβαση προς το κεντρικό πληρεξούσιο εξυπηρετητή να είναι διαφορετική μεταξύ τους.

Για παράδειγμα, μπορεί ένας διατμηματικός πληρεξούσιος εξυπηρετητής να επιτρέπει όλους τις URL αιτήσεις. Ο κεντρικός -από τη πολιτική της επιχείρησης ή του οργανισμού- να απαγορεύει τις URL αιτήσεις για συγκεκριμένα sites. Μία αίτηση ενός πελάτη προς αυτά θα προωθηθεί από το διατμηματικό προς το κεντρικό αλλά ο τελευταίος θα απαγορεύσει την περαιτέρω προώθηση.

Στο παραπάνω σχήμα έχουμε δύο επίπεδα από πληρεξούσιους εξυπηρετητές. Αυτό είναι ένα απλό παράδειγμα εξυπηρετητών πολλών επιπέδων (multilevel proxying). Στη πράξη και σε μεγαλύτερα μεγέθη μπορούμε να έχουμε πιο περίπλοκες και σε περισσότερα επίπεδα διατάξεις όπως φαίνεται στο επόμενο σχήμα:



3. Ειδικοί (Specialized)

Οι αρχιτεκτονικές που αποτελούνται από ειδικούς πληρεξούσιους εξυπηρετητές (specialized proxies) υλοποιούνται για να εξυπηρετήσουν συγκεκριμένους σκοπούς. Ένα απλό παράδειγμα ειδικού πληρεξούσιου εξυπηρετητή είναι αυτό που εκτελεί λογισμικά προγράμματα σε μία palmtop συσκευή. Αυτός ο τύπος εξυπηρετητή για παράδειγμα, μπορεί να μειώσει την ανάλυση μιας εικόνας και το πλήθος των χρωμάτων που χρησιμοποιεί για να είναι δυνατή η ανάγνωσή του από τη palmtop συσκευή. Έτσι μειώνεται η απαίτηση για διαθέσιμο εύρος ζώνης και ταυτόχρονα τα δεδομένα μετατρέπονται σε μορφή κατάλληλη για το αντίστοιχο λογικό ή λογισμικό.

Ακόμα μία ενδιαφέρουσα αρχιτεκτονική είναι ένας ειδικός πληρεξούσιος εξυπηρετητής που βρίσκεται μπροστά από το βασικό εξυπηρετητή. Αυτός ο εξυπηρετητής προωθεί όλη τη κυκλοφορία που λαμβάνει σε μία διαφορετική θύρα είτε στον ίδιο εξυπηρετητή είτε στο βασικό εξυπηρετητή, για να εκτελέσει ένα συγκεκριμένο σκοπό, όπως φιλτράρισμα των Java applets ή έλεγχος για ιούς.

Μία άλλη αρχιτεκτονική που ακολουθεί τη διάταξη της προηγούμενης περίπτωσης (μπροστά από το βασικό εξυπηρετητή), είναι οι επιταχυντές (accelerators). Ο τελικός εξυπηρετητής μπορεί να είναι ένας πληρεξούσιος αλλά συνήθως είναι ένας πραγματικός εξυπηρετητής. Ο σκοπός εδώ είναι ο εξής: ο επιταχυντής θα εκτελέσει γρήγορη επαναποθήκευση και I/O έτσι ώστε οι αιτήσεις να εξυπηρετούνται συχνότερα από τη κρυφή μνήμη του επιταχυντή αντί να προωθούνται στο πιο αργό τελικό εξυπηρετητή.

Ο επιταχυντής όταν παρουσιάστηκε αποτέλεσε μία καλή λύση για τα λογισμικά προγράμματα των εξυπηρετητών του Διαδικτύου σήμερα όμως λόγω των πολλών ανακατευθύνσεων που εκτελεί ίσως να κάνει τον εξυπηρετητή να φαίνεται πιο αργός.

Μερικές αρχιτεκτονικές εξυπηρετητών του παγκόσμιου ιστού χρησιμοποιούν τους επιταχυντές εσωτερικά στη διάταξή τους, βασικά για να επαναποθηκεύσουν προηγούμενες απαντήσεις και να τις ξαναχρησιμοποιήσουν για κάθε ίδια αίτηση. Άλλες, θέλουν τον επιταχυντή να τοποθετείται μπροστά από τον τελικό πληρεξούσιο επιταχυντή με σκοπό πληρέστερου φιλτραρίσματος του περιεχομένου των εισερχόμενων πακέτων.

Ένας τομέας που αναπτύσσεται με πολύ γρήγορους ρυθμούς είναι οι αρχιτεκτονικές πληρεξούσιων εξυπηρετητών που συνδυάζουν τη δυναμική επεξεργασία με τη κατανομή της πληροφορίας (active proxies). Μερικές εφαρμογές που έχουν αναπτυχθεί είναι:

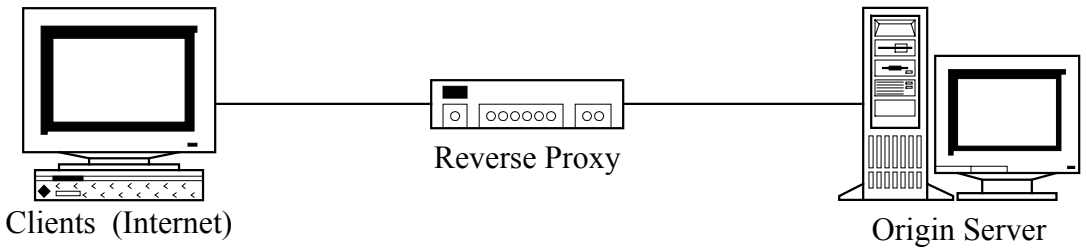
- Ο Crit-Link Mediator όπου ο κάθε χρήστης που κατεβάζει μία σελίδα, βλέπει στο πλαίσιό της σχόλια που αφορούν τη συγκεκριμένη σελίδα και έχουν γραφεί από προηγούμενους χρήστες που την ανάγνωσαν.
- Ο Lucent Personalized Web Assistant που αντικαθιστά τη πραγματική ταυτότητα του χρήστη με ένα ψευδώνυμο. Χρησιμοποιώντας τα χαραχτηριστικά ασφαλείας του HTTP οι χρήστες του εσωτερικού δικτύου συνδέονται με το πληρεξούσιο εξυπηρετητή ο οποίος επιτρέπει σ' αυτούς να έχουν πρόσβαση σε διάφορα sites αλλά χωρίς να είναι ορατή η ταυτότητά τους. Δύο σχετικές εφαρμογές είναι ο Anonymizer που αποκόπτει τα cookies, scripts κλπ και ο NoShit αποκόπτει τα γραφικά των διαφημίσεων.
- Content Translators όπου οι σελίδες μεταφράζονται στο επίπεδο των πληρεξούσιων και

• ο Remote Processing όπου για παράδειγμα τα Java applets εκτελούνται στο πληρεξούσιο εξυπηρετητή και τα αποτελέσματά του μόνο φαίνονται στους χρήστες.

4. Αντίστροφοι (Reverse)

Όταν ο πληρεξούσιος εξυπηρετητής λειτουργεί κατά τέτοιο τρόπο που να φαίνεται στους πελάτες σαν ένας κανονικός εξυπηρετητής του παγκόσμιου ιστού τότε αυτός ονομάζεται αντίστροφος πληρεξούσιος εξυπηρετητής (reverse proxy server). Έτσι οι πελάτες που συνδέονται μ' αυτόν θεωρούν ότι πρόκειται για το πραγματικό εξυπηρετητή, χωρίς να γνωρίζουν αν οι αιτήσεις προωθηθούν περαιτέρω σε κάποιον άλλο εξυπηρετητή ή μέσω άλλων πληρεξούσιων εξυπηρετητών.

Στη περίπτωση αυτή, ο πληρεξούσιος εξυπηρετητής εξυπηρετεί τις αιτήσεις των πελατών εκ μέρους του εξυπηρετητή του Διαδικτύου. Συνήθως ανήκει στον ίδιο οργανισμό ή επιχείρηση που διαχειρίζεται τον κύριο πραγματικό εξυπηρετητή. Αν θέλαμε να παραστήσουμε σχηματικά τη διάταξη αυτή θα παίρναμε το παρακάτω σχήμα:



Οι εξυπηρετητές αυτοί έχουν του εξής βασικούς σκοπούς:

- Αντιγραφή (replication) του περιεχομένου σε γεωγραφικά κατανεμημένες περιοχές.
- Αντιγραφή (replication) του περιεχομένου για load balancing.

Ειδικότερα, οι αντίστροφοι πληρεξούσιοι εξυπηρετητές μπορούν να χρησιμοποιηθούν για την εγκατάσταση πιστών αντιγράφων ενός κεντρικού εξυπηρετητή σε διαφορετικές και κατανεμημένες γεωγραφικές περιοχές. Ένα κλασσικό παράδειγμα είναι μία εταιρία με πολλά υποκαταστήματα σε διαφορετικά μέρη του κόσμου.

Ας υποθέσουμε ότι η εταιρία μας που βρίσκεται στη Θεσσαλονίκη, έχει ένα κεντρικό εξυπηρετητή που όλοι οι υπάλληλοι της εταιρίας χρησιμοποιούν για να παίρνουν τις πληροφορίες που χρειάζονται. Ένας αντίστροφος εξυπηρετητής μπορεί να εγκατασταθεί σε κάθε υποκατάστημα του κόσμου. Οι υπάλληλοι αυτών των γραφείων θα χρησιμοποιούν τον αντίστροφο εξυπηρετητή σαν να ήταν ο κεντρικός.

Δηλαδή, αν η διεύθυνση του κεντρικού ήταν *http://www.mycompany.com/*, τότε οι διευθύνσεις των υποκαταστημάτων θα ήταν:

http://www-rome.mycompany.com/ ,
http://www-athens.mycompany.com/
http://www-iraklio.mycompany.com/

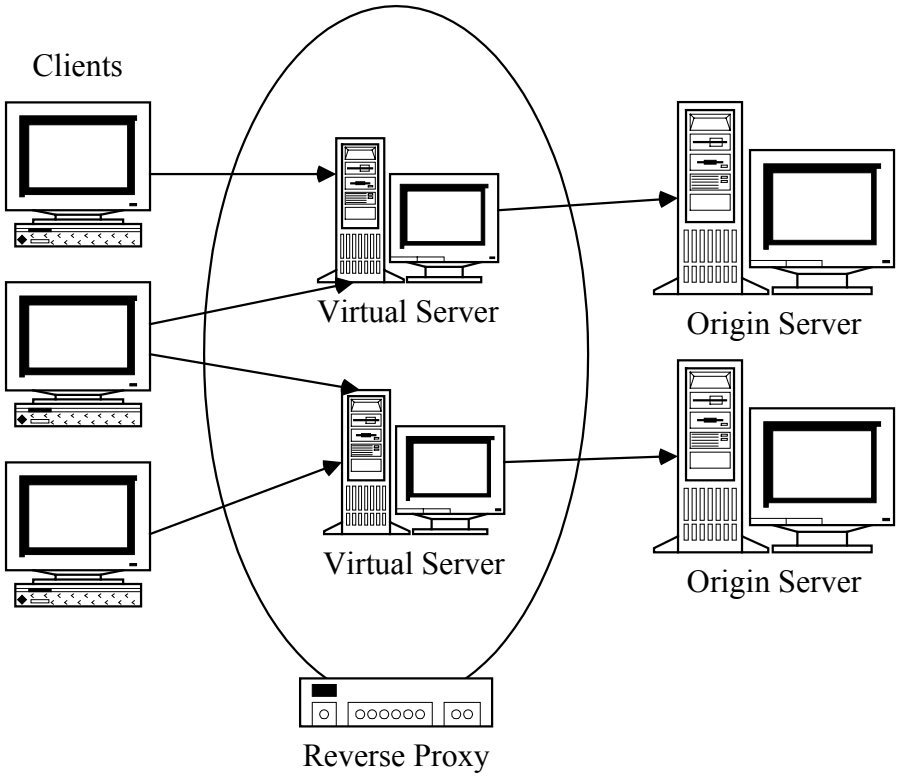
Βέβαια, εσωτερικά κάθε εξυπηρετητής που είναι ακριβές αντίγραφο του κεντρικού είναι ρυθμισμένος να λαμβάνει όλα τα περιεχόμενά του από το κεντρικό στη *www.mycompany.com*.

Οι αντίστροφοι πληρεξούσιοι εξυπηρετητές επίσης μπορεί να χρησιμοποιηθούν για να μοιράσουν το φόρτωμα σε εξυπηρετητές που παρουσιάζουν μεγάλη κίνηση. Οι αιτήσεις των πελατών κατανέμονται σε πολλαπλούς εξυπηρετητές χρησιμοποιώντας μεθόδους και τεχνικές για ομοιόμορφη κατανομή του φόρτου σ' αυτούς.

Ένας από τους εξυπηρετητές είναι ένας κανονικός εξυπηρετητής που λειτουργεί ως κύριος εξυπηρετητής. Το περιεχόμενο των δεδομένων αλλάζει μόνο σ' αυτό τον κύριο εξυπηρετητή. Οι υπόλοιποι εξυπηρετητές είναι αντίστροφοι πληρεξούσιοι που παίρνουν τα δεδομένα τους από τον κύριο εξυπηρετητή.

Αυτοί οι reverse εξυπηρετητές επαναποθηκεύουν τις ζητούμενες πληροφορίες στη κρυφή τους μνήμη και σύντομα οι περισσότερες αιτήσεις εξυπηρετούνται από αυτούς. Λόγω της λειτουργίας των reverse proxies επειδή εξυπηρετούν ένα ή περισσότερους πραγματικούς εξυπηρετητές μπορούν να αποθηκεύσουν όλες τις ζητούμενες URL's χωρίς να υπάρχει κίνδυνος να εξαντληθεί η κρυφή τους μνήμη. Γι' αυτό και η πιθανότητα για cache hit είναι 100%.

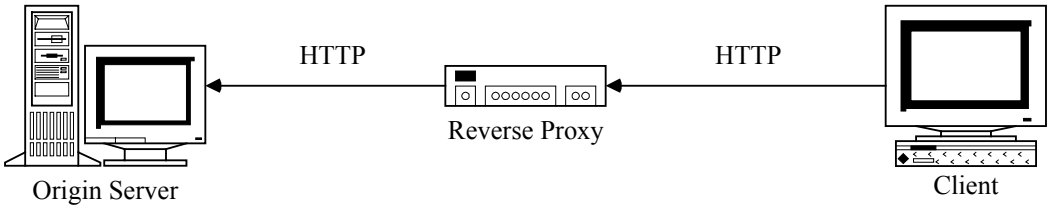
Μία ενδιαφέρουσα αρχιτεκτονική φαίνεται στο παρακάτω σχήμα:



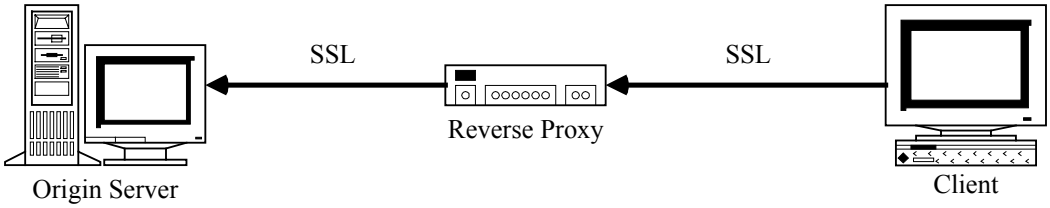
Η αρχιτεκτονική αυτή βασίζεται στη δυνατότητα να εμφανίζεται ο πληρεξούσιος εξυπηρετητής από τη μεριά του εξυπηρετητή του δικτύου, ως πολλαπλοί εικονικοί εξυπηρετητές. (Ο πληρεξούσιος εξυπηρετητής έχοντας δύο DNS aliases αντιστοιχεί σε καθεμιά από αυτές στους δύο πραγματικούς εξυπηρετητές).

Για να αυξηθεί η ασφάλεια μπορούμε αντί του HTTP πρωτοκόλλου να χρησιμοποιήσουμε το HTTPS πρωτόκολλο. Οι παρακάτω αρχιτεκτονικές περιγράφουν διαφορετικούς συνδυασμούς ασφαλών και μη συνδέσεων.

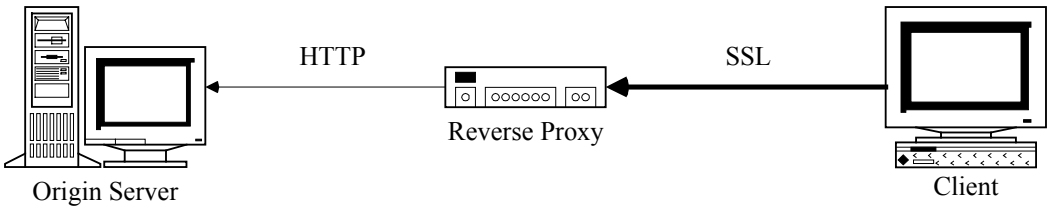
α. Μη ασφαλής αρχιτεκτονική.



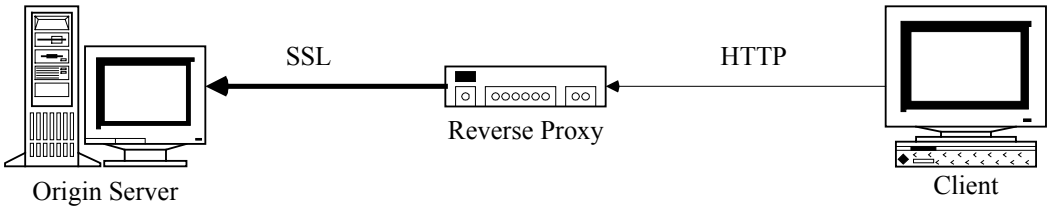
β. Αρχιτεκτονική που προσφέρει πλήρη ασφάλεια. Χρησιμοποιείται SSL για όλες τις εισερχόμενες και εξερχόμενες συνδέσεις.



γ. Αρχιτεκτονική που προσφέρει ασφάλεια για τις εισερχόμενες reverse proxy συνδέσεις. Σ' αυτή την εγκατάσταση ο εξυπηρετητής επιτρέπει τη κρυπτογράφηση των δεδομένων μεταξύ του πελάτη και του πληρεξούσιου εξυπηρετητή (το δίκτυο μεταξύ του αντίστροφου πληρεξούσιου και του πραγματικού εξυπηρετητή θεωρείται εξ' ορισμού ασφαλές).



δ. Αρχιτεκτονική που προσφέρει ασφάλεια μεταξύ του αντίστροφου και του πραγματικού εξυπηρετητή αλλά οι συνδέσεις των πελατών είναι μη ασφαλής. Μία αρχιτεκτονική που δεν εφαρμόζεται.



Συνοψίζοντας, οι πληρεξούσιοι εξυπηρετητές από τη μεριά του πραγματικού εξυπηρετητή αποτελούν μία πολύ καλή εναλλακτική λύση. Όσο μάλιστα η απόδοσή τους ανεβαίνει θα αποτελέσουν την καλύτερη λύση για το συγχρονισμό του περιεχομένου ανάμεσα σε ένα μεγάλο αριθμό εξυπηρετητών που βρίσκονται στην ίδια ή σε διαφορετικές γεωγραφικές περιοχές.

1.3. Circuit-Level

Ένας ακόμη συνηθισμένος τύπος είναι οι εξυπηρετητές επιπέδου-κυκλώματος (circuit-level proxy servers), που αποτελούν ένα πρόγραμμα λογισμικού στο επίπεδο συνόδου. Στη περίπτωση αυτή ο εξυπηρετητής δημιουργεί τη σύνδεση με την εφαρμογή που την απαιτεί και μετά απλά προωθεί τα δεδομένα προς και τις δύο κατευθύνσεις της σύνδεσης χωρίς να εμπλέκει το πρωτόκολλο επιπέδου εφαρμογής.

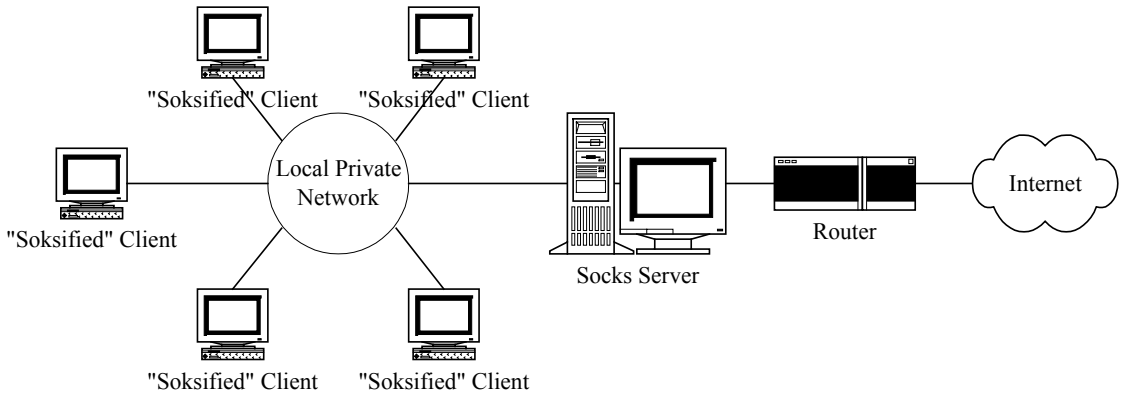
Αυτοί οι πληρεξούσιοι εξυπηρετητές βρίσκονται θεωρητικά ανάμεσα στους δρομολογητές και τους εξυπηρετητές επιπέδου εφαρμογής και αυτό διότι όπως αναφέραμε λειτουργούν στο επίπεδο συνόδου, σε υψηλότερο δηλαδή επίπεδο από αυτό των δρομολογητών και σε χαμηλότερο επίπεδο από τους εξυπηρετητές που βρίσκονται στο επίπεδο εφαρμογής. Το πιο διαδεδομένο και εφαρμοζόμενο πρωτόκολλο αυτής της αρχιτεκτονικής είναι το SOCKS.

Το SOCKS είναι ένα proxy πρωτόκολλο δικτύου που επιτρέπει τους χρήστες που βρίσκονται από τη μία μεριά του SOCKS εξυπηρετητή να έχουν πλήρη πρόσβαση με τους χρήστες που βρίσκονται από την άλλη μεριά χωρίς να απαιτείται από το χρήστη η απευθείας πρόσβαση με τη IP διεύθυνση. Ο SOCKS αναλαμβάνει να κάνει τη σύνδεση με αυτή την IP διεύθυνση και να ανακατευθύνει τις απαντήσεις προς το χρήστη, ενώ ταυτόχρονα εμποδίζει μη επιτρεπόμενες προσβάσεις που προέρχονται από το Διαδίκτυο προς τους εσωτερικούς χρήστες.

Το SOCKS πρωτόκολλο αναπτύχθηκε από τον David Koblas το 1990. Αμέσως μετά ήταν διαθέσιμο στην αγορά από τη εταιρία NEC's η έκδοση v4. Ακολούθησε η έκδοση v5, που επιπλέον παρείχε αναγνώριση ταυτότητας και υποστήριξης UDP.

Γενικά, η τεχνολογία SOCKS βασίζεται σε SOCKS εξυπηρετητές μέσω των οποίων περνάει όλη η κυκλοφορία του δικτύου. Αυτοί επιτρέπουν τις εφαρμογές των πελατών που χρησιμοποιούν το SOCKS πρωτόκολλο, να επικοινωνούν με τους SOCKS εξυπηρετητές και να κάνουν εφικτές τις συνδέσεις μεταξύ εσωτερικών και εξωτερικών δικτύων (Διαδίκτυο). Μπορούν να χρησιμοποιηθούν ακόμη για συνδέσεις συστημάτων που αποτελούν τμήματα εσωτερικών δικτύων.

Το επόμενο διάγραμμα περιγράφει μία συνηθισμένη εγκατάσταση που εφαρμόζει τη τεχνολογία SOCKS:



Συνοπτικά ο SOCKS v5 εκτελεί τις ακόλουθες βασικές λειτουργίες:

♦ Αναγνώριση ταυτοτήτων των πελατών.

♦ Διαχείριση των αιτήσεών τους.

♦ Δημιουργία ενός κυκλώματος εξυπηρετητή (proxy circuit) μεταξύ του πελάτη (application client) και του εξυπηρετητή (application server).

♦ Μεταγωγή πακέτων δεδομένων μεταξύ αυτών.

Τα βασικά πλεονεκτήματα του πρωτοκόλλου SOCKS v5 είναι:

♦ Ισχυρή αναγνώριση ταυτότητας.

♦ Ακεραιότητα και ασφάλεια δεδομένων.

♦ Υποστήριξη UDP.

♦ Υποστήριξη λειτουργίας σε πολλές "πλατφόρμες" (Unix, NT κλπ) και σε διαφορετικά περιβάλλοντα.

Γενικά, οι εξυπηρετητές στο επίπεδο εφαρμογής έχουν το μειονέκτημα να είναι ικανοί να διαχειριστούν μόνο ένα ή μία ομάδα λίγων καθορισμένων πρωτοκόλλων. Νέα ή λιγότερο γνωστά πρωτόκολλα δεν είναι δυνατό να εκτελεστούν χωρίς αναβάθμιση του λογισμικού (στη περίπτωση αυτή για κάθε νέο πρωτόκολλο θα πρέπει να χρησιμοποιήσουμε πρώτα ένα πληρεξούσιο εξυπηρετητή επιπέδου κυκλώματος (SOCKS) ή απλά να να προγραμματίσουμε τους δρομολογητές να επιτρέπουν τέτοιες συνδέσεις. Όταν αυτό το πρωτόκολλο θα υποστηριχθεί από ένα πληρεξούσιο επιπέδου εφαρμογής μπορεί να γίνει η αλλαγή. Από τη στιγμή αυτή θα έχουμε καλύτερο φιλτράρισμα και καλύτερους μηχανισμούς ελέγχου και πρόσβασης). Αυτό είναι το σημείο στο οποίο υπερτερούν οι εξυπηρετητές επιπέδου κυκλώματος. Από την άλλη βέβαια παρουσιάζουν το μειονέκτημα της μικρότερης ικανότητας

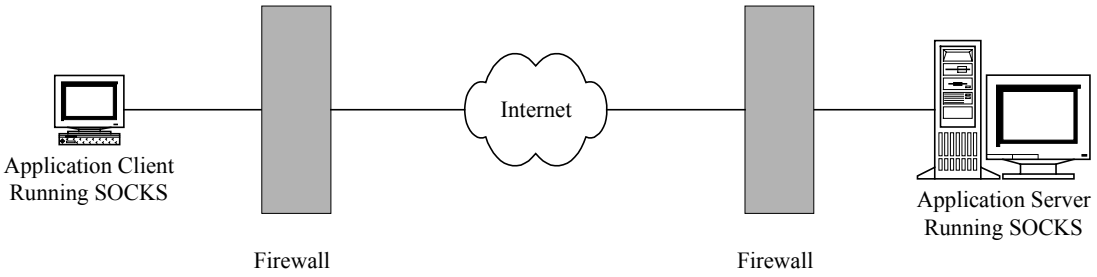
φιλτραρίσματος. Οι SOCKS και οι πληρεξούσιοι εξυπηρετητές επιπέδου εφαρμογής δεν αποτελούν ο ένας υποκατάστατο του άλλου. Αντίθετα συμπληρώνει ο ένας τον άλλο.

Η παρακάτω λίστα περιγράφει τις τρεις κύριες περιοχές όπου το SOCKS μπορεί να εφαρμοσθεί:

- *Εφαρμογές*,
- *Φράγματα* και
- *Διαχειριστές πρόσβασης στο Διαδίκτυο (Internet Access Managers).*

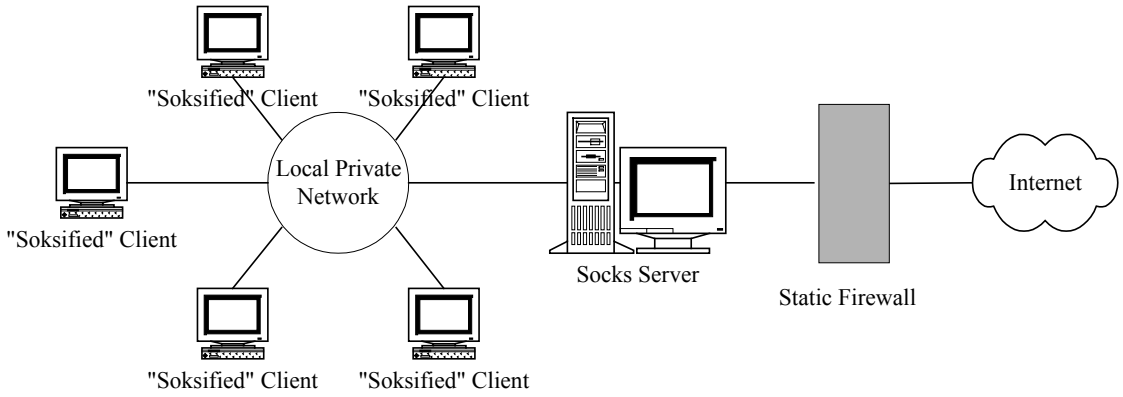
Αναλυτικότερα,

Στο πιο βασικό επίπεδο η τεχνολογία SOCKS v5 μπορεί να χρησιμοποιηθεί με εφαρμογές. Το πρωτόκολλο SOCKS v5 μπορεί να ενσωματωθεί σε εφαρμογές πελάτη-εξυπηρετητή.

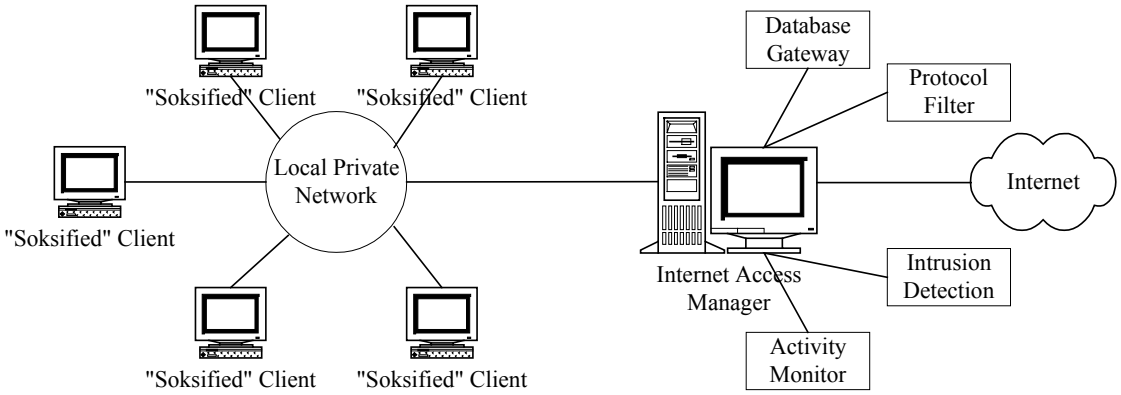


Οι εταιρίες Microsoft και Netscape χρησιμοποιούν SOCKS τεχνολογία ενσωματωμένη στις εμπορικές εφαρμογές των συστημάτων πλοήγησης Internet Explorer και Netscape Navigator αντίστοιχα.

Σε ένα υψηλότερο επίπεδο, το πρωτόκολλο SOCKS v5 μπορεί να χρησιμοποιηθεί εύκολα με φράγματα δίνοντας τη δυνατότητα στους διαχειριστές των δικτύων να εγκαταστήσουν δυναμικούς και αυστηρούς κανόνες καθώς και αποτελεσματικές πολιτικές ασφαλείας.



Τέλος, το πρωτόκολλο SOCKS v5 μπορεί να χρησιμοποιηθεί ως διαχειριστής πρόσβασης στο Διαδίκτυο προσφέροντας στους διαχειριστές να εφαρμόσουν αποτελεσματικούς κανόνες ασφαλείας και χρήσης. Η εγκατάσταση γίνεται ως εξής: με το πρωτόκολλο SOCKS v5 ως βάση, οι διαχειριστές των δικτύων μπορούν να εγκαταστήσουν πληρεξούσιους εξυπηρετητές που είναι εφοδιασμένοι με εργαλεία διαχείρισης όπως μόνιτορ πραγματικού χρόνου, φίλτρα πρωτοκόλλων, θύρες βάσης δεδομένων κλπ, όπως φαίνεται στο παρακάτω σχήμα:



2. Πλεονεκτήματα και Μειονεκτήματα

Παραπάνω αναλύθηκαν οι βασικοί τύποι και οι αρχιτεκτονικές που βασίζονται σ' αυτές, δόθηκαν τα βασικά χαραχτηριστικά και τα ιδιαίτερα πλεονεκτήματα και μειονεκτήματά τους καθώς και ο σκοπός χρησιμοποίησής τους. Αν θέλαμε τώρα να συνοψίσουμε τα βασικά πλεονεκτήματα και μειονεκτήματα των πληρεξούσιων εξυπηρετητών στο επίπεδο εφαρμογής, συνόδου και φιλτραρίσματος πακέτων θα είχαμε τους εξής πίνακες:

2.1. Φιλτράρισμα πακέτων

ΠΛΕΟΝΕΚΤΗΜΑΤΑ	ΜΕΙΟΝΕΚΤΗΜΑΤΑ
Εύκολη εφαρμογή: Η εγκατάσταση του συστήματος φιλτραρίσματος πακέτων είναι σχετικά απλή γιατί στηρίζεται στους ήδη υπάρχοντες δρομολογητές του δικτύου.	**Τα μετακινούμενα δεδομένα είναι εκτεθειμένα**: στις απευθείας συνδέσεις των χρηστών μεταξύ των δικτύων, με εργαλεία όπως για παράδειγμα τα packet-sniffers τα οποία έχουν τη δυνατότητα της άμεσης πρόσβασης σε πληροφορίες που περιλαμβάνονται στα πακέτα.
Υψηλή ταχύτητα: λόγω της απευθείας σύνδεσης μεταξύ εσωτερικών και εξωτερικών στοιχείων τα δεδομένα μεταφέρονται με υψηλές ταχύτητες.	**Δυσκολία προσαρμογής και εφαρμογής**: πιο σύνθετων κανόνων ασφαλείας σε χρήστες, ανάλογα με τα διαιώματα που επιθυμούμε να παραχωρήσουμε σ' αυτούς.
Η λειτουργία της ασφάλειας δεν γίνεται ορατή από τους τελικούς χρήστες: επειδή αυτή υλοποιείται στο επίπεδο του δρομολογητή του δικτύου. Συνέπεια αυτού είναι η ευκολότερη χρήση των εφαρμογών του πελάτη-χρήστη.	**Μη δυνατότητα αναγνώρισης ταυτότητας**: των πακέτων από συγκεκριμένους χρήστες.

2.2. Πληρεξούσιοι εξυπηρετητές στο επίπεδο εφαρμογής

Τα βασικά πλεονεκτήματα και μειονεκτήματα των πληρεξούσιων εξυπηρετητών στο επίπεδο εφαρμογής συνοψίζονται στον επόμενο πίνακα:

ΠΛΕΟΝΕΚΤΗΜΑΤΑ	ΜΕΙΟΝΕΚΤΗΜΑΤΑ
Ασφάλεια διευθύνσεων: Δεν υπάρχει απευθείας σύνδεση μεταξύ πελατών και εξυπηρετητών στο επίπεδο εφαρμογής. Οι πληρεξούσιοι εξυπηρετητές κρύβουν τη δομή διευθυνσοποίησης του εσωτερικού δικτύου και καθιστούν δύσκολο για αυτούς που επιθυμούν να εισβάλλουν σ' αυτό.	**Απαιτείται τροποποίηση προγράμματος**: στο πρόγραμμα λογισμικού από τη μεριά του πελάτη με αποτέλεσμα επιπλέον κόστος.
Ασφάλεια στο επίπεδο εφαρμογής: η επικοινωνία γίνεται χρησιμοποιώντας πρωτόκολλα στο επίπεδο εφαρμογής. Αυτό σημαίνει ότι ο εξυπηρετητής γνωρίζοντας καλά το πρωτόκολλο, μπορεί να ασκήξει αυστηρό έλεγχο στα μεταφερούμενα δεδομένα.	**Απαιτείται επικύρωση ταυτοποίησης**: για κάθε επίπεδο εφαρμογής. Αυτό σημαίνει ότι κάθε φορά που ο χρήστης ξεκινά μία νέα εφαρμογή απαιτείται νέα επικύρωση ταυτότητας (reauthentication) επιπέδου εφαρμογής.
Έλεγχος κυκλοφορίας στο επίπεδο εφαρμογής: επειδή η κυκλοφορία γίνεται μέσω του πληρεξούσιου εξυπηρετητή επιτρέπει στους διαχειριστές του δικτύου να ασκούν ολοκληρωμένο έλεγχο σ' αυτό.	**Απαιτείται δημιουργία νέου λογισμικού**: στη περίπτωση εισαγωγής νέων υπηρεσιών στο Διαδίκτυο, για κάθε εφαρμογή και για κάθε υπηρεσία.

2.3. Πληρεξούσιοι εξυπηρετητές στο επίπεδο συνόδου

Τα βασικά πλεονεκτήματα και μειονεκτήματα των πληρεξούσιων εξυπηρετητών στο επίπεδο συνόδου συνοψίζονται στον επόμενο πίνακα:

ΠΛΕΟΝΕΚΤΗΜΑΤΑ	ΜΕΙΟΝΕΚΤΗΜΑΤΑ
Ασφάλεια διευθύνσεων: δεν υπάρχει απευθείας σύνδεση μεταξύ πελατών και εξυπηρετητών στο επίπεδο εφαρμογής. Οι πληρεξούσιοι εξυπηρετητές κρύβουν τη δομή διευθυνσοποίησης του εσωτερικού δικτύου και καθιστούν δύσκολο για αυτούς που επιθυμούν να εισβάλλουν σ' αυτό.	**Απαιτείται τροποποίηση προγράμματος**: στο πρόγραμμα λογισμικού από τη μεριά του πελάτη και κυρίως στη μεταγωγή όλων των δεδομένων στο TCP/IP.
Μεγάλη προσαρμοστικότητα: επειδή οι πληρεξούσιοι εξυπηρετητές λειτουργούν στο επίπεδο συνόδου υποστηρίζουν ένα πολυ-πρωτοκολλικό περιβάλλον. Αν νέες υπηρεσίες προστεθούν στο Διαδίκτυο οι πληρεξούσιοι εξυπηρετητές θα τις υποστηρίξουν.	**Δε παρέχει συγκεντρωτική ασφάλεια και έλεγχο**: στον ίδιο βαθμό με τους πληρεξούσιους εξυπηρετητές στο επίπεδο πρωτοκόλλου.
Έλεγχος και ανάλυση κυκλοφορίας στο επίπεδο εφαρμογής: επειδή η κυκλοφορία γίνεται μέσω του πληρεξούσιου εξυπηρετητή επιτρέπει στους διαχειριστές του δικτύου να ασκούν ολοκληρωμένο έλεγχο σ' αυτό.	

Δ. ΔΙΑΧΕΙΡΙΣΗ

Η διαχείριση (*management*) των πληρεξούσιων εξυπηρετητών θα πρέπει να βασίζεται στους παρακάτω παράγοντες και να απαντά στα ερωτήματα που αυτοί καθορίζουν:

1. Για την επιλογή του κατάλληλου ή των κατάλληλων πληρεξούσιων εξυπηρετητών πρέπει να ληφθούν υπόψη οι εξής παράμετροι και ειδικότερα:

√ Τι δυνατότητες και λειτουργίες θέλουμε να μας παρέχει.

√ Ποιο το εκτιμώμενο φόρτωμα.

√ Ποιος ο εκτιμώμενος χρόνος εξυπηρέτησης ενός πελάτη.

√ Πως θα επιλέξουμε το τύπο λογικού και λογισμικού, από τα διαθέσιμα εμπορικά πακέτα και εφαρμογές.

√ Εμπορικές εφαρμογές.

2. Για την εγκατάσταση των πληρεξούσιων εξυπηρετητών πρέπει να αναλυθούν οι παρακάτω παράμετροι:

√ Πως και με ποια ιεραρχία θα κατανείμουμε τους πληρεξούσιους εξυπηρετητές,

√ Πως θα γίνεται η ανάθεση των εξυπηρετητών σε κάθε πελάτη.

3. Για την καλύτερη λειτουργία η οποία θα οδηγήσει σε καλύτερα αποτελέσματα πρέπει να εκτιμηθούν και να αναλυθούν οι εξής παράμετροι:

√ Πως θα διαμορφώσουμε τη κρυφή μνήμη των πληρεξούσιων εξυπηρετητών,

√ Πως θα επιτύχουμε καλύτερη και ομοιόμορφη κατανομή του φορτίου,

√ Πως θα επιτύχουμε τη βέλτιστη απόδοση στις παρακάτω λειτουργίες:

√ Φιλτράρισμα,

√ Έλεγχος κυκλοφορίας,

√ Έλεγχος πρόσβασης και

√ Ασφάλεια.

4. Τέλος, ποιες είναι οι απαραίτητες ενέργειες για να αντιμετωπισθούν τα τυχόν προβλήματα που θα παρουσιαστούν.

1. Επιλογή, Γενικά και Ειδικά Χαρακτηριστικά

1.1. Δυνατότητες και λειτουργίες

Ο πρώτος στόχος είναι να προσδιορίσουμε τους λόγους για τους οποίους απαιτείται η εγκατάσταση των πληρεξούσιων εξυπηρετητών. Με λίγα λόγια τί δυνατότητες και λειτουργίες μπορούν να μας προσφέρουν. Αυτές μπορούν να συνοψιστούν στις εξής:

- **Επαναποθήκευση.** Η επαναποθήκευση έχει δύο βασικά πλεονεκτήματα:

– Αυξάνει την απόδοση επαναποθηκεύοντας τη ζητούμενη πληροφορία, μειώνοντας το χρόνο απάντησης και μειώνοντας τον αριθμό των αιτήσεων γι' αυτήν σε απομακρυσμένους εξυπηρετητές. Τα ωφέλη είναι μεγαλύτερα σε συστήματα που παρουσιάζουν μεγάλη κυκλοφορία πληροφορίας. Όσο ο αριθμός των αιτήσεων αυξάνει, τόσο αυξάνει η πιθανότητα να υπάρχει αυτή στη κρυφή μνήμη (cache hit). Πρέπει επίσης να λάβουμε υπόψη ότι καλύτερη εκμετάλευση της επαναποθήκευσης έχουμε όσο αυτή υλοποιείται πιο κοντά στο πελάτη γιατί όσο πιο κοντά προς το πελάτη γίνει το cache hit τόσο πιο γρήγορα θα λάβει αυτός την απάντηση.

– Διατηρεί το διαθέσιμο εύρος στη σύνδεση προς το Διαδίκτυο σε χαμηλά επίπεδα γεγονός ιδιαίτερα σημαντικό όταν το διαθέσιμο εύρος ζώνης είναι περιορισμένο με συνέπεια να υπάρχει κίνδυνος συμφόρησης (bottleneck) ή η σύνδεση με το Διαδίκτυο είναι ακριβή. Επίσης, στη περίπτωση των μισθωμένων γραμμών δεν απαιτείται η αγορά επιπρόσθετων γραμμών.

- **Ασφάλεια.** Αντί να επιτρέπουμε την ελεύθερη κίνηση των πακέτων από και προς το εσωτερικό μας δίκτυο, είναι ασφαλέστερο να έχουμε ένα πληρεξούσιο εξυπηρετητή που ασκεί αυστηρές πολιτικές ασφαλείας. Ακόμα και αν έχουμε ένα φράγμα πάλι υφίσταται η ανάγκη για ένα πληρεξούσιο εξυπηρετητή που λειτουργεί σε επίπεδο εφαρμογών για να ασκηθεί ένας πιο ευέλικτος και έξυπνος έλεγχος.

- **Φιλτράρισμα.** Οι πληρεξούσιοι εξυπηρετητές μας παρέχουν πολλές δυνατότητες φιλτραρίσματος. Συγκεκριμένα μπορούμε να φιλτράρουμε αιτήσεις και να αποκλείσουμε αυτές που θεωρούμε ακατάλληλες, δηλαδή αιτήσεις που απαιτούν ακατάλληλο υλικό. Μπορούμε επίσης να εκτελέσουμε λειτουργίες φιλτραρίσματος με βάση τις ζητούμενες URL's ή τέλος μπορούμε να φιλτράρουμε το περιεχόμενο των απαντήσεων και να ελέξουμε για ιούς, Trojan Horses, ή applets.

- **Έλεγχος πρόσβασης.** Ο έλεγχος πρόσβασης μας παρέχει ένα τρόπο να επιτρέψουμε ή όχι επιλεκτικά, συγκεκριμένους χρήστες, τμήματα ή υποδίκτυα να έχουν πρόσβαση στο Διαδίκτυο. Η αναγνώριση ταυτότητας επίσης μας δίνει τη δυνατότητα, χρησιμοποιώντας

πληρεξούσιους εξυπηρετητές, να καθορίσουμε συγκεκριμένες πολιτικές πρόσβασης σε συγκεκριμένους χρήστες και ομάδες εργασίας.

• **Έλεγχος - Παρακολούθηση κυκλοφορίας.** Με την αναγνώριση ταυτότητας μπορούμε όχι μόνο να αποκλείσουμε από μερικούς χρήστες ή ομάδες εργασιών τη πρόσβαση σε συγκεκριμένα έγγραφα ή αρχεία, αλλά και να καθορίσουμε τη δυνατότητά τους στο διαθέσιμο ποσοστό κυκλοφορίας. Επίσης να βγάλουμε χρήσιμα συμπεράσματα για την απόδοση του πληρεξούσιου εξυπηρετητή, όπως το cache hit rate, ποιες είναι οι ώρες που δέχεται τις περισσότερες αιτήσεις, πόσο χρόνο χρειάζεται για να απαντηθεί η αίτηση ενός πελάτη και πολά άλλα, ώστε να βγάλουμε χρήσιμα συμπεράσματα για την απόδοσή του αλλά και την απόδοση του δικτύου γενικότερα, ποιες είναι οι περισσότερο ζητούμενες διευθύνσεις κ.ά.

1.2. Εκτιμώμενο φόρτωμα

Αφού καθορίσουμε το σκοπό ή τους σκοπούς του πληρεξούσιου εξυπηρετητή, θα πρέπει να εκτιμήσουμε το ποσό του φορτώματος που θα τεθεί να διαχειριστεί. Ο προσδιορισμός αυτός θα γίνει πιο εύκολα αν είναι γνωστά τα παρακάτω:

• **Ο αριθμός των χρηστών**, ο οποίος μπορεί να προσδιοριστεί με πολλούς τρόπους. Μπορεί να είναι ο συνολικός αριθμός των χρηστών ή ο συνολικός αριθμός των Η/Υ του δικτύου ή ακόμα καλύτερα ο αριθμός των χρηστών που χρησιμοποιούν το δίκτυο ταυτόχρονα σε μία δεδομένη χρονική στιγμή.

• **Η μελλοντική ανάπτυξη**, που μπορεί να καθοριστεί από τους στόχους του οργανισμού (επιχείρησης) και το ποια εκτιμούμε ότι θα είναι η μελλοντική εξέλιξη που προσφέρει η διαδικτύωσή του.

• **Το είδος της εργασίας**, δηλαδή αν ο παγκόσμιος ιστός αποτελεί ή θα αποτελέσει κεντρικό άξονα της εργασίας του οργανισμού. Αυτό συνεκτιμά δύο παραμέτρους, η χρήση του Διαδικτύου από τους υπαλλήλους του οργανισμού και ο εκτιμώμενος αριθμός των πελατών στο site του.

• **Το είδος των δεδομένων**, δηλαδή το μέγεθος των αρχείων που ζητούν οι πελάτες. Το μέσο μέγεθος των αντικειμένων του παγκόσμιου ιστού είναι 10-30 Kbytes. Αν είναι μεγαλύτερο από αυτό, το φόρτωμα θα αυξηθεί.

• **Ο αριθμός των προσβάσεων ανά δευτερόλεπτο, ώρα ή ημέρα**, δηλαδή ο καθορισμός του αριθμού των ταυτόχρονων αιτήσεων, σε συγκεκριμένες χρονικές περιόδους της ημέρας. Σε

ένα ήδη εγκατεστημένο πληρεξούσιο εξυπηρετητή αυτό μπορεί εύκολα να υπολογισθεί με τη βοήθεια διαφόρων βοηθημάτων. Για παράδειγμα ο proxy της Netscape χρησιμοποιεί το "sitemon" για να δώσει τον αριθμό των συνδέσεων.

• **Το μέγεθος των δεδομένων ανά δευτερόλεπτο, ώρα ή ημέρα**, όπου τρεις παράμετροι πρέπει να ληφθούν υπόψη:

– το ποσοστό των αιτήσεων που δεν εξυπηρετούνται από τη κρυφή μνήμη των πληρεξούσων εξυπηρετητών αλλά προωθείται προς τον απομακρυσμένο πραγματικό εξυπηρετητή και χρησιμοποιούν το εσωτερικό δίκτυο δύο φορές, μία για να προωθηθούν οι αιτήσεις προς το πραγματικό εξυπηρετητή και μία για τη μεταφορά των πακέτων από το πληρεξούσιο προς το πελάτη,

– το ποσοστό cache-hit, το οποίο σε συνάρτηση με τη προηγούμενη παράμετρο, έχει αποδειχθεί ότι είναι 30-60%, που σημαίνει ότι γι' αυτές τις αιτήσεις υπάρχει μία μοναδική μεταφορά στο δίκτυο, ενώ για το υπόλοιπο 70-40% αντίστοιχα, υπάρχει διπλή κυκλοφορία σ' αυτό και

– το διαθέσιμο εύρος του δικτύου που διαχειρίζεται τις εξερχόμενες συνδέσεις (η εξωτερική σύνδεση με το Διαδίκτυο) και αποτελεί το παράγοντα που περιορίζει το ποσό των δεδομένων που μεταφέρονται από τους πραγματικούς προς τους πληρεξούσιους εξυπηρετητές.

1.3. Εκτιμώμενος χρόνος εξυπηρέτησης

Ο χρόνος από τη στιγμή που η αίτηση έγινε δεκτή από τον πληρεξούσιο εξυπηρετητή μέχρι τη στιγμή που θα δεχθεί την επόμενη αίτηση, ονομάζεται χρόνος εξυπηρέτησης του πελάτη. Συνήθως το ποσοτικό μέγεθος που χρησιμοποιούμε για να εκτελέσουμε συγκρίσεις είναι ο αριθμός των αιτήσεων που μπορεί να διαχειριστεί ο εξυπηρετητής στη μονάδα του χρόνου (requests per second).

Αντίθετα από τους συνηθισμένους εξυπηρετητές που περιέχουν όλα τα δεδομένα "τοπικά", οι πληρεξούσιοι εξυπηρετητές πρέπει συνεχώς να συνδέονται με απομακρυσμένους εξυπηρετητές μέσω -πολλές φορές- αργών δικτύων. Μερικά τμήματα των αιτήσεων επίσης, προκαλούν επιπλέον καθυστερήσεις όπως για παράδειγμα οι σύνδεσμοι υπερκειμένων προς άλλα αντικείμενα με άκυρο ή μη ενεργό DNS ή ακόμα η κακή κατάσταση και εγκατάσταση των δικτύων.

Ο χρόνος εξυπηρέτησης της αίτησης του πελάτη εξαρτάται από τα εξής μεγέθη:

• *Ο αριθμός των αιτήσεων που άργησαν να εξυπηρετηθούν*, που επηρεάζει σημαντικά την ολική απόδοση του συστήματος ακόμα και αν είναι της τάξης του 5-10%.

• Η ρύθμιση που εκτελούμε όταν επιθυμούμε *επιπλέον σύνδεση (persistent connection)* με την ίδια διεύθυνση -ανεξάρτητα αν έχει ήδη εξυπηρετηθεί- προβλέποντας ότι πιθανό μία καινούρια αίτηση θα εξυπηρετηθεί από την ίδια διεύθυνση. Ακόμα και αν ο χρόνος αυτός είναι μικρός θα επηρεάσει σημαντικά την ολική απόδοση του συστήματος.

• *Ο χρόνος από τη στιγμή που ο πληρεξούσιος εξυπηρετητής δέχθηκε την αίτηση μέχρι να ολοκληρωθεί η ανάγνωσή της*. Αν η σύνδεση ξεκινήσει αλλά καμία αίτηση δε λαμβάνεται, η σύνδεση μετά από ένα ορισμένο χρονικό διάστημα θα τελειώσει. (Αυτό από πολλούς διαχειριστές αποτελεί ένδειξη ότι το εσωτερικό δίκτυο δέχεται τις προσπάθειες πρόσβασης σ' αυτό από το Διαδίκτυο).

• *Ο καθορισμός χρόνου της διάρκειας σύνδεσης με τον απομακρυσμένο πραγματικό εξυπηρετητή* που η πράξη έχει δείξει ότι δεν πρέπει να ξεπερνάει τα 15-30 δευτερόλεπτα. Αν έχουμε τη περίπτωση πληρεξούσιων εξυπηρετητών με πολλές IP διευθύνσεις, ο χρόνος αυτός πρέπει να είναι πολύ μικρός (2-3 δευτερόλεπτα).

• *Η οριοθέτηση του χρόνου απάντησης,* που βέβαια συνήθως δε καθορίζεται. Υπάρχουν όμως περιπτώσεις που το δίκτυό μας είναι πολύ αργό, οπότε σε μεγάλου μεγέθους έγγραφα θα πρέπει να καθορίσουμε ένα μέγιστο χρονικό διάστημα απάντησης.

• *Η υιοθέτηση και δημιουργία χρονικού διαστήματος* από τη στιγμή που ο πελάτης πατήσει το κουμπί STOP στο πρόγραμμα πλοήγησής του, όπου ο πληρεξούσιος εξυπηρετητής θα συνεχίσει να "κατεβάζει" το έγγραφο για να το επαναποθηκεύσει σε μελλοντική χρήση.

1.4. Επιλογή τύπου λογικού και λογισμικού

Όπως προαναφέραμε, δύο παράγοντες παίζουν σημαντικό ρόλο στην επιλογή του *λογικού* (hardware) και *λογισμικού* (software):

• *ο αριθμός των αιτήσεων ανά δευτερόλεπτο* και

• *το μέγεθος των δεδομένων που μεταφέρονται ανά δευτερόλεπτο*

υπολογισμένα και τα δύο τη χρονική περίοδο που έχουμε τη μεγαλύτερη δραστηριότητα και κυκλοφορία (peak load time). [1]

Πέρα από τα ανωτέρω και από αυτά που αναφέρθησαν στις προηγούμενες ενότητες υπάρχουν και άλλοι παράγοντες που επηρεάζουν την επιλογή αυτή και ουσιαστικά αντιστοιχούν στα φυσικά μεγέθη του συστήματος των πληρεξούσιων εξυπηρετητών.

Αυτά είναι:

1. Το μέγεθος της RAM που εξαρτάται από την αρχιτεκτονική του συστήματος.

2. Το μέγεθος του δίσκου.

3. Η ταχύτητα του δίσκου που θα πρέπει να είναι υψηλή και για να το πετύχουμε αυτό θα πρέπει να αφήνουμε ένα ποσοστό του μεγέθος του δίσκου ίσο με 10% ελεύθερο, να χρησιμοποιήσουμε SCSI controllers ή ακόμα και ένα σύστημα πολλαπλών δίσκων (disk array).

4. Το μέγεθος της κρυφής μνήμης που σύμφωνα με ένα πρακτικό κανόνα αντιστοιχεί σε κάθε χρήστη 5-40 MB.

5. Το μέγεθος της SWAP που εξαρτάται από την αρχιτεκτονική του εξυπηρετητή αλλά συνήθως ακολουθείται ο πρακτικός κανόνας που προτείνει να είναι διπλάσιος της RAM.

1.5. Εμπορικές εφαρμογές

Υπάρχουν πάρα πολλές εμπορικές εφαρμογές που είναι αδύνατο φυσικά να εξεταστούν και να αναλυθούν όλες. Στην ενότητα αυτή θα αναφερθούμε επιλεκτικά σε μερικές από αυτές -τις πέντε γνωστότερες-[2] και θα δοθούν τα πλεονεκτήματα και μειονεκτήματά τους καθώς και ένας γενικός κατάλογος με τις περισσότερες διαθέσιμες εμπορικές εφαρμογές[3] και τέλος μία γενική λίστα χαραχτηριστικών που μπορούν να αποτελέσουν κριτήρια επιλογής και σύγκρισης μεταξύ των πληρεξούσιων εξυπηρετητών.

- **Κατάλογος με τις πέντε γνωστότερες εμπορικές εφαρμογές**

1. *Microsoft Proxy Server 2.0.* Η εταιρεία Microsoft παρουσιάζει ένα εργαλείο που βασίζεται στο πρωτόκολλο το CARP. Παρέχει τη δυνατότητα array proxying, ώστε πολλοί

[1] Το μέγεθος αυτό στις περιπτώσεις των ISP αφορά ολόκληρες νύχτες ή Σαββατοκύριακα.

[2] Η σειρά με την οποία παρουσιάζονται είναι η σειρά αξιολόγησής τους, όπως δίδεται στο site της *www.internet.com.*

[3] Ο παραπάνω κατάλογος βρίσκεται στο site της *www.web-caching.com.*

υπολογιστές να έχουν πιθανά τη δυνατότητα να εξυπηρετήσουν μία κλήση (λειτουργεί σε πλατφόρμες NT)

2. *Netscape Proxy Server 3.5.* Η εταιρεία Netscape παρουσιάζει ένα ισχυρό και εύχρηστο εργαλείο. Η διαχείριση του εξυπηρετητή γίνεται μέσω WWW με απλό τρόπο ακόμα και σε πολύπλοκες διαδικασίες. Φιλτράρει τα URL, έχει τη δυνατότητα δημιουργίας χρηστών και ομάδες αυτών, περιορίζοντας την πρόσβαση σε συγκεκριμένα κείμενα ή sites, παρέχει δυνατότητα αντίστροφου πληρεξούσιου (Reverse Proxying). Υποστηρίζει το πρωτόκολλο ICP και CARP (λειτουργεί και σε πλαφόρμες UNIX και σε NT).

3. *Soft Router Plus* που η διαχείριση μπορεί να υλοποιηθεί εύκολα και με δυνατότητα υλοποίησής της από απόσταση (λειτουργεί και σε πλαφόρμες UNIX και σε NT).

4. *MidPoint Gateway, Sattellite, Companion, Teamer* ένα ολοκληρωμένο πακέτο που παρέχει ένα μεγάλο αριθμό λειτουργιών έχει δυνατότητες να φιλτράρει τα URL, περιορίζοντας την πρόσβαση σε συγκεκριμένα κείμενα ή sites, παρέχει τη δυνατότητα για trasparency, υποστηρίζει το πρωτόκολλο ICP (λειτουργεί σε πλατφόρμες UNIX, NT)

5. *Squid Internet Object Cache 2.0*. Ο εξυπηρετητής αυτός είναι από τους πλέον δημοφιλής. Διατίθεται ελεύθερα στο Διαδίκτυο και έχει σημαντική υποστήριξη από τη διεθνή δικτυακή κοινότητα. Χρησιμοποιεί ένα UDP πρωτόκολλο, το ICP για την επικοινωνία μεταξύ διαφόρων proxy cache εξυπηρετητών επιτυγχάνοντας τη μέγιστη ταχύτητα. Υποστηρίζει τον κλασσικό τρόπο σύνδεσης γονέα-παιδιού (parent relation) μεταξύ των εξυπηρετητών, αλλά και αδελφών (slibing relation), δηλαδή ο εξυπηρετητής ρωτά (μέσω του ICP) τους αδελφούς εάν έχουν το ζητούμενο αντικείμενο στην cache και σε θετική απάντηση το αντλούν από αυτούς. (Λειτουργεί σε πλατφόρμες UNIX)

- **Πλεονεκτήματα & Μειονεκτήματα των ανωτέρω**

  Microsoft Proxy Server 2.0.

Πλεονεκτήματα	Μειονεκτήματα
∗Υψηλή απόδοση ∗Αποτελεσματικές λειτουργίες φράγμα-τος ∗Δυναμικό φιλτράρισμα ∗Υποστήριξη RAS, VPN, SOCKS, HTTP1 ∗Υποστήριξη για πίνακες πολλαπλών εξυπηρετητών (proxy arrays) ∗Υποστήριξη για ιεραρχικό σχεδιασμό επαναποθήκευσης	∗Συμβατότητα μόνο με προϊόντα της Microsoft ∗Μη υποστήριξη UNIX ∗Μεγάλο κόστος αγοράς

2. Netscape Proxy Server 3.5

Πλεονεκτήματα	Μειονεκτήματα
∗Μεγάλες δυνατότητες φιλτραρίσματος ∗Συμβατότητα με UNIX ∗Δυναμικό φιλτράρισμα για ιούς ∗Υποστήριξη SOCKS v5 ∗Υποστήριξη για πίνακες πολλαπλών εξυ-πηρετητών (proxy arrays) ∗Υποστήριξη CARP	∗Σχετικά μειωμένες δυνατότητες και χαρα-χτηριστικά ασφαλείας

3. Soft Router Plus

Πλεονεκτήματα	Μειονεκτήματα
∗Εύκολο στην εγκατάσταση ∗Χρησιμοποίησή του σαν απλός TCP/IP δρομολογητής ∗Διαθέσιμο και σε Windows και Mac περιβάλλοντα ∗Διαχείριση από απόσταση	∗Η διαχείριση από απόσταση χρειάζεται βελτίωση

4. MidPoint Gateway, Sattellite, Companion, Teamer

Πλεονεκτήματα	Μειονεκτήματα
∗Η ολοκκληρωμένη έκδοση, παρέχει ένα πλήρη κατάλογο δυνατοτήτων και λειτουργιών	∗Μεγάλο κόστος αγοράς ∗Μη φιλικό περιβάλλον

5. Squid Internet Object Cache 2.0

Πλεονεκτήματα	Μειονεκτήματα
∗Πολύ καλή απόδοση για ιεραρχικό σχήμα ∗Ιδανικό για υποστήριξη πολλαπλής επαναποθήκευσης πληρεξούσιων εξυπηρετητών κατανεμειμένων σε δενδροειδές σχήμα ∗Ιδανικό για sites πολλαπλών εξυπηρετητών ∗Υποστήριξη URN, SNMP, HTTP1 persistent	∗Δεν υπάρχει έκδοση για NT

- **Διαθέσιμες εμπορικές εφαρμογές**

Οι διαθέσιμες εφαρμογές πληρεξούσιων εξυπηρετητών που κυκλοφορούν στο εμπόριο είναι:

Apache, Avirt, CacheBack, CacheFlow, CacheQube, CacheRaq, CERN/W3C, Cisco Cache Engine, CSM Proxy, DeleGate, DynaCache, FastLane, Harvest, IBM Web Traffic Express, InfoStorm, Inktomi Traffic Server, Jigsaw, JProxyma, Lagoon, Lucent's IPWorX, Microsoft Proxy, MOWS, NetCache, NetFilter, NetHawk, Netscape Proxy Server, Rebel.com NetWinder, Novell BorderManager FastCache, Novell Internet Caching System, Peregrine, Pushcache Cuttlefish, Roxen, SoftRouter Plus, Spaghetti, Squid, Sun Netra Proxy, TeraNode, Viking, Wcol, WebSpeed, and WinGate.

- **Κριτήρια επιλογής - σύγκρισης**

√ Στόχος αγοράς, σε τι οργανισμούς ή επιχειρήσεις απευθύνεται.

√ Ευκολία εγκατάστασης.

√ Απαίτηση ή όχι εκπαίδευσης.

√ Ευκολία στη χρήση και διαχείριση συστήματος.

√ Λειτουργικά συστήματα που υποστηρίζει.

√ Πρωτόκολλα που υποστηρίζει.

√ Δυνατότητα υποστήριξης πολλαπλών εξυπηρετητών.

√ Παροχή αποτελεσματικών τεχνικών κατανομής φόρτου.

√ Απαίτηση για επιπρόσθετο εξοπλισμό.

√ Υποστήριξη ελέγχου JavaScripts ή άλλων γενικά scripts.

√ Δυνατότητα αναλυτικής καταγραφής βασικών λειτουργιών και παροχής αναφορών.

√ Δυνατότητα λειτουργίας του εξυπηρετητή ως αντίστροφο.

√ Τεχνικές για την αποκατάσταση του συστήματος σε περίπτωση βλάβης.

√ Απόδοση (λόγος αριθμού αιτήσεων προς το δευτερόλεπτο).

√ Επεκτασιμότητα.

√ Κόστος αγοράς.

√ Κόστος συντήρησης.

2. Εγκατάσταση, Επιλογή Ιεραρχίας και Αρχιτεκτονικής

2.1. Επιλογή αρχιτεκτονικής και ιεραρχίας

Αφού καθορίσουμε τα χαραχτηριστικά του/των πληρεξούσιων εξυπηρετητών και ορισμένα σημαντικά μεγέθη, πρέπει να αποφασίσουμε για το είδος της ιεραρχίας και γεωγραφικής κατανομής των πληρεξούσιων εξυπηρετητών. Συγκεκριμένα, πρέπει να απαντήσουμε στις εξής ερωτήσεις:

• *Ιεραρχικό ή επίπεδο σχήμα;*

Αυτό εξαρτάται από το μέγεθος και τη γεωγραφική διασπορά του δικτύου. Ο βασικός κανόνας είναι ότι κάθε τμήμα (branch office-department) πρέπει να έχει το δικό του πληρεξούσιο εξυπηρετητή. Αν τα τμήματα αυτά συνδέονται με μισθωμένες γραμμές με το κεντρικό τότε θα πρέπει να υπάρχει ένας πληρεξούσιος εξυπηρετητής στο κεντρικό τμήμα, ο οποίος θα συνδέεται (chained) με τους υπόλοιπους των διαφόρων άλλων τμημάτων. Στη περίπτωση αυτή θα έχουμε μία δύο-επιπέδων ιεραρχική διάταξη. Στη περίπτωση όμως που ο οργανισμός ή η επιχείρηση βρίσκεται συγκεντρωμένη σε ένα μέρος μία επίπεδη διάταξη εξυπηρετητών αποτελεί τη καλύτερη λύση.

• *Ένας ή πολλοί πληρεξούσιοι εξυπηρετητές;*

Αν και ο κανόνας θέλει ένα πληρεξούσιο εξυπηρετητή για κάθε 3000 χρήστες, αυτό δε σημαίνει ότι ένας οργανισμός με 12000 υπαλλήλους θα χρειαστεί τέσσερις διατμηματικούς οι οποίοι θα συνδέονται με ένα κεντρικό πληρεξούσιο εξυπηρετητή. Αντίθετα οι τέσσερις μπορούν να συνδεθούν παράλληλα χρησιμοποιώντας το CARP ή κάποιο άλλο μηχανισμό που κάνει χρήση της συνάρτησης κατακερματισμού. Αυτό θα μας δίνει τη δυνατότητα να συνδυαστούν οι κρυφές μνήμες πολλών εξυπηρετητών σε ένα μόνο φυσικό σύστημα.

Γενικά, προτείνεται η δημιουργία ενός συμπλέγματος εξυπηρετητών, καθώς αυξάνει το συνολικό μέγεθος της κρυφής μνήμης και ελαττώνει το πρόβλημα της διπλο (ή πολυ)-εγγραφής της ίδιας πληροφορία σε περισσότερους από ένα πληρεξούσιους εξυπηρετητές. Έτσι τέσσερις εξυπηρετητές με 4 GB κρυφή μνήμη ο καθένας, θα παρέχουν συνολικά 16 GB κρυφής μνήμης αν συνδεθούν παράλληλα σχηματίζοντας ένα πίνακα (array) αντί να χρησιμοποιούσε ο καθένας τη δική του κρυφή μνήμη.

Αν και η σύνδεση των πληρεξούσιων εξυπηρετητών σε πίνακες συνήθως αποτελεί τη βέλτιστη λύση, υπάρχουν περιπτώσεις που η διαμόρφωση του δικτύου μας αναγκάζει να χρησιμοποιήσουμε διατμηματικούς εξυπηρετητές. Δε συμφέρει δηλαδή να περνάει όλη η

κυκλοφορία ενός οργανισμού ή μιας επιχείρησης από ένα πίνακα εξυπηρετητών διότι μπορεί να αποτελέσει πιθανό σημείο συμφόρησης (bottleneck) -για παράδειγμα σε περιπτώσεις υψηλού φορτώματος- ακόμα και αν οι εξυπηρετητές βρίσκονται σε διαφορετικά υποδίκτυα.

Σ' αυτές τις περιπτώσεις επιβάλλεται η χρήση διατμηματικών πληρεξούσιων εξυπηρετητών που θα βρίσκονται πιο κοντά στους πελάτες και θα τους εξυπηρετούν, χωρίς να είναι αναγκαία η προώθηση των αιτήσεών τους στο κεντρικό εξυπηρετητή.

2.2. Ανάθεση πελατών σε εξυπηρετητές

Ο τρόπος με τον οποίο θα γίνεται η ανάθεση των αιτήσεων των πελατών στους εξυπηρετητές αποτελεί ένα σημαντικό παράγοντα για το σχεδιασμό της εγκατάστασής τους και ασκεί βασική επίδραση στην επιτυχία ή όχι του συστήματος.

Η διαμόρφωση αυτή μπορεί να γίνει είτε αναθέτοντας κάθε χρήστη σε ένα καθορισμένο πληρεξούσιο εξυπηρετητή και αντιστοιχώντας το όνομά του σε πολλούς εξυπηρετητές (με το DNS round robin για παράδειγμα) είτε επιλέγοντας δυναμικά τον εξυπηρετητή κάνοντας χρήση έξυπνων προγραμμάτων ή JavaScripts που εκτός από την αποτελεσματικότερη ανάθεση παρέχουν επιπλέον και τη δυνατότητα να προσδιορίσουν αν κάποιος εξυπηρετητής βγει εκτός λειτουργίας, ώστε να μη στέλνουν σ' αυτόν επιπλέον αιτήσεις.

Αυτή τη τεχνική ακολουθούν τα *Proxy Auto-Configuration* (PAC) αρχεία που χρησιμοποιούνται από το σύστημα πλοήγησης Internet Explorer της Microsoft.

3. Βελτιστοποίηση των Βασικών Λειτουργιών

3.1. Διαμόρφωση της κρυφής μνήμης των πληρεξούσιων εξυπηρετητών

Ένας φυσικός πραγματικός ή πληρεξούσιος εξυπηρετητής συνήθως δεν μπορεί μόνος του να χειριστεί υψηλό αριθμό αιτήσεων. Το πρόβλημα αυτό μπορεί να επιλυθεί είτε εγκαθιστώντας πολλαπλούς εξυπηρετητές και κατανέμοντας το φόρτωμα ανάμεσα σ' αυτούς (load balancing), είτε δημιουργώντας ακριβές αντίγραφα των sites (mirror sites) επιτρέποντας το χρήστη να επιλέξει το κοντινότερο εξυπηρετητή. Αυτές οι λύσεις αμβλύνουν βέβαια την υπερφόρτωση στους εξυπηρετητές, αλλά από τη μία δεν έχουν τόσο καλά αποτελέσματα όσο η επαναποθήκευση από την άλλη δημιουργούν νέα ζητήματα που πρέπει να διευθετηθούν.

Όπως αναφέραμε η επαναποθήκευση αποτελεί μία από τις σημαντικότερες λειτουργίες ενός πληρεξούσιου εξυπηρετητή γι' αυτό θα πρέπει να δοθεί ιδιαίτερη προσοχή στο σχεδιασμό και στη διαχείρισή της. Τρεις παράμετροι θα πρέπει να εξεταστούν: ο τρόπος με τον οποίο θα γίνει η επαναποθήκευση, η αρχιτεκτονική που θα επιλεχθεί και το πως θα γίνει η διαχείριση της κρυφής μνήμης. Ειδικότερα:

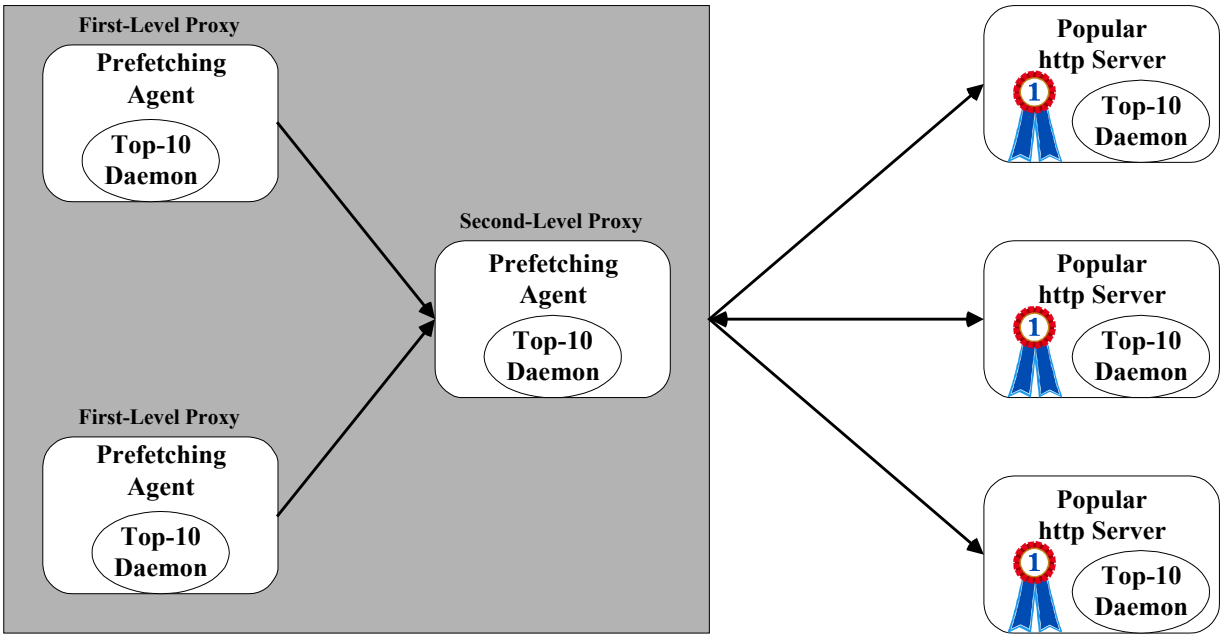
• Υπάρχουν τρεις κύριοι τρόποι επαναποθήκευσης:

– Ο κλασσικός και συνηθισμένος τρόπος είναι να αποθηκεύεται ένα έγγραφο στη κρυφή μνήμη, μόνο αν ζητηθεί από το χρήστη. Αν κάποιο έγγραφο δε ζητηθεί ποτέ, τότε η ύπαρξή του δε θα είναι γνωστή στο πληρεξούσιο εξυπηρετητή (*on demand caching*).

– Ένας δεύτερος τρόπος είναι ο πληρεξούσιος εξυπηρετητής αυτόματα και σε καθορισμένα χρονικά διαστήματα να επαναποθηκεύει συγκεκριμένες σελίδες ή ακόμα και ολόκληρα sites (*on command caching*). Το γεγονός αυτό επιτρέπει στον εξυπηρετητή, να διαχειρίζεται τα επαναποθηκευμένα έγγραφα ως ήδη ενημερωμένα (up-to-date) και να παραλείπονται όλοι οι έλεγχοι για κάθε έγγραφο ξεχωριστά (για το αν είναι ενημερωμένο ή όχι). Αυτοί οι έλεγχοι μπορούν να γίνουν σε χρονικές περιόδους που δεν υπάρχει μεγάλη δραστηριότητα όπως νυχτερινές ώρες και αργίες. Το αν θα επωφεληθούμε από τα πλεονεκτήματα που προσφέρει αυτός ο τρόπος επαναποθήκευσης εξαρτάται από το σκοπό που χρησιμοποιούμε τον εξυπηρετητή και το τρόπο χρήσης του Διαδικτύου. Αν δηλαδή έχουμε ένα πληρεξούσιο εξυπηρετητή για τον οποίο εκτιμούμε ότι θα δεχτεί πολύ φόρτωμα, τότε αυτός ο τρόπος αποτελεί μία συμφέρουσα λύση.

– Ο τρίτος τρόπος είναι ένας συνδυασμός των ανωτέρω. Συγκεκριμένα, θα αποθηκεύονται τα έγγραφα εκείνα τα οποία είναι δυνατό να ζητηθούν από το πελάτη (*prefetching*). Αυτός ο τρόπος συνίσταται δηλαδή στη προσπάθεια πρόβλεψης των εγγράφων που ο πελάτης επιθυμεί. Μία απλή εφαρμογή αυτής της σκέψης θα ήταν να επαναποθηκεύονται σελίδες και εικόνες που σχετίζονται με το συγκεκριμένο έγγραφο (links).

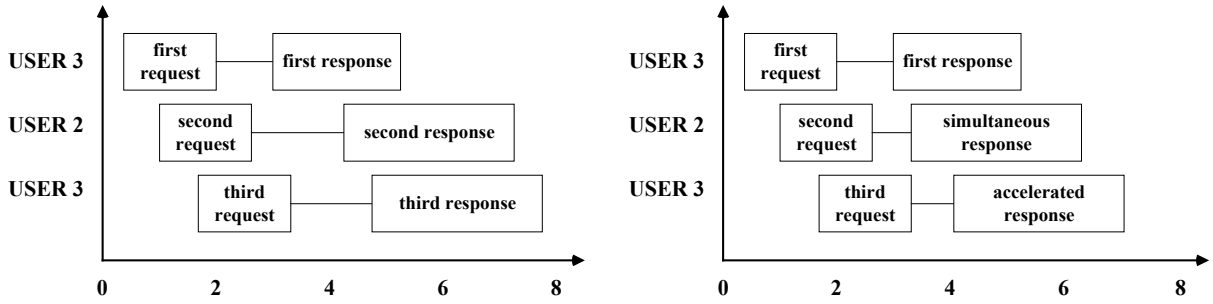
Μία ενδιαφέρουσα επίσης λύση είναι η πρόβλεψη αυτή να γίνεται βάση των ιδιαιτέρων χαραχτηριστικών του πελάτη σε συνδυασμό με τα δέκα πιο συχνά επαναποθηκευμένα έγγραφα που ανακτήθηκαν και περιέχονται στη κρυφή μνήμη του εξυπηρετητή. Στη περίπτωση αυτή αν μάλιστα υπάρχουν και πολλοί εξυπηρετητές, κάθε ένας από αυτούς θα προωθεί στους άλλους (που βρίσκονται σε υψηλότερο επίπεδο σε ένα ιεραρχικό σχήμα) τα δέκα πιο ζητούμενα έγγραφα με αποτέλεσμα η απόδοση να αυξάνεται σημαντικά (περισσότερο από 40%), ενώ η αύξηση κυκλοφορίας του δικτύου να διατηρείται σε χαμηλά επίπεδα (10%). Η διάταξη αυτή φαίνεται στο παρακάτω σχήμα:



Το Web-caching αποτελεί θέμα ερεύνης τόσο σε εθνικό επίπεδο (Εθνικό Δίκτυο Έρευνας και Τεχνολογίας), όσο και σε διεθνές όπου για το σκοπό αυτό έχουν δημιουργηθεί ομάδες εργασίας και ερευνητικές διακρατικές ομάδες με στόχο την ανάπτυξη μιας διεθνούς ιεραρχίας της επαναποθήκευσης του παγκόσμιου ιστού καθώς και την εγκατάσταση κοινών πλαισίων συνεργασίας που να διευκολύνουν την επικοινωνία και την ανταλλαγή δεδομένων μέσω του Διαδικτύου (όπως η ομάδα εργασίας της Ευρωπαϊκής Κοινότητας Έρευνας και Εκπαίδευσης σχετικά με τα Δίκτυα).

Επίσης αποτελεί ένα πεδίο έρευνας και για τις μεγάλες εταιρίες λογικού και λογισμικού. Μία εμπορική εφαρμογή που παρουσιάζει ενδιαφέρον είναι ο WebDoubler που αντίθετα από τους συμβατικούς πληρεξούσιους εξυπηρετητές που θεωρούν μία σελίδα επαναποθηκευμένη

στη κρυφή τους μνήμη μόνο μέχρι ολόκληρη η σελίδα ανακτηθεί από το πρώτο χρήστη, ο WebDoubler υπόσχεται ότι μπορεί να εξυπηρετήσει ταυτόχρονα όλους τους πελάτες που επιθυμούν εκείνη τη συγκεκριμένη στιγμή τη συγκεκριμένη σελίδα. Στο παρακάτω σχήμα φαίνεται αριστερά ο συμβατικός τρόπος και δεξιά η τεχνική που ακολουθεί ο WebDoubler.



• Σημαντικό ρόλο θα παίξει και η επιλογή της αρχιτεκτονικής με την οποία θα υλοποιηθεί η επαναποθήκευση. Υπάρχουν πάρα πολλές αρχιτεκτονικές, που η καθεμιά από αυτές παρουσιάζει πλεονεκτήματα και μειονεκτήματα. Εμείς θα αναφερθούμε στις σπουδαιότερες από αυτές που χρησιμοποιούνται σε εμπορικές εφαρμογές και είναι:

– Η κυρίαρχη αρχιτεκτονική για αρκετό καιρό ήταν το ιεραρχικό σχήμα του Squib συστήματος που όλοι οι εξυπηρετητές αποτελούσαν ένα "δενδροειδή" σχήμα και ICP συναλλαγές καθόριζαν το ποιος εξυπηρετητής θα εξυπηρετούσε καλύτερα την αίτηση του πελάτη. Ο Squib χρησιμοποιεί επίσης το μοντέλο εικονικής μνήμης (Virtual Memory Model) που διαχειρίζεται τη κρυφή μνήμη ως εικονική μνήμη.

– Η αρχιτεκτονική που χρησιμοποιεί τεχνικές κατάτμησης (hashing) και χρησιμοποιείται στο proxy της Netscape όπου τα επαναποθηκευμένα έγγραφα κατανέμονται σε καταλόγους και υποκαταλόγους ανάλογα με το τύπο και τη ζήτησή τους.

– Η αρχιτεκτονική CERN που δημιουργεί δομή στο πληρεξούσιο εξυπηρετητή ίδια με αυτή του πραγματικού εξυπηρετητή, δηλαδή στο πρώτο επίπεδο έχουμε το πρωτόκολλο, στο δεύτερο το όνομα του site κλπ.

• Τέλος, όσο αφορά στη διαχείριση της κρυφής μνήμης αυτό που αποτελεί το σημαντικότερο παράγοντα είναι η διαχείριση των επαναποθηκευμένων αρχείων δηλαδή ποια αρχεία είναι περισσότερο πιθανό να ζητηθούν στο μέλλον και ποια αρχεία θα πρέπει να διαγραφούν.

Ο κύριος στόχος μας είναι η βελτιστοποίηση της χρήσης της κρυφής μνήμης που αριθμητικά ισοδυναμεί με τη μεγιστοποίηση του λόγου:

cache hit / συνολικός αριθμός αιτήσεων

Για τη διαχείριση της κρυφής μνήμης χρησιμοποιούνται ορισμένοι αλγόριθμοι όπως:

1. Last Recently Used (LRU), που θα διαγράφει τα αντικείμενα στα οποία είχαμε μηδενική ή ελάχιστη πρόσβαση και θα διατηρεί τα πιο πρόσφατα αρχεία τα οποία είναι πιο πιθανό να ζητηθούν στο μέλλον.

2. Ο προηγούμενος αλγόριθμος χρησιμοποιεί το χρόνο που πέρασε από τη τελευταία πρόσβαση σαν το παράγοντα που καθορίζει την προτεραιότητα στα επαναποθηκευμένα αρχεία. Υπάρχουν όμως και άλλοι παράμετροι που την επηρεάζουν και χρησιμοποιούνται σα βάση από άλλους αλγόριθμους όπως:

– το πλήθος των πιο πρόσφατων προσβάσεων,

– το μέγεθος των αρχείων σε άμεση συνάρτηση με το διαθέσιμο εύρος του δικτύου και τη χωρητικότητα του δίσκου (τα μεγάλου μέγεθους αρχεία καταλαμβάνουν μεγαλύτερο χώρο στο δίσκο και καταναλώνουν περισσότερο εύρος οπότε θα έχουν χαμηλότερη προτεραιότητα) και

– ο χρόνος που χρειάστηκε για να ανακτηθεί ένα αρχείο (ένα αρχείο για το οποίο χρειάστηκαν ελάχιστα δευτερόλεπτα έχει μικρότερη προτεραιότητα από ένα αρχείο που απαιτήθηκε περισσότερος χρόνος).

3. Υπάρχουν και μερικοί άλλοι αλγόριθμοι όπως LAT (Latency Estimation Approach), HYB (Hybrid Algorithm).

3.2. Ομοιόμορφη κατανομή φορτώματος

Υπάρχουν πολλοί τρόποι και τεχνικές για να επιτευχθεί ομοιόμορφη κατανομή του φορτίου στους διάφορους εξυπηρετητές (*load balancing*). Εμείς θα αναφερθούμε σε τέσσερις μεθόδους:

- **Μέθοδος που βασίζεται στο DNS Round-Robin**

Στη περίπτωση αυτή πολλές και διαφορετικές μεταξύ τους IP διευθύνσεις (Η/Υ) μπορούν και επικοινωνούν με ένα μοναδικό DNS. Ο DNS θα αλλάζει συνεχώς τις IP διευθύνσεις των πελατών, ώστε κάθε πελάτης να εξυπηρετείται από διαφορετικό πληρεξούσιο εξυπηρετητή.

Εδώ όμως υπάρχει μία διαφορά των πληρεξούσιων σε σχέση με τους πραγματικούς εξυπηρετητές. Στους τελευταίους, το πρόβλημα του μοιράσματος του φορτίου, μπορεί να λυθεί χωρίζοντας τα περιεχόμενά τους σε λογικά τμήματα τα οποία μπορούν να είναι αποθηκευμένα σε διαφορετικούς εξυπηρετητές. Έτσι, μπορεί ένα site μιας εταιρίας να έχει τις γενικές πληροφορίες που αφορούν την εταιρία σε ένα εξυπηρετητή, τα προϊόντα σε άλλον, τις παραγγελίες σε τρίτο κλπ.

Στη περίπτωση των πληρεξούσιων εξυπηρετητών δεν υπάρχει η δυνατότητα να επιλέγουν αυτοί τι περιεχόμενο (πληροφορίες) θα περιέχει ο καθένας από αυτούς. Μία λύση που μπορεί να εφαρμοσθεί είναι η αυτόματη σύνθεση (*auto-configuration*) που μπορεί να είναι μία συνάρτηση JavaScript που εκτελείται από το πελάτη για κάθε ζητούμενη URL η οποία επιστρέφει τη διεύθυνση του πληρεξούσιου εξυπηρετητή που θα χρησιμοποιηθεί για να ανακτηθεί η απάντηση από τους εξυπηρετητές του Διαδικτύου.

- **Μέθοδος που βασίζεται στη συνάρτηση κατακερματισμού (hash function)**

Στη περίπτωση αυτή η hash τιμή υπολογίζεται από πληροφορίες που λαμβάνονται από τη URL και το αποτέλεσμα αυτής της τιμής χρησιμοποιείται για την επιλογή του κατάλληλου πληρεξούσιου εξυπηρετητή.

Γενικότερα, το πρόβλημα -που ώθησε σε νέες πιο πολύπλοκες και αποτελεσματικές τεχνικές- με τη προηγούμενη μέθοδο αφορά το τελικό μέγεθος της κρυφής μνήμης του συστήματος. Έτσι, εάν για παράδειγμα έχουμε πέντε παράλληλους πληρεξούσιους εξυπηρετητές που ο καθένας έχει κρυφή μνήμη ίση με 2 GB, η τελική κρυφή μνήμη θα είναι πάλι ίση με 2 GB γιατί οι αιτήσεις μπορούν να δρομολογηθούν σε ένα από το σύνολο των πληρεξούσιων εξυπηρετητών. Στη περίπτωση όμως που έχουμε τις μεθόδους που βασίζονται στη συνάρτηση κατακερματισμού, το τελικό μέγεθος της διαθέσιμης κρυφής μνήμης θα ισούται με το άθροισμα του καθενός (στη περίπτωσή μας 10 GB).

- **Μέθοδος που βασίζεται στη συνάρτηση κατακερματισμού (hash function) και στο πρωτόκολλο CARP (Cache Array Routing Protocol)**

Μία επέκταση της προηγούμενης τεχνικής είναι το πρωτόκολλο CARP που επιπλέον επιτρέπει σε πληρεξούσιους εξυπηρετητές να προστεθούν ή να αφαιρεθούν από τον πίνακα πληρεξούσιων εξυπηρετητών (proxy array) χωρίς να δημιουργηθούν προβλήματα μετάβασης στη καινούργια κατάσταση. Εάν για παράδειγμα έχουμε ένα πίνακα από έξη πληρεξούσιους εξυπηρετητές η hash συνάρτηση θα κατανείμει τις URL ισότιμα, δηλαδή καθένας θα αναλάβει να προωθήσει το 1/6 αυτών. Αν προστεθεί ένας νέος εξυπηρετητής τότε αυτός θα αναλάβει το

1/7 του συνολικού αριθμού των URL's. Το ίδιο φυσικά ισχύει εάν αντί να προσθέσουμε, αφαιρέσουμε ένα πληρεξούσιο εξυπηρετητή. Η συγκεκριμένη μέθοδος έχει τα καλύτερα αποτελέσματα γι' αυτό και προτιμάται από τις υπόλοιπες.

- **Μέθοδος που η επιλογή βασίζεται στο ICP (Internet Cache Protocol)**

Το ICP πρωτόκολλο χρησιμοποιείται για το προσδιορισμό του τρόπου επαναποθήκευσης των διαφόρων εγγράφων ανάμεσα στους εξυπηρετητές. Άρα, βασιζόμενοι σ' αυτό μπορούμε να καθορίσουμε τρόπους και τεχνικές που θα επιλέγουν αυτόν τον εξυπηρετητή που θα προσκομίσει πιο γρήγορα το ζητούμενο επαναποθηκευμένο έγγραφο. Το πρόβλημα που παρουσιάζει αυτή η μέθοδος είναι το γενικότερο πρόβλημα που παρουσιάζουν οι μέθοδοι που επιλέγουν τυχαία (random) ένα στοιχείο. Η περίπτωση να προωθούνται οι αιτήσεις στον ίδιο συνέχεια πληρεξούσιο εξυπηρετητή.

Εκτός από τις ανωτέρω τεχνικές, υπάρχουν και άλλες, όπως αυτές που βασίζονται στη δημιουργία ενός συμπλέγματος (cluster) πολλών εξυπηρετητών. Πολλές εταιρίες λογικού και λογισμικού υποστηρίζουν το clustering όπως η Cisco με το LocalDirector, η Resonate με το Dispatch κλπ. Λόγω της τρομακτικής αύξησης του Διαδικτύου, νέες τεχνολογίες και εμπορικές εφαρμογές εμφανίζονται στην αγορά μέρα με την ημέρα.

3.3. Φιλτράρισμα

Όπως είναι φανερό ο πληρεξούσιος εξυπηρετητής αποτελεί το ιδανικό σημείο στο οποίο μπορούμε να εφαρμόσουμε διαδικασίες και πολιτικές φιλτραρίσματος και αυτό γιατί αποτελεί το σημείο εκείνο, μέσα από το οποίο διέρχονται όλες οι αιτήσεις των πελατών από το εσωτερικό δίκτυο και των επιστροφών των απαντήσεων προς αυτό.

Οι πληρεξούσιοι εξυπηρετητές μπορούν να εφαρμόσουν αρκετούς τύπους φιλτραρίσματος όπως απαγόρευση αιτήσεων σε συγκεκριμένες URL's, με συγκεκριμένες λέξεις στο περιεχόμενο των επικεφαλλίδων ή ακόμα με λέξεις στο κείμενο των αιτήσεων. Μπορούν επίσης να υλοποιήσουν διαδικασίες φιλτραρίσματος και στις εισερχόμενες απαντήσεις απαγορεύοντας τη προώθηση προς τους πελάτες εγγράφων με συγκεκριμένες λέξεις στην επικεφαλίδα ή στο κείμενο και ακόμα ελέγχοντας για ιούς κλπ.

Ο πιο συνηθισμένος τύπος φιλτραρίσματος είναι η απαγόρευση ή όχι αιτήσεων προς συγκεκριμένα sites με τη χρήση της URL. Η ζητούμενη URL ελέγχεται αν ανήκει σε μία λίστα από ανεπιθύμητες URL's. Ανάλογα με το αποτέλεσμα του ελέγχου η αίτηση απορρίπτεται ή όχι.

Η λειτουργία αυτή υλοποιείται με δύο πολιτικές: είτε μπλοκάρονται συγκεκριμένες URL's ενώ άλλες επιτρέπονται είτε μπλοκάρονται όλες οι URL's εκτός απο τις επιθυμητές. Το πρόβλημα που γεννάται είναι: τι γίνεται αν ο αριθμός των URL's και στη μία και στην άλλη περίπτωση είναι πολύ μεγάλος; Αν αυτό συμβαίνει, η σειριακή αναζήτηση θα είναι πολύ χρονοβόρα. Μία προσέγγιση για να επιλυθεί το πρόβλημα αυτό θα ήταν αντί να χρησιμοποιούμε ολόκληρες τις URL's, να κάνουμε χρήση άλλων τιμών που θα αντιστοιχούν σ' αυτές και θα βρίσκονται σε ταξινομημένη σειρά. Έτσι η αναζήτηση θα εκτελείται πιο γρήγορα. Αυτή η λύση όμως -όπως είναι αντιληπτό- δε θα μπορούσε να λειτουργήσει σε περιπτώσεις που η λίστα των URL's θα χρησιμοποιούσε χαραχτήρες wildcards.

Ένας δεύτερος τρόπος φιλτραρίσματος αποτελεί ο έλεγχος των επικεφαλλίδων των αιτήσεων και συγκεκριμένα:

- φιλτράροντας το περιεχόμενο των επικεφαλλίδων αλλά επιτρέποντας τη περαιτέρω προώθηση των αιτήσεων,
- αντικαθιστώντας επικεφαλίδες με άλλες και
- απαγορεύοντας λόγω του περιεχομένου των επικεφαλλίδων τη προώθησή τους.

Η πρώτη περίπτωση εφαρμόζεται για στατιστικούς κυρίως λόγους ενώ οι δύο τελευταίες σε περιπτώσεις που δεν επιθυμούμε συγκεκριμένες πληροφορίες, που εμείς τις θεωρούμε διαβαθμισμένες, να γνωστοποιηθούν σε τρίτους. Επιπλέον ένας τρίτος τρόπος είναι ο έλεγχος του περιεχομένου των αιτήσεων. Στις εισερχόμενες απαντήσεις, θα έχουμε αντίστοιχα φιλτράρισμα στις επικεφαλίδες τους, στο περιεχόμενό τους και έλεγχο για ιούς, Trojan horses κλπ. Εδώ το πρόβλημα συνίσταται στο μέγεθος των δεδομένων.

Στη περίπτωση των ιών για παράδειγμα, απαιτείται πρώτα να κατέβει όλη η σελίδα και μετά να ελεχθεί και αφού ολοκληρωθεί ο έλεγχος τότε μόνο να προωθηθεί στο πελάτη. Ο χρήστης όμως όλο αυτό το διάστημα δεν βλέπει καμία απόκριση από τον υπολογιστή του. Υπάρχουν μερικοί τρόποι για αντιμετωπίσουμε αυτά τα προβλήματα:

- Επαναποθήκευση. Το μόνο σημείο στο οποίο εκτελείται το φιλτράρισμα όπως είδαμε είναι ο πληρεξούσιος εξυπηρετητής. Αν ο έλεγχος έχει ήδη προηγηθεί τότε η ανάκτηση των εγγράφων από τις κρυφές μνήμες των εξυπηρετητών δεν απαιτεί επιπλέον έλεγχο.
- Φιλτράρισμα σε άλλα υπολογιστικά συστήματα όπως για παράδειγμα σε άλλους εξυπηρετητές. Μπορούμε για παράδειγμα να χρησιμοποιήσουμε ένα επιταχυντή ο οποίος θα τοποθετηθεί μπροστά από το κεντρικό εξυπηρετητή του δικτύου μας και θα εκτελεί αυτός τις διαδικασίες φιλτραρίσματος.

3.4. Παρακολούθηση & έλεγχος κυκλοφορίας - Παροχή αναφορών

Όλες οι εμπορικές εφαρμογές πληρεξούσιων εξυπηρετητών προσφέρουν βοηθητικές εφαρμογές (utilities) με τις οποίες μπορούμε να παρακολουθήσουμε τις διάφορες λειτουργίες που εκτελούν και να έχουμε μία πλήρη εικόνα της κατάστασης του συστήματος (monitoring). Μπορούμε για παράδειγμα να δούμε το συνολικό τους φόρτωμα, να προσδιορίσουμε την απόδοσή τους, τη κατάσταση των εκτελούμενων εργασιών, τον αριθμό των συνδέσεων, κατάλογο με τις συχνότερα ζητούμενες URL's κλπ. Παρέχουν επίσης πληροφορίες που μας δίνονται με τη μορφή αναφορών όπως τον αριθμό επιτυχημένων συνδέσεων, το μέγεθος των δεδομένων που μετακινούνται ανάμεσα στα τμήματα του συστήματος, το ποσοστό των συνδέσεων που εξυπηρετήθηκαν από τους πληρεξούσιους ή πραγματικούς εξυπηρετητές, το χρόνο που απαιτήθηκε για τις συνδέσεις, στοιχεία δηλαδή που αν εκτιμηθούν σωστά μπορούν να βοηθήσουν όχι μόνο στην εξαγωγή στατιστικών αποτελεσμάτων αλλά κα στην αύξηση της απόδοσης του συστήματος. Παρακάτω παραθέτουμε μερικές αναφορές (οι τέσσερις πρώτες είναι παραδείγματα από το Netscape Proxy ενώ οι υπόλοιπες είναι πραγματικά στοιχεία από το Squib Proxy του Εθνικού Μετσόβειου Πολυτεχνείου):

1. *Ποσό δεδομένων που μετακινήθηκαν*:

DATA FLOW REPORT			
	Header	**Content**	**Total**
Client - Proxy	5 MB	0 MB	5 MB
Proxy - Client	5 MB	115 MB	120 MB
Proxy - Remote	5 MB	0 MB	5 MB
Remote - Proxy	2 MB	75 MB	77 MB

2. *Αριθμός αιτήσεων που εξυπηρετήθηκαν από τη κρυφή μνήμη ή από τους πραγματικούς εξυπηρετητές*:

REQUESTS AND CONNECTIONS REPORT

Total Requests	15000
Remote Connections	10000
Avoided Remote Connections	5000 (34%)

3. *Απόδοση της επαναποθήκευσης*:

CACHE PERFOMANCE REPORT			
Proxy cache hits	4000	25%	25 sec/request

4. *Χρόνος υλοποίησης εξυπηρέτησης*:

TRANSFER TIME REPORT	
Average Transaction Time	3 sec/request

5. *Αριθμός αιτήσεων και ποσοστό επιτυχίας του 1999*:

	Data Transferred (Mbytes)	**Data Transferred (Requests)**	**Hit Ratio**
January	69421.33	16916266	30.5%
February	103227.08	22808668	24.0%
March	107885.83	24721688	27.8%
April	88178.48	21445937	23.3%
May	110958.91	27684784	26.5%
June	74958.06	23406471	24.0%
July	51789.76	11616305	29.0%
August	36274.04	6512179	29.8%
September	45523.68	9236439	33.7%
October	50640.65	10141065	30.7%
November	52871.82	11856949	32.0%
December	29830.49	7654051	25.3%

6. *Γενικά στοιχεία του μηνός Δεκεμβρίου 1999*:

PROXY.GRNET.GR: Dec 1999 Traffic Report	
Number of hosts used proxy server	288

Number of sites accessed through proxy server	110457

PROXY.GRNET.GR: Dec 1999 Traffic Report		
Data transfer analysis	**Mbytes**	**Requests**
Caching data xfered from remote sites	22472.27	5448456
Non-caching data xfered from remote sites	1740.03	359110
Data xfered from cache	4904.23	1754659
Data xfered from parent/sibling cache	713.96	91826
Total Data offered to clients	29830.49	7654051
Hit Ratio	20.0%	25.3%

3.5 Ασφάλεια

Η ασφάλεια των δικτύων που έχουν πληρεξούσιους εξυπηρετητές περιλαμβάνει ουσιαστικά τη προστασία των δύο παρακάτω θεμάτων:

- *τη προστασία των Η/Υ συστημάτων του εσωτερικού δικτύου από το ίδιο το δίκτυο και τους κινδύνους που προέρχονται από το Διαδίκτυο* και
- *τη προστασία των δεδομένων που εισέρχονται στο εσωτερικό δίκτυο*.

Βασικά, ο σκοπός αυτός επιτυγχάνεται με τη χρήση των φραγμάτων και των πληρεξούσιων εξυπηρετητών ενώ τα δεδομένα προστατεύονται με τη κρυπτογράφησή τους (encryption).

Στο προηγούμενο κεφάλαιο δώθηκαν μερικές αρχιτεκτονικές που παρέχουν ασφάλεια. Στην ενότητα αυτή θα αναφερθούμε ειδικότερα για τις ενέργειες που οι διαχειριστές των δικτύων πρέπει να εκτελέσουν για να βελτιστοποιήσουν το παράγοντα αυτό. Βέβαια το θέμα ασφάλεια, είναι πολύ μεγάλο και περιλαμβάνει ένα μεγάλο αριθμό παραμέτρων που πρέπει να ληφθούν υπόψη. Εδώ όμως θα αναφερθούμε σ' αυτές που σχετίζονται με τους πληρεξούσιους εξυπηρετητές:

1. Είναι κοινός τόπος για τους διαχειριστές των UNIX δικτύων ότι ποτέ δε δουλεύουμε σαν superuser, αλλά με κάποιο άλλο όνομα (user ID). Επίσης μεγάλη προσοχή πρέπει να δοθεί στην ανάθεση του write permission στο φάκελλο και στα αρχεία διαμόρφωσης του πληρεξούσιου εξυπηρετητή.

2. Οι πληρεξούσιοι εξυπηρετητές θα πρέπει να δέχονται αιτήσεις μόνο από τους χρήστες του εσωτερικού δικτύου. Όλες οι αιτήσεις που προέρχονται από το Διαδίκτυο θα πρέπει να απορρίπτονται.

3. Παρόμοια, οι διατμηματικοί εξυπηρετητές θα δέχονται μόνο τις αιτήσεις από τους χρήστες που βρίσκονται στο συγκεκριμένο τμήμα.

4. Οι αντίστροφοι πληρεξούσιοι εξυπηρετητές και τα φράγματα από τη μεριά του Web site, θα πρέπει να έχουν εγκατασταθεί και διαμορφωθεί κατά τέτοιο τρόπο, ώστε να επιτρέπεται η σύνδεσή τους μόνο με συγκεκριμένους πραγματικούς εξυπηρετητές και όχι με άλλα υπολογιστικά συστήματα ώστε να αποτραπεί η πρόσβαση σ' αυτά και κατά συνέπεια στα αρχεία των συστημάτων τους και στα δεδομένα.

5. Όπως αναφέραμε οι SOCKS επιτρέπουν σε κάθε τύπο πρωτοκόλλου να περάσει, με συνέπεια ο πληρεξούσιος εξυπηρετητής να μην έχει τη πλήρη εικόνα των εκτελούμενων λειτουργιών σε επίπεδο πρωτοκόλλου. Η καλύτερη λύση είναι να επιτρέψουμε μόνο σε συγκεκριμένες θύρες (ports) να επικοινωνούν με συγκεκριμένα πρωτόκολλα.

6. Ιδιαίτερη προσοχή πρέπει να δοθεί σε ότι αφορά το καθορισμό πολιτικών ασφαλείας σχετικά με τα εισερχόμενα scripts της Java, JavaScripts και ActiveX, ώστε να φιλτράρονται και να ελέγχεται γενικά το περιεχόμενό τους. Σχεδόν όλοι οι πληρεξούσιοι εξυπηρετητές παρέχουν τέτοια βοηθήματα.

7. Τέλος, όπως αναφέρθηκε στην ενότητα για τη λειτουργία του φιλτραρίσματος, οι επικεφαλίδες μιας HTTP αίτησης μπορεί να περιέχουν πληροφορίες οι οποίες είναι διαβαθμισμένες και αφορούν το εσωτερικό δίκτυο ή τους χρήστες τους. Τέτοιες πληροφορίες μπορούν να θεωρηθούν οι παρακάτω:

- Εσωτερικές IP διευθύνσεις των χρηστών.
- Ονόματα των πληρεξούσιων εξυπηρετητών.
- Τοπολογία και αρχιτεκτονική του εσωτερικού δικτύου.
- Λειτουργικό σύστημα που χρησιμοποιεί ο πελάτης και έκδοση αυτού.
- Λογισμικό που χρησιμοποιεί ο πελάτης και έκδοση αυτού.
- Λογισμικό, τύπος και έκδοση του πληρεξούσιου εξυπηρετητή.
- Η ηλεκτρονική διεύθυνση του χρήστη και
- Στοιχεία της αναγνώρισης της ταυτότητας του χρήστη (login-password).

Όλες οι πληροφορίες που αναφέρθησαν μπορεί να περιέχονται στις επικεφαλίδες των αιτήσεων των πελατών και να προωθούνται στι Διαδίκτυο. Επειδή οι πληροφορίες αυτές αντιστοιχούν σε συγκεκριμένες λέξεις μπορούμε είτε στο επίπεδο του φράγματος είτε στο πληρεξούσιο εξυπηρετητή να τις τροποποιήσουμε ή να τις αποκρύψουμε.

8. Είναι πολύ σημαντικό γεγονός ότι σύμφωνα με το ICSA (International Computer Security Association) το 80% των δεδομένων αποσπάται παράτυπα ή παράνομα από τους εσωτερικούς χρήστες του δικτύου. Γι΄αυτό το λόγο απαιτείται η εγκατάσταση μίνι-φραγμάτων (mini-firewalls) και κατανομή τους σε όλη τη διάταξη του οργανισμού και όχι μόνο στα όρια του δικτύου/Διαδικτύου. Μερικοί πληρεξούσιοι εξυπηρετητές όπως ο Border Manager της Novell έχει τη δυνατότητα να ομαδοποιεί την ιεραρχική διάταξη των εσωτερικών δικτύων ανά τμήμα όπως Τμήμα Μάρκετινγκ, Τμήμα Πωλήσεων κλπ.

4. Συντήρηση - Προβλήματα

Όπως είναι φυσικό στις περισσότερες των περιπτώσεων που αφορούν προβλήματα που ενδέχεται να παρουσιαστούν, αυτά είναι ανάλογα με τα συγκεκριμένα εμπορικά πακέτα και είναι αδύνατο να αναλυθούν στα πλαίσια αυτής της εργασίας. Εδώ θα αναπτυχθούν εν συντομία αυτά που εκτιμούνται ότι σχετίζονται με τα περισσότερα από αυτά.

1. Επειδή οι πληρεξούσιοι εξυπηρετητές είναι ένα σημείο από το οποίο διέρχεται όλη η κυκλοφορία του δικτύου, είναι επιτακτική ανάγκη να σχεδιάσουμε τι πρέπει να γίνει σε περίπτωση που κάποιος ή κάποιοι από αυτούς βγουν εκτός λειτουργίας. Αν συμβεί αυτό, οι χρήστες δεν θα μπορέσουν να έχουν πλέον πρόσβαση στο δίκτυο.

Οι λύσεις που κυκλοφορούν στο εμπόριο προσανατολίζονται στην ανάθεση και μεταφορά του φόρτου εργασίας σε άλλον ή άλλους.

2. Επίσης πρέπει να αντιμετωπιστούν τυχόν προβλήματα με τη λειτουργία της επαναποθήκευσης όπως:

- να μην επαναποθηκεύονται καθόλου τα έγγραφα στη κρυφή μνήμη,
- να επαναποθηκεύονται τα έγγραφα αλλά να μην ελέγχονται καθόλου ή τέλος
- να ελέγχονται συνέχεια.

Όλες οι εμπορικές εφαρμογές παρέχουν διαγνωστικά εργαλεία για τον εντοπισμό των δυσλειτουργιών και την επιδιόρθωσή τους.

3. Τέλος, οι αναφορές που μας δίνονται μπορούν να αποτελέσουν ένα καλό βοήθημα για να διαχειριστούμε αποτελεσματικότερα και καλύτερα το σύστημα των πληρεξούσιων εξυπηρετητών και γενικότερα το εσωτερικό μας δίκτυο προβλέποντας τυχόν αδυναμίες και προβλήματα.

Ε. ΠΑΡΑΔΕΙΓΜΑΤΑ-ΠΡΟΤΑΣΕΙΣ

Το κεφάλαιο περιλαμβάνει την ανάλυση της εφαρμογής και διαχείρισης πληρεξούσιων εξυπηρετητών του συστήματος ΕΔΕΤ των ελληνικών ΑΕΙ και ΤΕΙ, που εκτιμώ ότι αποτελεί ένα πολύ καλό παράδειγμα και τη δυνατότητα συμμετοχής σ' αυτό ή νέας εγκατάστασης συστήματος πληρεξούσιων εξυπηρετητών στο δίκτυο του Πανεπιστημίου Μακεδονίας.

Βασικός κορμός του συστήματος πληρεξούσιων εξυπηρετητών του ΕΔΕΤ αποτελεί η εφαρμογή της πιλοτικής υπηρεσίας "National Proxy Cache Schema" η βασική λειτουργία της οποίας συνίσταται στην εφαρμογή της επαναποθήκευσης, της κράτησης δηλαδή αντιγράφων των συχνότερα αναζητούμενων θεμάτων στο τοπικό σύστημα ή σε κάποιο κοντινό εξυπηρετητή. Οι στόχοι είναι οι εξής: η εξοικονόμηση του διαθέσιμου εύρους του δικτύου, η βελτίωση του χρόνου απόκρισης - εξυπηρέτησης, η μείωση του κόστους διασύνδεσης με το διεθνές δίκτυο και τέλος η απόκτηση χρήσιμης τεχνογνωσίας και εμπειρίας πάνω στο αντικείμενο αυτό που το τελευταίο καιρό παρουσιάζει πολύ μεγάλο ενδιαφέρον.

Τα ιδρύματα που συμμετέχουν είναι:

- Εθνικό Μετσόβειο Πολυτεχνείο ως συντονιστής και
- Αριστοτέλειο Πανεπιστήμιο Θεσσαλονίκης,
- Γεωπονικό Πανεπιστήμιο,
- Δημοκρίτειο Πανεπιστήμιο Θράκης,
- Εθνικό και Καποδιστριακό Πανεπιστήμιο,
- Πανεπιστήμιο Ιωαννίνων,
- Πανεπιστήμιο Κρήτης,
- ΤΕΙ Αθηνών και τέλος
- ΤΕΙ Ηπείρου.

Η κεντρική ιδέα του παραπάνω συστήματος είναι η εξής:

Στο ΕΔΕΤ αυτή τη στιγμή υπάρχει ένας κεντρικός πληρεξούσιος εξυπηρετητής στο κεντρικό κόμβο του δικτύου στην Αθήνα με σκοπό να βρίσκεται πιο κοντά και πριν τις WAN συνδέσεις του ΕΔΕΤ με το Διαδίκτυο (σύνδεση με TEN-155, US και NL).

Οι συνδέσεις αυτές είναι οι πιο ακριβές του δικτύου και παρουσιάζουν καθημερινά φαινόμενα κορεσμού (ιδίως η γραμμή με US) οπότε σκοπός της εγκατάστασης του πληρεξούσιου εξυπηρετητή είναι η εξοικονόμηση του διαθέσιμου εύρους δικτύου και η βελτίωση του χρόνου απάντησης πάνω από τις συγκεκριμένες αυτές γραμμές.

Στο κεντρικό εξυπηρετητή του ΕΔΕΤ έχουν ελεύθερη πρόσβαση όλα τα δίκτυα των ιδρυμάτων που συνδέονται σε αυτό. Μπορούν δηλαδή να το χρησιμοποιήσουν είτε μεμονωμένοι χρήστες είτε οι πληρεξούσιοι των ιδρυμάτων σα "parent" ή "sibling" στη διεύθυνση proxy.grnet.gr:8080.

Θυμίζω εδώ ότι στο ιεραρχικό σχήμα οι σχέσεις, που μπορούν να αναπτυχθούν μεταξύ δύο εξυπηρετητών, είναι οι εξής:

- *γονέα-παιδιού (parent relations)*, όπου ο εξυπηρετητής, που λειτουργεί ως παιδί, ζητά από τον εξυπηρετητή-γονέα κάθε αντικείμενο, που δεν μπορεί να βρει στη δική του κρυφή μνήμη και αυτός με τη σειρά του, στην περίπτωση που δεν το έχει στη δική του κρυφή μνήμη, το φέρνει από το δίκτυο και το προωθεί στο παιδί του και
- *αδελφών (sibling relation)* όπου ένας εξυπηρετητής μπορεί να έχει έναν ή περισσότερους αδελφούς, τους οποίους ρωτά -κάνοντας χρήση του πρωτοκόλλου ICP- εάν διαθέτουν στη κρυφή τους μνήμη το ζητούμενο αντικείμενο. Αν το ζητούμενο αντικείμενο υπάρχει στη μνήμη κάποιου από τους εξυπηρετητές - αδελφούς, ο εξυπηρετητής, που έκανε την κλήση, θα το αντλήσει από αυτόν, αλλιώς θα πρέπει να το φέρει από το δίκτυο μόνος του.

Επιστρέφοντας στο παράδειγμά μας, ο *proxy.grnet.gr*, παράλληλα διατηρεί και ορισμένες σχέσεις "sibling" με πληρεξούσιους άλλων ελληνικών ISP's μέσω του AIX (τώρα με OTENET παλιότερα HOL και FORTHNET) αλλά το κέρδος δεν είναι σημαντικό και οι σχέσεις αυτές τείνουν να υποβαθμιστούν.

Ο κεντρικός cache εξυπηρετητής του ΕΔΕΤ βασίζεται στο Squid 1.1. Η πρόσβαση στον εξυπηρετητή επιτρέπεται στα συνδεμένα στο ΕΔΕΤ ακαδημαϊκά και ερευνητικά δίκτυα. Τα δίκτυα αυτά μπορούν να εγκαταστήσουν ένα τοπικό πληρεξούσιο εξυπηρετητή (συνιστάται η χρήση του Squid, το οποίο διατίθεται δωρεάν και μπορεί να βρεθεί στο σχετικό mirror του στην Ελλάδα: *ftp://ftp.ntua.gr/mirror/squid/*) ο οποίος θα ικανοποιεί τα αιτήματα των χρηστών τους και θα χρησιμοποιεί ως "γονέα" τον εξυπηρετητή του ΕΔΕΤ ή μπορεί καταρχήν να ορίσουν ένα CNAME (για παράδειγμα *proxy.uom.gr*) που να δείχνει στο *proxy.grnet.gr* με δυνατότητα αργότερα να γίνει ξεχωριστό μηχάνημα.

Οι ακόλουθοι πληρεξούσιοι χρησιμοποιούν ως "*γονέα*" τον εξυπηρετητή του ΕΔΕΤ:

- proxy.ntua.gr:8080
- proxy.uoa.gr:8080

- www.auth.gr:800
- proxy.cti.gr:8080
- proxy.aueb.gr:8080
- proxy.aegean.gr:8080
- proxy.uch.gr:8080
- demokritos.cc.duth.gr:8080
- proxy.aua.gr:8080
- proxy.ionio.gr:8080
- proxy.teiath.gr:8080

Δύο σημαντικά στοιχεία που προέκυψαν από τη χρήση του συστήματος είναι τα εξής:

1. Ένα πρώτο στοιχείο είναι η *ικανοποιητική απόδοση* του συστήματος. Για παράδειγμα σύμφωνα με τα στοιχεία του proxy server του Ε.Μ.Π., τα οποία καλύπτουν τους τελευταίους είκοσι μήνες λειτουργίας του (Φεβρ.1997-Σεπτ.1998), το *HIT RATIO* κυμαίνεται σε ικανοποιητικά επίπεδα, γύρω στο 50%, δηλαδή μία στις δύο κλήσεις εξυπηρετείται από την cache του server. Επίσης από τα στοιχεία του proxy του Α.Π.Θ. το ποσοστό επιτυχίας είναι 38-40%, ενώ σε δημοφιλή sites είναι πολύ μεγαλύτερο.

2. Ένα δεύτερο πολύ σημαντικό στοιχείο είναι το γεγονός ότι *οι χρήστες από μόνοι τους δε χρησιμοποιούν πολύ τους πληρεξούσιους εξυπηρετητές* (το ποσοστό του 1999 σύμφωνα με μετρήσεις του Ε.Μ.Π. κυμάνθηκε από 15% έως 20%).

Οι τρόποι, που χρησιμοποιούνται για την αύξηση του ποσοστού των χρηστών, οι οποίοι κάνουν χρήση των proxy cache υπηρεσιών είναι οι εξής:

- Ενημέρωση των χρηστών, σε συνδυασμό με την προσφορά κάποιου συστήματος, για αυτόματη διαμόρφωση των προγραμμάτων πλοήγησής τους (automatic proxy configuration). Η μέθοδος αυτή έχει πού καλά αποτελέσματα, γιατί προσφέρει ευκολία στους χρήστες, καλύτερη διαμόρφωση των προγραμμάτων πλοήγησής τους από κεντρικό ελεγχόμενο σημείο και δυνατότητα προσθήκης μηχανισμού για την περίπτωση διακοπής λειτουργίας του πληρεξούσιου εξυπηρετητή (fault tolerance).

- Ενημέρωση των χρηστών, προσφορά συστήματος automatic proxy configuration και ταυτόχρονη απαγόρευση της πρόσβασης στο παγκόσμιο ιστό, παρά μόνο αν αυτή πραγματοποιείται μέσω του πληρεξούσιου εξυπηρετητή (με κατάλληλη διαμόρφωση στον δρομολογητή του δικτύου που διαχειρίζεται τις συνδέσεις με τα εξωτερικά δίκτυα). Ανήκει

στις πλέον αυστηρές μεθόδους, με κύρια πλεονεκτήματα την απόλυτη εκμετάλλευση του πληρεξούσιου εξυπηρετητή, περνώντας όλη την HTTP κίνηση του εσωτερικού δικτύου μέσω αυτού και την πολύ εύκολη υλοποίηση του σχήματος από τεχνική άποψη. Τα μειονεκτήματα της παραπάνω μεδόδου είναι η μεγάλη ευθύνη για την συνεχή και ομαλή λειτουργία του εξυπηρετητή, ορισμένα θέματα παραβίασης της προσωπικής ελευθερίας και τέλος έχει μεγάλη αρχικά τουλάχιστον επιβάρυνση, για την ενημέρωση των χρηστών, για το νέο σχήμα και τις ρυθμίσεις που θα πρέπει να κάνουν στα προγράμματα τους.

- Η χρήση ενός μηχανισμού που ανακατευθύνει με διαφανή τρόπο την HTTP κίνηση, από και προς εξωτερικά δίκτυα, μέσω του πληρεξούσιου (transparent proxying). Η διαφορά με τη προηγούμενη μέθοδος είναι ότι δεν απαιτείται καμία απολύτως ενέργεια ή ρύθμιση από τους τελικούς χρήστες, γλιτώνοντας τους διαχειριστές του δικτύου από σημαντική επιβάρυνση.

Υιοθετώντας αυτή τη μέθοδο, οι διαχειριστές του συστήματος του ΕΔΕΤ έχουν ήδη εγκαταστήσει transparent σχήματα με μεγάλη επιτυχία στο Ε.Μ.Π. εδώ και έξη μήνες και τώρα μελετάται η φάση εγκατάστασής του σε όλο το σύστημα όπου θα χρησιμοποιηθούν τέσσερις εξυπηρετητές για καλύτερη κατανομή του φόρτου (load balancing), για το κόμβο της Αθήνας.

Πέρα από τη συμμετοχή στο εθνικό σχήμα του ΕΔΕΤ το Πανεπιστήμιο Μακεδονίας μπορεί να εγκαταστήσει στο δίκτυό του αυτοτελές πληρεξούσιο εξυπηρετητή. Στη περίπτωση αυτή θα πρέπει να καθορισθούν όλοι εκείνοι οι παράμετροι που αναλύθηκαν στο προηγούμενο κεφάλαιο.

Ειδικότερα εκτιμώ ότι ένας πληρεξούσιος εξυπηρετητής μπορεί να διαχειριστεί όλες τις αιτήσεις των συνδεόμενων χρηστών στο δίκτυο του Πανεπιστημίου. Θα απαιτηθεί ίσως και ένας δεύτερος σε περίπτωση που ο κύριος βγει εκτός λειτουργίας και δε θα υπάρχει επιπλέον πρόσβαση στο Διαδίκτυο, ή μπορεί για οικονομικούς λόγους να μην χρειαστεί και οι χρήστες να έχουν απευθείας πρόσβαση στο Διαδίκτυο χρησιμοποιώντας την αυτόματη διαμόρφωση των προγραμμάτων πλοήγησής τους. Στη πρώτη περίπτωση ο κύριος θα είναι *proxy1.uom.gr* και ο δεύτερος *proxy2.uom.gr* και οι δύο θα βρίσκονται στη πόρτα 8080.

Ανάλογα με το πόσο ασφάλεια επιθυμούμε να προσφέρουμε στο δίκτυο του Πανεπιστημίου θα πρέπει να σχεδιάσουμε και την κατάλληλη αρχιτεκτονική. Έτσι αυτή μπορεί να περιλαμβάνει ένα πληρεξούσιο και ένα δρομολογητή ή για ακόμα μεγαλύτερη ασφάλεια ο πληρεξούσιος δρομολογητής μπορεί να βρίσκεται σε μία DMZ ζώνη ανάμεσα σε δύο δρομολογητές απαγορεύοντας όλες τις εισερχόμενες συνδέσεις εκτός από τα e-mails, news και

μπλοκάροντας όλες τις εξερχόμενες συνδέσεις (και οι δύο περιπτώσεις έχουν αναλυθεί στο πρώτο κεφάλαιο).

ΣΤ. ΣΥΜΠΕΡΑΣΜΑΤΑ

Καθώς ο αριθμός των εταιριών και οργανισμών που συνδέονται με το Διαδίκτυο αυξάνεται με πολύ γρήγορους ρυθμούς, η διαχείριση των πληροφοριών γίνεται απαιτητικότερη

και δυσκολότερη. Η τεχνολογία των πληρεξούσιων εξυπηρετητών που αναπτύχθηκε τα τελευταία χρόνια προσφέρει μία ιδανική τεχνική λύση και αποτελεί ένα πολύ σημαντικό βοήθημα για τους διαχειριστές των δικτύων.

Λόγω της θέσης του –κεντρικό σημείο, μέσω του οποίου διέρχεται όλη η κυκλοφορία από και προς το εσωτερικό δίκτυο- ο πληρεξούσιος εξυπηρετητής μπορεί να επαναποθηκεύσει τα συχνότερα ζητούμενα έγγραφα και να τα διαθέσει γρηγορότερα στους πελάτες-χρήστες, να εκτελέσει λειτουργίες φιλτραρίσματος, έλεγχου πρόσβασης και να εφαρμόσει πολιτικές ασφάλειας. Η διαφάνεια (transparency) τέλος, αποτελεί ένα από τα σημαντικότερα πλεονεκτήματα που κερδίζουμε από τη χρήση του. Ο χρήστης έχει την εντύπωση ότι αλληλεπιδρά απευθείας με το πραγματικό εξυπηρετητή και το αντίστροφο.

Οι περισσότεροι πληρεξούσιοι παρέχουν υπηρεσίες και λειτουργίες για όλες τις εφαρμογές όπως HTTP, FTP, Telnet και άλλα πρωτόκολλα του Διαδικτύου και υπόσχονται ότι μπορούν να βελτιώσουν την αποδοτικότητα των εσωτερικών δικτύων κατά 30-50%.

Οι διαθέσιμοι τύποι (packet filters, application & circuit level proxies) και οι αντίστοιχες αρχιτεκτονικές τους, καθώς και οι διαθέσιμες εμπορικές εφαρμογές, αποτελούν για τις εταιρίες-οργανισμούς, ολοκληρωμένες λύσεις καλύπτοντας τις ανάγκες τους και αντιμετωπίζοντας τα προβλήματα που δημιουργούνται από τη διαδικτύωσή τους. Δύο σημαντικοί παράμετροι επηρεάζουν σημαντικά την εγκατάσταση, αρχιτεκτονική, λειτουργία και διαχείριση των πληρεξούσιων εξυπηρετητών: *η δομή και οι στόχοι του οργανισμού-εταιρίας*.

Οι διαχειριστές των δικτύων ενθαρρύνουν –και σε πολλές περιπτώσεις υποχρεώνουν- τους χρήστες τους να έχουν πρόσβαση στο Διαδίκτυο μέσω, μόνο, του πληρεξούσιου εξυπηρετητή. Το αποτέλεσμα είναι, όλοι οι πελάτες να έχουν πλήρη πρόσβαση στο Διαδίκτυο εύκολα, γρήγορα και με ασφάλεια.

Ζ. ΒΙΒΛΙΟΓΡΑΦΙΑ

1.http://www.whatis.com/proxy.html: όπου δίνεται ο ορισμός του πληρεξούσιου εξυπηρετητή και αναλύονται οι βασικές έννοιες, τα βασικά χαραχτηριστικά και οι γενικές λειτουργίες του όπως επαναποθήκευση, φιλτράρισμα κλπ.

2.http://www.vms.process.com/help/helpproxy.html: όπου αναλύονται οι γενικές ιδιότητες των εξυπηρετητών, ο τρόπος με τον οποίο εκτελείται μία κανονική συναλλαγή στο Διαδίκτυο και μία συναλλαγή μέσω ενός πληρεξούσιου εξυπηρετητή και τα πλεονεκτήματα και μειονεκτήματα της επαναποθήκευσης.

3.http://www.metalab.unc.edu/mdw/HOWTO/proxy.html: όπου δίνονται οι βασικοί τύποι των πληρεξούσιων εξυπηρετητών καθώς και αρχιτεκτονικές που προσφέρουν στο εσωτερικό δίκτυο ασφάλεια.

4.http://www.webopedia.internet.com/helpproxy.html: όπου παρουσιάζονται οι κύριοι σκοποί των πληρεξούσιων εξυπηρετητών.

5.http://www.socks.nec.com: πληροφορίες για τη τεχνολογία SOCKS και το πρωτόκολλο SOCKS, οι δυνατότητες και λειτουργίες των εκδόσεων v4 και v5, τα μειονεκτήματα της v4 που ώθησαν στο σχεδιασμό της νέας έκδοσης, το τρόπο λειτουργίας τους, οι μηχανισμοί με τους οποίους παρέχεται η ασφάλεια στα εσωτερικά δίκτυα και τα πλεονεκτήματα και μειονεκτήματα που έχουν σε σχέση με τους application-level proxy servers.

6.http://www.summitonline.com/security/papers/aventail1.html: πληροφορίες για τη τεχνολογία SOCKS v5, το πρωτόκολλο SOCKS v5, αναλυτικές συγκρίσεις με τους άλλους τύπους φραγμάτων, γενικά χαραχτηριστικά και λειτουργίες των SOCKS v5, εφαρμογές και αρχιτεκτονικές που βασίζονται σ' αυτά.

7.http://cedpa-k12.org/databus-issues/v36nt/proxy.html: όπου αναφέρεται η βασική λειτουργία της επαναποθήκευσης και τα βασικά πλεονεκτήματα και μειονεκτήματα των πληρεξούσιων εξυπηρετητών.

8.http://www.web-caching.com/proxy.html: στο site αυτό περιέχονται οι αρχιτεκτονικές επαναποθήκευσης, μελέτες και έρευνες γι' αυτήν, καθώς και μία αρκετά ενημερωμένη λίστα διαθέσιμων εμπορικών εφαρμογών, σύγκριση των τεχνικών χαραχτηριστικών και δυνατοτήτων τους και τέλος ένα κατάλογο χαραχτηριστικών βάση των οποίων μπορούμε να αξιολογήσουμε και να επιλέξουμε τον κατάλληλο πληρεξούσιο εξυπηρετητή.

9.http://www.nepean.uws.edu.au/its/comms/projects/proxies/: τα πλεονεκτήματα από τη χρήση των πληρεξούσιων εξυπηρετητών και η πρόταση χρήσης του Squib Internet Object Cache για την εγκατάσταση κατανεμημένων εξυπηρετητών σε ιεραρχικό σχήμα.

10.http://www.maxum.com/WebDoubler/FAQ/Perfomance.html: όπου μέσω μιας εμπορικής εφαρμογής το WebDoubler παρουσιάζεται μία νέα τεχνική επαναποθήκευσης που

υπόσχεται ότι μειώνει κατά πολύ το χρόνο απάντησης και ανάκτησης σελίδων που ζητούνται ταυτόχρονα από πολλούς χρήστες.

11.http://hotwired.lycos.com/packet/garfinkel/96/51/index2a.html: οι πληρεξούσιοι εξυπηρετητές αποτελούν μία πολύ καλή τεχνική λύση αλλά μερικές φορές όπως για παράδειγμα στις διαφημιστικές εταιρίες δημιουργούν πονοκεφάλους. Οι μεγάλες εταιρίες δεν είναι σε θέση να γνωρίζουν τον ακριβή αριθμό των ατόμων που επισκέπτονται το site τους.

12.http://www.gunet.gr: παρουσιάζεται η υπηρεσία National Proxy Cache Schema, στα πλαίσια του δικτύου GUNET, μιας συνολικής υπηρεσίας, σε εθνικό επίπεδο, για proxy caching που θα καλύπτει με τον καλύτερο δυνατόν τρόπο τις ανάγκες των ιδρυμάτων και των φορέων που συμμετέχουν στο GUNET. Περιέχονται επίσης στατιστικά στοιχεία από το proxy του Ε.Μ.Π.

13.http://www.rad.com/networks/1998/proxy.html: όπου απαριθμούνται τα μειονεκτήματα της επαναποθήκευσης των πληρεξούσιων εξυπηρετητών και δίνονται γραφήματα τα οποία αναπαριστούν αποτελέσματα που έβγαλαν μετρήσεις του Πανεπιστημίου Kaiserslauten και αφορούν τις σχέσεις των εξής μεγεθών: το μέγεθος της κρυφής μνήμης, οι αλγόριθμοι επαναποθήκευσης, το μέγεθος των εγγράφων κ.ά.

14.http://www.microsoft.com/catalog/proxy: στο site της Microsoft δίνεται μία αναλυτική αναφορά για το Proxy Server 2.0, τα γενικά και τεχνικά χαραχτηριστικά, τις βασικές λειτουργίες, τις διαθέσιμες αρχιτεκτονικές και πρωτόκολλα που υποστηρίζει και τις επιπλέον δυνατότητες που υποστηρίζει σε σχέση με την προηγούμενη έκδοση.

15.http://help.netscape.com/faqs.html: όπου αναλύεται ο Proxy Server της Netscape, τα γενικά και τεχνικά χαραχτηριστικά, τις βασικές λειτουργίες, τις διαθέσιμες αρχιτεκτονικές και πρωτόκολλα που υποστηρίζει κλπ.

16.http://compnetworking.about.com: όπου έχουμε τη σύγκριση των δύο γνωστότερων εμπορικών proxy εφαρμογών της Microsoft και της Netscape.

17.http://squid.nlanr.net/: το πολύ γνωστό πακέτο του squid, υποστήριξη, εκδόσεις, τεκμηρίωση και οδηγίες εγκατάστασης και διαχείρισης.

18.http://www.cs.caltech.edu/papers/: οι ειδικές αρχιτεκτονικές και οι πειραματικές και εμπορικές εφαρμογές που προκύπτουν απ' αυτές καθώς και οι εφαρμογές που συνδυάζουν την ενεργητική επεξεργασία με τη κατανομή των πληροφοριών (active proxies).

19.*World Wide Web Proxies* by Ari Luotonen, Kevin Altis: η εργασία αυτή αποτέλεσε από τη στιγμή που πρωτοπαρουσιάστηκε τον Απρίλιο του 1994, το σημείο αναφοράς για τις

επόμενες επιστημονικές μελέτες και τη βάση για αρκετές εμπορικές εφαρμογές. Στην εργασία αυτή δίδεται μία πρώτη παρουσίαση των βασικών λειτουργιών και χαραχτηριστικών των πληρεξούσιων εξυπηρετητών.

20.*SOCKS v5* by Joe Paone: όπου δίνεται μία πλήρη ανάλυση των υπηρεσιών, των δυνατοτήτων και των πλεονεκτημάτων από τη χρήση του ανωτέρω προτοκόλλου.

21.*The influence of geographical and culltural issues on the cache proxy server* by Vivrglio F. Almeida, Marcio G.Cesario, Rodrig C.Fonseca, WagnerMeira Jr, Cristina D.Mutra: στην εργασία αυτή αποδεικνύεται βάση πραγματικών στοιχείων η επίδραση των γεωγραφικών και κοινωνικών παραμέτρων στη φόρτωση των πληρεξούσιων εξυπηρετητών.

22.*Intermediaries: new places for producing and manipulating Web content* by R. Barret, P.P.Maglio: προτείνεται μία νέα προσέγγιση στο σχεδιασμό και τη δημιουργία εφαρμογών του WWW με την εγκατάσταση ενδιάμεσων (Intermediaries) υπολογιστικών στοιχείων καθώς και ορισμένα παραδείγματα.

23.*A Top-10 Approach toprefetching on the Web*: το μέγιστο ποσοστό επαναποθήκευσης είναι περίπου 40-50% δηλαδή ένα στα δύο έγγραφα. Ο Π. Μαρκάτος και η Ε. Χρονάκη παρουσιάζουν μία τεχνική επαναποθήκευσης που λαμβάνει υπόψη τα γνωστότερα και περισσότερο ζητούμενα έγγραφα με τα ιδιαίτερα χαραχτηριστικά των χρηστών.

24.http://ircache.nlanr.net/Cache/Workshop97/Papers/Jeffery/jeffery.html: στην εργασία αυτή οι συντάκτες προτείνουν το σχεδιασμό κατανεμημένων πληρεξούσιων εξυπηρετητών (Proxy-sharing Proxy Server) που εκτός από την επαναποθηκευμένη πληροφορία θα κρατούνται και μερικά άλλα στοιχεία που θα που θα οδηγήσουν στη μείωση της κυκλοφορίας του παγκόσμιου ιστού.

-