LABORATORY WORK NO. 7
DOMAIN NAME SYSTEM

1. Objectives

The aims of this laboratory are: understanding name resolution, understanding DNS namespace, understanding DNS components: servers, zones, resolvers, resource records, understanding how a DNS query works: resolution methods, query steps, caching; deploying DNS servers, configuring DNS clients.

2. Theoretical considerations

2.1 Introduction

Every host that runs TCP/IP must have a unique IP address that is used when communicating with other computers. Computers operate easily with IP addresses, but people do not; users would rather identify systems by name. To facilitate effective and efficient communication, users need to be able to refer to a computer by name and still have their computer use IP addresses transparently.

In the early days o ARPANET there were only a small number of computers attached to the network. The Network Information Center (NIC) was responsible for compiling a single file named HOSTS.TXT that contained the names and addresses of every computer. Administrators would e-mail updates to Standford Research Institute, which would then update the HOSTS.TXT file. ARPANET users would then download the new version of the file using FTP and convert it for local use (for example, copy HOST.TXT/Etc/Hosts on UNIX system).

As ARPANET grew, it became obvious that this approach would not scale for the following reasons: the bandwidth consumed in transmitting update versions of an ARPANET wide host file was proportional to the square of the number of hosts in the ARPANET, the static flat host file also meant that no two computers on the entire ARPANET could have the same name, the nature of the underlying network was changing. As the ARPANET continued to grow, it became clear that a better solution was required. Several proposals were generated based on the concept of a distributed naming service, based on a hierarchical namespace. RFCs 882 and 883 emerged, which describe the design for a domain name system based on a distributed database containing generalized resource information. This design is described in RFCs 1034 and 1035.

At its core, the DNS is still a list of names and their IP addresses, but instead of storing all the information in one place, the DNS distributes it among servers all over the Internet.

2.2 Exploring DNS

2.2.1 DNS Namespace. Domain names

The naming system on which DNS is based is a hierarchical and logical tree structure called the DNS namespace. The DNS namespace has a unique root that can have any number of subdomains, each node and leaf in the domain namespace tree represents a named domain. Each domain can have additional child domains.

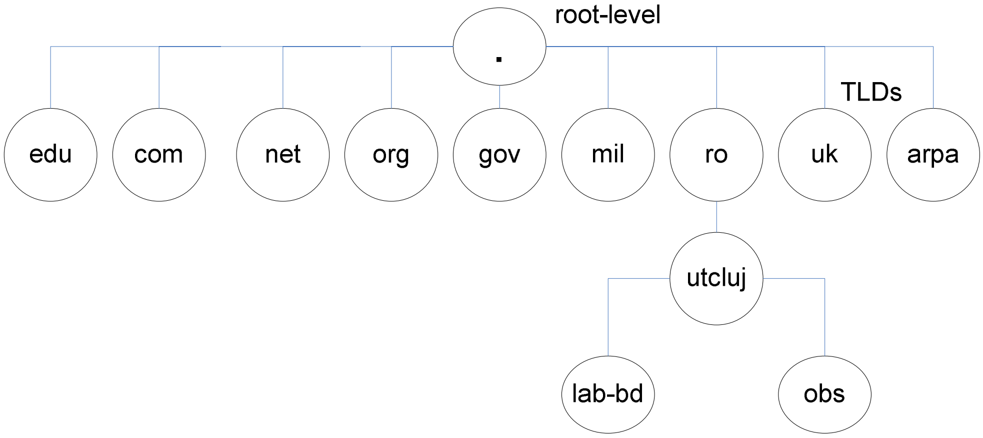

Figure 7.1 *DNS hierarchy*

The hierarchical nature of the DNS domain namespace is designed to make it possible for any DNS server on the Internet to use a minimum number of queries to locate the authoritative source for any domain name. This efficiency is possible because the domains at each level are responsible for maintaining information about the domains at the next lower level.

Each node in the DNS tree has a separate name, referred to in RFC 1034 as a label. Each DNS label can be from 1 through 63 characters in length, with the root domain having a length of zero characters.
A specific node's domain name is the list of labels in the path from the node to the DNS namespace root. DNS convention is that labels that compose a domain name are read left to right – from the most specific to the root (for example, host1.design.companya.com). This full name is also known as the fully qualified domain name (FQDN).

The domains directly bellow the root are called top-level domains (TLDs). There are three categories of TLSs:

- ARPA: this is a special domain used for IP_address-to-name mappings (referred to as reverse lookup);
- Generic TLDs (organizational domains): domains named using a three-character code that indicates the primary function or activity of the organizations contained in the DNS domain (for example, com, edu, info, gov, mil, museum, biz, aero, net, org);
- Two-letter country-based domain (geographical domains): based on the ISO country name, used principally by companies and organizations outside the USA (for example, ro, uk, dk, sg).

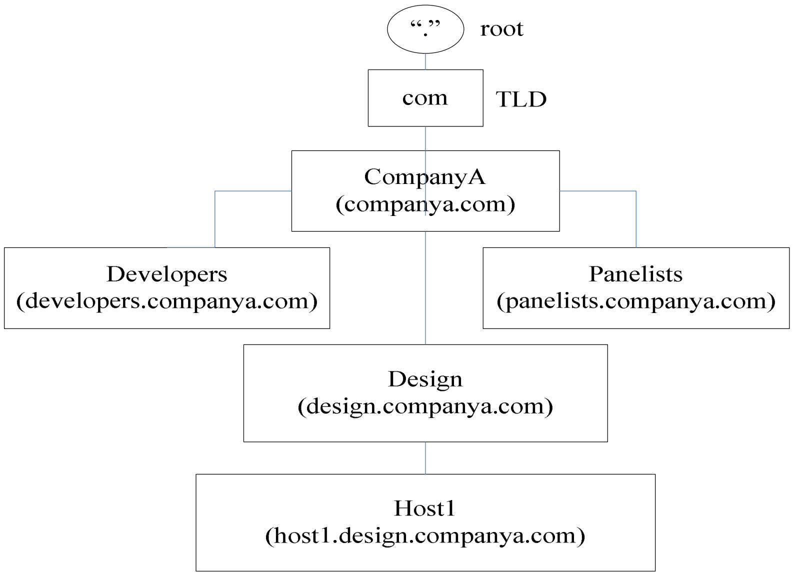

Figure 7.2 *Domains and subdomains*

2.2.2 DNS Components

DNS relies on the proper configuration of DNS servers, zones, resolvers, and resource records.

2.2.2.1 DNS Servers

A DNS server is a computer that runs a DNS server program, such as the DNS Server service or Berkley Internet Name Domain (BIND). DNS servers contain DNS database information about some portion of the DNS domain tree structure and resolve name resolution queries issued by DNS clients. When queried, DNS servers can provide the requested information, provide a pointer to another server that can help resolve the query, or respond that the information is unavailable or does not exist.

The root name servers are the highest-level DNS servers in the namespace, and they maintain information about the top-level domains. Software developers preconfigure all DNS server implementations with the IP addresses of multiple root name servers.

Name servers that hold all information on hosts within a zone are referred to as master name servers. Any query for a host within this zone will end up at one of these master name servers. Master servers must be fairly well synchronized. Thus, the zone's network administrator must make one the primary server, which loads its zone information from data files, and make the others secondary servers, which transfer the zone data from the primary server at regular intervals.

2.2.2.2 DNS Zones

A zone is a contiguous portion of the domain namespace for which a DNS server has authority to resolve DNS queries. You can divide the DNS namespace into zones which store information about one or more DNS domains. For each DNS domain name included in a zone, the zone becomes the authoritative source for information about the domain.

In many DNS server implementations, zone data is stored in a flat database or text file.
Zone files contain the necessary information that a DNS server references to perform two different tasks: resolving host names to IP addresses or

resolving IP addresses to host names. This information is stored as resource records that populate the zone file.

2.2.2.3 DNS resolvers

A DNS resolver is a service that uses the DNS protocol to query for information from DNS servers. In many implementations, the resolver is part of the DNS Client service.

An IP host that needs to contact another host on a regular basis needs to resolve a particular DNS name many times (for example, the name of the mail server). To avoid having send queries to a DNS server each time the host wants to resolve the name, the DNS Client service implements a local cache of DNS information.

The DNS Client service caches resource records from query response that the DNS Client service receives. The information is held for a set TTL and, when present, is used to answer subsequent queries and avoid sending the query to a DNS server.
You can use the `ipconfig` command with the `/displaydns` option to display the current DNS resolver cache contents.

2.2.2.4 Resource records

A resource record (RR) is a record containing information relating to a domain that the DNS database can hold and that a DNS client can retrieve and use.
Each DNS server contains the RRs relating to those portions of the DNS namespace for which it is authoritative. There are various resource record types that are defined for a zone. Each type of resource record contains different types of data.

DNS servers can contain the following types of resource records:

- A – host name: used to hold a specific host’s IP address;
- NS – name server: defines the servers that are authoritative for a certain zone or contain the zone file for that domain;
- CNAME – canonical name: used to make an alias name for a host; allows to provide additional names to a server that already has a name in an A record (for example, if the server called webserver.cs.utcluj should host the web site for cs.utcluj, this server should have the common name www.webserver.cs.utcluj;

to do this you can create a CNAME resource record that maps the name www to webserver);

- MX – mail exchanger: specifies the server to which e-mail applications can deliver mail (for example, if you have a mail server running on a computer named mail.cs.utcluj and you want all of the mail for user_name@cs.utcluj to be delivered to this mail server, you need to add an MX record to the zone for cs.utcluj that points to the mail server for that domain);
- SOA – start of authority: used to determine the DNS server that is the primary server for a DNS zone and to store other zone property information; the SOA resource record is the first resource record that is created when you add a new zone;
- PTR – pointer: used for reverse lookup – resolving an IP address into a domain name using the in-addr.arpa domain;
- SRV – service locator: provides the ability to find a server providing a specific service.

2.3 Query process

DNS uses a client/server model in which the DNS server contains information about a portion of the DNS namespace and provides this information to clients. A DNS client queries a DNS server for information about the DNS namespace. This server can, in turn, query other DNS servers to answer the client’s query.

When a DNS server receives a DNS request, it attempts to locate the requested information within its own database by searching its local zone files. The DNS server can also search in its local cache, which is comprised of results from previous queries. If the request cannot be answered by using either the local zone files or the cache, further communication with other DNS servers is required.

2.3.1 Query types

There are two types of DNS queries that are performed in DNS:

- iterative: a query made from a client to a DNS server in which the server returns the best answer that it can provide based on its cache or zone data. If the queried server does not have an exact match for the request, it provides a pointer to an authoritative server in a lower level of the domain namespace. The client then queries the authoritative server to which it was referred. The client continues this process until it locates a server that is authoritative for the requested name, an error occurs, or time-out condition is met;

- recursive: a query made from a client to a DNS server in which the server assumes the full workload and responsibility for providing a complete answer to the query, if an answer can be found anywhere in the known DNS hierarchy. If a request cannot be answered by using local zone or cached data, the server will perform separate iterative queries to other servers, on behalf of the clients, to assist in answering the recursive query.

In general, host computers issue recursive queries against DNS servers. The host assumes that the DNS server either knows the answer to the query or can find the answer. On the other hand, DNS server issues iterative queries against other DNS servers if it is unable to answer a recursive query from cached or locally stored information.

2.3.2 Lookup types

Most queries sent to a DNS server involve a search based on the DNS name of another computer as stored in an address A RR. This type of query expects an IP address as the resource data for the answered response and it is generally referred to as *forward query*.

DNS also provides a reverse-lookup process, which enables a host to determine another host’s name based on its IP address. To allow for *reverse queries*, a special domain, in-addr.arpa, was defined and reserved in the Internet DNS namespace. Subdomains within the in-addr.arpa domain are named using the reverse ordering of the numbers in the dotted-decimal notation of IP addresses. For example, the reverse-lookup zone for the subnet 192.168.100.0 is 100.168.192.in-addr.arpa.

2.3.3 DNS query steps

The following example illustrates a DNS query process. In the example, the client queries its preferred DNS server, which then performs recursion by querying hierarchically superior DNS servers. In the example, the DNS client and all DNS servers are assumed to have empty caches.

In the example, a client somewhere on the Internet needs to resolve the name example.utcluj.ro to an IP address.

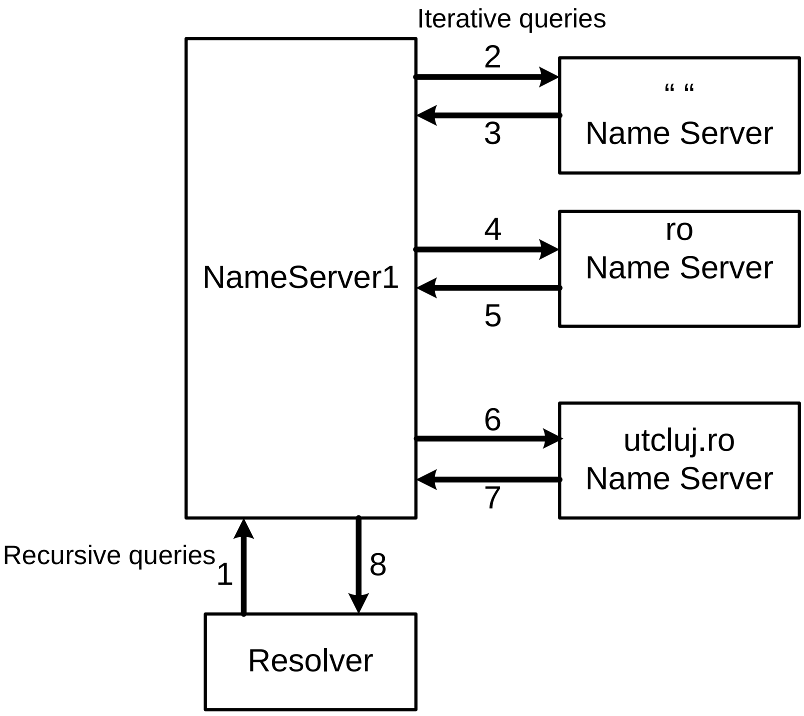

Figure 7.3 *DNS query example*

1. The client computer generates a request for the IP address of example.utcluj.ro by sending a recursive query to the DNS server that it is configured to use (NameServer1).
2. NameServer1 checks its cache and zones for the answer but does not find it, so it contacts a server authoritative for the Internet (that is, the root server) with a query for example.utcluj.ro.
3. The server at the root of the Internet does not know the answer, so it responds with a referral to a server authoritative for the ro domain.
4. Nameserver1 contacts a server authoritative for the .ro domain with a query for example.utcluj.ro.
5. The server authoritative for the ro domain does not know the exact answer, so it responds with a referral to a server authoritative for the utcluj.ro domain.

6. NameServer1 contacts the server authoritative for the utcluj.ro domain with a query for example.utcluj.ro.
7. The server authoritative for the utcluj.ro domain does know the answer. It responds with the requested IP address.
8. NameServer1 responds to the client query with the IP address for example.utcluj.ro.

2.3.4 Caching

Both the DNS client service and the DNS server service maintain caches. Caching provides a way to improve DNS performance and to substantially reduce DNS-related query traffic on the network.

The DNS client is also called the DNS resolver cache. Whenever the DNS client service starts, all host-name-to-IP-address aping contained in a static file named Hosts are preloaded into the DNS resolver cache. In addition to the entries in the Hosts file, the DNS resolver cache also includes entries the client has received in response to a query from DNS servers. The DNS resolver cache is emptied whenever the DNS Client service is stopped.

As DNS servers make recursive queries on behalf of clients, they temporarily cache resource records. These cached records contain information acquired in the process of answering on behalf of a client. Later, when other clients place new queries that request information matching cached resource records, the DNS server can use the cached information to answer these queries.

The DNS server cache is cleared whenever the DNS server service is stopped. In addition, you can clear the DNS cache manually.

A Time to Live value applies to all cached resource records, whether in the DNS resolver cache or the DNS server cache. As long as the TTL for a cached record does not expire, a DNS resolver or server can continue to use that record to answer queries. By default, the TTL is 1 hour, but this parameter can be adjusted at both the zone and record level.

2.3.5 Nslookup

Nslookup is a program to query Internet domain name servers. Nslookup has two modes: interactive and non-interactive. Interactive mode allows the user to query name servers for information about various hosts and domains or to print a list of hosts in a domain. Non-interactive mode is used to print just the name and requested information for a host or domain.

```
nslookup [ -option ] [ host-to-find | - [ server ]]
```

3. Lab activity

3.1 Go to page http://www.root-servers.org to find out which are the root servers.
3.2 Install the DNS Server Service.
3.3 Configure the new DNS Server: create a zone, create resource records for this zone.
3.4 Configure DNS Client settings.
3.5 Test your configuration querying for a resource record that you have defined in the zone on the server.
3.6 Test the command `nslookup` *`domain_name`*.
3.7 Test the command `nslookup` *`IP_address`*.

Notes