# Chapter 1

# Domain Name System

## 1.1 DNS fundamentals

Domain Name System, DNS, is used to find IP-addresses from a logical name using the concept of the Host Lookup Table. The Host Lookup Table, HLT, was a file placed on every computer connected to the network which contained the IP-address and the logical name for each computer connected. When a computer on the network needs to communicate with another connected computer the HLT was used to map the human friendly - and easy to remember - logical name with the IP address [**?**, History section].
When a new computer joined the network the HLT was updated, and all the other computers on the network needed to add the logical name and associated IP address which, due to the expansion of connecting computers to the network, became an obstacle and hindrance for the flow. To solve this the DNS system was invented in 1982 [**?**, History section].
DNS, one of the largest distributed systems, is like a big telephone book which everyone uses to lookup IP-addresses from logical names, through an address record (A record) [**?**, p. 209-210], rather than everyone keeping track of all connected addresses.

To communicate with a computer (or device) on a network the hostname is used to identify the computer. To find the computer's hostname the command `hostname` will show the logical human readable name for the computer.

In Linux the command `nm-tool` will access the NetworkManager Tool and among others, show the IP-address, MAC-address, connection state and DNS-server for the computer. This is shown in Figure 1.1[1].
You can run a similar command in Windows; `ipconfig /all` to access IP-address, DNS-server, MAC-Address ect. shown in Figure 1.2.

[1] Note that this is on a virtual machine which do not show as much as a native Linux machine will

```
limro@ubuntu:~$ nm-tool

NetworkManager Tool

State: connected (global)

- Device: eth0  [Wired connection 1] ----
------------
  Type:              Wired
  Driver:            vmxnet
  State:             connected
  Default:           yes
  HW Address:        00:0C:29:D5:8E:55

  Capabilities:
    Carrier Detect:  yes
    Speed:           1000 Mb/s

  Wired Properties
    Carrier:         on

  IPv4 Settings:
    Address:         192.168.92.128
    Prefix:          24 (255.255.255.0)
    Gateway:         192.168.92.2

    DNS:             192.168.92.2
```

Figure 1.1: Use of the command `nm-tool`

```
C:\Users\Becks>ipconfig /all

Windows IP Configuration

   Host Name . . . . . . . . . . . . : Becks-PC
   Primary Dns Suffix  . . . . . . . :
   Node Type . . . . . . . . . . . . : Hybrid
   IP Routing Enabled. . . . . . . . : No
   WINS Proxy Enabled. . . . . . . . : No
   DNS Suffix Search List. . . . . . : iha.dk
   System Quarantine State . . . . . : Not Restricted


Wireless LAN adapter Wireless Network Connection:

   Connection-specific DNS Suffix  . : iha.dk
   Description . . . . . . . . . . . : Broadcom 4313 802.11b/g/n
   Physical Address. . . . . . . . . : E0-2A-82-A7-2E-43
   DHCP Enabled. . . . . . . . . . . : Yes
   Autoconfiguration Enabled . . . . : Yes
   Link-local IPv6 Address . . . . . : fe80::29e7:43ec:7fc0:49c4%14(Preferred)
   IPv4 Address. . . . . . . . . . . : 10.193.2.193(Preferred)
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Lease Obtained. . . . . . . . . . : 16. april 2013 12:05:07
   Lease Expires . . . . . . . . . . : 17. april 2013 00:55:45
   Default Gateway . . . . . . . . . : 10.193.2.1
   DHCP Server . . . . . . . . . . . : 10.88.1.95
   DHCPv6 IAID . . . . . . . . . . . : 383789698
   DHCPv6 Client DUID. . . . . . . . : 00-01-00-01-18-F8-51-A0-64-31-50-0E-14-76

   DNS Servers . . . . . . . . . . . : 10.20.255.36
                                       10.20.255.33
   NetBIOS over Tcpip. . . . . . . . : Enabled
```

Figure 1.2: Use of the command `ipconfig /all`

To detect an IP-address and determine the latency to the server use the `ping` command. `ping` will target either a webserver name to get the IP-address or target the IP-address directly without asking the DNS-server. This is shown on figure 1.3

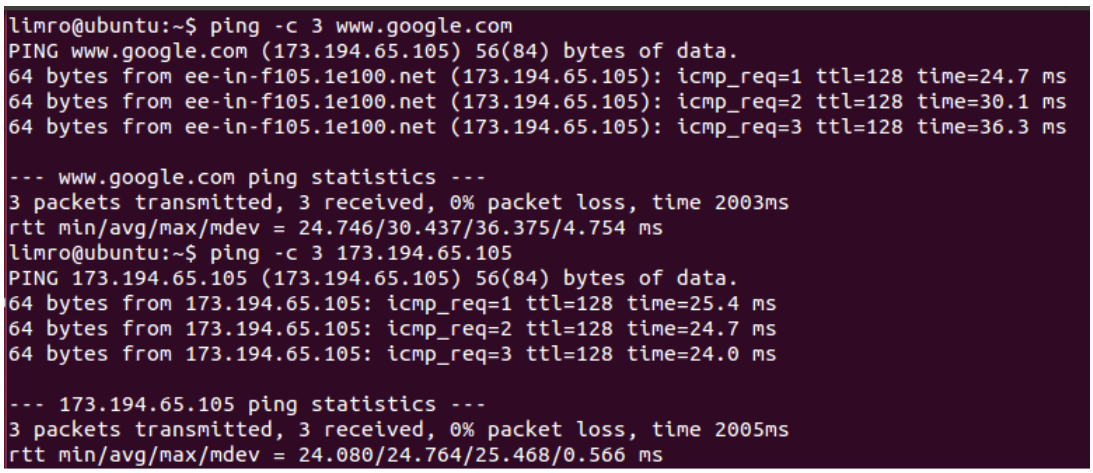


Figure 1.3: Use of the command `ping -c 3 www.google.com`

Before making a lookup at the DNS server the system will check local HLT file, located in */etc/hosts*, to see if any redirects or A records are listed. Redirections are created by typing the static IP-address followed by the new logical name as shown in code snippet below.

**Code snippet**

```
# Redirections
212.130.55.139 vejr #Resolve vejr to "www.dmi.dk"
159.20.6.38 nyhed #Resolve nyhed to "www.dr.dk"
157.55.46.241 mail #Resolve mail to "www.hotmail.com"
```

When requesting a webserver through a DNS, the root servers first redirect to the top-level domain server, TLD, which contains the '.com', '.net', '.dk' etc [**?**, p. 192]. From here the domain server can return the IP-address for the name server to which the client can connect. This will then (most likely) recursively or (less likely) iterative[2] return the specified IP-address to the client.

Each DNS server is responsible for looking up domains in nonoverlapping parts called 'zones'. A zone is implemented by a separate name server [**?**, p. 202-205].

When looking for a host name there can be multiple units with the same logical name. However it is possible to specify a name for a unit to unambiguously unique with a '.' (dot) at the end of the logical address which is called a fully qualified domain name (FQDN) [**?**].
Multiple computers could have the name *myComputer* but if looking for a computer on a network called *work.net.* there can only be one computer with the name *myComputer.work.net.*

[2]See section 1.2, Name resolution

## 1.2 Name resolution

When looking up an IP-address it can be resolved in two different ways. In iterative name resolution the root DNS server return the IP-address for the a DNS server which contains information of the requested country code. Afterwards the client will ask this DNS server for the IP-address and so on until the client the requested IP-address. This DNS query/response transaction type of resolution requires the client involvement for each request to a DNS server which leads to lower performance cost for the DNS servers [**?**, p. 205-209].

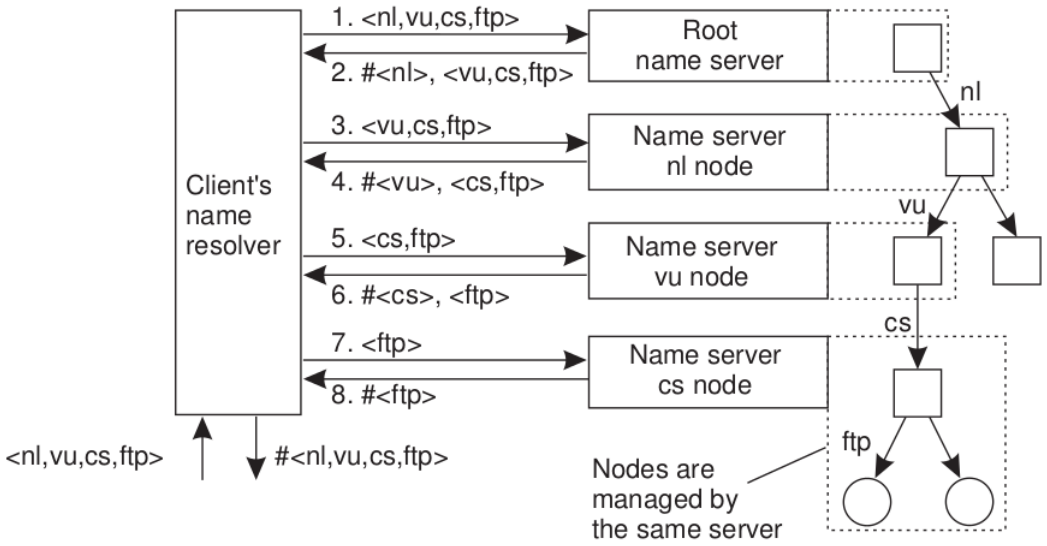


Figure 1.4: Iterative request for an IP-address

In recursive name resolution the root DNS server will ask the country coded DNS server for the IP-address of the website. The country coded DNS server ask the another DNS server which contains specific information of the domain and so on, until the website is identified. The IP-address is returned recursively to the root DNS server and back to the client. Caching the returned IP-address can lower the performance cost drastically, since a lookup is unnecessary for the same request next time.

Therefore the client is only involved when asking for and obtaining the IP-address which reduce performance cost for the client [**?**, p. 205-209].

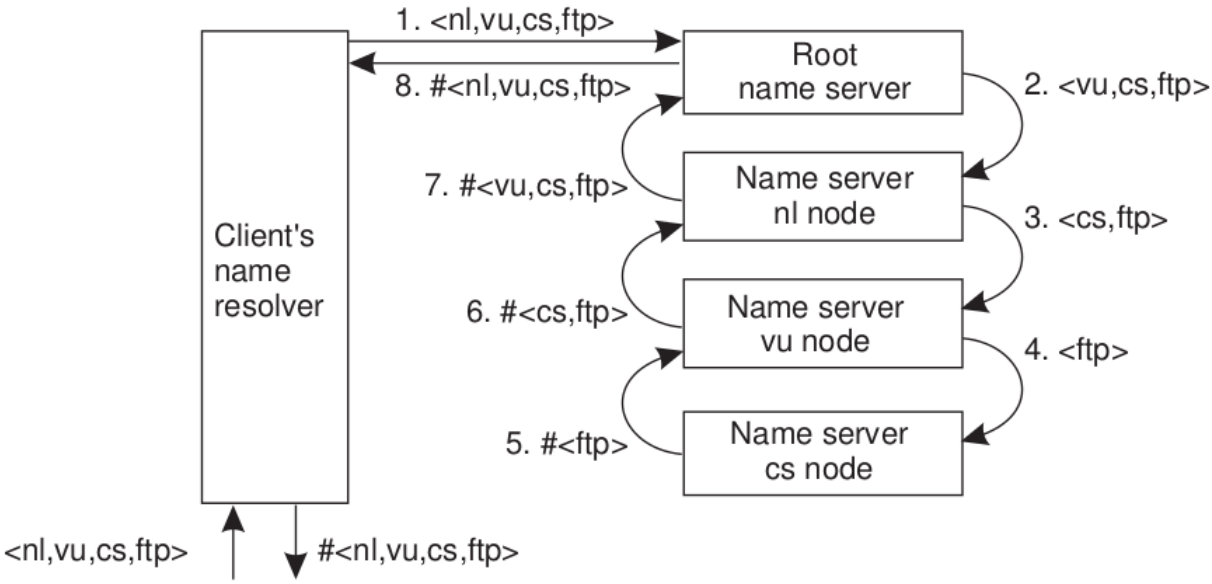


Figure 1.5: Iterative request for an IP-address

## 1.3 DNS security extentions

Due to security issues various governments, research organizations and others have developed a specification and associated protocol called DNS Security extensions (DNSSEC) which protects DNS querty/response transactions [**?**, p. 84].

Two main security threats exist for DNS in the context of query/response transactions. Attackers can

- cheat domain name servers and respond to DNS queries and alter DNS responses
- alter the DNS responses stored in caching name servers.

[**?**, p. 84]

DNSSEC was designed to protect the users from obtaining corrupted DNS data. It contains a set of extensions to DNS which provide origin authentication of DNS data, authenticated denial of existence, and data integrity. An administrative entity in the domain name system that provides DNS services for a group of domains is called a *zone*. If a DNS server uses DNSSEC it is marked as a *signed zone* [**?**].

A signed zone is a DNS server zone which includes a digital signature for all content it returns. It verifies that the underlying responses have requested resource records, special resource records that carry the digital signatures associated with the requested resources and it contains a DNSKey which include a public key which can verify the signature [**?**].

Using signed zones as authentication requires a DNSSEC-aware caching name server which start from a trusted public key stored within it self, a *trust anchor*. This establish a chain of trusted DNS servers with DNSSEC implemented to the public key of the zone. It is also possible to use a *trust anchor list* which contains a list over trusted anchors [**?**].

Currently the boundaries of the secure DNS is at the caching name servers as there is no end-to-end security. It would be preferable if an application could decide weather the requested data was corrupted or not. To accommodate this it it recommended to develop standardized formats or APIs that enabled caching name servers to communicate the security status (information about the outcome of the signature verification) to for example web or mail servers. It could also be met if the web servers performed the signature verification although it could lead to lower performance when small networked devices are used [**?**].

### 1.3.1 Censorship and misuse of DNS

In 1998 China started the Golden Shield Project because they feared inciting of internet users overthrowing the government, harm national unification, spread rumours e.g. [**?**] and began, with various techniques, to block the Chinese from websites and searches on the internet.
One of the methods used is by setting up *wiretap* that listens to everything sent out from Chinese internet service providers (ISPs) and DNS resolvers and sending back a fake DNS reply – a technique called *DNS injection* [**?**].

This is one example[3] where DNSSEC could prove valuable – reassuring the client the correct IP-address.

[3] Looking aside from other techniques the Golden Shield provide