DNS – 4

Introduction :

The DNS server, or Domain Name System, is a system that allows domain names (for example, www.example.com) to be translated into IP addresses (like 192.0.2.1), making it easier to browse the Internet. This is a protocol that usually works on port 53, using both UDP and TCP protocols. DNS servers are essential to ensure efficient communication between clients (computers, smartphones, etc.) and servers hosting websites. The importance of a DNS server lies in its ability to facilitate access to websites, by allowing users to connect to the servers hosting these sites simply by using their domain name. Without DNS servers, users would have to remember and enter IP addresses to access sites, which would be very inconvenient. In this text, we will discuss the technical aspects of how a DNS server works, the different queries and types of DNS servers, as well as the practical implementation of a DNS server within the framework of our project.

How a DNS server work ?

A DNS (Domain Name System) server is a domain name system that acts as an intermediary between clients and servers hosting websites. Its primary role is to translate domain names into IP addresses, allowing users to browse the Internet using easy-to-remember names rather than numeric IP addresses.

The domain name resolution process begins when the user enters a domain name in their browser. The client then sends a request to the local DNS server (usually provided by the Internet access provider) to obtain the IP address associated with the domain name. If the local DNS server does not cache this information, it queries DNS servers higher in the hierarchy until it gets the answer. The DNS hierarchy begins with the root servers, which are at the top of the structure and contain information about top-level domain (TLD) name servers, such as .com, .org, or .net. TLD servers contain information about second-level domain name servers, which are usually operated by individual companies or organizations.

There are two types of DNS queries: recursive and iterative. When a client sends a recursive query to a DNS server, it asks the server to provide the IP address associated with the domain name or forward the query to other servers until the answer is found. The DNS server then responds with the IP address or an error, if the domain name is not resolved. In an iterative query, the client asks the DNS server to provide the IP address associated with the domain name, without requiring the server to forward the query to other servers. If the DNS server does not know the IP address, it returns information about the higher DNS server in the hierarchy. The client can then send a new iterative request to this superior server.

There are several types of DNS servers, each with a specific role in the DNS hierarchy:

- Root Server: This is the first server consulted in the domain name resolution process. It contains information about TLD servers and directs requests to them.
- TLD Server: This server stores information about second-level domain name servers and is responsible for managing top-level domains (such as .com, .org, etc.).
- Local DNS server: This is the DNS server to which clients send their queries directly. It can be managed by the Internet service provider or a company.

In our project :

As part of this project, a DNS server was set up to respond to DNS queries and resolve domain names to IP addresses. The DNS server code was written in Python and uses the socket module to handle network communications. The socket module was used to create a UDP server and listen for DNS queries on port 53. The DNS server IP address was set to "127.0.0.9" and the fixed IP address to return for queries was set to "127.0.0.104" (the different addresses can change).

Socket initialization: The socket was initialized using socket.AF_INET (for IPv4) and socket.SOCK_DGRAM (for UDP). The socket has been bound to the IP address and port defined earlier.

Implementation of DNS resolution functions: Several functions have been implemented to handle the DNS resolution process, such as getflags(), getquestiondomain(), buildquestion(), rectobytes(), and buildresponse.()

-Main loop: The DNS server operates continuously, waiting for incoming queries, processing those queries, and returning appropriate responses.

-DNS server test: The DNS server can be tested by using tools like nslookup or dig to send DNS queries and check if the responses are correct and consistent with expectations.

Conclusion :

In this text, we have presented the concept and operation of DNS servers, which are essential for facilitating access to websites and ensuring smooth communication between clients and servers. We covered the technical aspects of how DNS servers work, including the domain name resolution process, types of DNS queries (recursive and iterative), and the different types of DNS servers (root servers, TLD servers, and local/local DNS servers). authoritative).

Additionally, we looked at a practical case of creating a DNS server using Python and the socket module. This project made it possible to implement the knowledge acquired on DNS servers and to better understand the DNS system.

DNS servers play a crucial role in the stable and fast functioning of the Internet, by allowing users to easily connect to servers hosting websites using domain names instead of numerical IP addresses. Without DNS servers, internet browsing would be much less convenient and potentially slower, as users would have to remember and type in IP addresses to access sites.