



Computer Oriented Project

Server Implementations on Campus Wide Networks BITS Goa – Case Study

Aalap Tripathy

Faculty Guide : Mr Mangesh Bedekar



ACKNOWLEDGEMENT

It's a moment of intense pleasure to express my gratitude towards everyone who directly or indirectly helped me during this project work. I am thankful to **Dr D M Kulkarni**, Co-ordinator Computer Centre, BITS Pilani Goa Campus, for providing me an opportunity to work on various tasks which has led to this project for over two years now. My sincere gratitude to **Mr. MANGESH BEDEKAR** to have shown me what team work and dedication can lead to. Despite his busy schedule, he always found time out for his sharp review of the development I made during the day. It is truly an honour for me to have been associated with such co-operative and extraordinarily brilliant persons. Of course much of my work wouldn't have been possible for the assistance of the staff at Computer Centre mainly **Mr Anjaneya Sardesai., Ms Sandhya Samant, Mr Santosh S.** Also **Mr Kunal Tyagi** who was instrumental in sorting out many of the complex network issues which troubled my work. Thank you, Sirs.

Aalap Tripathy
aalap@bits-go.a.ac.in

INTRODUCTION

This project is essentially a comprehensive documentation of the setup and various stages of trouble shooting involved during the setup of Arrayed Proxies, Domain Name Service (DNS), Light-weight Directory Access Protocol (LDAP), FTP Services, Web Server and Mail Server.

I will include all material I referred, including those which proved to be ineffective. Often I have tried numerous approaches to solving a given error I encountered all of which I am including. The final configuration lines are also included in the text (with adequate explanation where required).

I have not included the configuration of the Windows based Web server for <http://www.bits-go.a.ac.in> and <http://mail.bits-go.a.ac.in> - the mail server because we intend to move to other applications which will give the same solution. Also, I am including the planning stages for the LDAP Server which as of now remains incomplete.

Also for the Windows based DHCP server, I have included only the functional parts which would enhance our understanding of how our network performs. Although I wasn't involved in the setup of the DHCP server directly, I am still including the functional aspects of it which I and Bedekar Sir have had to modify to make some systems operational (viz the mail server)

All the configuration that you would be reading are mostly "at this time" states of the servers. Since, this is a functional network, many of the systems might (as requirements dictate) be modified to suit the conditions.

Also, I have included in **bold** some of the actions which possibly need to be taken in view of this report. This are actions which I intended to take, have taken (but failed), want to be taken for better & logical function of the servers.

CONTENTS

SI No.	Description
1	IP Addressing on network
2	DHCP Server
3	DNS Server
4	SQUID PROXY Caching
5	LDAP Server Planning
6	LDAP Server Deployment
7	Future Activities
8	Bibliography
9	Appendices

IP ADDRESSING ON BITS NETWORK



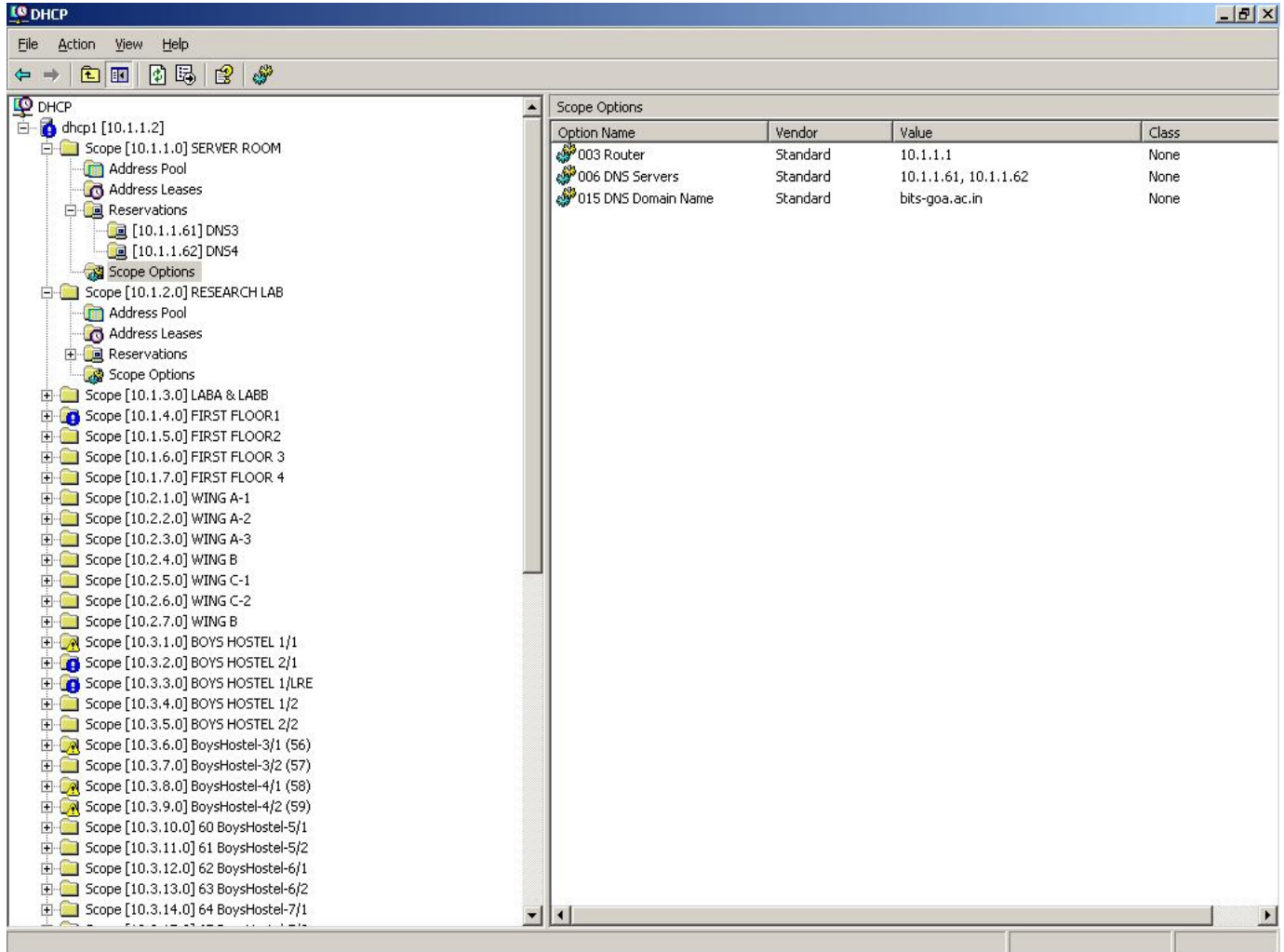
BITS IP Addressing

- `10.0.0.0/8' is a description of the BITS Goa Network !!!
- Contains any address from 10.0.0.0 to 10.255.255.255 (over 16 million addresses!)
- 10.0.0.0/16 is mostly 1 hostel or a combination of nearby hostels
- 10.0.0.0/24 is most generally closest rooms in a hostel or classroom/faculty chambers in a corridor

DHCP SERVER

BITS Goa Network has a redundant dual mode DHCP servers defined on IP addresses 10.1.1.1 & 10.1.1.2. These run the DHCP Service on Windows 2003 Server Edition. Since this has been configured on a Windows Machine, many of the features are invoked by just “pressing the button”. I have also explored Linux based DHCP Servers which I am not documenting here.

I am showing below a screenshot of the DHCP Service running on 10.1.1.2



The left hand pane shows the various “scopes” defined. New Scopes can be defined by simply going to Action → New Scope

Definitions like 10.1.1.0 are the default broadcast addresses for all network points connected to the router 10.1.1.1 (defined on the Right Pane)

For the 10.1.1.x range (as shown) I gave “IP Reservation” for the primary and secondary DNS Servers (10.1.1.61 and 10.1.1.62). Here I obtained the MAC address of the NIC Interface connected to both the machines and defined that whenever the NIC with this predefined IP Address requests an address lease, assign it the IP address 10.1.1.61 or 10.1.1.62 whatever the case may be.

It is very essential we replicate the same for all machines which we have defined as static IP Addresses. This is because unless otherwise specified, the DHCP server will unknowingly produce and IP clash in which either or both of the machines

will cease to function. Also it will generate un-necessary log/error messages every time such an action occurs. **Also the maintenance of the address leases must be done regularly. We also need to verify the logs generated by the DHCP Server for possible conflicts.**

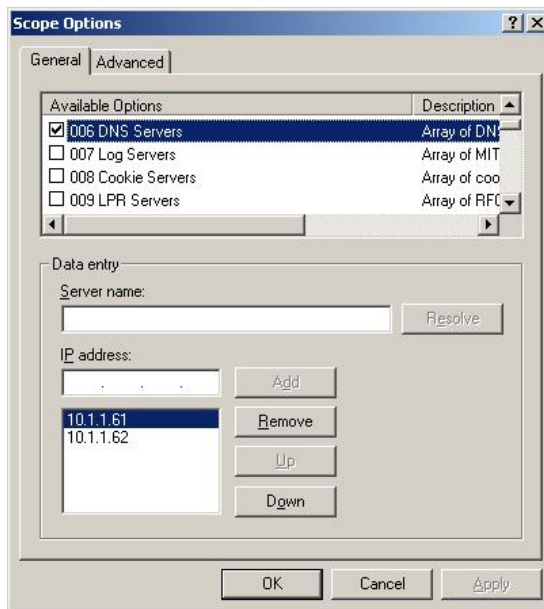
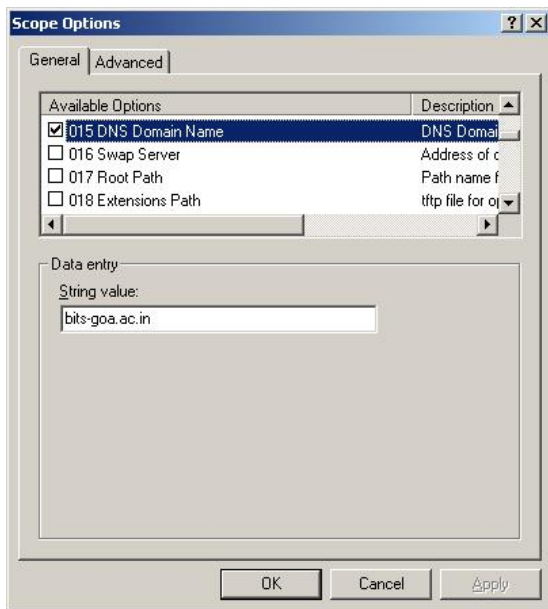
The MAC address of an interface can be obtained by ipconfig /all (Windows) or ifconfig (LINUX) as the physical address.

```

Ethernet adapter Local Area Connection:

    Connection-specific DNS Suffix  . : bits-goa.ac.in
    Description . . . . . : Realtek RTL8139/810x Family Fast Eth
    Physical Address. . . . . : 00-0B-6A-70-4D-60
    DHCP Enabled. . . . . : Yes
    Autoconfiguration Enabled . . . . . : Yes
    IP Address. . . . . : 192.168.1.100
    Subnet Mask . . . . . : 255.255.255.0
    Default Gateway . . . . . : 192.168.1.1
    DHCP Server . . . . . : 192.168.1.1
    DNS Servers . . . . . : 10.1.1.61
                           10.1.1.62
    Lease Obtained. . . . . : Thursday, November 30, 2006 7:04:08
    Lease Expires . . . . . : Friday, December 01, 2006 7:04:08 AM
  
```

The next significant change I made was to include a definition for the DNS Servers in the DHCP Broadcast itself. This meant that the user need not define the DNS server he was required to use. **In case the ip addresses of the DNS servers are to be changed, this must be reflected in the Scope Definition of the DHCP Server.** Otherwise the user may manually override the default DNS Servers in his configuration – (on linux /etc/resolv.conf on windows – TCP/IP Properties)



Also I defined the DNS Domain Name or the Connection Specific DNS Suffix as bits-goa.ac.in. This meant that everytime a computer NAME (not IP address) was queried by a device on our network, it would appened .bits-goa.ac.in to the name. Now a DNS query as shown below on an Ethreal Packet Capture is done subsequently to the DNS Server(s) (which are the master zones for bit-goa.ac.in). Simply put now we could make http or ftp queries based on name (viz <http://dakiya> for mail server and <http://www> for the bits-goa.ac.in website or <ftp://orion> for the Academics FTP Server) Everytime the computer making such queries(based on the settings done here in the DHCP server)

would append .bits-go.a.in to such names and make a DNS Query. This is resolved by the DNS Server into an IP address (refer section on DNS Servers) to which the relevant request is then made.

```

E:\WINDOWS\system32\cmd.exe
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

E:\Documents and Settings\aalap>ipconfig

Windows IP Configuration

Ethernet adapter Local Area Connection 2:

    Connection-specific DNS Suffix . : bits-go.a.in
    IP Address . . . . . : 192.168.1.100
    Subnet Mask . . . . . : 255.255.255.0
    Default Gateway . . . . . : 192.168.1.1
  
```



Realtek RTL8139/810x Family Fast Ethernet NIC: Capturing - Ethereal

No.	Time	Source	Destination	Protocol	Info
1418	171.327318	10.3.1.99	10.3.1.255	NBNS	Name query NB WPAD.<00>
1419	171.46625	Cisco_c2:23:94	CDP/VTP	CDP	Cisco Discovery Protocol
1420	171.61776	Cisco_c2:23:94	01:00:0c:cc:cc:cd	STP	Conf. Root = 8192/00:0f:f8:ef:e9:73 Cost = 11 Port = 0x8014
1421	171.94710	Cisco_c2:23:94	Cisco_c2:23:94	LOOP	Reply
1422	172.07908	Cisco_c2:23:94	01:00:0c:cc:cc:cd	STP	Conf. Root = 8192/00:0f:f8:ef:e9:b1 Cost = 11 Port = 0x8014
1423	172.16980	10.3.1.28	10.3.1.255	BROWSE	Get Backup List Request
1424	172.16986	10.3.1.28	10.3.1.255	NBNS	Name query NB WORKGROUP<1b>
1425	172.17010	10.3.1.60	10.3.1.255	NBNS	Name query NB SHARIEF<00>
1426	172.17022	10.3.1.28	Broadcast	ARP	who has 10.3.1.60? Tell 10.3.1.28
1427	172.53218	10.3.1.22	Broadcast	ARP	who has 10.3.1.41? Tell 10.3.1.22
1428	172.53357	10.15.3.161	Broadcast	ARP	who has 10.15.3.1? Tell 10.15.3.161
1429	172.55020	10.3.1.60	Broadcast	ARP	who has 10.3.1.41? Tell 10.3.1.60
1430	172.55863	10.3.1.156	Broadcast	ARP	who has 10.3.1.41? Tell 10.3.1.156
1431	172.56017	10.3.1.13	Broadcast	ARP	who has 10.3.1.41? Tell 10.3.1.13
1432	172.61831	10.15.3.2	224.0.0.2	HSRP	Hello (state Standby)
1433	172.81941	10.3.1.28	10.3.1.255	NBNS	Name query NB WORKGROUP<1b>
1434	173.03548	10.3.1.40	10.15.1.5	TCP	2020 > 2748 [PSH, ACK] Seq=1040 Ack=1341 Win=64195 Len=32
1435	173.12527	10.15.1.5	10.3.1.40	TCP	2748 > 2020 [PSH, ACK] Seq=1341 Ack=1072 Win=65000 Len=32
1436	173.12728	10.3.1.3	224.0.0.2	HSRP	Hello (state Standby)
1437	173.26492	10.3.1.40	10.15.1.5	TCP	2020 > 2748 [ACK] Seq=1072 Ack=1373 Win=64163 Len=0
1438	173.42746	10.3.1.40	10.1.1.61	DNS	Standard query A yahoo.com.bits-go.a.in
1439	173.42781	10.1.1.61	10.3.1.40	DNS	Standard query response, No such name
1440	173.42965	10.3.1.40	10.1.1.61	DNS	Standard query A yahoo.com
1441	173.43004	10.1.1.61	10.3.1.40	DNS	Standard query response A 216.109.112.135 A 66.94.234.13
1442	173.56395	10.15.3.2	224.0.0.2	HSRP	Hello (state Active)
1443	173.61615	Cisco_c2:23:94	01:00:0c:cc:cc:cd	STP	Conf. Root = 8192/00:0f:f8:ef:e9:73 Cost = 11 Port = 0x8014
1444	173.66937	10.3.1.28	10.3.1.255	NBNS	Name query NB WORKGROUP<1b>
1445	173.86846	10.3.1.40	10.3.1.15	TCP	2041 > netbios-ssn [SYN] Seq=0 Ack=0 Win=65535 Len=0 MSS=1460

Name: ns1.yahoo.com
 Type: A (Host address)
 Class: IN (0x0001)
 Time to live: 1 day, 16 hours, 6 minutes, 54 seconds
 Data length: 4
 Addr: 66.218.71.63

During the course of preparation of the report I encountered a strange packet movement because of setting a connection specific DNS Suffix. Refer nos. 1438 to 1440 marked in blue. Everytime I made a query for a public website say yahoo.com the originating computer by virtue of having a DNS suffix defined appended a .bits-go.a.in to it. Of course this would result in a failed DNS Query. Luckily, it again makes a standard A query for yahoo.com to which receives a suitable query response in 1441. And therefore it works. However, this point

requires a bit for tweaking. The above packet analysis shows conclusively that we are making 1 failed DNS query for every successful query done !! We definitely need to analyse how to remove this error/overhead by changes in server configuration or in the Network Protocol itself !! Of course dropping the Connection Specific DNS Suffix would do the trick, however it wouldn't make the bits-goa.ac.in zone as defined on the DNS Nameserver (see section : DNS) effective. So, my question is : Can we give a NULL zone definition in the DNS config ?? This point needs a bit more work.

DOMAIN NAME SERVICE

There are now 4 separate DNS Servers on the BITS Network. 2 each for internal and external use. For each such combination of DNS servers one is “primary” or “master” and the other is “secondary” or slave. The nomenclature primary/secondary is used in context of windows whereas LINUX based servers use the nomenclature master and slave.

How a DNS Query is Made : Referring the figure shown alongside, we make a DNS query for say yahoo.com. We get a non authoritative answer from the yahoo.com nameserver that is available at ip addresses 216.109.112.135 or/and 66.94.234.13

```
Default Server: dns3.bits-go.a.in
Address: 10.1.1.61

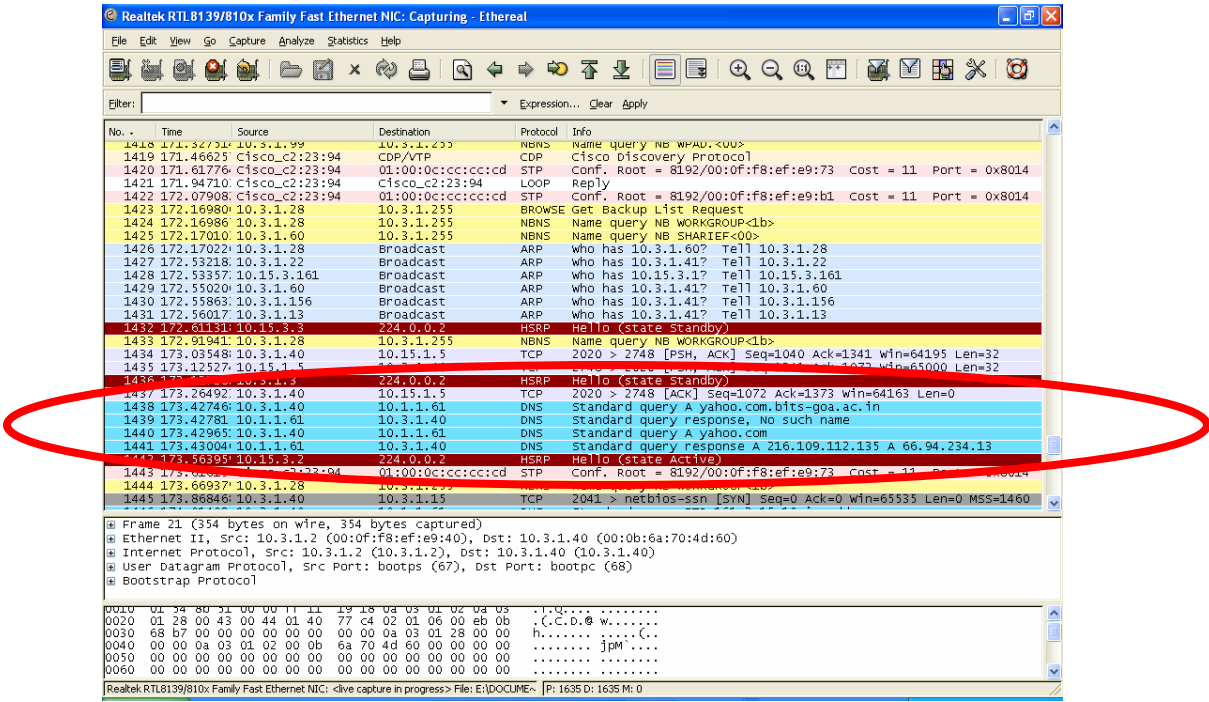
> yahoo.com
Server: dns3.bits-go.a.in
Address: 10.1.1.61

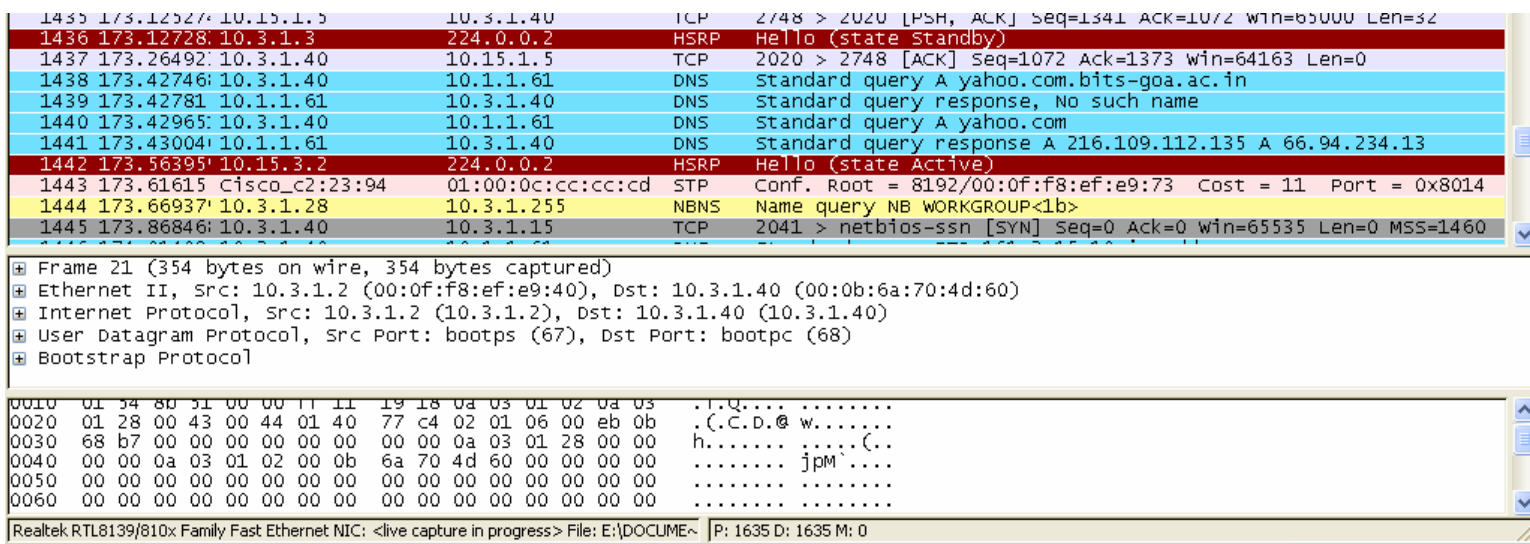
Non-authoritative answer:
Name: yahoo.com
Addresses: 216.109.112.135,
66.94.234.13

> 216.109.112.135
Server: dns3.bits-go.a.in
Address: 10.1.1.61

Name: w2.rc.vip.dcn.yahoo.com
Address: 216.109.112.135
```

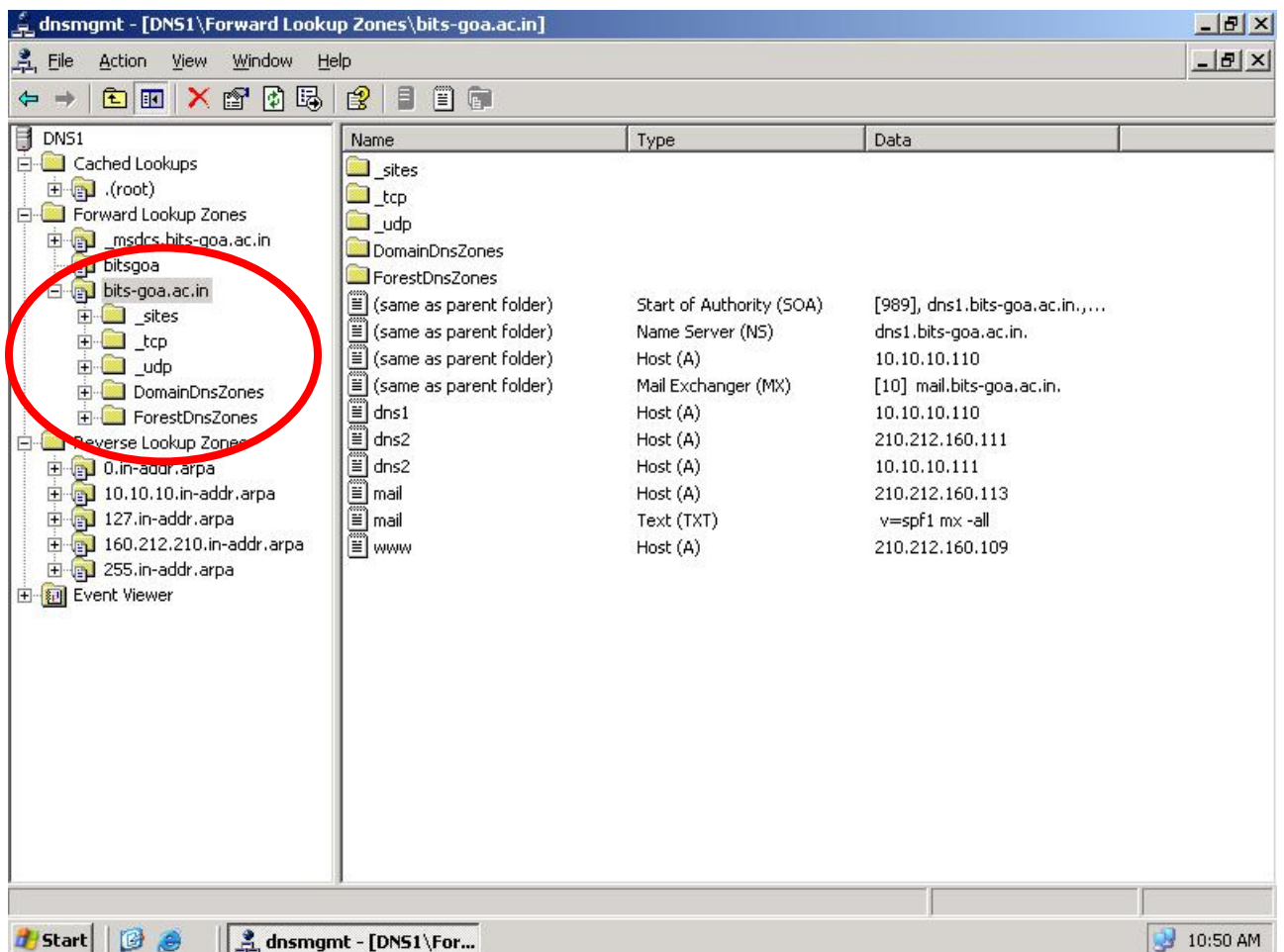
How the DNS Query Travels on the packet level :





I have used Ethereal Packet Capture. This shows the packet level movement of an actual DNS Query. And the response received from the DNS Server. The contents of the packet can also be analysed with reference to the Internet Protocol & is not included in this report. Also the above figure includes an understandable description of what is actually happening in the packets sent and received from a client.

External DNS Servers : I now describe some of the salient features of the DNS Service running on Windows 2003 Server



In Windows parlance, the primary DNS Server on our network is 10.10.10.110 registered globally as dns1.bits-go.a.in This server is connected to the DMZ (Demilitarized zone) of the router (10.1.1.1) As per configuration, this machine's packets after NAT (Network Address Translation) are routed to the public domain IP address 210.212.160.110. Similar configuration is employed for the secondary nameserver

Here, I observed that, the switch to where the DMZ is defined was not physically connected to the DMZ port of the router. This configuration was done on the basis of commands. A specific port was supposedly defined as the DMZ port with adequate permissions defined in the firewall.

Our Primary nameserver is registered for the bits-go.a.in domain with the registrars on the given IP Addresses (210.212.160.110 & 210.212.160.111) The computer running as the DNS1 (or DNS2, DNS1 is shown) is also registered as dns1.bits-go.a.in (& similarly dns2.bits-go.a.in). This machine runs the DNS Service where the bits-go.a.in domain is defined as shown in the left pane.

The entries made under this domain are as under :

Name	Type	Data
_sites		
_tcp		
_udp		
DomainDnsZones		
ForestDnsZones		
(same as parent folder)	Start of Authority (SOA)	[989], dns1.bits-go.a.in,...
(same as parent folder)	Name Server (NS)	dns1.bits-go.a.in.
(same as parent folder)	Host (A)	10.10.10.110
(same as parent folder)	Mail Exchanger (MX)	[10] mail.bits-go.a.in.
dns1	Host (A)	10.10.10.110
dns2	Host (A)	210.212.160.111
dns2	Host (A)	10.10.10.111
mail	Host (A)	210.212.160.113
mail	Text (TXT)	v=spf1 mx -all
www	Host (A)	210.212.160.109

- This nameserver announces itself as the Start of Authority (SOA) (i.e. authoritative nameserver for the bits-go.a.in domain. This declaration has to be corroborated by atleast one external source. That is the nameserver for the ac.in domains must include an entry for bits-go (i.e. bits-go.a.in) and point it to the primary nameserver (i.e. 210.212.160.110). **Although I have not seen how the .ac.in registrar maintains its entries, I am sure that their configuration will be as defined**
- With an NS entry, this computer defines to all others that it is a nameserver.
- It has host entries for its domain. Ideally since it caters to the external domain, all the IP Address mapping (as seen under the Data Column) **MUST be External (publically routable IP Addresses). Here is a source of ERROR WHICH I HAVE REPORTED MANY TIMES BUT CORRECTIVE ACTION HAS NOT BEEN TAKEN. For dns2 entry there is an error as BOTH public and private ip addresses are mapped. Because of the way DNS functions, it might point to one or the other of the entry at a given time. Every time the private IP**

address is wrongly referred to, it will result in a **FAILED QUERY** and the action will not proceed. **THIS ERROR MUST BE RECTIFIED IMMEDIATELY.**

- For mail,bits-go.a.ac.in & www.bits-go.a.ac.in the corresponding publicly routable entries are also done.
- THIS effectively means that everytime we query www.bits-go.a.ac.in (or anything else in this list say - xyz.bits-go.a.ac.in) a query is made to a number of public DNS servers until one (i.e this one) **AUTHORITATIVELY REPLIES** that it is available at **210.212.160.109**

Since this is a windows based DNS service, we can see the Root Hints by going to the properties dialogue box. These entries are the same as those shown in Appendix D.

Since this DNS server is also designed to answer to clients for Non Authoritative answers, it begins it queries from the root servers down to an authoritative answer or till it finds a cached answer (described below)

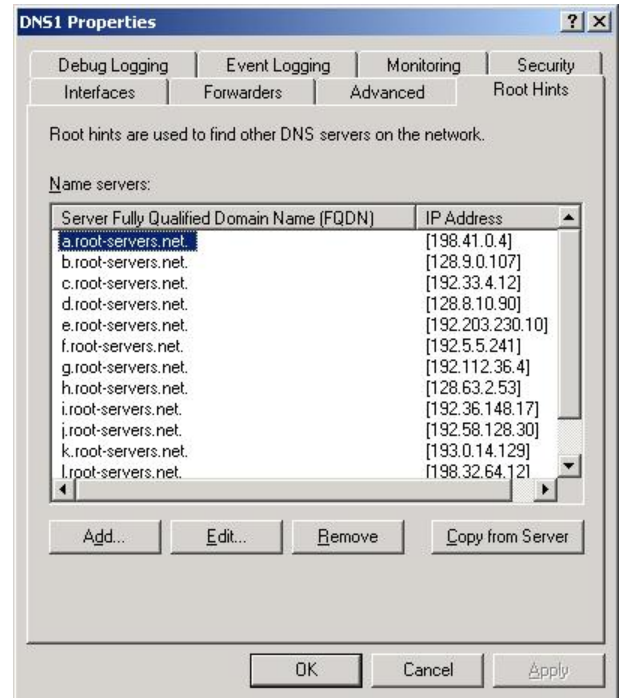
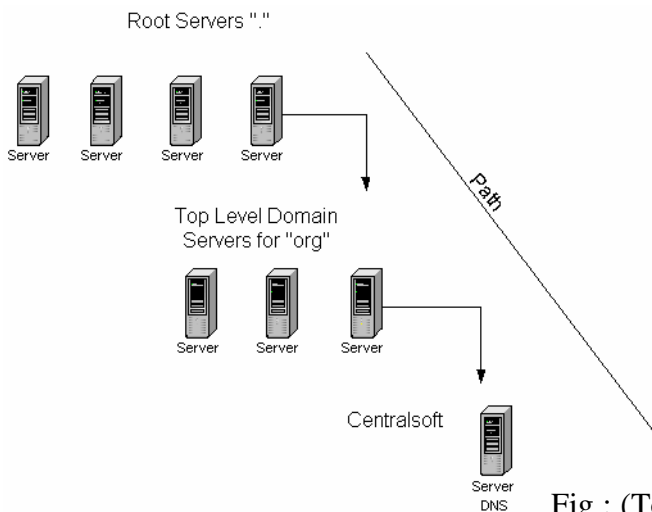
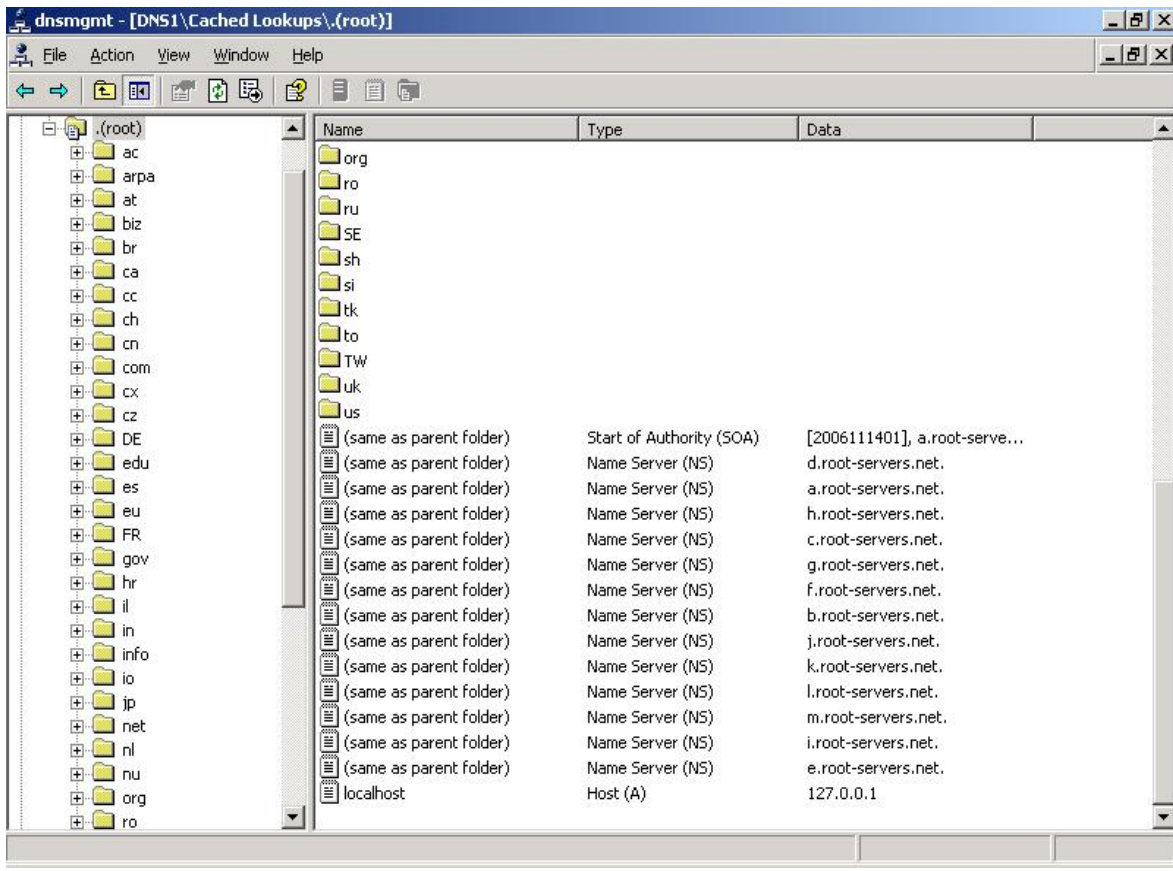
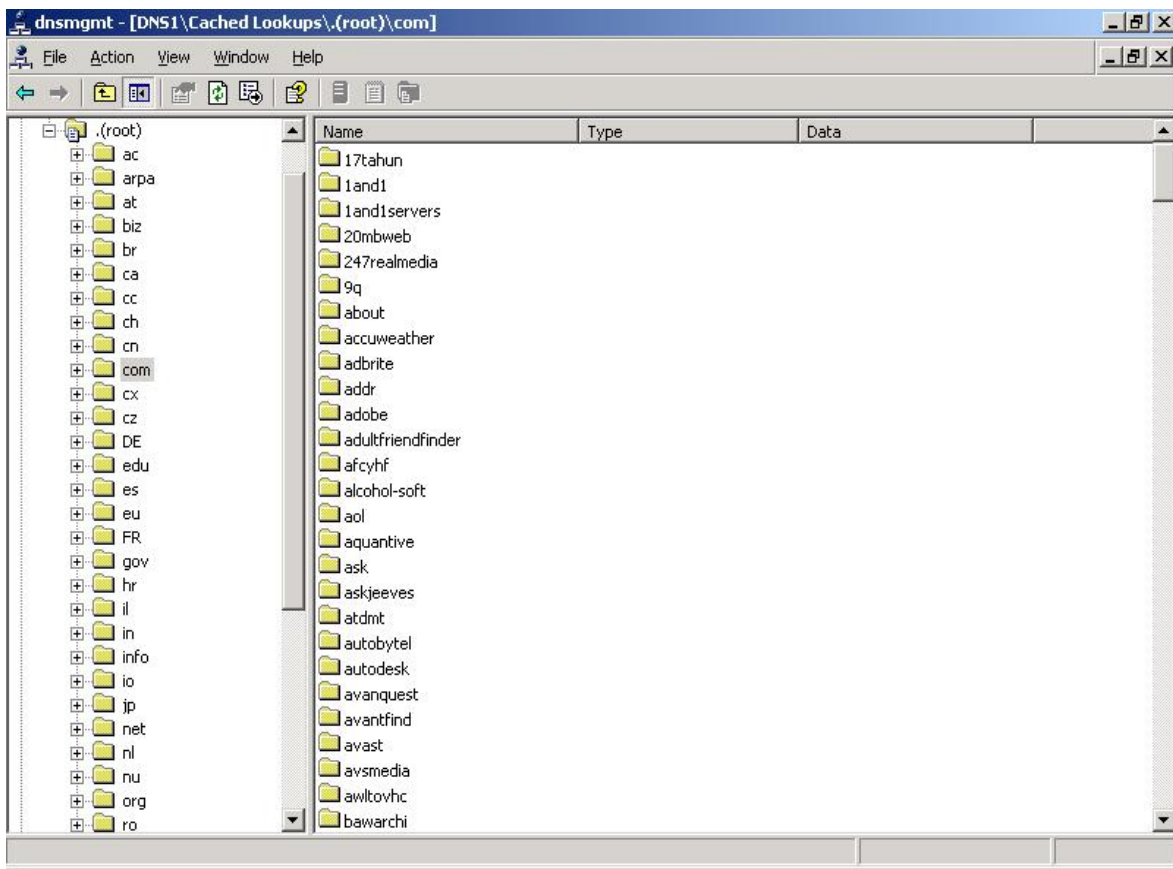


Fig : (Top)- Root Hints (Left)-Shows the progress of a DNS Query from the root servers to a given Centralsoft.org nameserver (bottom) shows the results of a standard nslookup or host query from a client

Cached Entries : Since most modern servers store regularly made DNS queries in their cache for quicker reference, we can conclude that a DNS Server actually performs better over time (**provided its configurations are maintained**) The figure below shows the cached entries sorted alphabetically with respect to domain and then subdomain (root level is first). This is a hierarchical tree which describes the STORED entries. **This can be used to determine what queries have been made on the BITS Network internally or externally Also an intelligent understanding of the DNS Logs can describe who (which IP) made the query & when. This is a ready made resource base available to check for penetration testers and hack attacks.**

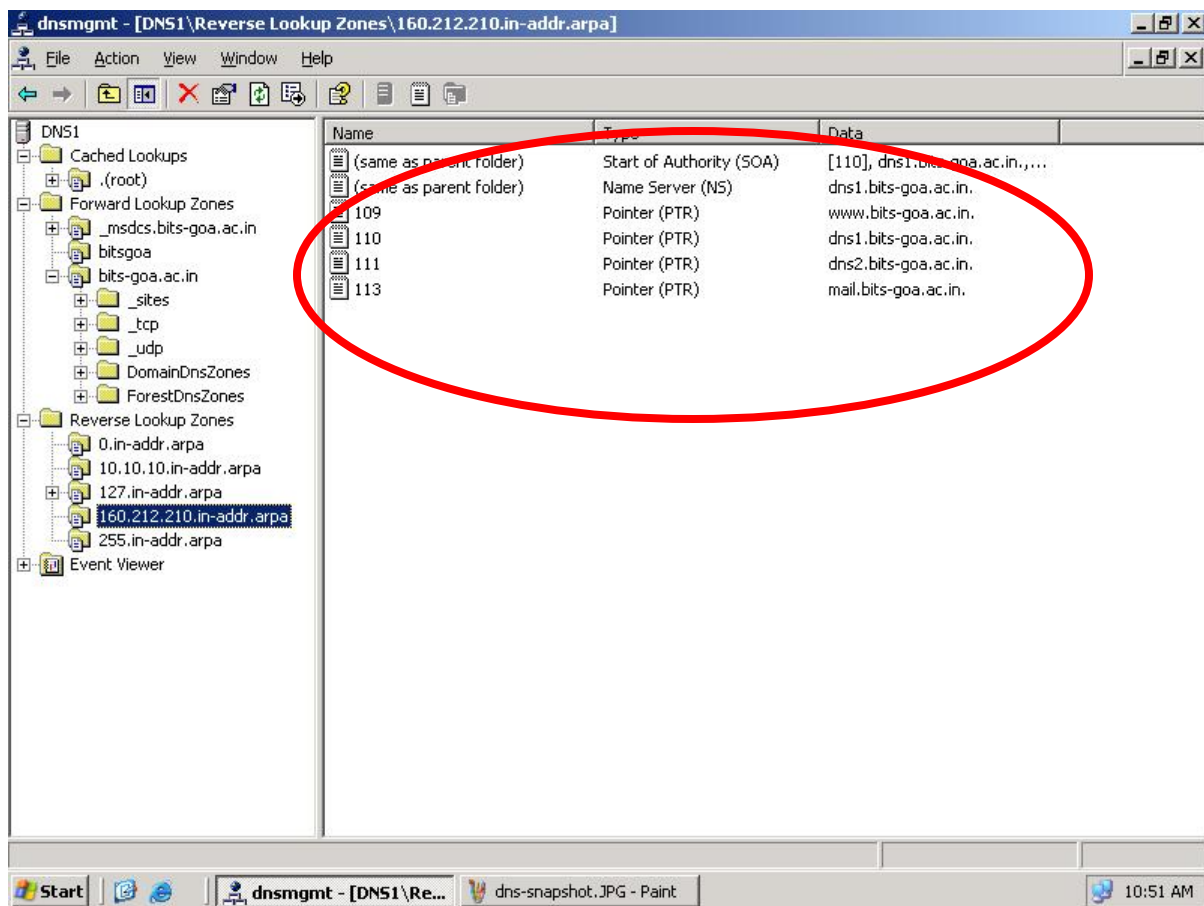


(fig : (Top),- Root level entries for all top level domains. Note also it maintains a list of the root servers as it is regularly queried. (Bottom) Shows a listing of the xyz.com's queried regularly from this DNS server



As of the time of writing this report, we have changed the primary default DNS Server for all the IN-NETWORK queries to the newly installed internal DNS Server, therefore the cached entries here refer to the period when this server was queried (bystatically setting the DNS Servers)

Reverse Lookup Entries: These are the listing of the DNS Server's Answers to IP address queries made to it which if it is configured as a Reverse DNS Server, it will answer to authoritatively. For the Primary Nameserver the reverse lookup will be inverse of 210.212.160.x i.e.160.212.210.in-addr.arpa. **On Microsoft Windows® based machines these entries are created automatically, however it is always better to check.** The NS & SOA Authorities have the same meaning as described previously. The reverse entries are denoted by PTR type of records. This basically maps all IP addresses back to its Fully Qualified Domain Name (FQDN). **It is essential to have an accurate reverse entry for the mail server. This is for SPAM PROTECTION.** The section marked shows reverse entries for the DNS servers themselves and the www & mail servers. **It is essential for a properly configured system to have reverse entries.**



Similar information as described above will also be available from the secondary nameserver (i.e. 10.10.10.111 or 210.212.160.111)

Internal DNS Server

Why ? Terms of reference : Prior to the setup of internal DNS Server, we were using the public DNS servers provided by our internet service provider (viz 61.1.128.5 & 61.1.96.69). Two other DNS server's (primary and secondary) were also created on Windows 2000 Server computers. These were publicly available as 210.212.160.110 & 210.212.160.111 – internally as 10.10.10.110 & 10.10.10.111 for the primary and secondary respectively.

The problem was in assigning the domains which were entirely private hence the need for internal DNS servers arose. To ensure that we used open source software, I chose BIND implementation on a fedora core 3 base.

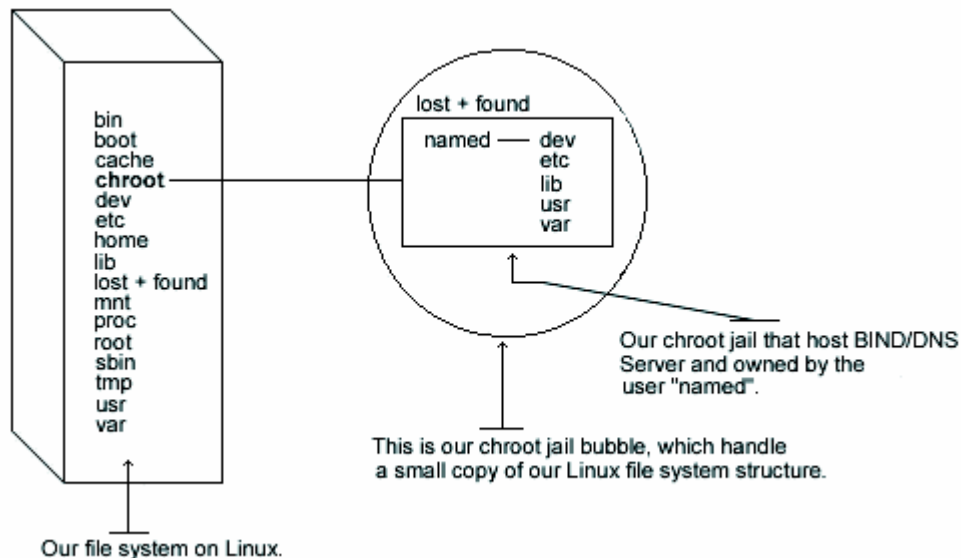
Here in under I am putting forward my understanding of the DNS process for easy replication and understanding later. There were no ready manuals or complete howto's available for this task. The numerous howto's available were often incomplete, outdated, or too complicated.

The description is very specific to the BIND (Refer Appendix) “named” (pronounced name-D) process as implemented on Fedora Core 3.

The chroot jail : The biggest problem faced here was locating where the configuration file is supposed to exist. Intuition told me /var/named/ But this turned out to be blank. This actually contains dummy files which are mirrors of the actual contents inside the chroot folder. **No modification of files present in /var/named will work, we have to navigate inside the chroot jail to make any required changes. Any changes made in the root folder will be lost as they are read only files.**



Ensures that if the system is ever compromised, the attacker will not have access to the entire file system. The attacker might feel that he has compromised the system but actually he has just exposed himself – as his activity has been logged !!



A terminal window titled 'root@dns5:/var/named/chroot/var/named/slaves'. The window has a menu bar with 'Edit', 'View', 'Terminal', 'Tabs', and 'Help'. The terminal shows a prompt '[root@dns5 slaves]# pwd' followed by the output '/var/named/chroot/var/named/slaves'. A red circle highlights the output path, and a blue oval highlights the terminal window title.

To access a folder slaves inside named we actually have to navigate to /var/named/chroot/var/named/slaves

Making the chroot jail effective

- This is important because running it as root defeats the purpose of the jail, and using a different user id that already exists on the system can allow services to access each others' resources.
- Check the /etc/passwd and /etc/group files for a free UID/GID number available.
- In my case, I used number 53 and the name named.

```
[root@dns4] /#useradd -c DNS Server -u 53 -s /bin/false -r -d /chroot/named named 2>/dev/null || :
```

Internal DNS Server (Master)

Making a DNS Server

- Configuring the DNS Service alone is not the only criteria for running a DNS Server.
- For the service to run effectively, the computer running the said service must be configured to refer to itself (127.0.0.1 – local loopback) or to its own ip address for DNS queries.
- Therefore the system will refer to its own authoritative section for defining its answers.

Relevant configuration details for configuring Authoritative Zone (named.conf)

- **REFER APPENDIX E**
- Among a large amount of tweaking of the configuration files, I had to define the primary zone as bits-go.a.in. Infact, I could have chosen any zone name for the zone definition. **But it is possible that these internal servers might some day need to be made public, I chose the same name as our external FQDN**
- I defined it as the primary nameserver by : type master
- I needed to create a file which would contain the actual entries for my newly defined zone which I named bits-go.a.zone. Any filename can be used for this purpose provided its contents are readable and understandable by the named service (i.e. syntax is maintained)
- I define the notify option to be “yes” i.e. it notifies any changes made on it to other slave DNS servers (10.1.1.62 is configured as one)
- I have placed no restrictions on WHO (which IP) can query it. Similar case for update queries.
- It is important to define **allow-transfer {10.1.1.62;}**; This is **REQUIRED** for defining a secondary or slave nameserver to this machine. This way the machine with IP 10.1.1.62 is allowed to access the entries on the primary nameserver through its named service and update its contents.

```
zone "bits-go.a.in" IN {  
    type master;
```

```

file "bits-goa.zone";
notify yes;
allow-query {any;};
allow-update {any;};
allow-transfer {10.1.1.62;};
};

```

Example of A Reverse Zone Authoritative Zone Definition

- Refer any of Appendix G to K
- Unlike Windows, here the named.conf needs to be explicitly told that we are defining a reverse zone.
- **For every reverse zone created, we need to add separate chunks of config similar to the one below**
- I have quoted the reverse zone configuration for the 10.1.1.x range.
- Type, notify & allow transfer have the same meaning as earlier.
- The file where the actual definition is written mentioned as “pri.1.1.10.in-addr.arpa” Any file name could be chosen for this
- **A study of standard naming conventions made me chose this particular schematic for naming the files.**

```

zone "1.1.10.in-addr.arpa" IN {
    type master;
    notify yes;
    file "pri.1.1.10.in-addr.arpa";
    allow-transfer {10.1.1.62;};
};

```

Content Description of a Forward Zone file

Content Description of a Reverse Zone file

```

$TTL 3D
@      IN      SOA    ns1.bits-goac.in. admin.bits-goac.in.(
                          200607213 ; Serial
                          3600      ; Refresh seconds
                          3600      ; retry, seconds
                          3600      ; expire, seconds
                          3600)     ; minimum, seconds

```

- \$TTL defined the Time to Live which I defined as 3 Days
- @ IN SOA is part of standard syntax but defined the configuration to “start at Start of authority ns1.bits-goac.in. (The last dot is essential)
- **EVERYTIME A CHANGE IS DONE IN THE ZONE FILES THE SERIAL NUMBER IS TO BE MODIFIED TO REFLECT THE CHANGES DONE. IT IS IN THE FORMAT YYYYMMDDM.(Where M defines the mth time in a day when a change was done)**
- **THIS IS ESSENTIAL BECAUSE A SECONDARY NAME SERVER CHECKS THE SERIAL NUMBER OF THE ZONE FILE BEFORE “DECIDING” WHETHER IT IS REQUIRED FOR MIGRATION**

```

62 PTR dns4.bits-go.a.ac.in.
222 PTR studentnet.bits-go.a.ac.in
223 PTR orion.bits-go.a.ac.in
225 PTR titan.bits-go.a.ac.in
220 PTR library.bits-go.a.ac.in

```

- The meaning of these lines is similar to that described under the section External Nameserver.

How the nameserver actually performs under a nslookup

```

E:\WINDOWS\system32\cmd.exe - nslookup
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

E:\Documents and Settings\aalap>nslookup
Default Server: dns3.bits-go.a.ac.in
Address: 10.1.1.61

> 10.1.1.225
Server: dns3.bits-go.a.ac.in
Address: 10.1.1.61

Name: titan.bits-go.a.ac.in.1.1.10.in-addr.arpa
Address: 10.1.1.225

> titan
Server: dns3.bits-go.a.ac.in
Address: 10.1.1.61

Name: titan.bits-go.a.ac.in
Address: 10.1.1.225

> -

```

A reverse query made for an IP address translated to the FQDN of the Machine as shown red. Similarly a forward query translates to an ip address. Note that although the test mentions titan it actually travels to the nameserver as titan.bits-go.a.ac.in This is due to the configuration done on the DHCP network specific to the BITS Goa Internal network.

Internal DNS Server (Slave)

Making the Secondary DNS Server

- Similar to the primary nameserver, this computer must be configured to refer to itself (127.0.0.1 – local loopback) or to its own ip address for DNS queries.
- Under no circumstances should it refer to the master for resolving its queries
- In case this is done, the system will still function under normal circumstances but then the secondary nameserver won't be an independent DNS server (though deriving its entries from the primary). When the master fails to respond, this computer will also fail. This is often a standard error encountered in many situations.

How is the Secondary DNS Config Different ?

- Because I never make the entries which it finally answers on it
- It is supposed to prefetch the primary DNS Servers entries as and when they change and keep onto local cache.
- My named.conf configuration is critical here

The Critical Lines in named.conf

```
// query-source address * port 53;
```

```

allow-notify {10.1.1.61;};
recursive-clients 6000;
// the above line was added by RJ/AS/RS on 27/10/2006
};

//
// a caching only nameserver config
//
controls {
    inet 10.1.1.62 allow { any; } keys { rndckey; };
};

```

- These lines would be similar to those on primary named except for a few additions & changes
- I allow notify from the primary name server 10.1.1.61 of changes made on it. these are sent out as broadcasts
- The most critical line is where I define that allow 10.1.1.62 to share access keys with master 10.1.1.61 rndckey is a file which stores a key which is common to both the servers.

```

// Segment added to make This m/c a slave for bits-
goa.ac.in Internal Zone It seeks its addresses from
10.1.1.61 which is defined to be the master
zone "bits-goac.in" IN {
    type slave;
    file "slaves/bits-goac.zone";
    masters { 10.1.1.61; };
};

```

- The lines shown in RED above define the secondary nameserver as one which can authoritatively answer for the bits-goac.in zone (if the master doesn't answer/cannot answer/this server is specifically queried)
- It is a type slave
- Its domain definition files are to be stored in the location slaves/bits-goac.zone Here this definition is purely arbitrary. Any other folder or any other file name can be used. I have used this file names for my consistency.
- Most significantly, I define the masters for the zone. **We can have multiple masters also !! This must be tried !!**

Content Description of a Reverse Zone file(for slave)

```

zone "1.1.10.in-addr.arpa" IN {
    type slave;
    file "slaves/pri.1.1.10.in-addr.arpa";
    masters { 10.1.1.61; };
};

```

- **Besides filename and definition as slave, the description of masters is also done**

Important factors

- Here the zone files as in the master zone are transferred physically to the location mentioned in my zone definition i.e. slaves/<filename>
- All this occurs within the boundaries of the chroot jail
- The obtained files are not included in Appendices

- The following figure shows the contents of dns5 (the secondary slave nameserver) Here the files are shown which are obtained from the primary nameserver.
- Note only three reverse zone files are shown as the rest were not defined in the master zone definition in named.conf when this report was written.

```

root@ dns5:/var/named/chroot/var/named/slaves
File Edit View Terminal Tabs Help
[root@dns5 slaves]# pwd
/var/named/chroot/var/named/slaves
[root@dns5 slaves]# ls -al
total 64
drwxrwx--- 2 named named 4096 Nov  9 21:04 .
drwxrwxrwx 4 root named 4096 Jul 20 07:22 ..
-rw----- 1 named named  676 Nov 14 00:16 bits-go.a.zone
-rw----- 1 named named  586 Aug 31 00:43 bits-go.a.zone~
-rw----- 1 named named  392 Nov 14 00:05 pri.10.10.10.in-addr.arpa
-rw----- 1 named named  512 Nov 14 00:07 pri.1.1.10.in-addr.arpa
-rw----- 1 named named  352 Nov 14 00:08 pri.2.1.10.in-addr.arpa
-rw----- 1 named named  352 Nov 14 00:04 pri.3.1.10.in-addr.arpa
[root@dns5 slaves]# █

```

Procedure to make these DNS servers public : (Future prospects)

- All entries on the DNS servers correspond to internal IP addresses only.
- To make their contents publicly visible we need to change them to their corresponding global public ip addresses (i.e. those starting with 210.212.160.***)
- The corresponding computers need to be cleared for public access by the firewall.
- Since we have registered nameservers as 210.212.160.110 & 210.212.160.111 – the machines currently running as 10.1.1.61 & 10.1.1.62 need to be moved to such publicly defined ip addresses.
- Also the named.conf file needs to have its server 10.1.1.61 entries changed to 210.212.160.110 (or 10.1.1.62 to 210.212.160.111 as the case may be)

SQUID CACHING PROXY



We use Squid Web Proxy which is...

- a full-featured Web proxy cache
- free, open-source software
- the result of many contributions by unpaid (and paid) volunteers

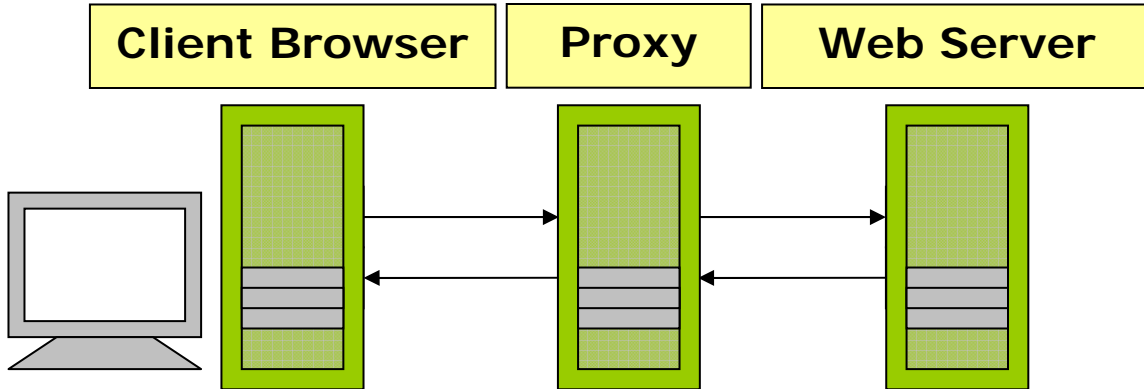


fig : Proxy as an intermediary for Out of Network requests

Proxies

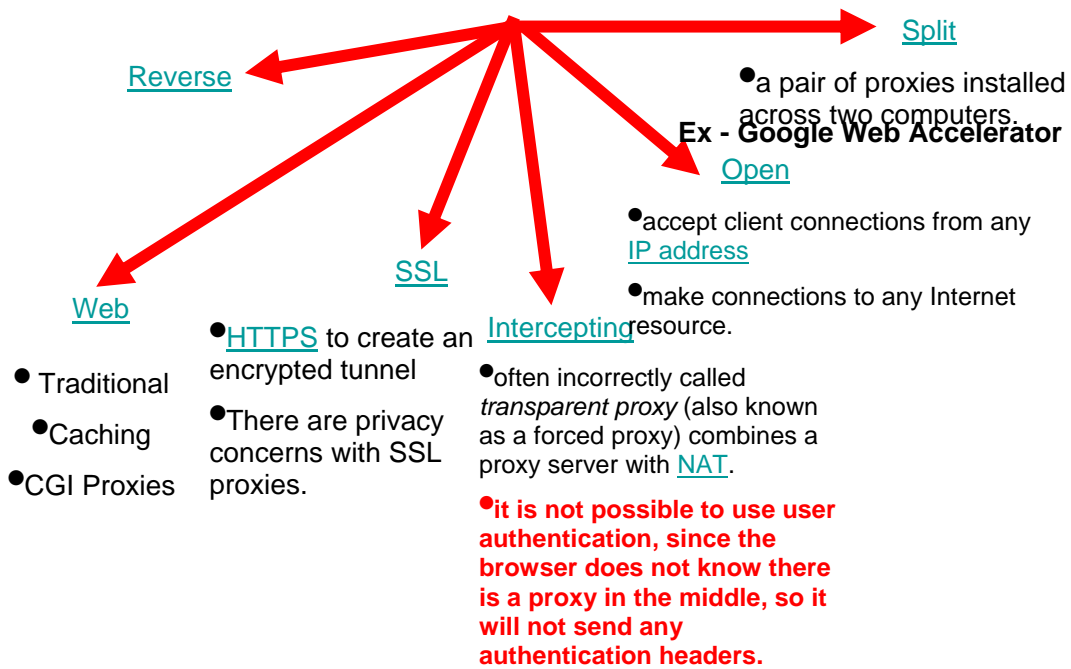
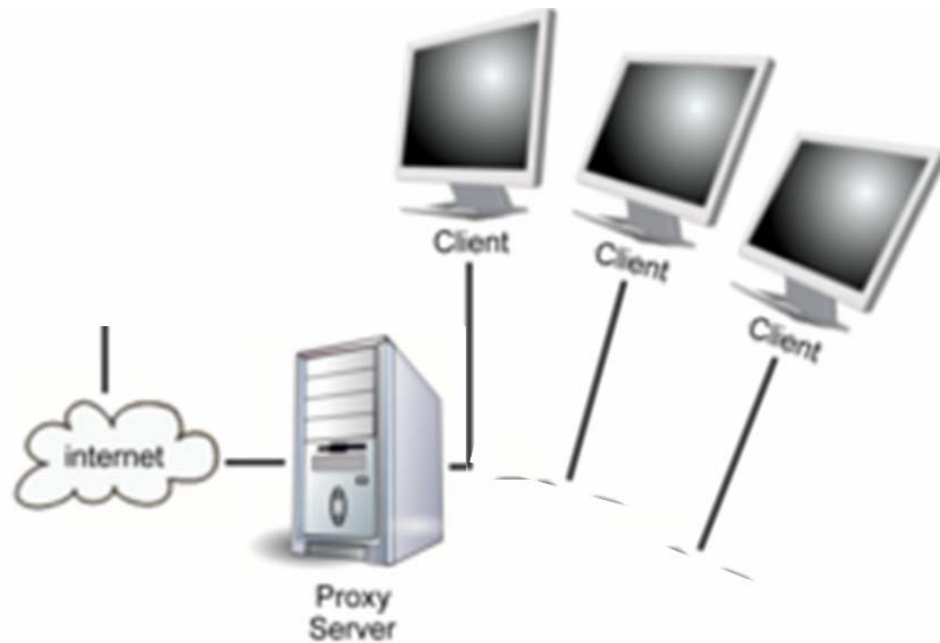


fig : The different type of proxies with functional description of each



Advantage of Using a Web Proxy

- **Improves Performance:**
 - it saves the results of all requests for a certain amount of time (caching)
- **Filter Requests:**
 - Pages to be accessed can be limited
 - Ports / Services Accessed can be controlled
 - Timing of Web Access can be controlled
- **Bandwidth Control:**
 - **Most Important Mandate in the system currently setup on campus**

Caching & Caching Algorithms – Existence of Expiration Algorithms

- Two simple [cache algorithms](#) are Least Recently Used (LRU) and Least Frequently Used (LFU).
- LRU removes the documents that have been left the longest, while LFU removes the least popular documents. The algorithms can also be combined.

Explanation of the most Critical squid.conf lines

Basic Lines

```
http_port 10.1.2.100:8080
# The socket addresses where Squid will listen for HTTP
client requests.
```

- The proxy defined to respond to requests sent only on 10.1.2.100 at port 8080. The IP is optional. Standard ports used are 8080, 3128, 6345. however any non standard port can be used here. For convenience we have chosen port 80

```
cache_mem 100 MB
# maximum_object_size 409600 KB
# Objects larger than this size will NOT be saved on disk.
# minimum_object_size 0 KB
```

```
# Knowingly done so that everything is actually stored.
This is for faster operation
visible_hostname BITSGOA
```

- I define the cache memory to be 100 MB. This was based on the memory size limitations of the host computer
- Objects larger than 409 MB to not be stored in cache
- All objects to be stored in cache
- For all practical purposes, the proxy will be referred to as “BITSGOA” This was just a random name given. **Subsequently, this should be changed to BH1proxy, GH1proxy**

```
cache_replacement_policy lru
memory_replacement_policy lru
```

- LRU Caching algorithm was used. LRU removes the documents that have been left the longest, while LFU removes the least popular documents. **The algorithms can also be combined in SQUID proxy. We must try this out.**

```
# cache_access_log /var/log/squid/access.log
# TAG: cache_access_log
# Logs the client request activity. Contains an entry
for
every HTTP and ICP queries received. To disable, enter
"none".
log_fqdn on
```

- Besides many other logs, the account of who (which IP or username) that makes a request in particular format is stored in a location defined in cache_access_log

Critical Proxy Configuration Lines

```
auth_param basic program /usr/lib/squid/ncsa_auth
/etc/squid/squid_passwd
auth_param basic children 300
auth_param basic realm BITS GOA PROXY
auth_param basic credentialsttl 1 minute
```

- We have used ncsa_authentication scheme. This is the most basic
- **This makes passwords transmitted as cleartext. This is a critical security violation. That is one of the reasons why I chose to move to an LDAP based authentication**
- credentialsttl defines that passwords to be asked every 1 minute for each new request made. This is for user’s security of password

Access Control Definition on the Proxy Server

```
# ACCESS CONTROLS
acl ncsa_users proxy_auth REQUIRED
acl all src 10.0.0.0/255.0.0.0
```

```
acl labs src 10.1.0.0/255.255.0.0 10.2.0.0/255.255.0.0
acl hostels src 10.3.0.0/255.255.0.0 10.4.0.0/255.255.0.0
acl SSL_ports port 443 563
acl Safe_ports port 80          # http
acl day_time time 8:30-17:30
acl night_time time 17:30-24:00 0:00-8:30
acl other_time time 17:30-21:00
```

- Access control lists are defined by the tag acl. This defines the allowed access from the proxy.

```
acl all src 10.0.0.0/255.0.0.0
```

- This line defines the network. This can also be written as 10.0.0.0/8

```
acl labs src 10.1.0.0/255.255.0.0 10.2.0.0/255.255.0.0
acl hostels src 10.3.0.0/255.255.0.0 10.4.0.0/255.255.0.0
```

- These lines define the labs and the hostels with appropriate subnet masks.

```
acl SSL_ports port 443 563
acl Safe_ports port 80          # http
```

- This is an example of the ports which are allowed. Of course this list has been comprehensively updated.

```
acl day_time time 8:30-17:30
acl night_time time 17:30-24:00 0:00-8:30
acl other_time time 17:30-21:00
```

- This is where timings are defined
- **Days of the week can also be defined as MWF etc. This will make it possible for the proxy to turn itself ON/OFF automatically based on the timing. This is to be done in the http_access tag as described below.**

http_access directives – Most Critical instructions

- **The order of lines are very important. SQUID read the http_access lines from top to bottom everytime a request is made to it. Therefore the sequence of checks must also be efficient.**
- **The last line is always converse of the previous line. This is assumed by default.**

```
http_access allow ncsa_users
http_access allow labs day_time other_time
http_access allow hostels night_time
http_access deny banned
http_access deny !Safe_ports
#http_access deny all
#Last line. By default. The final directive is the reverse
of the last okayed directive
```

- ncsa_users allowed means that a splash screen asking username and password dialogue box appears every time a query is made. In case it succeeds the remaining lines are read. In case it fails, access denied violation occurs and a default page is displayed
- **The standard pages for cache errors can be modified. They are at a specified location mentioned in the config file itself. Custom built error messages can also be displayed.**
- Labs are allowed at all times. Hostels are allowed automatically once an access is made at the defined night time.
- http_access deny!Safe_ports is an efficient line. This denies anything that is not on the safe_ports list
- http_access deny all is commented. The default is the inverse of this line.

LDAP SERVER PLANNING

"LDAP is a client-server protocol for accessing a directory service. It was initially used as a front-end to X.500, but can also be used with stand-alone and other kinds of directory servers."

- LDAP lets us "locate organizations, individuals, and other resources such as files and devices in a network, whether on the Internet or on a corporate intranet,"
- An LDAP directory can be distributed among many servers on a network, then replicated and synchronized regularly. An LDAP server is also known as a Directory System Agent (DSA).
- LDAP was developed at the University of Michigan; it's "lightweight" in contrast to DAP, a part of the older X.500 directory protocol for networks.
- It can be used to route email in large organizations as well as look up people and machines across public or private networks.

Many current email clients, including Microsoft Outlook, Eudora, and Netscape Communicator, use some form of LDAP database to look up email addresses. Internic and Infospace are two examples of big public look-up services built with LDAP.

Applications of LDAP

- Internet applications
- Centralized or distributed white pages
- ISP on-line subscriber directory
- Intranet applications
- Internal white pages
- Certificate and CRL distribution
- System/network management database
- For use by people through WWW gateways/clients
- Telephone number, email address lookup
- Can also return photos, spoken names, URLs
- Naming and distribution model allows the directory to contain information from multiple organizations
- Same data can be used by programs
- sendmail extension checks LDAP for addressing
- Netscape, other WWW servers validate user
- Directory synchronization: combining address databases from multiple mail systems
- Dynamic directory extension can be used where information is frequently changing
- Microsoft NetMeeting and other clients will register user in directories of everyone on-line
- Other people can search for that user, based on their name or other attributes
- Terminal capabilities can be determined from directory before communication starts

LDAP SERVER IMPLEMENTATION PROPOSAL

OpenLDAP Software is an open source implementation of the **L**ightweight **D**irectory **A**ccess **P**rotocol.

The suite includes:

- slapd - stand-alone LDAP daemon (server)
- slurpd - stand-alone LDAP update replication daemon
- libraries implementing the LDAP protocol, and utilities, tools, and sample clients.

- Basic Configuration of the slap.conf file has been done completely.
- We now need to integrate code in squid for making this work based on the authserver

```
auth_param basic program /usr/local/squid/libexec/squid_ldap_auth -h  
authserver.bits-go.a.ac.in -b "ou=People,dc=bits-go.a,dc=ac,dc=in"
```

- A script called migrate_passwd.pl to (available in ldap) to be used to generate a LDAP Data Interchange Format (LDIF) file of a type similar to the one shown below
- The mechanism for converting data in one format to another hasn't yet been completed
- Also, a web based mechanism to modify the entries on the LDAP Server needs to be worked out.

```
dn: uid=rsharma,ou=People,dc=barc,dc=ernet,dc=in  
uid: rsharma  
cn: R.Sharma  
sn: R.Sharma  
mail: rsharma@barc.ernet.in  
objectClass: person  
objectClass: organizationalPerson  
objectClass: inetOrgPerson  
objectClass: posixAccount  
objectClass: top  
objectClass: shadowAccount  
userPassword: {crypt}$1$87pxG10I$nt/693FjAtdCwQ5E9cg1m1  
shadowLastChange: 13342  
shadowMin: 5  
shadowMax: 90  
shadowWarning: 10  
shadowInactive: 0  
shadowExpire: -1  
shadowFlag: -1  
loginShell: /bin/bash  
uidNumber: 7881  
gidNumber: 111  
homeDirectory: /usr2/rsharma  
gecos: R.Sharma,,COMP. D,E&IG,22370
```

FUTURE ACTIVITIES

- In house Mail Server Development
 - Having group based mail ids (ex – a EEE student would have <name_of_student>@eee.bits-go.a.ac.in)
 - These servers would be administered in the Group itself.
- LDAP Server Deployment
- Integration of LDAP Server with SQUID PROXY & Mail Server
- Cascading Proxies & “Atleast one proxy per hostel”
- Have multiple interfaces (one for incoming & one for outgoing) for each proxy. This is to enhance the performance.
- Decentralization of the website – <http://www.bits-go.a.ac.in>
 - making groups like swd.bits-go.a.ac.in (DNS entry) + hosting it physically at the location of the Student Welfare Department
 - Having group homepages hosted in the department itself viz (cs.bits-go.a.ac.in) & (eee.bits-go.a.ac.in)

BIBLIOGRAPHY

The bibliography here does not include all the references used as they are included mostly in the appendices. The links mentioned here are mostly referred items.

1. http://www.linuxhomenetworking.com/wiki/index.php/Quick_HOWTO_:_Ch18_:_Configuring_DNS
2. http://www.brandonhutchinson.com/Miscellaneous_BIND_notes.html
3. <http://www.revsys.com/writings/quicktips/bind-permission.html>
Enhancement and Validation of Squid's Cache Replacement Policy
Dilley, John; Arlitt, Martin; Perret, Stephane
HPL-1999-69
990527 External
4. <http://ask.yahoo.com/20000414.html>
Tosa
Milpitas, California
5. <http://www.apricot.net/apricot97/apII/Presentations/LDAProtocol/>
Mark Wahl
President, Critical Angle Inc.
6. <http://www.ietf.org/html.charters/asisd-charter.html>
LDAPv3 Home Page
7. <http://www.critical-angle.com/ldapworld/ldapv3.html>
LDAPv2 Implementations List
8. <http://www.critical-angle.com/dir/lisurvey.html>

APPENDICES

A	Notes which were used to configure the secondary DNS configuration
B	Notes which were ALSO useful to configure the secondary DNS
C	DNS Basics
D	Root Server Files held on a DNS Server
E	Contents of named.conf for the slave nameserver 10.1.1.61
F	Contents of bits-go.a.zone
G	Contents of the pri.1.1.10.in-addr.arpa
H	Contents of the pri.2.1.10.in-addr.arpa
I	Contents of the pri.3.1.10.in-addr.arpa
J	Contents of the pri.4.1.10.in-addr.arpa
K	Contents of the pri.10.10.10.in-addr.arpa
L	Contents of named.conf for the slave nameserver 10.1.1.62

APPENDIX A

Notes which were used to configure the secondary DNS configuration
Source : http://www.brandonhutchinson.com/Miscellaneous_BIND_notes.html

Miscellaneous BIND notes

BIND tracing

The BIND trace file is named *named.run* and is located in */var/named* by default.

Enabling:

ndc trace (BIND 8.x)

rndc trace (BIND 9.x)

kill -USR1 *named_PID* (either version)

Disabling:

ndc notrace (BIND 8.x)

rndc notrace (BIND 9.x)

kill -USR2 *named_PID* (either version)

Flushing individual zone information from the BIND cache

BIND of course clears all cache information with **rndc flush**. It is possible to clear individual zone or resource record information with **rndc flushname**.

Example:

rndc flushname example.com example.net

BIND changes for Fedora Core 2

If your BIND name server acts as a slave for DNS zones, you have to change your BIND configuration to locate slave zone files within the *slaves* subdirectory of your zone data file directory. Otherwise, you may receive errors:

```
Jun 15 11:29:29 host named[6428]: dumping master file: tmp-XXXXPF7BBb: open: permission denied
```

```
Jun 15 11:29:29 host named[6428]: transfer of 'zone/IN' from IP_address#53: failed while receiving responses: permission denied
```

```
Jun 15 11:29:29 host named[6428]: transfer of 'zone/IN' from IP_address#53: end of transfer
```

Old configuration:

```
zone "zone" IN {
    type slave;
    file "zone";
    masters { IP_address; };
};
```

New configuration:

```
zone "zone" IN {
    type slave;
    file "slaves/zone";
    masters { IP_address; };
};
```

Back to brandonhutchinson.com.

Last modified: 11/11/2005

Appendix B

Notes which were ALSO useful to configure the secondary DNS

Source : http://www.brandonhutchinson.com/Miscellaneous_BIND_notes.html

transfer of 'revsys.com/IN' from 69.44.154.136#53: failed while receiving responses:
permission denied

From this error I assumed that my master (aka primary) server was not setup correctly to allow transfers from the secondary. This is normally done with the following configuration option:

```
allow-transfer { 192.168.0.2; };
```

Where 192.168.0.2 is the IP address of the secondary (slave) DNS server.

After I had double checked this configuration on the master to make sure it was there and that I had not done something dumb like typo the IP address. On the surface everything seemed to be perfectly setup, but I was still getting the error.

Eventually I realized that the error was **not** a permission denied error from the remote master server, but from the local DNS server. The error turned out to be a file permission error in the default layout of BIND on a Fedora Core system.

Around the time of Fedora Core 3 the default configuration for BIND is setup to chroot the daemon into it's own filesystem space to help avoid and contain any security breaches. This is a great feature.

To fix your *permission denied* error on your secondary or slave DNS server all you need to do is change the permissions of your data directory to include group write permissions. On my system that directory is /var/named/chroot/var/named/. You can do this with a simple:

```
chmod 775 /var/named/chroot/var/named
```

or

```
chmod g+w /var/named/chroot/var/named
```

It should be noted that you will only run into this error on a secondary or slave DNS server if you have the secondary store it's slave information in a file. This happens when a slave is configured like this:

```
zone "example.com" IN {  
    type slave;  
    file "secondary-example.com";  
    masters { 192.168.0.1; };  
};
```

Another option, that in many ways is more correct on a Fedora Core system, is to store your secondary zone files in the slaves/ directory. This directory is in /var/named/chroot/var/named/ and already has the proper permissions for you. So instead of file "secondary-example.com"; you would simply say: file "slaves/secondary-example.com";.

Hopefully this shows you how to resolve this particular error. These suggestions have been tested on Fedora Core 3 and Core 4. If you find any errors or have any suggestions regarding this information please feel free to E-mail me at frank@revsys.com.

Appendix C

Basic DNS related Definitions

Source :

http://www.linuxhomenetworking.com/wiki/index.php/Quick_HOWTO.../main.css

DNS Domains

Everyone in the world has a first name and a last, or family, name. The same thing is true in the DNS world: A family of Web sites can be loosely described a **domain**. For example, the domain linuxhomenetworking.com has a number of children, such as www.linuxhomenetworking.com and mail.linuxhomenetworking.com for the Web and mail servers, respectively.

BIND

BIND is an acronym for the **Berkeley Internet Name Domain** project, which is a group that maintains the DNS-related software suite that runs under Linux. The most well known program in BIND is named, the daemon that responds to DNS queries from remote machines.

DNS Clients

A DNS client doesn't store DNS information; it must always refer to a DNS server to get it. The only DNS configuration file for a DNS client is the `/etc/resolv.conf` file, which defines the IP address of the DNS server it should use. You shouldn't need to configure any other files. You'll become well acquainted with the `/etc/resolv.conf` file soon.

Authoritative DNS Servers

Authoritative servers provide the definitive information for your DNS domain, such as the names of servers and Web sites in it. They are the last word in information related to your domain.

How DNS Servers Find Out Our Site Information

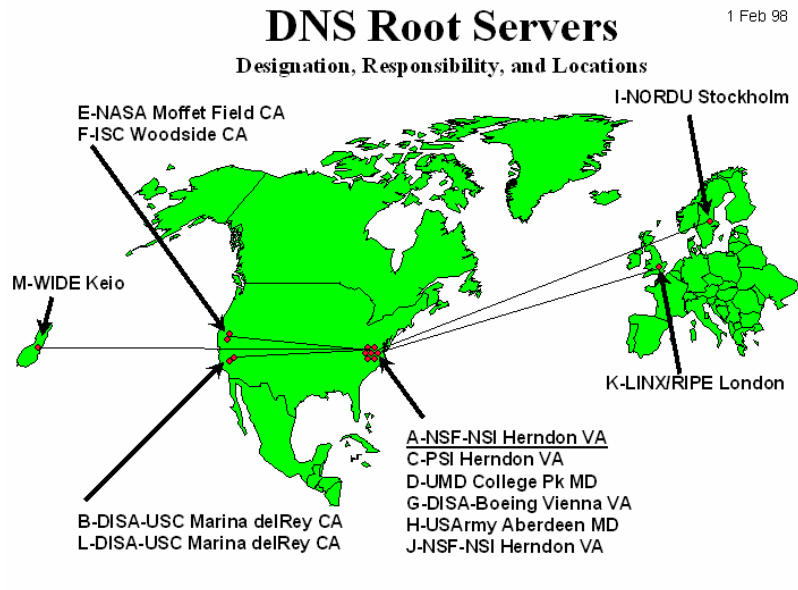
There are 13 root authoritative DNS servers (super duper authorities) that all DNS servers query first. These root servers know all the authoritative DNS servers for all the main domains - .com, .net, and the rest. This layer of servers keep track of all the DNS servers that Web site systems administrators have assigned for their sub domains.

For example, when you register your domain my-site.com, you are actually inserting a record on the .com DNS servers that point to the authoritative DNS servers you assigned for your domain. (More on how to register your site later.).

When To Use A DNS Caching Name Server

Most servers don't ask authoritative servers for DNS directly, they usually ask a **caching DNS server** to do it on their behalf. The caching DNS servers then store (or cache), the most frequently requested information to reduce the lookup overhead of subsequent queries.

If you want to advertise your Web site `www.my-site.com` to the rest of the world, then a regular DNS server is what you require. Setting up a caching DNS server is fairly straightforward and works whether or not your ISP provides you with a static or dynamic Internet IP address.



Location of the Various Root Servers (All perform roughly the same function)



(L) ICANN – Internet Corporation of Assigned Names & Numbers – a quasi-governmental body which ensures generality of the Internet. (R) Picture of the a - Rootserver

Summary of Root Server Info



Root Servers

- There are 13 root authoritative DNS servers (super duper authorities) that all DNS servers query first.
- These root servers know all the authoritative DNS servers for all the main domains - .com, .net, and the rest.
- This layer of servers keep track of all the DNS servers that Web site systems administrators have assigned for their sub domains.

APPENDIX D

Here is a typical list of root servers held by a typical name server: Refer Appendix C for more information on the Root Servers

Source : <http://www.internic.net>

Here is a typical list of root servers held by a typical name server: We note how each and every root server is explicitly defined by its IP Address

```
; This file holds the information on root name servers
; needed to initialize cache of Internet domain name
; servers (e.g. reference this file in the
; "cache . <file>" configuration file of BIND domain
; name servers).
;
; This file is made available by InterNIC registration
; services under anonymous FTP as
;   file           /domain/named.root
;   on server      FTP.RS.INTERNIC.NET
; -OR- under Gopher at RS.INTERNIC.NET
;   under menu     InterNIC Registration Services (NSI)
;   submenu       InterNIC Registration Archives
;   file          named.root
;
; last update:    Aug 22, 1997
; related version of root zone:  1997082200
;
;
; formerly NS.INTERNIC.NET
;
.           3600000 IN  NS  A.ROOT-SERVERS.NET.
A.ROOT-SERVERS.NET. 3600000 A  198.41.0.4
;
; formerly NS1.ISI.EDU
;
.           3600000 NS  B.ROOT-SERVERS.NET.
B.ROOT-SERVERS.NET. 3600000 A  128.9.0.107
;
; formerly C.PSI.NET
;
.           3600000 NS  C.ROOT-SERVERS.NET.
C.ROOT-SERVERS.NET. 3600000 A  192.33.4.12
;
; formerly TERP.UMD.EDU
;
.           3600000 NS  D.ROOT-SERVERS.NET.
D.ROOT-SERVERS.NET. 3600000 A  128.8.10.90
;
; formerly NS.NASA.GOV
;
.           3600000 NS  E.ROOT-SERVERS.NET.
E.ROOT-SERVERS.NET. 3600000 A  192.203.230.10
;
; formerly NS.ISC.ORG
;
.           3600000 NS  F.ROOT-SERVERS.NET.
F.ROOT-SERVERS.NET. 3600000 A  192.5.5.241
;
; formerly NS.NIC.DDN.MIL
;
```

```
.          3600000      NS  G.ROOT-SERVERS.NET.
G.ROOT-SERVERS.NET.  3600000      A   192.112.36.4
;
; formerly AOS.ARL.ARMY.MIL
;
.          3600000      NS  H.ROOT-SERVERS.NET.
H.ROOT-SERVERS.NET.  3600000      A   128.63.2.53
;
; formerly NIC.NORDU.NET
;
.          3600000      NS  I.ROOT-SERVERS.NET.
I.ROOT-SERVERS.NET.  3600000      A   192.36.148.17
;
; temporarily housed at NSI (InterNIC)
;
.          3600000      NS  J.ROOT-SERVERS.NET.
J.ROOT-SERVERS.NET.  3600000      A   198.41.0.10
;
; housed in LINX, operated by RIPE NCC
;
.          3600000      NS  K.ROOT-SERVERS.NET.
K.ROOT-SERVERS.NET.  3600000      A   193.0.14.129
;
; temporarily housed at ISI (IANA)
;
.          3600000      NS  L.ROOT-SERVERS.NET.
L.ROOT-SERVERS.NET.  3600000      A   198.32.64.12
;
; housed in Japan, operated by WIDE
;
.          3600000      NS  M.ROOT-SERVERS.NET.
M.ROOT-SERVERS.NET.  3600000      A   202.12.27.33
; End of File
```



```
        allow-update { none; };
};

zone "255.in-addr.arpa" IN {
    type master;
    file "named.broadcast";
    allow-update { none; };
};

zone "0.in-addr.arpa" IN {
    type master;
    file "named.zero";
    allow-update { none; };
};

zone "bits-go.a.in" IN {
    type master;
    file "bits-go.a.zone";
    notify yes;
    allow-query {any;};
    allow-update {any;};
    allow-transfer {10.1.1.62;};
};

zone "1.1.10.in-addr.arpa" IN {
    type master;
    notify yes;
    file "pri.1.1.10.in-addr.arpa";
    allow-transfer {10.1.1.62;};
};

zone "10.10.10.in-addr.arpa" IN {
    type master;
    notify yes;
    file "pri.10.10.10.in-addr.arpa";
    allow-transfer {10.1.1.62;};
};

zone "2.1.10.in-addr.arpa" IN {
    type master;
    notify yes;
    file "pri.2.1.10.in-addr.arpa";
    allow-transfer {10.1.1.62;};
};

zone "3.1.10.in-addr.arpa" IN {
    type master;
    notify yes;
    file "pri.3.1.10.in-addr.arpa";
    allow-transfer {10.1.1.62;};
};

zone "info" IN {
    type master;
    file "test.zone";
    notify yes;
    allow-query {any;};
    allow-update {any;};
    allow-transfer {10.1.1.62;};
};

include "/etc/rndc.key";
```

APPENDIX F

Contents of the bits-go.a.zone file
Source : Self Written

```
;
; Zone File for bits-go.a.ac.in
; The Full Zone File
;
$TTL 3D
@      IN      SOA    ns1.bits-go.a.ac.in. admin.bits-go.a.ac.in.(
                200608228 ; Serial
                3600      ; Refresh seconds
                3600      ; retry, seconds
                3600      ; expire, seconds
                3600)     ; minimum, seconds

                NS      dns4.bits-go.a.ac.in.
www      A      10.10.10.109
studentnet A      10.1.1.222
orion    A      10.1.1.223
proxy    A      10.1.1.20
proxy    A      10.1.1.21
proxy    A      10.1.1.22
titan    A      10.1.1.225
glimpses06 A      10.1.1.222
library  A      10.1.4.220
S1       A      10.1.1.58
S2       A      10.1.1.59
mailbox  A      10.1.1.57
bits-go.a.ac.in IN MX 10 mailbox.bits-go.a.ac.in.

dns3     A      10.1.1.61
dns4     A      10.1.1.62

dakiya   A      10.10.10.113
central  A      10.10.10.112
mail     CNAME   dakiya
;election CNAME   election
```

APPENDIX G

Contents of the pri.1.1.10.in-addr.arpa
Source : Self Written

```
;
; Reverse Zone File for bits-go.a.ac.in
; Note Made By Aalap as Internal DNS server only
;
; The Full Reverse Zone File
;
$TTL 3D
@      IN      SOA    ns1.bits-go.a.ac.in. admin.bits-go.a.ac.in.(
                                200607213    ; Serial
                                3600          ; Refresh seconds
                                3600          ; retry, seconds
                                3600          ; expire, seconds
                                3600)         ; minimum, seconds
      NS      dns4.bits-go.a.ac.in.;

61     PTR     dns3.bits-go.a.ac.in.
62     PTR     dns4.bits-go.a.ac.in.
222    PTR     studentnet.bits-go.a.ac.in
223    PTR     orion.bits-go.a.ac.in
225    PTR     titan.bits-go.a.ac.in
220    PTR     library.bits-go.a.ac.in
```

APPENDIX H

Contents of the pri.2.1.10.in-addr.arpa

Source : Self Written

```
;
; Reverse Zone File for bits-go.a.c.in
; Note Made By AS as Internal DNS server only
;
; The Full Reverse Zone File
;
$TTL 3D
@      IN      SOA    ns1.bits-go.a.c.in. admin.bits-go.a.c.in.(
                200608211   ; Serial
                3600        ; Refresh seconds
                3600        ; retry, seconds
                3600        ; expire, seconds
                3600)       ; minimum, seconds
      NS      dns4.bits-go.a.c.in. ;

20     PTR    proxy.bits-go.a.c.in.
;18    PTR    election.bits-go.a.c.in
```

APPENDIX I

Contents of the pri.3.1.10.in-addr.arpa
Source : Self Written

```
;
; Reverse Zone File for bits-go.a.c.in
; Note Made By AS as Internal DNS server only
;
; The Full Reverse Zone File
;
$TTL 3D
@      IN      SOA    ns1.bits-go.a.c.in. admin.bits-go.a.c.in.(
                200607211    ; Serial
                3600         ; Refresh seconds
                3600         ; retry, seconds
                3600         ; expire, seconds
                3600)        ; minimum, seconds
        NS      dns4.bits-go.a.c.in.;

21     PTR     proxy.bits-go.a.c.in.
```

APPENDIX J

Contents of the pri.4.1.10.in-addr.arpa
Source : Self Written

```
;
; Reverse Zone File for bits-go.a.c.in
; Note Made By AS as Internal DNS server only
;
; The Full Reverse Zone File
;
$TTL 3D
@      IN      SOA    ns1.bits-go.a.c.in. admin.bits-go.a.c.in.(
                200607213  ; Serial
                3600      ; Refresh seconds
                3600      ; retry, seconds
                3600      ; expire, seconds
                3600)     ; minimum, seconds
NS     dns4.bits-go.a.c.in.;

54     PTR     library1.bits-go.a.c.in
```

APPENDIX K

Contents of the pri.10.10.10.in-addr.arpa
Source : Self Written

```
;
; Reverse Zone File for bits-go.a.c.in
; Note Made By AS as Internal DNS server only
;
; The Full Reverse Zone File
;
$TTL 3D
@      IN      SOA    ns1.bits-go.a.c.in. admin.bits-go.a.c.in.(
                200607194  ; Serial
                3600      ; Refresh seconds
                3600      ; retry, seconds
                3600      ; expire, seconds
                3600)     ; minimum, seconds
      NS      dns4.bits-go.a.c.in.;

113   PTR     dakiya.bits-go.a.c.in.
112   PTR     central.bits-go.a.c.in.
```



```
zone "255.in-addr.arpa" IN {
    type master;
    file "named.broadcast";
    allow-update { none; };
};

zone "0.in-addr.arpa" IN {
    type master;
    file "named.zero";
    allow-update { none; };
};
// Segment added by a to make This m/c a slave for bits-go.a.in
Internal Zone It seeks its addresses from 10.1.1.61 which is defined to
be the master
zone "bits-go.a.in" IN {
    type slave;
    file "slaves/bits-go.a.zone";
    masters { 10.1.1.61; };
};
zone "1.1.10.in-addr.arpa" IN {
    type slave;
    file "slaves/pri.1.1.10.in-addr.arpa";
    masters { 10.1.1.61; };
};
zone "2.1.10.in-addr.arpa" IN {
    type slave;
    file "slaves/pri.2.1.10.in-addr.arpa";
    masters { 10.1.1.61; };
};
zone "3.1.10.in-addr.arpa" IN {
    type slave;
    file "slaves/pri.3.1.10.in-addr.arpa";
    masters { 10.1.1.61; };
};
zone "10.10.10.in-addr.arpa" IN {
    type slave;
    file "slaves/pri.10.10.10.in-addr.arpa";
    masters { 10.1.1.61; };
};
include "/etc/rndc.key";
```

Appendix M

The Description of a resolv.conf file

Source :

http://www.linuxhomenetworking.com/wiki/index.php/Quick_HOWTO.../main.css

The /etc/resolv.conf File

DNS clients (servers not running BIND) use the `/etc/resolv.conf` file to determine both the location of their DNS server and the domains to which they belong. The file generally has two columns; the first contains a keyword, and the second contains the desired values separated by commas. See Table 18.1 for a list of keywords.

Table 18.1 Keywords In /etc/resolv.conf

Keyword	Value
Nameserver	IP address of your DNS nameserver. There should be only one entry per "nameserver" keyword. If there is more than one nameserver, you'll need to have multiple "nameserver" lines.
Domain	The local domain name to be used by default. If the server is <code>bigboy.my-site.com</code> , then the entry would just be <code>my-site.com</code>
Search	If you refer to another server just by its name without the domain added on, DNS on your client will append the server name to each domain in this list and do an DNS lookup on each to get the remote servers' IP address. This is a handy time saving feature to have so that you can refer to servers in the same domain by only their servername without having to specify the domain. The domains in this list must separated by spaces.

Take a look at a sample configuration in which the client server's main domain is `my-site.com`, but it also is a member of domains `my-site.net` and `my-site.org`, which should be searched for shorthand references to other servers. Two name servers, `192.168.1.100` and `192.168.1.102`, provide DNS name resolution:

```
search my-site.com my-site.net my-site.org
```

```
nameserver 192.168.1.100
```

```
nameserver 192.168.1.102
```

The first domain listed after the search directive must be the home domain of your network, in this case `my-site.com`. Placing a domain and search entry in the `/etc/resolv.conf` is redundant, therefore.

Appendix N

The squid.conf file

Source : Mostly Edited from the default file

```
# WELCOME TO SQUID 2
# -----
#
# This is the default Squid configuration file. You may wish
# to look at the Squid home page (http://www.squid-cache.org/)
# for the FAQ and other documentation.
#
# The default Squid config file shows what the defaults for
# various options happen to be. If you don't need to change the
# default, you shouldn't uncomment the line. Doing so may cause
# run-time problems. In some cases "none" refers to no default
# setting at all, while in other cases it refers to a valid
# option - the comments for that keyword indicate if this is the
# case.
#
#
# NETWORK OPTIONS
# -----
# -----
# TAG: http_port
# Usage:      port
#           hostname:port
#           1.2.3.4:port
#
# The socket addresses where Squid will listen for HTTP client
# requests. You may specify multiple socket addresses.
# There are three forms: port alone, hostname with port, and
# IP address with port. If you specify a hostname or IP
# address, then Squid binds the socket to that specific
# address. This replaces the old 'tcp_incoming_address'
# option. Most likely, you do not need to bind to a specific
# address, so you can use the port number alone.
#
# The default port number is 3128.
#
# If you are running Squid in accelerator mode, then you
# probably want to listen on port 80 also, or instead.
#
# The -a command line option will override the *first* port
# number listed here. That option will NOT override an IP
# address, however.
#
# You may specify multiple socket addresses on multiple lines.
#
# If you run Squid on a dual-homed machine with an internal
# and an external interface then we recommend you to specify the
# internal address:port in http_port. This way Squid will only be
# visible on the internal address.
#
#Default:
# http_port 3128
http_port 10.1.2.100:8080
# TAG: https_port
# Usage: [ip:]port cert=certificate.pem [key=key.pem]
[options...]
#
# The socket address where Squid will listen for HTTPS client
```

```
# requests.
#
# This is really only useful for situations where you are
running
# squid in accelerator mode and you want to do the SSL work at
the
# accelerator level.
#
# You may specify multiple socket addresses on multiple lines,
# each with their own SSL certificate and/or options.
#
# Options:
#
# cert= Path to SSL certificate (PEM format)
#
# key= Path to SSL private key file (PEM format)
# if not specified, the certificate file is
# assumed to be a combined certificate and
# key file
#
# version= The version of SSL/TLS supported
# 1 automatic (default)
# 2 SSLv2 only
# 3 SSLv3 only
# 4 TLSv1 only
#
# cipher= Colon separated list of supported ciphers
#
# options= Various SSL engine options. The most important
# being:
# NO_SSLv2 Disallow the use of SSLv2
# NO_SSLv3 Disallow the use of SSLv3
# NO_TLSv1 Disallow the use of TLSv1
# See src/ssl_support.c or OpenSSL documentation
# for a more complete list.
#
#Default:
# none

# TAG: ssl_unclean_shutdown
# Some browsers (especially MSIE) bugs out on SSL shutdown
# messages.
#
#Default:
# ssl_unclean_shutdown off

# TAG: icp_port
# The port number where Squid sends and receives ICP queries to
# and from neighbor caches. Default is 3130. To disable use
# "0". May be overridden with -u on the command line.
#
#Default:
# icp_port 3130

# TAG: htcp_port
# Note: This option is only available if Squid is rebuilt with the
# --enable-htcp option
#
# The port number where Squid sends and receives HTCP queries to
# and from neighbor caches. Default is 4827. To disable use
# "0".
#
#Default:
# htcp_port 4827

# TAG: mcast_groups
```



```

# cache_peer parent.foo.net      parent      3128  3130  [proxy-
only]
# cache_peer sib1.foo.net        sibling      3128  3130  [proxy-
only]
# cache_peer sib2.foo.net        sibling      3128  3130  [proxy-
only]
#
#         type:  either 'parent', 'sibling', or 'multicast'.
#
# proxy_port:  The port number where the cache listens for proxy
#               requests.
#
# icp_port:   Used for querying neighbor caches about
#               objects.  To have a non-ICP neighbor
#               specify '7' for the ICP port and make sure the
#               neighbor machine has the UDP echo port
#               enabled in its /etc/inetd.conf file.
#
# options: proxy-only
#           weight=n
#           ttl=n
#           no-query
#           default
#           round-robin
#           multicast-responder
#           closest-only
#           no-digest
#           no-netdb-exchange
#           no-delay
#           login=user:password | PASS | *:password
#           connect-timeout=nn
#           digest-url=url
#           allow-miss
#           max-conn
#
# use 'proxy-only' to specify that objects fetched
# from this cache should not be saved locally.
#
# use 'weight=n' to specify a weighted parent.
# The weight must be an integer.  The default weight
# is 1, larger weights are favored more.
#
# use 'ttl=n' to specify a IP multicast TTL to use
# when sending an ICP queries to this address.
# Only useful when sending to a multicast group.
# Because we don't accept ICP replies from random
# hosts, you must configure other group members as
# peers with the 'multicast-responder' option below.
#
# use 'no-query' to NOT send ICP queries to this
# neighbor.
#
# use 'default' if this is a parent cache which can
# be used as a "last-resort." You should probably
# only use 'default' in situations where you cannot
# use ICP with your parent cache(s).
#
# use 'round-robin' to define a set of parents which
# should be used in a round-robin fashion in the
# absence of any ICP queries.
#
# 'multicast-responder' indicates that the named peer
# is a member of a multicast group.  ICP queries will
# not be sent directly to the peer, but ICP replies
# will be accepted from it.
#
#

```

```
# 'closest-only' indicates that, for ICP_OP_MISS
# replies, we'll only forward CLOSEST_PARENT_MISSES
# and never FIRST_PARENT_MISSES.
#
# use 'no-digest' to NOT request cache digests from
# this neighbor.
#
# 'no-netdb-exchange' disables requesting ICMP
# RTT database (NetDB) from the neighbor.
#
# use 'no-delay' to prevent access to this neighbor
# from influencing the delay pools.
#
# use 'login=user:password' if this is a
personal/workgroup
# proxy and your parent requires proxy authentication.
# Note: The string can include URL escapes (i.e. %20 for
# spaces). This also means that % must be written as %%.
#
# use 'login=PASS' if users must authenticate against
# the upstream proxy. This will pass the users
credentials
# as they are to the peer proxy. This only works for the
# Basic HTTP authentication scheme. Note: To combine this
# with proxy_auth both proxies must share the same user
# database as HTTP only allows for one proxy login.
# Also be warned that this will expose your users proxy
# password to the peer. USE WITH CAUTION
#
# use 'login=*:password' to pass the username to the
# upstream cache, but with a fixed password. This is
meant
# to be used when the peer is in another administrative
# domain, but it is still needed to identify each user.
# The star can optionally be followed by some extra
# information which is added to the username. This can
# be used to identify this proxy to the peer, similar to
# the login=username:password option above.
#
# use 'connect-timeout=nn' to specify a peer
# specific connect timeout (also see the
# peer_connect_timeout directive)
#
# use 'digest-url=url' to tell Squid to fetch the cache
# digest (if digests are enabled) for this host from
# the specified URL rather than the Squid default
# location.
#
# use 'allow-miss' to disable Squid's use of only-if-
cached
# when forwarding requests to siblings. This is
primarily
# useful when icp_hit_stale is used by the sibling. To
# extensive use of this option may result in forwarding
# loops, and you should avoid having two-way peerings
# with this option. (for example to deny peer usage on
# requests from peer by denying cache_peer_access if the
# source is a peer)
#
# use 'max-conn' to limit the amount of connections
Squid
# may open to this peer.
#
# NOTE: non-ICP neighbors must be specified as 'parent'.
#
#Default:
```

```

# none

# TAG: cache_peer_domain
#   Use to limit the domains for which a neighbor cache will be
#   queried.  Usage:
#
#   cache_peer_domain cache-host domain [domain ...]
#   cache_peer_domain cache-host !domain
#
#   For example, specifying
#
#       cache_peer_domain parent.foo.net      .edu
#
#   has the effect such that UDP query packets are sent to
#   'bigserver' only when the requested object exists on a
#   server in the .edu domain.  Prefixing the domainname
#   with '!' means that the cache will be queried for objects
#   NOT in that domain.
#
#   NOTE: * Any number of domains may be given for a cache-host,
#         * either on the same or separate lines.
#         * When multiple domains are given for a particular
#         * cache-host, the first matched domain is applied.
#         * Cache hosts with no domain restrictions are queried
#         * for all requests.
#         * There are no defaults.
#         * There is also a 'cache_peer_access' tag in the ACL
#         * section.
#
#Default:
# none

# TAG: neighbor_type_domain
#   usage: neighbor_type_domain parent|sibling domain domain ...
#
#   Modifying the neighbor type for specific domains is now
#   possible.  You can treat some domains differently than the the
#   default neighbor type specified on the 'cache_peer' line.
#   Normally it should only be necessary to list domains which
#   should be treated differently because the default neighbor type
#   applies for hostnames which do not match domains listed here.
#
#EXAMPLE:
#   cache_peer parent cache.foo.org 3128 3130
#   neighbor_type_domain cache.foo.org sibling .com .net
#   neighbor_type_domain cache.foo.org sibling .au .de
#
#Default:
# none

# TAG: icp_query_timeout      (msec)
#   Normally Squid will automatically determine an optimal ICP
#   query timeout value based on the round-trip-time of recent ICP
#   queries.  If you want to override the value determined by
#   Squid, set this 'icp_query_timeout' to a non-zero value.  This
#   value is specified in MILLISECONDS, so, to use a 2-second
#   timeout (the old default), you would write:
#
#       icp_query_timeout 2000
#
#Default:
# icp_query_timeout 0

# TAG: maximum_icp_query_timeout      (msec)
#   Normally the ICP query timeout is determined dynamically.  But
#   sometimes it can lead to very large values (say 5 seconds).

```

```

# Use this option to put an upper limit on the dynamic timeout
# value. Do NOT use this option to always use a fixed (instead
# of a dynamic) timeout value. To set a fixed timeout see the
# 'icp_query_timeout' directive.
#
#Default:
# maximum_icp_query_timeout 2000

# TAG: mcast_icp_query_timeout      (msec)
# For Multicast peers, Squid regularly sends out ICP "probes" to
# count how many other peers are listening on the given multicast
# address. This value specifies how long Squid should wait to
# count all the replies. The default is 2000 msec, or 2
# seconds.
#
#Default:
# mcast_icp_query_timeout 2000

# TAG: dead_peer_timeout            (seconds)
# This controls how long Squid waits to declare a peer cache
# as "dead." If there are no ICP replies received in this
# amount of time, Squid will declare the peer dead and not
# expect to receive any further ICP replies. However, it
# continues to send ICP queries, and will mark the peer as
# alive upon receipt of the first subsequent ICP reply.
#
# This timeout also affects when Squid expects to receive ICP
# replies from peers. If more than 'dead_peer' seconds have
# passed since the last ICP reply was received, Squid will not
# expect to receive an ICP reply on the next query. Thus, if
# your time between requests is greater than this timeout, you
# will see a lot of requests sent DIRECT to origin servers
# instead of to your parents.
#
#Default:
# dead_peer_timeout 10 seconds

# TAG: hierarchy_stoplist
# A list of words which, if found in a URL, cause the object to
# be handled directly by this cache. In other words, use this
# to not query neighbor caches for certain objects. You may
# list this option multiple times.
#We recommend you to use at least the following line.
hierarchy_stoplist cgi-bin ?

# TAG: no_cache
# A list of ACL elements which, if matched, cause the request to
# not be satisfied from the cache and the reply to not be cached.
# In other words, use this to force certain objects to never be
# cached.
#
# You must use the word 'DENY' to indicate the ACL names which
# should
# NOT be cached.
#
#We recommend you to use the following two lines.
acl QUERY urlpath_regex cgi-bin \?
no_cache deny QUERY

# OPTIONS WHICH AFFECT THE CACHE SIZE
# -----
# -----

# TAG: cache_mem (bytes)
# NOTE: THIS PARAMETER DOES NOT SPECIFY THE MAXIMUM PROCESS SIZE.

```

```

# IT ONLY PLACES A LIMIT ON HOW MUCH ADDITIONAL MEMORY SQUID WILL
# USE AS A MEMORY CACHE OF OBJECTS. SQUID USES MEMORY FOR OTHER
# THINGS AS WELL. SEE THE SQUID FAQ SECTION 8 FOR DETAILS.
#
# 'cache_mem' specifies the ideal amount of memory to be used
# for:
#     * In-Transit objects
#     * Hot Objects
#     * Negative-Cached objects
#
# Data for these objects are stored in 4 KB blocks. This
# parameter specifies the ideal upper limit on the total size of
# 4 KB blocks allocated. In-Transit objects take the highest
# priority.
#
# In-transit objects have priority over the others. When
# additional space is needed for incoming data, negative-cached
# and hot objects will be released. In other words, the
# negative-cached and hot objects will fill up any unused space
# not needed for in-transit objects.
#
# If circumstances require, this limit will be exceeded.
# Specifically, if your incoming request rate requires more than
# 'cache_mem' of memory to hold in-transit objects, Squid will
# exceed this limit to satisfy the new requests. When the load
# decreases, blocks will be freed until the high-water mark is
# reached. Thereafter, blocks will be used to store hot
# objects.
#
#Default:
cache_mem 100 MB

# TAG: cache_swap_low (percent, 0-100)
# TAG: cache_swap_high (percent, 0-100)
#
# The low- and high-water marks for cache object replacement.
# Replacement begins when the swap (disk) usage is above the
# low-water mark and attempts to maintain utilization near the
# low-water mark. As swap utilization gets close to high-water
# mark object eviction becomes more aggressive. If utilization is
# close to the low-water mark less replacement is done each time.
#
# Defaults are 90% and 95%. If you have a large cache, 5% could be
# hundreds of MB. If this is the case you may wish to set these
# numbers closer together.
#
#Default:
# cache_swap_low 90
# cache_swap_high 95

# TAG: maximum_object_size (bytes)
# Objects larger than this size will NOT be saved on disk. The
# value is specified in kilobytes, and the default is 4MB. If
# you wish to get a high BYTES hit ratio, you should probably
# increase this (one 32 MB object hit counts for 3200 10KB
# hits). If you wish to increase speed more than you want to
# save bandwidth you should leave this low.
#
# NOTE: if using the LFUDA replacement policy you should increase
# this value to maximize the byte hit rate improvement of LFUDA!
# See replacement_policy below for a discussion of this policy.
#
#Default:
# maximum_object_size 409600 KB

# TAG: minimum_object_size (bytes)

```

```

#   Objects smaller than this size will NOT be saved on disk.  The
#   value is specified in kilobytes, and the default is 0 KB, which
#   means there is no minimum.
#
#Default:
# minimum_object_size 0 KB

# TAG: maximum_object_size_in_memory      (bytes)
#   Objects greater than this size will not be attempted to kept
in
#   the memory cache. This should be set high enough to keep
objects
#   accessed frequently in memory to improve performance whilst
low
#   enough to keep larger objects from hoarding cache_mem .
#
#Default:
# maximum_object_size_in_memory 80000 KB

# TAG: ipcache_size      (number of entries)
# TAG: ipcache_low      (percent)
# TAG: ipcache_high     (percent)
#   The size, low-, and high-water marks for the IP cache.
#
#Default:
# ipcache_size 1024
# ipcache_low 90
# ipcache_high 95

# TAG: fqdn_cache_size (number of entries)
#   Maximum number of FQDN cache entries.
#
#Default:
# fqdn_cache_size 1024

# TAG: cache_replacement_policy
#   The cache replacement policy parameter determines which
#   objects are evicted (replaced) when disk space is needed.
#
#   lru          : Squid's original list based LRU policy
#   heap GDSF   : Greedy-Dual Size Frequency
#   heap LFUDA  : Least Frequently Used with Dynamic Aging
#   heap LRU    : LRU policy implemented using a heap
#
#   Applies to any cache_dir lines listed below this.
#
#   The LRU policies keeps recently referenced objects.
#
#   The heap GDSF policy optimizes object hit rate by keeping smaller
#   popular objects in cache so it has a better chance of getting a
#   hit.  It achieves a lower byte hit rate than LFUDA though since
#   it evicts larger (possibly popular) objects.
#
#   The heap LFUDA policy keeps popular objects in cache regardless
of
#   their size and thus optimizes byte hit rate at the expense of
#   hit rate since one large, popular object will prevent many
#   smaller, slightly less popular objects from being cached.
#
#   Both policies utilize a dynamic aging mechanism that prevents
#   cache pollution that can otherwise occur with frequency-based
#   replacement policies.
#
#   NOTE: if using the LFUDA replacement policy you should increase
#   the value of maximum_object_size above its default of 4096 KB to
#   to maximize the potential byte hit rate improvement of LFUDA.

```

```
#
#   For more information about the GDSF and LFUDA cache replacement
#   policies see http://www.hpl.hp.com/techreports/1999/HPL-1999-69.html
#   and http://fog.hpl.external.hp.com/techreports/98/HPL-98-173.html.
#
#Default:
cache_replacement_policy lru
```

```
# TAG: memory_replacement_policy
#   The memory replacement policy parameter determines which
#   objects are purged from memory when memory space is needed.
#
#   See cache_replacement_policy for details.
#
#Default:
memory_replacement_policy lru
```

```
# LOGFILE PATHNAMES AND CACHE DIRECTORIES
```

```
# -----
# -----
```

```
# TAG: cache_dir
#   Usage:
#
#   cache_dir Type Directory-Name Fs-specific-data [options]
#
#   cache_dir diskd Maxobjsize Directory-Name MB L1 L2 Q1 Q2
#
#   You can specify multiple cache_dir lines to spread the
#   cache among different disk partitions.
#
#   Type specifies the kind of storage system to use. Only "ufs"
#   is built by default. To enable any of the other storage systems
#   see the --enable-storeio configure option.
#
#   'Directory' is a top-level directory where cache swap
#   files will be stored. If you want to use an entire disk
#   for caching, then this can be the mount-point directory.
#   The directory must exist and be writable by the Squid
#   process. Squid will NOT create this directory for you.
#
#   The ufs store type:
#
#   "ufs" is the old well-known Squid storage format that has always
#   been there.
#
#   cache_dir ufs Directory-Name Mbytes L1 L2 [options]
#
#   'Mbytes' is the amount of disk space (MB) to use under this
#   directory. The default is 100 MB. Change this to suit your
#   configuration. Do NOT put the size of your disk drive here.
#   Instead, if you want Squid to use the entire disk drive,
#   subtract 20% and use that value.
#
#   'Level-1' is the number of first-level subdirectories which
#   will be created under the 'Directory'. The default is 16.
#
#   'Level-2' is the number of second-level subdirectories which
#   will be created under each first-level directory. The default
#   is 256.
#
#   The aufs store type:
#
```

```

# "aufs" uses the same storage format as "ufs", utilizing
# POSIX-threads to avoid blocking the main Squid process on
# disk-I/O. This was formerly known in Squid as async-io.
#
# cache_dir aufs Directory-Name Mbytes L1 L2 [options]
#
# see argument descriptions under ufs above
#
# The diskd store type:
#
# "diskd" uses the same storage format as "ufs", utilizing a
# separate process to avoid blocking the main Squid process on
# disk-I/O.
#
# cache_dir diskd Directory-Name Mbytes L1 L2 [options] [Q1=n]
[Q2=n]
#
# see argument descriptions under ufs above
#
# Q1 specifies the number of unacknowledged I/O requests when Squid
# stops opening new files. If this many messages are in the queues,
# Squid won't open new files. Default is 64
#
# Q2 specifies the number of unacknowledged messages when Squid
# starts blocking. If this many messages are in the queues,
# Squid blocks until it receives some replies. Default is 72
#
# Common options:
#
# read-only, this cache_dir is read only.
#
# max-size=n, refers to the max object size this storedir supports.
# It is used to initially choose the storedir to dump the object.
# Note: To make optimal use of the max-size limits you should order
# the cache_dir lines with the smallest max-size value first and
the
# ones with no max-size specification last.
#
#Default:
cache_dir ufs /var/spool/squid 10000 16 256

# TAG: cache_access_log
# Logs the client request activity. Contains an entry for
# every HTTP and ICP queries received. To disable, enter "none".
#
#Default:
# cache_access_log /var/log/squid/access.log

# TAG: cache_log
# Cache logging file. This is where general information about
# your cache's behavior goes. You can increase the amount of data
# logged to this file with the "debug_options" tag below.
#
#Default:
# cache_log /var/log/squid/cache.log

# TAG: cache_store_log
# Logs the activities of the storage manager. Shows which
# objects are ejected from the cache, and which objects are
# saved and for how long. To disable, enter "none". There are
# not really utilities to analyze this data, so you can safely
# disable it.
#
#Default:
# cache_store_log /var/log/squid/store.log

```

```

# TAG: cache_swap_log
#   Location for the cache "swap.log." This log file holds the
#   metadata of objects saved on disk. It is used to rebuild the
#   cache during startup. Normally this file resides in each
#   'cache_dir' directory, but you may specify an alternate
#   pathname here. Note you must give a full filename, not just
#   a directory. Since this is the index for the whole object
#   list you CANNOT periodically rotate it!
#
#   If %s can be used in the file name then it will be replaced with
#   a
#   a representation of the cache_dir name where each / is replaced
#   with '.'. This is needed to allow adding/removing cache_dir
#   lines when cache_swap_log is being used.
#
#   If have more than one 'cache_dir', and %s is not used in the name
#   then these swap logs will have names such as:
#
#       cache_swap_log.00
#       cache_swap_log.01
#       cache_swap_log.02
#
#   The numbered extension (which is added automatically)
#   corresponds to the order of the 'cache_dir' lines in this
#   configuration file. If you change the order of the 'cache_dir'
#   lines in this file, then these log files will NOT correspond to
#   the correct 'cache_dir' entry (unless you manually rename
#   them). We recommend that you do NOT use this option. It is
#   better to keep these log files in each 'cache_dir' directory.
#
#Default:
# none

# TAG: emulate_httpd_log      on|off
#   The Cache can emulate the log file format which many 'httpd'
#   programs use. To disable/enable this emulation, set
#   emulate_httpd_log to 'off' or 'on'. The default
#   is to use the native log format since it includes useful
#   information that Squid-specific log analyzers use.
#
#Default:
# emulate_httpd_log off

# TAG: log_ip_on_direct      on|off
#   Log the destination IP address in the hierarchy log tag when
#   going
#   direct. Earlier Squid versions logged the hostname here. If you
#   prefer the old way set this to off.
#
#Default:
# log_ip_on_direct on

# TAG: mime_table
#   Pathname to Squid's MIME table. You shouldn't need to change
#   this, but the default file contains examples and formatting
#   information if you do.
#
#Default:
# mime_table /etc/squid/mime.conf

# TAG: log_mime_hdrs      on|off
#   The Cache can record both the request and the response MIME
#   headers for each HTTP transaction. The headers are encoded
#   safely and will appear as two bracketed fields at the end of
#   the access log (for either the native or httpd-emulated log
#   formats). To enable this logging set log_mime_hdrs to 'on'.

```

```
#
#Default:
# log_mime_hdrs off

# TAG: useragent_log
# Note: This option is only available if Squid is rebuilt with the
#       --enable-useragent-log option
#
#       Squid will write the User-Agent field from HTTP requests
#       to the filename specified here.  By default useragent_log
#       is disabled.
#
#Default:
# none

# TAG: referer_log
# Note: This option is only available if Squid is rebuilt with the
#       --enable-referer-log option
#
#       Squid will write the Referer field from HTTP requests to the
#       filename specified here.  By default referer_log is disabled.
#
#Default:
# none

# TAG: pid_filename
#       A filename to write the process-id to.  To disable, enter "none".
#
#Default:
# pid_filename /var/run/squid.pid

# TAG: debug_options
#       Logging options are set as section,level where each source file
#       is assigned a unique section.  Lower levels result in less
#       output, Full debugging (level 9) can result in a very large
#       log file, so be careful.  The magic word "ALL" sets debugging
#       levels for all sections.  We recommend normally running with
#       "ALL,1".
#
#Default:
# debug_options ALL,1

# TAG: log_fqdn on|off
#       Turn this on if you wish to log fully qualified domain names
#       in the access.log.  To do this Squid does a DNS lookup of all
#       IP's connecting to it.  This can (in some situations) increase
#       latency, which makes your cache seem slower for interactive
#       browsing.
#
#Default:
# log_fqdn off
log_fqdn on

# TAG: client_netmask
#       A netmask for client addresses in logfiles and cachemgr output.
#       Change this to protect the privacy of your cache clients.
#       A netmask of 255.255.255.0 will log all IP's in that range with
#       the last digit set to '0'.
#
#Default:
# client_netmask 255.255.255.255
```

```
# OPTIONS FOR EXTERNAL SUPPORT PROGRAMS
```

```
# -----  
-----
```

```
# TAG: ftp_user
#   If you want the anonymous login password to be more informative
#   (and enable the use of picky ftp servers), set this to something
#   reasonable for your domain, like wwwuser@somewhere.net
#
#   The reason why this is domainless by default is that the
#   request can be made on the behalf of a user in any domain,
#   depending on how the cache is used.
#   Some ftp server also validate that the email address is valid
#   (for example perl.com).
#
#Default:
# ftp_user Squid@

# TAG: ftp_list_width
#   Sets the width of ftp listings. This should be set to fit in
#   the width of a standard browser. Setting this too small
#   can cut off long filenames when browsing ftp sites.
#
#Default:
# ftp_list_width 32

# TAG: ftp_passive
#   If your firewall does not allow Squid to use passive
#   connections, then turn off this option.
#
#Default:
# ftp_passive on

# TAG: ftp_sanitycheck
#   For security and data integrity reasons Squid by default performs
#   sanity checks of the addresses of FTP data connections ensure the
#   data connection is to the requested server. If you need to allow
#   FTP connections to servers using another IP address for the data
#   connection then turn this off.
#
#Default:
# ftp_sanitycheck on

# TAG: cache_dns_program
# Note: This option is only available if Squid is rebuilt with the
#       --disable-internal-dns option
#
#       Specify the location of the executable for dnslookup process.
#
#Default:
# cache_dns_program /usr/lib/squid/

# TAG: dns_children
# Note: This option is only available if Squid is rebuilt with the
#       --disable-internal-dns option
#
#       The number of processes spawn to service DNS name lookups.
#       For heavily loaded caches on large servers, you should
#       probably increase this value to at least 10. The maximum
#       is 32. The default is 5.
#
#       You must have at least one dnsserver process.
#
#Default:
# dns_children 5

# TAG: dns_retransmit_interval
#   Initial retransmit interval for DNS queries. The interval is
#   doubled each time all configured DNS servers have been tried.
```

```
#
#
#Default:
# dns_retransmit_interval 5 seconds

# TAG: dns_timeout
#   DNS Query timeout. If no response is received to a DNS query
#   within this time then all DNS servers for the queried domain
#   is assumed to be unavailable.
#
#Default:
# dns_timeout 5 minutes

# TAG: dns_defnames    on|off
# Note: This option is only available if Squid is rebuilt with the
#       --disable-internal-dns option
#
#       Normally the 'dnsserver' disables the RES_DEFNAMES resolver
#       option (see res_init(3)). This prevents caches in a hierarchy
#       from interpreting single-component hostnames locally. To allow
#       dnsserver to handle single-component names, enable this
#       option.
#
#Default:
# dns_defnames off

# TAG: dns_nameservers
#   Use this if you want to specify a list of DNS name servers
#   (IP addresses) to use instead of those given in your
#   /etc/resolv.conf file.
#   On Windows platforms, if no value is specified here or in
#   the /etc/resolv.conf file, the list of DNS name servers are
#   taken from the Windows registry, both static and dynamic DHCP
#   configurations are supported.
#
#   Example: dns_nameservers 10.0.0.1 192.172.0.4
#
#Default:
# none

# TAG: hosts_file
#   Location of the host-local IP name-address associations
#   database. Most Operating Systems have such a file: under
#   Un*X it's by default in /etc/hosts MS-Windows NT/2000 places
#   that in %SystemRoot%(by default
#   c:\winnt)\system32\drivers\etc\hosts, while Windows 9x/ME
#   places that in %windir%(usually c:\windows)\hosts
#
#   The file contains newline-separated definitions, in the
#   form ip_address_in_dotted_form name [name ...] names are
#   whitespace-separated. lines beginning with an hash (#)
#   character are comments.
#
#   The file is checked at startup and upon configuration. If
#   set to 'none', it won't be checked. If append_domain is
#   used, that domain will be added to domain-local (i.e. not
#   containing any dot character) host definitions.
#
#Default:
# hosts_file /etc/hosts

# TAG: diskd_program
#   Specify the location of the diskd executable.
#   Note that this is only useful if you have compiled in
#   diskd as one of the store io modules.
#
```

```
#Default:
# diskd_program /usr/lib/squid/diskd

# TAG: unlinkd_program
#     Specify the location of the executable for file deletion process.
#
#Default:
# unlinkd_program /usr/lib/squid/unlinkd

# TAG: pinger_program
# Note: This option is only available if Squid is rebuilt with the
#       --enable-icmp option
#
#     Specify the location of the executable for the pinger process.
#
#Default:
# pinger_program /usr/lib/squid/

# TAG: redirect_program
#     Specify the location of the executable for the URL redirector.
#     Since they can perform almost any function there isn't one
#     included.
#     See the FAQ (section 15) for information on how to write one.
#     By default, a redirector is not used.
#
#Default:
# none

# TAG: redirect_children
#     The number of redirector processes to spawn. If you start
#     too few Squid will have to wait for them to process a backlog of
#     URLs, slowing it down. If you start too many they will use RAM
#     and other system resources.
#
#Default:
# redirect_children 5

# TAG: redirect_rewrites_host_header
#     By default Squid rewrites any Host: header in redirected
#     requests. If you are running an accelerator then this may
#     not be a wanted effect of a redirector.
#
#Default:
# redirect_rewrites_host_header on

# TAG: redirector_access
#     If defined, this access list specifies which requests are
#     sent to the redirector processes. By default all requests
#     are sent.
#
#Default:
# none

# TAG: auth_param
#     This is used to pass parameters to the various authentication
#     schemes.
#     format: auth_param scheme parameter [setting]
#
#     auth_param basic program /usr/bin/ncsa_auth /usr/etc/passwd
#     would tell the basic authentication scheme it's program
parameter.
#
#     The order that authentication prompts are presented to the
client_agent
#     is dependant on the order the scheme first appears in config
file.
```

```
# IE has a bug (it's not rfc 2617 compliant) in that it will use
the basic
# scheme if basic is the first entry presented, even if more secure
schemes
# are presented. For now use the order in the file below. If other
browsers
# have difficulties (don't recognise the schemes offered even if
you are using
# basic) then either put basic first, or disable the other schemes
(by commenting
# out their program entry).
#
# Once an authentication scheme is fully configured, it can only be
shutdown
# by shutting squid down and restarting. Changes can be made on the
fly and
# activated with a reconfigure. I.E. You can change to a different
helper,
# but not unconfigure the helper completely.
#
# === Parameters for the basic scheme follow. ===
#
# "program" cmdline
# Specify the command for the external authenticator. Such a
# program reads a line containing "username password" and replies
# "OK" or "ERR" in an endless loop. If you use an authenticator,
# make sure you have 1 acl of type proxy_auth. By default, the
# authenticate_program is not used.
#
# If you want to use the traditional proxy authentication,
# jump over to the ../auth_modules/NCSA directory and
# type:
#
#     % make
#     % make install
#
# Then, set this line to something like
#
# auth_param basic program /usr/bin/ncsa_auth /usr/etc/passwd
#
# "children" numberofchildren
# The number of authenticator processes to spawn (no default).
# If you start too few Squid will have to wait for them to
# process a backlog of usercode/password verifications, slowing
# it down. When password verifications are done via a (slow)
# network you are likely to need lots of authenticator
# processes.
#
# auth_param basic children 5
#
# "realm" realmstring
# Specifies the realm name which is to be reported to the
# client for the basic proxy authentication scheme (part of
# the text the user will see when prompted their username and
# password). There is no default.
#
# auth_param basic realm Squid proxy-caching web server
#
# "credentialsttl" timetolive
# Specifies how long squid assumes an externally validated
# username:password pair is valid for - in other words how
# often the helper program is called for that user. Set this
# low to force revalidation with short lived passwords. Note
# that setting this high does not impact your susceptibility
# to replay attacks unless you are using an one-time password
# system (such as SecureID). If you are using such a system,
# you will be vulnerable to replay attacks unless you also
# use the max_user_ip ACL in an http_access rule.
#
```

```

# === Parameters for the digest scheme follow ===
#
# "program" cmdline
# Specify the command for the external authenticator. Such
# a program reads a line containing "username":"realm" and
# replies with the appropriate H(A1) value base64 encoded.
# See rfc 2616 for the definition of H(A1). If you use an
# authenticator, make sure you have 1 acl of type proxy_auth.
# By default, authentication is not used.
#
# If you want to use build an authenticator,
# jump over to the ../digest_auth_modules directory and choose the
# authenticator to use. It it's directory type
#     % make
#     % make install
#
# Then, set this line to something like
#
# auth_param digest program /usr/bin/digest_auth_pw
/usr/etc/digpass
#
#
# "children" numberofchildren
# The number of authenticator processes to spawn (no default).
# If you start too few Squid will have to wait for them to
# process a backlog of H(A1) calculations, slowing it down.
# When the H(A1) calculations are done via a (slow) network
# you are likely to need lots of authenticator processes.
# auth_param digest children 5
#
# "realm" realmstring
# Specifies the realm name which is to be reported to the
# client for the digest proxy authentication scheme (part of
# the text the user will see when prompted their username and
# password). There is no default.
# auth_param digest realm Squid proxy-caching web server
#
# "nonce_garbage_interval" timeinterval
# Specifies the interval that nonces that have been issued
# to client_agent's are checked for validity.
#
# "nonce_max_duration" timeinterval
# Specifies the maximum length of time a given nonce will be
# valid for.
#
# "nonce_max_count" number
# Specifies the maximum number of times a given nonce can be
# used.
#
# "nonce_strictness" on|off
# Determines if squid requires increment-by-1 behaviour for
# nonce counts (on - the default), or strictly incrementing
# (off - for use when useragents generate nonce counts that
# occasionally miss 1 (ie, 1,2,4,6)).
#
# === NTLM scheme options follow ===
#
# "program" cmdline
# Specify the command for the external ntlm authenticator.
# Such a program reads a line containing the uuencoded NEGOTIATE
# and replies with the ntlm CHALLENGE, then waits for the
# response and answers with "OK" or "ERR" in an endless loop.
# If you use an ntlm authenticator, make sure you have 1 acl
# of type proxy_auth. By default, the ntlm authenticator_program
# is not used.
#
#

```

```

#   auth_param ntlm program /usr/bin/ntlm_auth
#
#   "children" numberofchildren
#   The number of authenticator processes to spawn (no default).
#   If you start too few Squid will have to wait for them to
#   process a backlog of credential verifications, slowing it
#   down. When credential verifications are done via a (slow)
#   network you are likely to need lots of authenticator
#   processes.
#   auth_param ntlm children 5
#
#   "max_challenge_reuses" number
#   The maximum number of times a challenge given by a ntlm
#   authentication helper can be reused. Increasing this number
#   increases your exposure to replay attacks on your network.
#   0 means use the challenge only once. (disable challenge
#   caching) See max_ntlm_challenge_lifetime for more information.
#   auth_param ntlm max_challenge_reuses 0
#
#   "max_challenge_lifetime" timespan
#   The maximum time period that a ntlm challenge is reused
#   over. The actual period will be the minimum of this time
#   AND the number of reused challenges.
#   auth_param ntlm max_challenge_lifetime 2 minutes
#
#Recommended minimum configuration:
#auth_param digest program <uncomment and complete this line>
#auth_param digest children 5
#auth_param digest realm Squid proxy-caching web server
#auth_param digest nonce_garbage_interval 5 minutes
#auth_param digest nonce_max_duration 30 minutes
#auth_param digest nonce_max_count 50
#auth_param ntlm program <uncomment and complete this line to activate>
#auth_param ntlm children 5
#auth_param ntlm max_challenge_reuses 0
#auth_param ntlm max_challenge_lifetime 2 minutes
auth_param basic program /usr/lib/squid/nscache_auth
/etc/squid/squid_passwd
auth_param basic children 300
auth_param basic realm BITS GOA PROXY
auth_param basic credentialsttl 1 minute

# TAG: authenticate_cache_garbage_interval
#   The time period between garbage collection across the
#   username cache. This is a tradeoff between memory utilisation
#   (long intervals - say 2 days) and CPU (short intervals -
#   say 1 minute). Only change if you have good reason to.
#
#Default:
# authenticate_cache_garbage_interval 1 hour

# TAG: authenticate_ttl
#   The time a user & their credentials stay in the logged in
#   user cache since their last request. When the garbage
#   interval passes, all user credentials that have passed their
#   TTL are removed from memory.
#
#Default:
# authenticate_ttl 1 hour

# TAG: authenticate_ip_ttl
#   If you use proxy authentication and the 'max_user_ip' ACL,
#   this directive controls how long Squid remembers the IP

```

```

# addresses associated with each user. Use a small value
# (e.g., 60 seconds) if your users might change addresses
# quickly, as is the case with dialups. You might be safe
# using a larger value (e.g., 2 hours) in a corporate LAN
# environment with relatively static address assignments.
#
#Default:
# authenticate_ip_ttl 0 seconds

# TAG: external_acl_type
# This option defines external acl classes using a helper program
# to look up the status
#
# external_acl_type name [options] FORMAT.. /path/to/helper
[helper arguments..]
#
# Options:
#
# ttl=n          TTL in seconds for cached results (defaults to
3600
#                for 1 hour)
# negative_ttl=n
#                TTL for cached negative lookups (default same
#                as ttl)
# concurrency=n  Concurrency level / number of processes spawn
#                to service external acl lookups of this type.
# cache=n        result cache size, 0 is unbounded (default)
#
# FORMAT specifications
#
# %LOGIN         Authenticated user login name
# %IDENT         Ident user name
# %SRC           Client IP
# %DST          Requested host
# %PROTO        Requested protocol
# %PORT         Requested port
# %METHOD        Request method
# %{Header}     HTTP request header
# %{Hdr:member} HTTP request header list member
# %{Hdr:;member}
#                HTTP request header list member using ; as
#                list separator. ; can be any non-alphanumeric
#                character.
#
# In addition, any string specified in the referencing acl will
# also be included in the helper request line, after the specified
# formats (see the "acl external" directive)
#
# The helper receives lines per the above format specification,
# and returns lines starting with OK or ERR indicating the validity
# of the request and optionally followed by additional keywords
with
# more details.
#
# General result syntax:
#
# OK/ERR keyword=value ...
#
# Defined keywords:
#
# user=          The users name (login)
# error=         Error description (only defined for ERR results)
#
# Keyword values need to be enclosed in quotes if they may contain
# whitespace, or the whitespace escaped using \. Any quotes or \
# characters within the keyword value must be \ escaped.

```

```

#
#Default:
# none

# OPTIONS FOR TUNING THE CACHE
# -----
-----

# TAG: wais_relay_host
# TAG: wais_relay_port
#     Relay WAIS request to host (1st arg) at port (2 arg).
#
#Default:
# wais_relay_port 0

# TAG: request_header_max_size      (KB)
#     This specifies the maximum size for HTTP headers in a request.
#     Request headers are usually relatively small (about 512 bytes).
#     Placing a limit on the request header size will catch certain
#     bugs (for example with persistent connections) and possibly
#     buffer-overflow or denial-of-service attacks.
#
#Default:
# request_header_max_size 10 KB

# TAG: request_body_max_size (KB)
#     This specifies the maximum size for an HTTP request body.
#     In other words, the maximum size of a PUT/POST request.
#     A user who attempts to send a request with a body larger
#     than this limit receives an "Invalid Request" error message.
#     If you set this parameter to a zero (the default), there will
#     be no limit imposed.
#
#Default:
# request_body_max_size 0 KB

# TAG: refresh_pattern
#     usage: refresh_pattern [-i] regex min percent max [options]
#
#     By default, regular expressions are CASE-SENSITIVE.  To make
#     them case-insensitive, use the -i option.
#
#     'Min' is the time (in minutes) an object without an explicit
#     expiry time should be considered fresh. The recommended
#     value is 0, any higher values may cause dynamic applications
#     to be erroneously cached unless the application designer
#     has taken the appropriate actions.
#
#     'Percent' is a percentage of the objects age (time since last
#     modification age) an object without explicit expiry time
#     will be considered fresh.
#
#     'Max' is an upper limit on how long objects without an explicit
#     expiry time will be considered fresh.
#
#     options: override-expire
#               override-lastmod
#               reload-into-ims
#               ignore-reload
#
#     override-expire enforces min age even if the server
#     sent a Expires: header. Doing this VIOLATES the HTTP
#     standard.  Enabling this feature could make you liable
#     for problems which it causes.
#
#

```

```

#         override-lastmod enforces min age even on objects
#         that was modified recently.
#
#         reload-into-ims changes client no-cache or ``reload``
#         to If-Modified-Since requests. Doing this VIOLATES the
#         HTTP standard. Enabling this feature could make you
#         liable for problems which it causes.
#
#         ignore-reload ignores a client no-cache or ``reload``
#         header. Doing this VIOLATES the HTTP standard. Enabling
#         this feature could make you liable for problems which
#         it causes.
#
# Basically a cached object is:
#
#         FRESH if expires < now, else STALE
#         STALE if age > max
#         FRESH if lm-factor < percent, else STALE
#         FRESH if age < min
#         else STALE
#
# The refresh_pattern lines are checked in the order listed here.
# The first entry which matches is used. If none of the entries
# match, then the default will be used.
#
# Note, you must uncomment all the default lines if you want
# to change one. The default setting is only active if none is
# used.
#
#Suggested default:
refresh_pattern ^ftp:          1440 20% 10080
refresh_pattern ^gopher:      1440 0% 1440
refresh_pattern .              0 20% 4320

# TAG: quick_abort_min (KB)
# TAG: quick_abort_max (KB)
# TAG: quick_abort_pct (percent)
# The cache by default continues downloading aborted requests
# which are almost completed (less than 16 KB remaining). This
# may be undesirable on slow (e.g. SLIP) links and/or very busy
# caches. Impatient users may tie up file descriptors and
# bandwidth by repeatedly requesting and immediately aborting
# downloads.
#
# When the user aborts a request, Squid will check the
# quick_abort values to the amount of data transfered until
# then.
#
# If the transfer has less than 'quick_abort_min' KB remaining,
# it will finish the retrieval.
#
# If the transfer has more than 'quick_abort_max' KB remaining,
# it will abort the retrieval.
#
# If more than 'quick_abort_pct' of the transfer has completed,
# it will finish the retrieval.
#
# If you do not want any retrieval to continue after the client
# has aborted, set both 'quick_abort_min' and 'quick_abort_max'
# to '0 KB'.
#
# If you want retrievals to always continue if they are being
# cached then set 'quick_abort_min' to '-1 KB'.
#
#Default:
# quick_abort_min 16 KB

```

```
# quick_abort_max 16 KB
# quick_abort_pct 95

# TAG: negative_ttl      time-units
#   Time-to-Live (TTL) for failed requests.  Certain types of
#   failures (such as "connection refused" and "404 Not Found") are
#   negatively-cached for a configurable amount of time.  The
#   default is 5 minutes.  Note that this is different from
#   negative caching of DNS lookups.
#
#Default:
# negative_ttl 5 minutes

# TAG: positive_dns_ttl  time-units
#   Time-to-Live (TTL) for positive caching of successful DNS
lookups.
#   Default is 6 hours (360 minutes).  If you want to minimize the
#   use of Squid's ipcache, set this to 1, not 0.
#
#Default:
# positive_dns_ttl 6 hours

# TAG: negative_dns_ttl  time-units
#   Time-to-Live (TTL) for negative caching of failed DNS lookups.
#
#Default:
# negative_dns_ttl 5 minutes

# TAG: range_offset_limit (bytes)
#   Sets a upper limit on how far into the the file a Range request
#   may be to cause Squid to prefetch the whole file.  If beyond this
#   limit then Squid forwards the Range request as it is and the
result
#   is NOT cached.
#
#   This is to stop a far ahead range request (lets say start at
17MB)
#   from making Squid fetch the whole object up to that point before
#   sending anything to the client.
#
#   A value of -1 causes Squid to always fetch the object from the
#   beginning so that it may cache the result. (2.0 style)
#
#   A value of 0 causes Squid to never fetch more than the
#   client requested. (default)
#
#Default:
# range_offset_limit 0 KB

# TIMEOUTS
# -----
-----

# TAG: connect_timeout time-units
#   Some systems (notably Linux) can not be relied upon to properly
#   time out connect(2) requests.  Therefore the Squid process
#   enforces its own timeout on server connections.  This parameter
#   specifies how long to wait for the connect to complete.  The
#   default is two minutes (120 seconds).
#
#Default:
# connect_timeout 2 minutes

# TAG: peer_connect_timeout time-units
#   This parameter specifies how long to wait for a pending TCP
```

```
# connection to a peer cache. The default is 30 seconds. You
# may also set different timeout values for individual neighbors
# with the 'connect-timeout' option on a 'cache_peer' line.
#
#Default:
# peer_connect_timeout 30 seconds

# TAG: read_timeout time-units
# The read_timeout is applied on server-side connections. After
# each successful read(), the timeout will be extended by this
# amount. If no data is read again after this amount of time,
# the request is aborted and logged with ERR_READ_TIMEOUT. The
# default is 15 minutes.
#
#Default:
# read_timeout 15 minutes

# TAG: request_timeout
# How long to wait for an HTTP request after initial
# connection establishment.
#
#Default:
# request_timeout 5 minutes

# TAG: persistent_request_timeout
# How long to wait for the next HTTP request on a persistent
# connection after the previous request completes.
#
#Default:
# persistent_request_timeout 1 minute

# TAG: client_lifetime time-units
# The maximum amount of time that a client (browser) is allowed to
# remain connected to the cache process. This protects the Cache
# from having a lot of sockets (and hence file descriptors) tied up
# in a CLOSE_WAIT state from remote clients that go away without
# properly shutting down (either because of a network failure or
# because of a poor client implementation). The default is one
# day, 1440 minutes.
#
# NOTE: The default value is intended to be much larger than any
# client would ever need to be connected to your cache. You
# should probably change client_lifetime only as a last resort.
# If you seem to have many client connections tying up
# filedescriptors, we recommend first tuning the read_timeout,
# request_timeout, persistent_request_timeout and quick_abort
values.
#
#Default:
# client_lifetime 1 day

# TAG: half_closed_clients
# Some clients may shutdown the sending side of their TCP
# connections, while leaving their receiving sides open.
# Sometimes,
# Squid can not tell the difference between a half-closed and a
# fully-closed TCP connection. By default, half-closed client
# connections are kept open until a read(2) or write(2) on the
# socket returns an error. Change this option to 'off' and Squid
# will immediately close client connections when read(2) returns
# "no more data to read."
#
#Default:
# half_closed_clients on

# TAG: pconn_timeout
```

```

#     Timeout for idle persistent connections to servers and other
#     proxies.
#
#Default:
# pconn_timeout 120 seconds

# TAG: ident_timeout
#     Maximum time to wait for IDENT lookups to complete.
#
#     If this is too high, and you enabled IDENT lookups from untrusted
#     users, then you might be susceptible to denial-of-service by
having
#     many ident requests going at once.
#
#Default:
# ident_timeout 10 seconds

# TAG: shutdown_lifetime      time-units
#     When SIGTERM or SIGHUP is received, the cache is put into
#     "shutdown pending" mode until all active sockets are closed.
#     This value is the lifetime to set for all open descriptors
#     during shutdown mode.  Any active clients after this many
#     seconds will receive a 'timeout' message.
#
#Default:
# shutdown_lifetime 30 seconds

# ACCESS CONTROLS
# -----
# -----

# TAG: acl
#     Defining an Access List
#
#     acl aclname acltype string1 ...
#     acl aclname acltype "file" ...
#
#     when using "file", the file should contain one item per line
#
#     acltype is one of the types described below
#
#     By default, regular expressions are CASE-SENSITIVE.  To make
#     them case-insensitive, use the -i option.
#
#     acl aclname src          ip-address/netmask ... (clients IP address)
#     acl aclname src          addr1-addr2/netmask ... (range of addresses)
#     acl aclname dst          ip-address/netmask ... (URL host's IP
address)
#     acl aclname myip         ip-address/netmask ... (local socket IP
address)
#
#     acl aclname srcdomain    .foo.com ...      # reverse lookup, client
IP
#     acl aclname dstdomain    .foo.com ...      # Destination server from
URL
#     acl aclname srcdom_regex [-i] xxx ...      # regex matching client
name
#     acl aclname dstdom_regex [-i] xxx ...      # regex matching server
#     # For dstdomain and dstdom_regex a reverse lookup is tried if
a IP
#     # based URL is used. The name "none" is used if the reverse
lookup
#     # fails.
#
#     acl aclname time         [day-abbrevs] [h1:m1-h2:m2]

```

```

#       day-abbrevs:
#       S - Sunday
#       M - Monday
#       T - Tuesday
#       W - Wednesday
#       H - Thursday
#       F - Friday
#       A - Saturday
#       h1:m1 must be less than h2:m2
#       acl aclname url_regex [-i] ^http:// ... # regex matching on
whole URL
#       acl aclname urlpath_regex [-i] \.gif$ ... # regex matching on URL
path
#       acl aclname port      80 70 21 ...
#       acl aclname port      0-1024 ... # ranges allowed
#       acl aclname myport    3128 ... # (local socket TCP port)
#       acl aclname proto     HTTP FTP ...
#       acl aclname method    GET POST ...
#       acl aclname browser   [-i] regexp ...
#       # pattern match on User-Agent header
#       acl aclname referer_regex [-i] regexp ...
#       # pattern match on Referer header
#       # Referer is highly unreliable, so use with care
#       acl aclname ident     username ...
#       acl aclname ident_regex [-i] pattern ...
#       # string match on ident output.
#       # use REQUIRED to accept any non-null ident.
#       acl aclname src_as    number ...
#       acl aclname dst_as    number ...
#       # Except for access control, AS numbers can be used for
#       # routing of requests to specific caches. Here's an
#       # example for routing all requests for AS#1241 and only
#       # those to mycache.mydomain.net:
#       acl asexample dst_as 1241
#       # cache_peer_access mycache.mydomain.net allow asexample
#       # cache_peer_access mycache.mydomain.net deny all
#
#       acl aclname proxy_auth username ...
#       acl aclname proxy_auth_regex [-i] pattern ...
#       # list of valid usernames
#       # use REQUIRED to accept any valid username.
#       #
#       # NOTE: when a Proxy-Authentication header is sent but it is
not
#       # needed during ACL checking the username is NOT logged
#       # in access.log.
#       #
#       # NOTE: proxy_auth requires a EXTERNAL authentication program
#       # to check username/password combinations (see
#       # authenticate_program).
#       #
#       # WARNING: proxy_auth can't be used in a transparent proxy. It
#       # collides with any authentication done by origin servers. It
may
#       # seem like it works at first, but it doesn't.
#
#       acl aclname snmp_community string ...
#       # A community string to limit access to your SNMP Agent
#       # Example:
#       #
#       #       acl snmppublic snmp_community public
#
#       acl aclname maxconn number
#       # This will be matched when the client's IP address has
#       # more than <number> HTTP connections established.
#

```

```

#     acl aclname max_user_ip [-s] number
#     # This will be matched when the user attempts to log in from
more
#     # than <number> different ip addresses. The authenticate_ip_ttl
#     # parameter controls the timeout on the ip entries.
#     # If -s is specified then the limit is strict, denying browsing
#     # from any further IP addresses until the ttl has expired.
Without
#     # -s Squid will just annoy the user by "randomly" denying
requests.
#     # (the counter is then reset each time the limit is reached and
a
#     # request is denied)
#     # NOTE: in acceleration mode or where there is mesh of child
proxies,
#     # clients may appear to come from multiple addresses if they
are
#     # going through proxy farms, so a limit of 1 may cause user
problems.
#
#     acl aclname req_mime_type mime-type1 ...
#     # regex match againsts the mime type of the request generated
#     # by the client. Can be used to detect file upload or some
#     # types HTTP tunnelling requests.
#     # NOTE: This does NOT match the reply. You cannot use this
#     # to match the returned file type.
#
#     acl aclname rep_mime_type mime-type1 ...
#     # regex match against the mime type of the reply recieved by
#     # squid. Can be used to detect file download or some
#     # types HTTP tunnelling requests.
#     # NOTE: This has no effect in http_access rules. It only has
#     # effect in rules that affect the reply data stream such as
#     # http_reply_access.
#
#     acl acl_name external class_name [arguments...]
#     # external ACL lookup via a helper class defined by the
#     # external_acl_type directive.
#
#Examples:
#acl myexample dst_as 1241
#acl password proxy_auth REQUIRED
#acl fileupload req_mime_type -i ^multipart/form-data$
#acl javascript rep_mime_type -i ^application/x-javascript$
#
#Recommended minimum configuration:
acl ncsa_users proxy_auth REQUIRED
acl all src 10.0.0.0/255.0.0.0
acl labs src 10.1.0.0/255.255.0.0 10.2.0.0/255.255.0.0
acl hostels src 10.3.0.0/255.255.0.0 10.4.0.0/255.255.0.0
acl manager proto cache_object
acl localhost src 127.0.0.1/255.255.255.255
acl to_localhost dst 127.0.0.0/8
acl SSL_ports port 443 563
acl Safe_ports port 80          # http
acl Safe_ports port 21          # ftp
acl Safe_ports port 5221        #Gtalk installed on demand
acl Safe_ports port 443 563     # https, snews
acl Safe_ports port 70          # gopher
acl Safe_ports port 210         # wais
acl Safe_ports port 5050-5070   # yim
#acl Safe_ports port 1025-65535  # unregistered ports
acl Safe_ports port 280         # http-mgmt
acl Safe_ports port 488         # gss-http
acl Safe_ports port 403         # orkut
acl Safe_ports port 591         # filemaker

```

```

acl Safe_ports port 777          # multiling http
acl CONNECT method CONNECT
acl day_time time 8:30-17:30
acl night_time time 17:30-24:00 0:00-8:30
acl other_time time 17:30-21:00
acl banned src 10.1.7.147 10.1.7.81 10.3.4.70 10.4.7.179 10.3.4.10
10.4.7.32 10.4.7.74 10.4.7.36 10.3.1.64 10.3.2.143
#acl stop_url url_regex www.12345.com
# TAG: http_access
#     Allowing or Denying access based on defined access lists
#
#     Access to the HTTP port:
#     http_access allow|deny [!]aclname ...
#
#     NOTE on default values:
#
#     If there are no "access" lines present, the default is to deny
#     the request.
#
#     If none of the "access" lines cause a match, the default is the
#     opposite of the last line in the list.  If the last line was
#     deny, then the default is allow.  Conversely, if the last line
#     is allow, the default will be deny.  For these reasons, it is a
#     good idea to have an "deny all" or "allow all" entry at the end
#     of your access lists to avoid potential confusion.
#
#Default:
# http_access deny all
#
#Recommended minimum configuration:
#
# Only allow cachemgr access from localhost
http_access allow ncsa_users
#http_access allow ncsa_users
#http_access allow all
http_access allow labs day_time other_time
http_access allow hostels night_time
http_access deny banned
#http_access allow ncsa_users
#http_access deny all
#http_access allow manager localhost
#http_access allow manager
# Deny requests to unknown ports
http_access deny !Safe_ports
# Deny CONNECT to other than SSL ports
http_access deny CONNECT !SSL_ports
#http_access deny stop_url
#
# We strongly recommend to uncomment the following to protect innocent
# web applications running on the proxy server who think that the only
# one who can access services on "localhost" is a local user
#http_access deny to_localhost
#
# INSERT YOUR OWN RULE(S) HERE TO ALLOW ACCESS FROM YOUR CLIENTS
#
# Exampe rule allowing access from your local networks. Adapt
# to list your (internal) IP networks from where browsing should
# be allowed
#acl our_networks src 192.168.1.0/24 192.168.2.0/24
#acl our_networks src 10.1.1.0/255.255.255.0
#http_access allow our_networks
#
# And finally deny all other access to this proxy
#http_access allow localhost
#http_access deny all

```

```

# TAG: http_reply_access
#     Allow replies to client requests. This is complementary to
http_access.
#
#     http_reply_access allow|deny [!] aclname ...
#
#     NOTE: if there are no access lines present, the default is to
allow
#     all replies
#
#     If none of the access lines cause a match, then the opposite
of the
#     last line will apply. Thus it is good practice to end the
rules
#     with an "allow all" or "deny all" entry.
#
#Default:
# http_reply_access allow all
#
#Recommended minimum configuration:
#
# Insert your own rules here.
#
#
# and finally allow by default
http_reply_access allow all

# TAG: icp_access
#     Allowing or Denying access to the ICP port based on defined
#     access lists
#
#     icp_access allow|deny [!]aclname ...
#
#     See http_access for details
#
#Default:
# icp_access deny all
#
#Allow ICP queries from everyone
icp_access allow all

# TAG: miss_access
#     Use to force your neighbors to use you as a sibling instead of
#     a parent. For example:
#
#         acl localclients src 172.16.0.0/16
#         miss_access allow localclients
#         miss_access deny !localclients
#
#     This means that only your local clients are allowed to fetch
#     MISSES and all other clients can only fetch HITS.
#
#     By default, allow all clients who passed the http_access rules
#     to fetch MISSES from us.
#
#Default setting:
# miss_access allow all

# TAG: cache_peer_access
#     Similar to 'cache_peer_domain' but provides more flexibility by
#     using ACL elements.
#
#     cache_peer_access cache-host allow|deny [!]aclname ...
#
#     The syntax is identical to 'http_access' and the other lists of
#     ACL elements. See the comments for 'http_access' below, or

```

```

# the Squid FAQ (http://www.squid-cache.org/FAQ/FAQ-10.html).
#
#Default:
# none

# TAG: ident_lookup_access
# A list of ACL elements which, if matched, cause an ident
# (RFC 931) lookup to be performed for this request. For
# example, you might choose to always perform ident lookups
# for your main multi-user Unix boxes, but not for your Macs
# and PCs. By default, ident lookups are not performed for
# any requests.
#
# To enable ident lookups for specific client addresses, you
# can follow this example:
#
# acl ident_aware_hosts src 198.168.1.0/255.255.255.0
# ident_lookup_access allow ident_aware_hosts
# ident_lookup_access deny all
#
# Only src type ACL checks are fully supported. A src_domain
# ACL might work at times, but it will not always provide
# the correct result.
#
#Default:
# ident_lookup_access deny all

# TAG: tcp_outgoing_tos
# Allows you to select a TOS/Diffserv value to mark outgoing
# connections with, based on the username or source address
# making the request.
#
# tcp_outgoing_tos ds-field [!]aclname ...
#
# Example where normal_service_net uses the TOS value 0x00
# and normal_service_net uses 0x20
#
# acl normal_service_net src 10.0.0.0/255.255.255.0
# acl good_service_net src 10.0.1.0/255.255.255.0
# tcp_outgoing_tos 0x00 normal_service_net 0x00
# tcp_outgoing_tos 0x20 good_service_net
#
# TOS/DSCP values really only have local significance - so you
should
# know what you're specifying. For more, see RFC 2474
#
# The TOS/DSCP byte must be exactly that - a byte, value 0 - 255,
or
# "default" to use whatever default your host has.
#
# Processing proceeds in the order specified, and stops at first
fully
# matching line.
#
#Default:
# none

# TAG: tcp_outgoing_address
# Allows you to map requests to different outgoing IP addresses
# based on the username or sourceaddress of the user making
# the request.
#
# tcp_outgoing_address ipaddr [!]aclname] ...
#
# Example where requests from 10.0.0.0/24 will be forwarded
# with source address 10.1.0.1, 10.0.2.0/24 forwarded with

```

```

# source address 10.1.0.2 and the rest will be forwarded with
# source address 10.1.0.3.
#
# acl normal_service_net src 10.0.0.0/255.255.255.0
# acl good_service_net src 10.0.1.0/255.255.255.0
# tcp_outgoing_address 10.0.0.1 normal_service_net
# tcp_outgoing_address 10.0.0.2 good_service_net
# tcp_outgoing_address 10.0.0.3
#
# Processing proceeds in the order specified, and stops at first
fully
# matching line.
#
#Default:
# none

# TAG: reply_body_max_size bytes allow|deny acl acl...
# This option specifies the maximum size of a reply body. It
# can be used to prevent users from downloading very large files,
# such as MP3's and movies. When the reply headers are recieved,
# the reply_body_max_size lines are processed, and the first line
with
# a result of "allow" is used as the maximum body size for this
reply.
# This size is then checked twice. First when we get the reply
headers,
# we check the content-length value. If the content length value
exists
# and is larger than the allowed size, the request is denied and
the
# user receives an error message that says "the request or reply
# is too large." If there is no content-length, and the reply
# size exceeds this limit, the client's connection is just closed
# and they will receive a partial reply.
#
# WARNING: downstream caches probably can not detect a partial
reply
# if there is no content-length header, so they will cache
# partial responses and give them out as hits. You should NOT
# use this option if you have downstream caches.
#
# WARNING: A maximum size larger than the size of squid's error
messages
# will cause an infinite loop and crash squid. Ensure that the
smallest
# non-zero value you use is greater that the maximum header size
plus
# the size of your largest error page.
#
# If you set this parameter to zero (the default), there will be
# no limit imposed.
#
#Default:
# reply_body_max_size 0 allow all

# ADMINISTRATIVE PARAMETERS
# -----
# -----

# TAG: cache_mgr
# Email-address of local cache manager who will receive
# mail if the cache dies. The default is "webmaster."
#cache_mgr root
#
#Default:

```

```

# cache_mgr root
cache_mgr root
# TAG: cache_effective_user
# TAG: cache_effective_group
#
# If the cache is run as root, it will change its effective/real
# UID/GID to the UID/GID specified below. The default is to
# change to UID to squid and GID to the default group of squid.
#
# If Squid is not started as root, the default is to keep the
# current UID/GID, and only the GID can be changed to any of
# the groups the user starting Squid is member of. Note that if
# Squid is not started as root then you cannot set http_port to
# a value lower than 1024.
#
#cache_effective_user squid
#cache_effective_group squid
#
#Default:
# cache_effective_user squid
# cache_effective_group squid

# TAG: visible_hostname
# If you want to present a special hostname in error messages, etc,
# then define this. Otherwise, the return value of gethostname()
# will be used. If you have multiple caches in a cluster and
# get errors about IP-forwarding you must set them to have
individual
# names with this setting.
#
#Default:
# none
visible_hostname BITSGOA

# TAG: unique_hostname
# If you want to have multiple machines with the same
# 'visible_hostname' then you must give each machine a different
# 'unique_hostname' so that forwarding loops can be detected.
#
#Default:
# none

# TAG: hostname_aliases
# A list of other DNS names that your cache has.
#
#Default:
# none

# OPTIONS FOR THE CACHE REGISTRATION SERVICE
# -----
#
# This section contains parameters for the (optional) cache
# announcement service. This service is provided to help
# cache administrators locate one another in order to join or
# create cache hierarchies.
#
# An 'announcement' message is sent (via UDP) to the registration
# service by Squid. By default, the announcement message is NOT
# SENT unless you enable it with 'announce_period' below.
#
# The announcement message includes your hostname, plus the
# following information from this configuration file:
#
# http_port

```

```

#         icp_port
#         cache_mgr
#
# All current information is processed regularly and made
# available on the Web at http://www.ircache.net/Cache/Tracker/.
#
# TAG: announce_period
# This is how frequently to send cache announcements. The
# default is `0' which disables sending the announcement
# messages.
#
# To enable announcing your cache, just uncomment the line
# below.
#
#Default:
# announce_period 0
#
#To enable announcing your cache, just uncomment the line below.
#announce_period 1 day
#
# TAG: announce_host
# TAG: announce_file
# TAG: announce_port
# announce_host and announce_port set the hostname and port
# number where the registration message will be sent.
#
# Hostname will default to 'tracker.ircache.net' and port will
# default default to 3131. If the 'filename' argument is given,
# the contents of that file will be included in the announce
# message.
#
#Default:
# announce_host tracker.ircache.net
# announce_port 3131
#
# HTTPD-ACCELERATOR OPTIONS
# -----
# -----
#
# TAG: httpd_accel_host
# TAG: httpd_accel_port
# If you want to run Squid as an httpd accelerator, define the
# host name and port number where the real HTTP server is.
#
# If you want IP based virtual host support then specify the
# hostname as "virtual". This will make Squid use the IP address
# where it accepted the request as hostname in the URL.
#
# If you want virtual port support then specify the port as "0".
#
# NOTE: enabling httpd_accel_host disables proxy-caching and
# ICP. If you want these features enabled also, then set
# the 'httpd_accel_with_proxy' option.
#
#Default:
# httpd_accel_port 80
#
# TAG: httpd_accel_single_host      on|off
# If you are running Squid as an accelerator and have a single
# backend
# server then set this to on. This causes Squid to forward the
# request
# to this server irregardles of what any redirectors or Host
# headers
# says.

```

```

#
# Leave this at off if you have multiple backend servers, and use a
# redirector (or host table or private DNS) to map the requests to
the
# appropriate backend servers. Note that the mapping needs to be a
# 1-1 mapping between requested and backend (from redirector)
domain
# names or caching will fail, as cacing is performed using the
# URL returned from the redirector.
#
# See also redirect_rewrites_host_header.
#
#Default:
# httpd_accel_single_host off

# TAG: httpd_accel_with_proxy on|off
# If you want to use Squid as both a local httpd accelerator
# and as a proxy, change this to 'on'. Note however that your
# proxy users may have trouble to reach the accelerated domains
# unless their browsers are configured not to use this proxy for
# those domains (for example via the no_proxy browser configuration
# setting)
#
#Default:
# httpd_accel_with_proxy off

# TAG: httpd_accel_uses_host_header on|off
# HTTP/1.1 requests include a Host: header which is basically the
# hostname from the URL. The Host: header is used for domain based
# virtual hosts. If your accelerator needs to provide domain based
# virtual hosts on the same IP address then you will need to turn
this
# on.
#
# Note that Squid does NOT check the value of the Host header
matches
# any of your accelerated server, so it may open a big security
hole
# unless you take care to set up access controls proper. We
recommend
# that this option remain disabled unless you are sure of what you
# are doing.
#
# However, you will need to enable this option if you run Squid
# as a transparent proxy. Otherwise, virtual servers which
# require the Host: header will not be properly cached.
#
#Default:
# httpd_accel_uses_host_header off

# MISCELLANEOUS
# -----
-----

# TAG: dns_testnames
# The DNS tests exit as soon as the first site is successfully
looked up
#
# This test can be disabled with the -D command line option.
#
#Default:
# dns_testnames netscape.com internic.net nlanr.net microsoft.com

# TAG: logfile_rotate
# Specifies the number of logfile rotations to make when you

```

```
# type 'squid -k rotate'. The default is 10, which will rotate
# with extensions 0 through 9. Setting logfile_rotate to 0 will
# disable the rotation, but the logfiles are still closed and
# re-opened. This will enable you to rename the logfiles
# yourself just before sending the rotate signal.
#
# Note, the 'squid -k rotate' command normally sends a USR1
# signal to the running squid process. In certain situations
# (e.g. on Linux with Async I/O), USR1 is used for other
# purposes, so -k rotate uses another signal. It is best to get
# in the habit of using 'squid -k rotate' instead of 'kill -USR1
# <pid>'.
#
#logfile_rotate 0
#
#Default:
# logfile_rotate 0

# TAG: append_domain
# Appends local domain name to hostnames without any dots in
# them. append_domain must begin with a period.
#
# Be warned that there today is Internet names with no dots in
# them using only top-domain names, so setting this may
# cause some Internet sites to become unavailable.
#
#Example:
# append_domain .yourdomain.com
#
#Default:
# none
append_domain .bits-go.a.ac.in
# TAG: tcp_rcv_bufsize (bytes)
# Size of receive buffer to set for TCP sockets. Probably just
# as easy to change your kernel's default. Set to zero to use
# the default buffer size.
#
#Default:
# tcp_rcv_bufsize 0 bytes

# TAG: err_html_text
# HTML text to include in error messages. Make this a "mailto"
# URL to your admin address, or maybe just a link to your
# organizations Web page.
#
# To include this in your error messages, you must rewrite
# the error template files (found in the "errors" directory).
# Wherever you want the 'err_html_text' line to appear,
# insert a %L tag in the error template file.
#
#Default:
# none

# TAG: deny_info
# Usage: deny_info err_page_name acl
# Example: deny_info ERR_CUSTOM_ACCESS_DENIED bad_guys
#
# This can be used to return a ERR_ page for requests which
# do not pass the 'http_access' rules. A single ACL will cause
# the http_access check to fail. If a 'deny_info' line exists
# for that ACL then Squid returns a corresponding error page.
#
# You may use ERR_ pages that come with Squid or create your own
pages
# and put them into the configured errors/ directory.
#
```

```

# Alternatively you can tell Squid to reset the TCP connection
# by specifying TCP_RESET.
#
#Default:
# none

# TAG: memory_pools    on|off
#   If set, Squid will keep pools of allocated (but unused) memory
#   available for future use.  If memory is a premium on your
#   system and you believe your malloc library outperforms Squid
#   routines, disable this.
#
#Default:
# memory_pools on

# TAG: memory_pools_limit    (bytes)
#   Used only with memory_pools on:
#   memory_pools_limit 50 MB
#
#   If set to a non-zero value, Squid will keep at most the specified
#   limit of allocated (but unused) memory in memory pools. All
free()
#   requests that exceed this limit will be handled by your malloc
#   library. Squid does not pre-allocate any memory, just safe-keeps
#   objects that otherwise would be free()d. Thus, it is safe to set
#   memory_pools_limit to a reasonably high value even if your
#   configuration will use less memory.
#
#   If not set (default) or set to zero, Squid will keep all memory
it
#   can. That is, there will be no limit on the total amount of
memory
#   used for safe-keeping.
#
#   To disable memory allocation optimization, do not set
#   memory_pools_limit to 0. Set memory_pools to "off" instead.
#
#   An overhead for maintaining memory pools is not taken into
account
#   when the limit is checked. This overhead is close to four bytes
per
#   object kept. However, pools may actually _save_ memory because of
#   reduced memory thrashing in your malloc library.
#
#Default:
# none

# TAG: forwarded_for    on|off
#   If set, Squid will include your system's IP address or name
#   in the HTTP requests it forwards.  By default it looks like
#   this:
#
#           X-Forwarded-For: 192.1.2.3
#
#   If you disable this, it will appear as
#
#           X-Forwarded-For: unknown
#
#Default:
# forwarded_for on

# TAG: log_icp_queries on|off
#   If set, ICP queries are logged to access.log. You may wish
#   do disable this if your ICP load is VERY high to speed things
#   up or to simplify log analysis.
#

```

```
#Default:
# log_icp_queries on

# TAG: icp_hit_stale on|off
#   If you want to return ICP_HIT for stale cache objects, set this
#   option to 'on'.  If you have sibling relationships with caches
#   in other administrative domains, this should be 'off'.  If you
only
#   have sibling relationships with caches under your control, then
#   it is probably okay to set this to 'on'.
#   If set to 'on', then your siblings should use the option "allow-
miss"
#   on their cache_peer lines for connecting to you.
#
#Default:
# icp_hit_stale off

# TAG: minimum_direct_hops
#   If using the ICMP pinging stuff, do direct fetches for sites
#   which are no more than this many hops away.
#
#Default:
# minimum_direct_hops 4

# TAG: minimum_direct_rtt
#   If using the ICMP pinging stuff, do direct fetches for sites
#   which are no more than this many rtt milliseconds away.
#
#Default:
# minimum_direct_rtt 400

# TAG: cachemgr_passwd
#   Specify passwords for cachemgr operations.
#
#   Usage: cachemgr_passwd password action action ...
#
#   Some valid actions are (see cache manager menu for a full list):
#       5min
#       60min
#       asndb
#       authenticator
#       cbdata
#       client_list
#       comm_incoming
#       config *
#       counters
#       delay
#       digest_stats
#       dns
#       events
#       filedescriptors
#       fqdnocache
#       histograms
#       http_headers
#       info
#       io
#       ipcache
#       mem
#       menu
#       netdb
#       non_peers
#       objects
#       pconn
#       peer_select
#       redirector
#       refresh
```

```
#         server_list
#         shutdown *
#         store_digest
#         storedir
#         utilization
#         via_headers
#         vm_objects
#
# * Indicates actions which will not be performed without a
#   valid password, others can be performed if not listed here.
#
# To disable an action, set the password to "disable".
# To allow performing an action without a password, set the
# password to "none".
#
# Use the keyword "all" to set the same password for all actions.
#
#Example:
# cachemgr_passwd secret shutdown
# cachemgr_passwd lessssssssecret info stats/objects
# cachemgr_passwd disable all
#
#Default:
# none

# TAG: store_avg_object_size (kbytes)
#   Average object size, used to estimate number of objects your
#   cache can hold. See doc/Release-Notes-1.1.txt. The default is
#   13 KB.
#
#Default:
# store_avg_object_size 13 KB

# TAG: store_objects_per_bucket
#   Target number of objects per bucket in the store hash table.
#   Lowering this value increases the total number of buckets and
#   also the storage maintenance rate. The default is 50.
#
#Default:
# store_objects_per_bucket 20

# TAG: client_db on|off
#   If you want to disable collecting per-client statistics, then
#   turn off client_db here.
#
#Default:
# client_db on

# TAG: netdb_low
# TAG: netdb_high
#   The low and high water marks for the ICMP measurement
#   database. These are counts, not percents. The defaults are
#   900 and 1000. When the high water mark is reached, database
#   entries will be deleted until the low mark is reached.
#
#Default:
# netdb_low 900
# netdb_high 1000

# TAG: netdb_ping_period
#   The minimum period for measuring a site. There will be at
#   least this much delay between successive pings to the same
#   network. The default is five minutes.
#
#Default:
# netdb_ping_period 5 minutes
```

```
# TAG: query_icmp      on|off
#   If you want to ask your peers to include ICMP data in their ICP
#   replies, enable this option.
#
#   If your peer has configured Squid (during compilation) with
#   '--enable-icmp' then that peer will send ICMP pings to origin
server
#   sites of the URLs it receives.  If you enable this option then
the
#   ICP replies from that peer will include the ICMP data (if
available).
#   Then, when choosing a parent cache, Squid will choose the parent
with
#   the minimal RTT to the origin server.  When this happens, the
#   hierarchy field of the access.log will be
#   "CLOSEST_PARENT_MISS".  This option is off by default.
#
#Default:
# query_icmp off

# TAG: test_reachability  on|off
#   When this is 'on', ICP MISS replies will be ICP_MISS_NOFETCH
#   instead of ICP_MISS if the target host is NOT in the ICMP
#   database, or has a zero RTT.
#
#Default:
# test_reachability off

# TAG: buffered_logs    on|off
#   cache.log log file is written with stdio functions, and as such
#   it can be buffered or unbuffered.  By default it will be
unbuffered.
#   Buffering it can speed up the writing slightly (though you are
#   unlikely to need to worry unless you run with tons of debugging
#   enabled in which case performance will suffer badly anyway..).
#
#Default:
# buffered_logs off

# TAG: reload_into_ims on|off
#   When you enable this option, client no-cache or ``reload''
#   requests will be changed to If-Modified-Since requests.
#   Doing this VIOLATES the HTTP standard.  Enabling this
#   feature could make you liable for problems which it
#   causes.
#
#   see also refresh_pattern for a more selective approach.
#
#Default:
# reload_into_ims off

# TAG: always_direct
#   Usage: always_direct allow|deny [!]aclname ...
#
#   Here you can use ACL elements to specify requests which should
#   ALWAYS be forwarded directly to origin servers.  For example,
#   to always directly forward requests for local servers use
#   something like:
#
#       acl local-servers dstdomain my.domain.net
#       always_direct allow local-servers
#
#   To always forward FTP requests directly, use
#
#       acl FTP proto FTP
```

```

#         always_direct allow FTP
#
# NOTE: There is a similar, but opposite option named
# 'never_direct'. You need to be aware that "always_direct deny
# foo" is NOT the same thing as "never_direct allow foo". You
# may need to use a deny rule to exclude a more-specific case of
# some other rule. Example:
#
#         acl local-external dstdomain external.foo.net
#         acl local-servers dstdomain .foo.net
#         always_direct deny local-external
#         always_direct allow local-servers
#
# This option replaces some v1.1 options such as local_domain
# and local_ip.
#
#Default:
# none

# TAG: never_direct
# Usage: never_direct allow|deny [!]aclname ...
#
# never_direct is the opposite of always_direct. Please read
# the description for always_direct if you have not already.
#
# With 'never_direct' you can use ACL elements to specify
# requests which should NEVER be forwarded directly to origin
# servers. For example, to force the use of a proxy for all
# requests, except those in your local domain use something like:
#
#         acl local-servers dstdomain .foo.net
#         acl all src 0.0.0.0/0.0.0.0
#         never_direct deny local-servers
#         never_direct allow all
#
# or if squid is inside a firewall and there is local intranet
# servers inside the firewall then use something like:
#
#         acl local-intranet dstdomain .foo.net
#         acl local-external dstdomain external.foo.net
#         always_direct deny local-external
#         always_direct allow local-intranet
#         never_direct allow all
#
# This option replaces some v1.1 options such as inside_firewall
# and firewall_ip.
#
#Default:
# none

# TAG: header_access
# Usage: header_access header_name allow|deny [!]aclname ...
#
# WARNING: Doing this VIOLATES the HTTP standard. Enabling
# this feature could make you liable for problems which it
# causes.
#
# This option replaces the old 'anonymize_headers' and the
# older 'http_anonymizer' option with something that is much
# more configurable. This new method creates a list of ACLs
# for each header, allowing you very fine-tuned header
# mangling.
#
# You can only specify known headers for the header name.
# Other headers are reclassified as 'Other'. You can also
# refer to all the headers with 'All'.

```

```
#
# For example, to achieve the same behaviour as the old
# 'http_anonymizer standard' option, you should use:
#
#     header_access From deny all
#     header_access Referer deny all
#     header_access Server deny all
#     header_access User-Agent deny all
#     header_access WWW-Authenticate deny all
#     header_access Link deny all
#
# Or, to reproduce the old 'http_anonymizer paranoid' feature
# you should use:
#
#     header_access Allow allow all
#     header_access Authorization allow all
#     header_access Cache-Control allow all
#     header_access Content-Encoding allow all
#     header_access Content-Length allow all
#     header_access Content-Type allow all
#     header_access Date allow all
#     header_access Expires allow all
#     header_access Host allow all
#     header_access If-Modified-Since allow all
#     header_access Last-Modified allow all
#     header_access Location allow all
#     header_access Pragma allow all
#     header_access Accept allow all
#     header_access Accept-Charset allow all
#     header_access Accept-Encoding allow all
#     header_access Accept-Language allow all
#     header_access Content-Language allow all
#     header_access Mime-Version allow all
#     header_access Retry-After allow all
#     header_access Title allow all
#     header_access Connection allow all
#     header_access Proxy-Connection allow all
#     header_access All deny all
#
# By default, all headers are allowed (no anonymizing is
# performed).
#
#Default:
# none

# TAG: header_replace
# Usage: header_replace header_name message
# Example: header_replace User-Agent Nutscape/1.0 (CP/M; 8-bit)
#
# This option allows you to change the contents of headers
# denied with header_access above, by replacing them with
# some fixed string. This replaces the old fake_user_agent
# option.
#
# By default, headers are removed if denied.
#
#Default:
# none

# TAG: icon_directory
# Where the icons are stored. These are normally kept in
# /usr/share/squid/icons
#
#Default:
# icon_directory /usr/share/squid/icons
```

```
# TAG: error_directory
#   Directory where the error files are read from.
#   /usr/lib/squid/errors contains sets of error files
#   in different languages. The default error directory
#   is /etc/squid/errors, which is a link to one of these
#   error sets.
#
#   If you wish to create your own versions of the error files,
#   either to customize them to suit your language or company,
#   copy the template English files to another
#   directory and point this tag at them.
#
#error_directory /usr/share/squid/errors
#
#Default:
# error_directory /usr/share/squid/errors

# TAG: minimum_retry_timeout (seconds)
#   This specifies the minimum connect timeout, for when the
#   connect timeout is reduced to compensate for the availability
#   of multiple IP addresses.
#
#   When a connection to a host is initiated, and that host has
#   several IP addresses, the default connection timeout is reduced
#   by dividing it by the number of addresses. So, a site with 15
#   addresses would then have a timeout of 8 seconds for each
#   address attempted. To avoid having the timeout reduced to the
#   point where even a working host would not have a chance to
#   respond, this setting is provided. The default, and the
#   minimum value, is five seconds, and the maximum value is sixty
#   seconds, or half of connect_timeout, whichever is greater and
#   less than connect_timeout.
#
#Default:
# minimum_retry_timeout 5 seconds
minimum_retry_timeout 30 seconds
# TAG: maximum_single_addr_tries
#   This sets the maximum number of connection attempts for a
#   host that only has one address (for multiple-address hosts,
#   each address is tried once).
#
#   The default value is three tries, the (not recommended)
#   maximum is 255 tries. A warning message will be generated
#   if it is set to a value greater than ten.
#
#Default:
# maximum_single_addr_tries 3

# TAG: snmp_port
#   Squid can now serve statistics and status information via SNMP.
#   A value of "0" disables SNMP support. If you wish to use SNMP,
#   set this to "3401" to use the normal SNMP support.
#
#Default:
# snmp_port 0

# TAG: snmp_access
#   Allowing or denying access to the SNMP port.
#
#   All access to the agent is denied by default.
#   usage:
#
#   snmp_access allow|deny [!]aclname ...
#
#Example:
# snmp_access allow snmppublic localhost
```

```
# snmp_access deny all
#
#Default:
# snmp_access deny all

# TAG: snmp_incoming_address
# TAG: snmp_outgoing_address
#   Just like 'udp_incoming_address' above, but for the SNMP port.
#
#   snmp_incoming_address is used for the SNMP socket receiving
#   messages from SNMP agents.
#   snmp_outgoing_address is used for SNMP packets returned to SNMP
#   agents.
#
#   The default snmp_incoming_address (0.0.0.0) is to listen on all
#   available network interfaces.
#
#   If snmp_outgoing_address is set to 255.255.255.255 (the default)
#   then it will use the same socket as snmp_incoming_address. Only
#   change this if you want to have SNMP replies sent using another
#   address than where this Squid listens for SNMP queries.
#
#   NOTE, snmp_incoming_address and snmp_outgoing_address can not
have
#   the same value since they both use port 3401.
#
#Default:
# snmp_incoming_address 0.0.0.0
# snmp_outgoing_address 255.255.255.255

# TAG: as_whois_server
#   WHOIS server to query for AS numbers. NOTE: AS numbers are
#   queried only when Squid starts up, not for every request.
#
#Default:
# as_whois_server whois.ra.net
# as_whois_server whois.ra.net

# TAG: wccp_router
#   Use this option to define your WCCP ``home'' router for
#   Squid. Setting the 'wccp_router' to 0.0.0.0 (the default)
#   disables WCCP.
#
#Default:
# wccp_router 0.0.0.0

# TAG: wccp_version
#   According to some users, Cisco IOS 11.2 only supports WCCP
#   version 3. If you're using that version of IOS, change
#   this value to 3.
#
#Default:
# wccp_version 4

# TAG: wccp_incoming_address
# TAG: wccp_outgoing_address
#   wccp_incoming_address Use this option if you require WCCP
#   messages to be received on only one
#   interface. Do NOT use this option if
#   you're unsure how many interfaces you
#   have, or if you know you have only one
#   interface.
#
#   wccp_outgoing_address Use this option if you require WCCP
#   messages to be sent out on only one
#   interface. Do NOT use this option if
```

```

#             you're unsure how many interfaces you
#             have, or if you know you have only one
#             interface.
#
#             The default behavior is to not bind to any specific address.
#
#             NOTE, wccp_incoming_address and wccp_outgoing_address can not
have
#             the same value since they both use port 2048.
#
#Default:
# wccp_incoming_address 0.0.0.0
# wccp_outgoing_address 255.255.255.255

# DELAY POOL PARAMETERS (all require DELAY_POOLS compilation option)
# -----
-----

# TAG: delay_pools
#     This represents the number of delay pools to be used.  For
example,
#     if you have one class 2 delay pool and one class 3 delays pool,
you
#     have a total of 2 delay pools.
#
#Default:
# delay_pools 0
#delay_pools 1
# TAG: delay_class
#     This defines the class of each delay pool.  There must be exactly
one
#     delay_class line for each delay pool.  For example, to define two
#     delay pools, one of class 2 and one of class 3, the settings
above
#     and here would be:
#
#Example:
# delay_pools 2           # 2 delay pools
# delay_class 1 2        # pool 1 is a class 2 pool
# delay_class 2 3        # pool 2 is a class 3 pool
#
#     The delay pool classes are:
#
#             class 1           Everything is limited by a single
aggregate
#             bucket.
#
#             class 2           Everything is limited by a single aggregate
#             bucket as well as an "individual" bucket chosen
#             from bits 25 through 32 of the IP address.
#
#             class 3           Everything is limited by a single
aggregate
#             bucket as well as a "network" bucket chosen
#             from bits 17 through 24 of the IP address and a
#             "individual" bucket chosen from bits 17 through
#             32 of the IP address.
#
#     NOTE: If an IP address is a.b.c.d
#           -> bits 25 through 32 are "d"
#           -> bits 17 through 24 are "c"
#           -> bits 17 through 32 are "c * 256 + d"
#
#Default:
# none

```

```

#delay_class 1 1
# TAG: delay_access
# This is used to determine which delay pool a request falls into.
# The first matched delay pool is always used, i.e., if a request
falls
# into delay pool number one, no more delay are checked, otherwise
the
# rest are checked in order of their delay pool number until they
have
# all been checked. For example, if you want some_big_clients in
delay
# pool 1 and lotsa_little_clients in delay pool 2:
#
#Example:
# delay_access 1 allow some_big_clients
# delay_access 1 deny all
# delay_access 2 allow lotsa_little_clients
# delay_access 2 deny all
#
#Default:
# none
#delay_access 1 allow all
#delay_access 1 deny all
# TAG: delay_parameters
# This defines the parameters for a delay pool. Each delay pool
has
# a number of "buckets" associated with it, as explained in the
# description of delay_class. For a class 1 delay pool, the syntax
is:
#
#delay_parameters pool aggregate
#
# For a class 2 delay pool:
#
#delay_parameters pool aggregate individual
#
# For a class 3 delay pool:
#
#delay_parameters pool aggregate network individual
#
# The variables here are:
#
# pool a pool number - ie, a number between 1 and the
# number specified in delay_pools as used in
# delay_class lines.
#
# aggregate the "delay parameters" for the aggregate bucket
# (class 1, 2, 3).
#
# individual the "delay parameters" for the individual
# buckets (class 2, 3).
#
# network the "delay parameters" for the network
buckets
# (class 3).
#
# A pair of delay parameters is written restore/maximum, where
restore is
# the number of bytes (not bits - modem and network speeds are
usually
# quoted in bits) per second placed into the bucket, and maximum is
the
# maximum number of bytes which can be in the bucket at any time.
#
# For example, if delay pool number 1 is a class 2 delay pool as in
the

```

```
# above example, and is being used to strictly limit each host to
64kbps
# (plus overheads), with no overall limit, the line is:
#
#delay_parameters 1 -1/-1 8000/8000
#
# Note that the figure -1 is used to represent "unlimited".
#
# And, if delay pool number 2 is a class 3 delay pool as in the
above
# example, and you want to limit it to a total of 256kbps (strict
limit)
# with each 8-bit network permitted 64kbps (strict limit) and each
# individual host permitted 4800bps with a bucket maximum size of
64kb
# to permit a decent web page to be downloaded at a decent speed
# (if the network is not being limited due to overuse) but slow
down
# large downloads more significantly:
#
#delay_parameters 2 32000/32000 8000/8000 600/64000
#
# There must be one delay_parameters line for each delay pool.
#
#Default:
# none
#delay_parameters 1 40000/50000

# TAG: delay_initial_bucket_level (percent, 0-100)
# The initial bucket percentage is used to determine how much is
put
# in each bucket when squid starts, is reconfigured, or first
notices
# a host accessing it (in class 2 and class 3, individual hosts and
# networks only have buckets associated with them once they have
been
# "seen" by squid).
#
#Default:
# delay_initial_bucket_level 50

# TAG: incoming_icp_average
# TAG: incoming_http_average
# TAG: incoming_dns_average
# TAG: min_icp_poll_cnt
# TAG: min_dns_poll_cnt
# TAG: min_http_poll_cnt
# Heavy voodoo here. I can't even believe you are reading this.
# Are you crazy? Don't even think about adjusting these unless
# you understand the algorithms in comm_select.c first!
#
#Default:
# incoming_icp_average 6
# incoming_http_average 4
# incoming_dns_average 4
# min_icp_poll_cnt 8
# min_dns_poll_cnt 8
# min_http_poll_cnt 8

# TAG: max_open_disk_fds
# To avoid having disk as the I/O bottleneck Squid can optionally
# bypass the on-disk cache if more than this amount of disk file
# descriptors are open.
#
# A value of 0 indicates no limit.
```

```
#
#Default:
# max_open_disk_fds 0

# TAG: offline_mode
#   Enable this option and Squid will never try to validate cached
#   objects.
#
#Default:
# offline_mode off

# TAG: uri_whitespace
#   What to do with requests that have whitespace characters in the
#   URI. Options:
#
#   strip: The whitespace characters are stripped out of the URL.
#           This is the behavior recommended by RFC2616.
#   deny:  The request is denied. The user receives an "Invalid
#           Request" message.
#   allow: The request is allowed and the URI is not changed. The
#           whitespace characters remain in the URI. Note the
#           whitespace is passed to redirector processes if they
#           are in use.
#   encode: The request is allowed and the whitespace characters
are
#           encoded according to RFC1738. This could be considered
#           a violation of the HTTP/1.1
#           RFC because proxies are not allowed to rewrite URI's.
#   chop:  The request is allowed and the URI is chopped at the
#           first whitespace. This might also be considered a
#           violation.
#
#Default:
# uri_whitespace strip

# TAG: broken_posts
#   A list of ACL elements which, if matched, causes Squid to send
#   an extra CRLF pair after the body of a PUT/POST request.
#
#   Some HTTP servers has broken implementations of PUT/POST,
#   and rely on an extra CRLF pair sent by some WWW clients.
#
#   Quote from RFC 2068 section 4.1 on this matter:
#
#   Note: certain buggy HTTP/1.0 client implementations generate an
#   extra CRLF's after a POST request. To restate what is
explicitly
#   forbidden by the BNF, an HTTP/1.1 client must not preface or
follow
#   a request with an extra CRLF.
#
#Example:
# acl buggy_server url_regex ^http://....
# broken_posts allow buggy_server
#
#Default:
# none

# TAG: mcast_miss_addr
# Note: This option is only available if Squid is rebuilt with the
#       -DMULTICAST_MISS_STREAM option
#
#   If you enable this option, every "cache miss" URL will
#   be sent out on the specified multicast address.
#
#   Do not enable this option unless you are absolutely
```

```
# certain you understand what you are doing.
#
#Default:
# mcast_miss_addr 255.255.255.255

# TAG: mcast_miss_ttl
# Note: This option is only available if Squid is rebuilt with the
#       -DMULTICAST_MISS_TTL option
#
#       This is the time-to-live value for packets multicasted
#       when multicasting off cache miss URLs is enabled. By
#       default this is set to 'site scope', i.e. 16.
#
#Default:
# mcast_miss_ttl 16

# TAG: mcast_miss_port
# Note: This option is only available if Squid is rebuilt with the
#       -DMULTICAST_MISS_STREAM option
#
#       This is the port number to be used in conjunction with
#       'mcast_miss_addr'.
#
#Default:
# mcast_miss_port 3135

# TAG: mcast_miss_encode_key
# Note: This option is only available if Squid is rebuilt with the
#       -DMULTICAST_MISS_STREAM option
#
#       The URLs that are sent in the multicast miss stream are
#       encrypted. This is the encryption key.
#
#Default:
# mcast_miss_encode_key XXXXXXXXXXXXXXXXXXXX

# TAG: nonhierarchical_direct
#       By default, Squid will send any non-hierarchical requests
#       (matching hierarchy_stoplist or not cachable request type) direct
#       to origin servers.
#
#       If you set this to off, then Squid will prefer to send these
#       requests to parents.
#
#       Note that in most configurations, by turning this off you will
only
#       add latency to these request without any improvement in global
hit
#       ratio.
#
#       If you are inside an firewall then see never_direct instead of
#       this directive.
#
#Default:
# nonhierarchical_direct on

# TAG: prefer_direct
#       Normally Squid tries to use parents for most requests. If you by
some
#       reason like it to first try going direct and only use a parent if
#       going direct fails then set this to on.
#
#       By combining nonhierarchical_direct off and prefer_direct on you
#       can set up Squid to use a parent as a backup path if going direct
#       fails.
#
```

```
#Default:
# prefer_direct off

# TAG: strip_query_terms
#   By default, Squid strips query terms from requested URLs before
#   logging. This protects your user's privacy.
#
#Default:
# strip_query_terms on
strip_query_terms off
# TAG: coredump_dir
#   By default Squid leaves core files in the directory from where
#   it was started. If you set 'coredump_dir' to a directory
#   that exists, Squid will chdir() to that directory at startup
#   and coredump files will be left there.
#
#Default:
# coredump_dir none
#
# Leave coredumps in the first cache dir
coredump_dir /var/spool/squid

# TAG: redirector_bypass
#   When this is 'on', a request will not go through the
#   redirector if all redirectors are busy. If this is 'off'
#   and the redirector queue grows too large, Squid will exit
#   with a FATAL error and ask you to increase the number of
#   redirectors. You should only enable this if the redirectors
#   are not critical to your caching system. If you use
#   redirectors for access control, and you enable this option,
#   then users may have access to pages that they should not
#   be allowed to request.
#
#Default:
# redirector_bypass off

# TAG: ignore_unknown_nameservers
#   By default Squid checks that DNS responses are received
#   from the same IP addresses that they are sent to. If they
#   don't match, Squid ignores the response and writes a warning
#   message to cache.log. You can allow responses from unknown
#   nameservers by setting this option to 'off'.
#
#Default:
# ignore_unknown_nameservers on

# TAG: digest_generation
# Note: This option is only available if Squid is rebuilt with the
#       --enable-cache-digests option
#
#   This controls whether the server will generate a Cache Digest
#   of its contents. By default, Cache Digest generation is
#   enabled if Squid is compiled with USE_CACHE_DIGESTS defined.
#
#Default:
# digest_generation on

# TAG: digest_bits_per_entry
# Note: This option is only available if Squid is rebuilt with the
#       --enable-cache-digests option
#
#   This is the number of bits of the server's Cache Digest which
#   will be associated with the Digest entry for a given HTTP
#   Method and URL (public key) combination. The default is 5.
#
#Default:
```

```
# digest_bits_per_entry 5

# TAG: digest_rebuild_period (seconds)
# Note: This option is only available if Squid is rebuilt with the
#       --enable-cache-digests option
#
#       This is the number of seconds between Cache Digest rebuilds.
#
#Default:
# digest_rebuild_period 1 hour

# TAG: digest_rewrite_period (seconds)
# Note: This option is only available if Squid is rebuilt with the
#       --enable-cache-digests option
#
#       This is the number of seconds between Cache Digest writes to
#       disk.
#
#Default:
# digest_rewrite_period 1 hour

# TAG: digest_swapout_chunk_size (bytes)
# Note: This option is only available if Squid is rebuilt with the
#       --enable-cache-digests option
#
#       This is the number of bytes of the Cache Digest to write to
#       disk at a time. It defaults to 4096 bytes (4KB), the Squid
#       default swap page.
#
#Default:
# digest_swapout_chunk_size 4096 bytes

# TAG: digest_rebuild_chunk_percentage (percent, 0-100)
# Note: This option is only available if Squid is rebuilt with the
#       --enable-cache-digests option
#
#       This is the percentage of the Cache Digest to be scanned at a
#       time. By default it is set to 10% of the Cache Digest.
#
#Default:
# digest_rebuild_chunk_percentage 10

# TAG: chroot
#       Use this to have Squid do a chroot() while initializing. This
#       also causes Squid to fully drop root privileges after
#       initializing. This means, for example, that if you use a HTTP
#       port less than 1024 and try to reconfigure, you will get an
#       error.
#
#Default:
# none

# TAG: client_persistent_connections
# TAG: server_persistent_connections
#       Persistent connection support for clients and servers. By
#       default, Squid uses persistent connections (when allowed)
#       with its clients and servers. You can use these options to
#       disable persistent connections with clients and/or servers.
#
#Default:
# client_persistent_connections on
# server_persistent_connections on

# TAG: pipeline_prefetch
#       To boost the performance of pipelined requests to closer
#       match that of a non-proxied environment Squid can try to fetch
```

```
# up to two requests in parallel from a pipeline.
#
# Defaults to off for bandwidth management and access logging
# reasons.
#
#Default:
# pipeline_prefetch off

# TAG: extension_methods
# Squid only knows about standardized HTTP request methods.
# You can add up to 20 additional "extension" methods here.
#
#Default:
# none

# TAG: request_entities
# Squid defaults to deny GET and HEAD requests with request
# entities,
# as the meaning of such requests are undefined in the HTTP
# standard
# even if not explicitly forbidden.
#
# Set this directive to on if you have clients which insists
# on sending request entities in GET or HEAD requests.
#
#Default:
# request_entities off

# TAG: high_response_time_warning (msec)
# If the one-minute median response time exceeds this value,
# Squid prints a WARNING with debug level 0 to get the
# administrators attention. The value is in milliseconds.
#
#Default:
# high_response_time_warning 0

# TAG: high_page_fault_warning
# If the one-minute average page fault rate exceeds this
# value, Squid prints a WARNING with debug level 0 to get
# the administrators attention. The value is in page faults
# per second.
#
#Default:
# high_page_fault_warning 0

# TAG: high_memory_warning
# If the memory usage (as determined by mallinfo) exceeds
# value, Squid prints a WARNING with debug level 0 to get
# the administrators attention.
#
#Default:
# high_memory_warning 0

# TAG: store_dir_select_algorithm
# Set this to 'round-robin' as an alternative.
#
#Default:
# store_dir_select_algorithm least-load

# TAG: forward_log
# Note: This option is only available if Squid is rebuilt with the
# -DWIP_FWD_LOG option
#
# Logs the server-side requests.
#
# This is currently work in progress.
```

```
#
#Default:
# none

# TAG: ie_refresh      on|off
#   Microsoft Internet Explorer up until version 5.5 Service
#   Pack 1 has an issue with transparent proxies, wherein it
#   is impossible to force a refresh. Turning this on provides
#   a partial fix to the problem, by causing all IMS-REFRESH
#   requests from older IE versions to check the origin server
#   for fresh content. This reduces hit ratio by some amount
#   (~10% in my experience), but allows users to actually get
#   fresh content when they want it. Note that because Squid
#   cannot tell if the user is using 5.5 or 5.5SP1, the behavior
#   of 5.5 is unchanged from old versions of Squid (i.e. a
#   forced refresh is impossible). Newer versions of IE will,
#   hopefully, continue to have the new behavior and will be
#   handled based on that assumption. This option defaults to
#   the old Squid behavior, which is better for hit ratios but
#   worse for clients using IE, if they need to be able to
#   force fresh content.
#
#Default:
# ie_refresh off

# TAG: vary_ignore_expire  on|off
#   Many HTTP servers supporting Vary gives such objects
#   immediate expiry time with no cache-control header
#   when requested by a HTTP/1.0 client. This option
#   enables Squid to ignore such expiry times until
#   HTTP/1.1 is fully implemented.
#   WARNING: This may eventually cause some varying
#   objects not intended for caching to get cached.
#
#Default:
# vary_ignore_expire off

# TAG: sleep_after_fork    (microseconds)
#   When this is set to a non-zero value, the main Squid process
#   sleeps the specified number of microseconds after a fork()
#   system call. This sleep may help the situation where your
#   system reports fork() failures due to lack of (virtual)
#   memory. Note, however, that if you have a lot of child
#   processes, then these sleep delays will add up and your
#   Squid will not service requests for some amount of time
#   until all the child processes have been started.
#
#Default:
# sleep_after_fork 0
```