**Domain Name System (DNS)**

1. **Introduction to DNS**
   1. **Addressing**

* There are different levels of addresses are used in an interent using TCP/IP protocol.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Sl. No.** | **Parameter** | **Physical (or Link) Address** | **Logical (or IP) Address** | **Port Adress** |
| 1. | Definition | It is the address of a node as defined by its LAN or WAN | It is the address of a universal addressing system in which, each host can be identified uniquely, regardless of the underlying physical network. |  |
| 2. | Size | Ethernet uses a 6-byte (48-bit) physical address that is imprinted on the network interface card (NIC) card.  It is fixed. | The IP address in interent is currently a 4-byte (32-bit) address.  It may be changed. | A port address is a 16-bit address |
| 3. | Representation | Normally written as 12 Hexa-decimal digits, with every two bytes separated by a colon.  07:01:02:01:2C:4B | In IPv4, normally written as 4 decimal numbers, with each number representing 1 byte.  132.24.75.9 | It is represented by one decimal number  753 |
| 4. | Visibility | It is local & fixed for a node. | It is global, but not fixed for a host.  No two publicly addressed and visible hosts on the interent can have the same IP address. | It is local |

* Dd
  1. **Definition of DNS**
* To identify an entity, TCP/IP protocols use the IP address, which uniquely identiifes the connection of a host to the internet.
* These IP addresses are represented in dotted deciaml numbers.
* However, people prefer to use names instead of numeric addresses.Therefore, a system is required that can map a name to an address or an address to a name. This system is called DNS.
  1. **Name Space**
* The name spcace is the is most fundamental part of any name system, defines the ways that the names themselves are created.
* The name space tells us what form names may take, and provides the rules for how they are created. Most importantly, it specifies the architecture of the names—the internal structure of names themselves.
* To be unambiguous, the names assigned to machines must be carefully selected from a name space with complete control over the binding between the names and IP address.
* A name space that maps each address to a unique name can be organised in two ways.

1. **Flat name space:** Here, a name is assigned to an address. A name in this space is a sequence of characters without structure.
2. **Hierarchical name space:** Here, a name is assigned to an address. A name in this space is made of several parts with a structure. The first part can define the nature of the organisation, the second part can define the name of the organisation, the third part can define departments in the organisation and so on. In th
   1. **Domain Name Space**

* To have a hierachical name space, a domain name space was designed.
* In this design the names are defined in an inverted-tree structure with the root at the top.
* A name space that maps each address to a unique name can be organised in two ways.

1. **Label:** Each noe in the tree has a label, which is a string with a max. Of 63 characters. The root level is a null string. The tree can have only 128 levels : level 0 (rrot) to level 127.
2. **Domain Name:** Each node in the tree has a domain name. The domain names are always read from the node upto the root. A domain name is a sequence of levels separated by dots. If a level is terminated by a null string, it is called FQDN (Full Qualified Domain Name). An FQDN is a domain name that contains the full name of a host. It contains all levels, from the most specific to most general. If a label is not terminated by a null string, it is calleda partially qualified domain name. It is used when the name to be resolved belongs to the name site as the client.
3. **Domain:** A domain is a subtree of the domain name space. The name of the domain is the domain name of the node at the top of the subtree.
   1. **Name Servers**

* Servers are computers that run software to facilitate various kinds of network activities; the software packages that enable such activities are sometimes also called servers.
* The name servers are the servers that contains the information contained in the domain name space.
* The information of DNS, are stored in hierarchy of name servers in distributed manner, which uses the client–server model.
* Thee are not stored in a single server because it is inefficient as responding to requests from all over the world places a heavy load on the system. Also it is not reliable because any failure makes the data inaccessable.

- Why not centralize?

- Soingle point of failure - Traffic volume

- Distant centralize data base

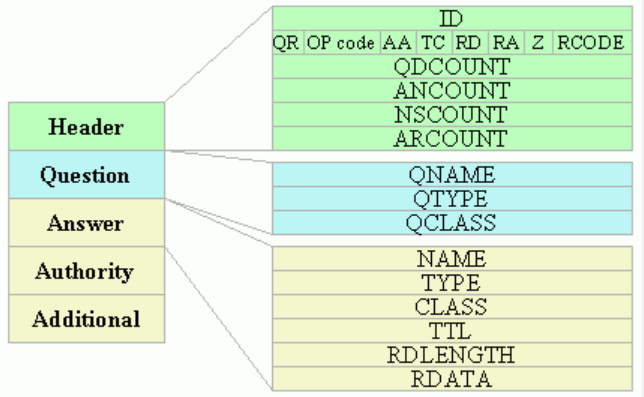
- Maintenace - does not scale

* No server has all name to IP address mapping. In other way we can say, the complete domain name hierarchy, can not be stored on a single server, it is divided into many servers.
* **Zone:** The responsibility of a name server is called a zone. A zone is defined as a contiguous part of the entire tree. If the server accepts the responsibility for a domain and does not divide the domain into smaller domains, the domain and zone refer the same thing.
  1. **Name Servers**
* Servers are computers that run software to facilitate various kinds of network activities; the software packages that enable such activities are sometimes also called servers.
* The name servers are the servers that contains the information contained in the domain name space.
* **Domain name servers** keep a record of your domain name and what IP address (server) it should point to.

1. **DNS Message Format** 
   1. **Overview**

* DNS is used to map **human-readable domain names** (such as example.com) to **machine-readable IP addresses** (like 93.184.216.34).
* To use DNS, we send a query to a DNS server. This query contains the domain name we’re looking up. The DNS server tries to look up that domain name’s IP address in its internal data store. If it finds it, it returns it. If it can’t find it, the server will forward the query to a different DNS server, which will repeat this process until the IP is found.
* DNS messages are usually sent using the **UDP protocol.**
* The DNS protocol uses a common message format for all exchanges between client and server or between servers. The DNS messages are encapsulated over UDP or TCP using the "well-known port number" 53. DNS uses UDP for message smaller than 512 bytes (common requests and responses). DNS uses TCP for bigger exchange (i.e. zone transfer).
  1. **DNS Message**
* There are two types of amessages: **queries & responses/replies** and they both have the **same format**.
* Each **DNS message** consists of a **header** and four sections: **question, answer, authority & additional**.
* The header field “flags” controls the content of these four sections.

|  |  |
| --- | --- |
| **Header** | => describe the type of message and provide  important information about it. |
| **Question** | => The question (s) for the name server |
| **Answer** | => Resource Records (RRs) answering the question |
| **Authority** | => RRs pointing toward an authority |
| **Additional** | => RRs holding additional information |



* **DNS Message Header Format :** The header is the most important part of any message, since it is where critical control fields are carried. In DNS messages, the Header section carries several key control flags, and is also where we find out which of the other sections are even being used in the message. Examining the Header can help us understand several of the nuances of how messaging works in DNS.
* The **format of the Header section** used in all DNS messages is described in detail below, where fields are used differently by the client and server in an exchange,

|  |  |  |  |
| --- | --- | --- | --- |
| **Sl. No.** | **Field name** | **Size (bits)** | **Description** |
| 1 | **ID** | 16 | **Identifier:** A 16-bit identification field generated by the device that creates the DNS query. It is copied by the server into the response, so it can be used by that device to match that query to the corresponding reply received from a DNS server. |
| 2 | **QR** | 1 | **Query/Response Flag:** Differentiates between queries and responses. Set to 0 when the query is generated; changed to 1 when that query is changed to a response by a replying server |
| 3 | **Opcode** | 4 | It specifies the type of query the message is carrying. This message is set by the creater of the query and copied unchanged into the response.  The possibilities are:  0: Standard query  1: Inverse query  2: Server status request  3-15: Reserved for future use |
| 4 | **AA** | 1 | **Authoritative Answer Flag:** this bit is available only for DNS replies. It tells whether the server is authoritative for the requested domain name.  AA = 1; the DNS server is authoritative on the domain name  AA = 0; the DNS server is not authoritative on the domain name |
| 5 | **TC** | 1 | **Truncation Flag:** tells whether the message is truncated or not. A DNS message is truncated when it can not fit in a single UDP datagram with a maximum size of 512 Bytes. |
| 6 | **RD** | 1 | **Recursion Desired:** When set in a query, requests that the server receiving the query attempt to answer the query recursively, if the server supports recursive resolution. The value of this bit is not changed in the response.  => expresses the querying host’s desire to make a recursive query or not.  If set to 1, then it means “the querying host desires a recursive query”. This flag is copied in the response too. |
| 7 | **RA** | 1 | **Recursion Available:** Set to 1 or cleared to 0 in a response to indicate whether the server creating the response supports recursive queries. This can then be noted by the device that sent the query for future use.  =>appears only in DNS responses. It tells whether the name server that receives the query can do recursive queries or not  If set to 1, this flag says “I can do recursive queries”,  if set to 0, this flag says “Sorry dude I can not do recursive queries”. |
| 8 | **Z** | 3 | **Zero:** Three reserved bits set to zero. This flag is developed for future use. |
| 9 | **RCode** | 4 | **Response Code:** This field is used to indicate if the query was answered successfully or if some sort of error occured.  This flag is only valid in a DNS reply. It tells whether the response contains errors or not.  if RCODE = 0, then there are no errors,  else: there is an error |
| 10 | **QDCount** | 16 | **Question Count:** Specifies the number of questions in the Question section of the message. |
| 11 | **ANCount** | 16 | **Answer Record Count:** Specifies the number of resource records in the Answer section of the message. |
| 12 | **NSCount** | 16 | **Authority Record Count:** Specifies the number of resource records in the Authority section of the message. (“NS” stands for “name server”) |
| 13 | **ARCount** | 16 | **Additional Record Count:** Specifies the number of resource records in the Additional section of the message. |
|  |  |  |  |

* The **question section** has the format:

|  |  |
| --- | --- |
| **QNAME** | **QNAME:** This contains the URL who’s IP address we wish to find. It is encoded as a series of ‘labels’. Each label corresponds to a section of the URL. The URL example.com contains two sections, example, and com. It is the domain name encoded in labels. |
| **QTYPE** | **=>** determines the type of the query. |
| **QCLASS** | **=>** determines the class of the query. Usually QCLASS has the value “IN” to mean “INternet”. |

* The **answer section** has a resource record format:

|  |  |
| --- | --- |
| **NAME** | => has the same format as QNAME |
| **TYPE** | => has the same format as QTYPE |
| **CLASS** | => has the same format as QCLASS |
| **TTL** | => **Time To Live :** describes how much time – in seconds- can this record be cached before it must be discarded |
| **RDLENGTH** | => describes the length of the RDATA field |
| **RDATA** | => contains the resource itself. For example, if the RR is of type A, then RDATA is an IPv4 address. If the RR is of type NS, then RDATA is a name server alias hostname. |

* 1. **How a domain name is represented inside the DNS packet**
* To understand how QNAME is represented (and also NAME and RDATA), we need to understand the way a domain name is represented in a DNS packet.
* **A domain name is represented in the form of labels separated by dots (.):**

**{label}.{label}.{label}**

* A label can be of two types:

1. Data label
2. Compression label

|  |  |
| --- | --- |
| **Data label** | **Compression label** |
| A **data label** is composed of :   * **Label length:** a byte that describes the length of the current label (just before the dot). The value ranges from 0 to 63. * **the bytes of the current label.** Their maximum size is 63 bytes. | A DNS reply message can contain the same domain name multiple times. This repetition is a waste of bits in the message. A compression technique can be used to reduce the number used bits and replace the repeated domain name by a pointer. For more visit  <http://www.keyboardbanger.com/dns-message-format-name-compression/>  <https://routley.io/tech/2017/12/28/hand-writing-dns-messages.html> |
| For example, the domaine name “www.microsoft.com” has three labels and is encoded in the DNS message in this format:  3 ‘w’ ‘w’ ‘w’ 9 ‘m’ ‘i’ ‘c’ ‘r’ ‘o’ ‘s’ ‘o’ ‘f’ ‘t’ 3 ‘c’ ‘o’ ‘m’ 0  Notice the trailing “0”. This is called the label of length 0 and means the root label. Remember that all DNS domain names end with a root domain; the dot. | <https://www.cisco.com/c/en/us/about/security-center/dns-best-practices.html> |
| This domain name encoding is found in the following fields: QNAME, NAME and RDATA |  |

* V

1. **Resource Records (RRs)** 
   1. **Overview**

* A DNS record is a database record used to map a URL to an IP address.
* DNS records are stored in DNS servers and work to help users connect their websites to the outside world. When the URL is entered and searched in the browser, that URL is forwarded to the DNS servers and then directed to the specific Web server. This Web server then serves the queried website outlined in the URL or directs the user to an email server that manages the incoming mail.
* DNS stands for Domain Name System, which is the largest digital database in the world, containing information about every web site on the internet. Every web site online has an IP address that is its actual internet location, and this number is used to locate the web site within the database.
* The data that tells the web server how to respond to your input is known as the **DNS records, or zone files**. These records play a vital role in the functionality of the internet, and any aspiring internet technology expert should learn the following facts about DNS records and how they are used.
* The DNS records tool retrieves the domain name records for a specified domai
  1. **Overview**
* **DNS Records Explained:** DNS records are basically mapping files that tell the DNS server which IP address each domain is associated with, and how to handle requests sent to each domain. When someone visits a web site, a request is sent to the DNS server and then forwarded to the web server provided by a web hosting company, which contain the data contained on the site.
* Various strings of letters are used as commands that dictate the actions of the DNS server, and these strings of commands are called **DNS syntax.** Some DNS records syntax that are commonly used in nearly all DNS record configurations are A, AAAA, CNAME, MX, PTR, NS, SOA, SRV, TXT, and NAPTR.

|  |  |  |  |
| --- | --- | --- | --- |
| **Sl. No.** | **DNS Record** | **Full Form** | **Use** |
| 1 | **A** | Address | It is the most basic type of DNS record and are **used to point a domain or subdomain to an IP address.** |
| 2 | **CNAME** | Canonical Name | It **forwards one domain or subdomain to another domain**, does NOT provide an IP address.  It is used in **lieu of an A record**, when a domain or subdomain is an **alias of another domain**. Imagine a scavenger hunt where each clue points to another clue, and the final clue points to the treasure. A domain with a CNAME record is like a clue which can point you to another clue (another domain with a CNAME record) or to the treasure (a domain with an A record). For example, suppose www.example.com has a CNAME record with a value of ‘example.com’ (without the ‘www’). This means when a DNS server hits the DNS records for www.example.com, it actually triggers another DNS lookup to example.com, returning example.com’s IP address. In this case we would say that example.com is the canonical name (or true name) of blog.example.com. All CNAME records must point to a domain, never to an IP address. |
| 3 | **MX** | Mail Exchange | It **directs email to a mail server.** The MX record indicates how email messages should be routed in accordance with Simple Mail Transfer Protocol (SMTP, the standard protocol for all email.) |
| 4 | **NS** | Name Server | It is used to specify the authoritative nameservers for the domain.  This record **indicates which DNS server is authoritative for that domain** (which server contains the actual DNS records). A domain will often have multiple NS records which can indicate primary and backup name servers for that domain. |
| 5 | **PTR** | Pointer | It is used to map an Ipv4 address to the CNAME on the host.  Pointer records (PTR) are used for reverse lookups. For example, to make 192.168.0.1 resolve to www.yourdomain.com, the record would look like: 1.0.168.192.in-addr.arpa PTR www.yourdomain.com. |