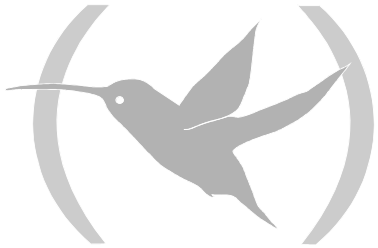
Teldat Router

DNS Client

Doc. *DM723-I* Rev. *10.00*
March, 2003

INDEX

Chapter 1
Domain Name System



1. Introduction

The Domain Name System, better known as DNS, is a standard protocol described in the RFCs 1034 and 1035. This permits network users to use simple hierarchical names in order to refer to other devices. In this way, the user can obviate the IP address associated to the device and refer to it with a name that is easier to remember. Additionally, this also simplifies changing the IP address of a device: address changes should only be notified to the DNS server in charge of this device as they are transparent to the user who continues to refer to the device with the same name.

DNS is an application protocol and uses both UDP as well as TCP. The clients send the DNS servers their queries through UDP in order to speed up communication and only use TCP in cases where a truncated response arrives.

The DNS uses the concept of *distributed names space*. The symbolic names are grouped in *authority zones* or more commonly *zones*. In each of these zones, one or more devices have the task of maintaining a database of symbolic names and IP addresses and of providing server function for the clients who wish to translate the symbolic names to IP addresses. These local *name servers* logically interconnect in a *domains* hierarchic tree. Each zone contains a part of the tree or *subtree* and the names of this zone are administered independently to those of other zones. The authority over zones is delegated in the name servers. At the points where a domain containing a subtree which falls into a different zone, we say that the name servers with authority over the superior domain *delegates authority* to the name servers with authority over the subdomains. The name servers can also delegate authority within themselves; in this case the name space is still divided into zones but the authority of both is executed by the same server.

The result of this scheme is as follows:

- Instead of having a central server for the database, the work implicated is divided between the Hosts in the length and breadth of the names space.

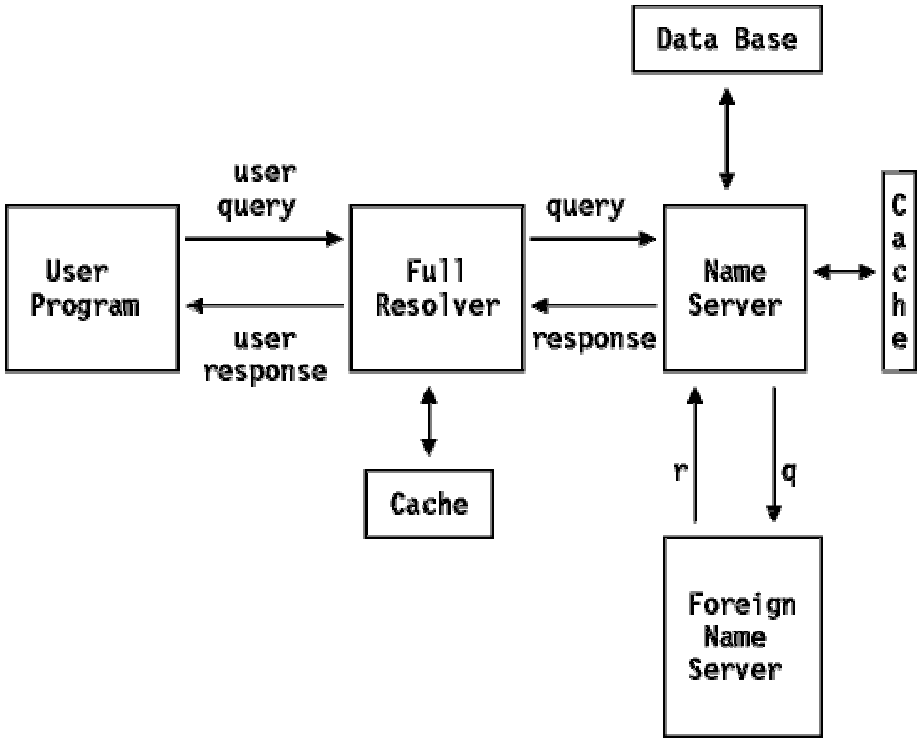
- The authority to create and change symbolic names of the host and the responsibility to maintain a database for these corresponds to the proprietor organization of the zone containing these.
- From the user point of view, there is only one database that deals with the resolution of addresses.

2. Resolution of domains

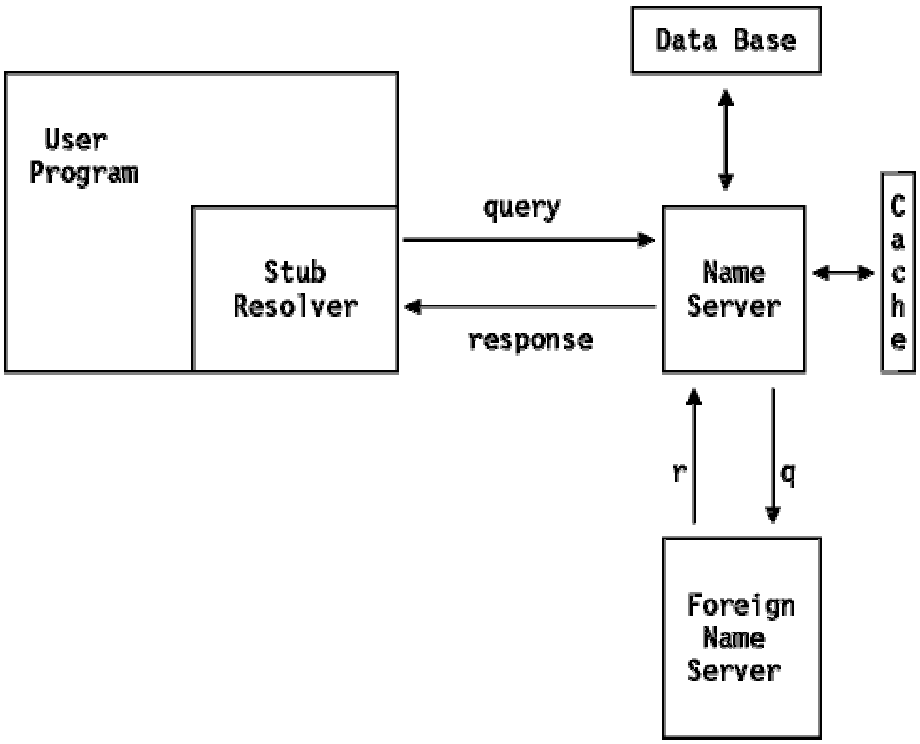
The resolution of domain names is a client-server process. The most common DNS client carries out petitions to the name servers to obtain the IP address associated to a domain name. These types of petitions are known as standard petitions. Apart from these petitions, there are also inverse petitions in order to obtain the name of a domain from the IP address and the generic petitions in order to obtain some additional data on a domain.

Two types of resolvers can be distinguished:

- Full resolver: the resolver really carries out the necessary petitions to retrieve the information required. This analyzes the responses received from the server in order to see if it has received the response to the executed petition or a delegation to another server. In this latter case, new petitions are deployed until the required response is achieved.
- Simple resolver (stub resolver): delegates the petition resolution in a full resolver. The resolver has the IP address of a series of servers capable of carrying out the complete petition process configured: deploys the required petition and waits for a response to be received to the petition. This does not admit responses which include delegations in other servers.



Operational scheme of the full resolver



Operational scheme of the stub resolver

2.1. Domain names resolver functionality

The petitions over domain names can be of two types: *recursive* or *iterative* (also known as *non-recursive*). One flag bit in the query specifies if the client wants a recursive query and one flag bit in the response indicates if the server supports recursive petitions. The difference between a recursive query and an iterative appears when the server receives a request to which it cannot give a complete response. A recursive query demands that the server in turn deploys a query to determine the looked for information and subsequently return this to the client. An iterative query implies that the name server must return the information that it has as well as a list of additional servers with which the client may contact in order to complete his query.

The domain name responses may be of two types: *authoritive* and *non-authoritive*. A flag bit in the response indicates which response type this is. When a name server receives a query for a domain in an area where it has authority, it returns a response with an active flag bit. If it does not have authority in this zone, the reaction depends on whether the recursive flag is active or not.

If the recursive flag is active and the server supports this, the query is directed to another name server. This will be the server with authority over the query domain or one of the root name servers. If the second server does not return an authoritive response, the process is repeated.

When a server (or a full resolver) receives a response, this is cached in order to improve the performance of the repeated queries. The cache entry is stored with a maximum specified time in the response in the 32 bits *TTL* ("*time to live*") field. The typical value here is 172,800 seconds i.e. two days.

If the recursive flag is not active or the server does not support recursive queries, the information that it has in its cache is returned together with a list of servers capable of giving authoritive responses.

2.2. Functionality of the domain name server

Each name server has *authority* for zero or more zones. There are three types of name servers:

- primary: a primary name server loads from the disk the information from a zone and has authority over this.
- secondary: a secondary name server has authority over a zone but retrieves the information from this zone from a primary server using a process known as *zone transfer*. For this to remain synchronized, the secondary name servers regularly query the primary servers (usually every three hours) and re-execute the zone transfer if the primary has been updated. A name server can operate as a primary or a secondary server for multiple domains or act as a primary for some and as secondary for others. A primary or secondary server carries out all the functions of a cache server.
- cache: a name server that does not have authority for any zone is known as a cache server. This retrieves all its data from the primary or secondary servers. This, at the least, requires an NS record (Name Server) in order to appoint a server from which it can initially retrieve the information.

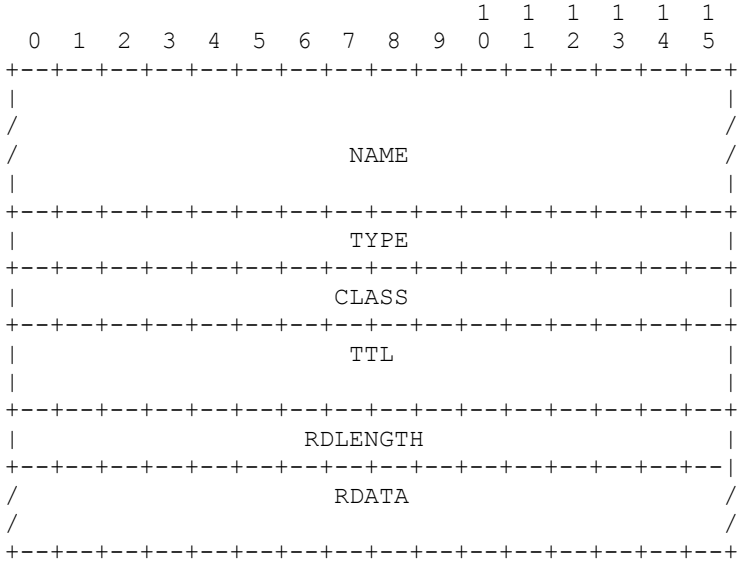
When a domain is registered in the root and establishes a separate authority zone, the following rules are applied:

- the domain must be registered in the root administrator.
- There must be an administrator identifier for the domain.
- There must be at least two name servers with authority for the zone so they are accessible both inside and outside the domain in order to avoid any possible weak point.

It is also recommended that the name servers which delegate authority also apply these rules as they are responsible for the behavior of the delegated name servers.

2.3. DNS resource records

The database distributed from the DNS is composed of *RRs* ("*resource records*"). These provide mapping between the domain names and the *network objects*. The most common network objects are the hosts' addresses, however the DNS is designed to take a wide range of distinct objects. The general resource record format is as follows:



NAME

This is the domain name to which the record refers. The DNS rules are very general as regards the composition of the domain names. A domain name consists of a series of labels made up of alphanumerical characters or hyphens, each label having a length of between 1 and 63 characters beginning with an alphabetical character. The domain names are usually represented by separating the labels through a period. In the messages, each label includes a byte at the beginning indicating the length of this label. All the names end with a zero length label indicating the root domain. The domain names are not sensitive to upper or lower case.

CLASS

Identifies the protocol family. Class 1 (IN) is used for Internet.

TYPE

Identifies the type of record resource. Type 1 (A) identifies a host address.

TTL

This is the *"time-to-live"* or the time in seconds that the record will be valid in the name server cache. This is stored in the DNS as a value of 32 bits without signed. 86400 (one day) is the typical value for records that note an IP address.

RDLENGTH

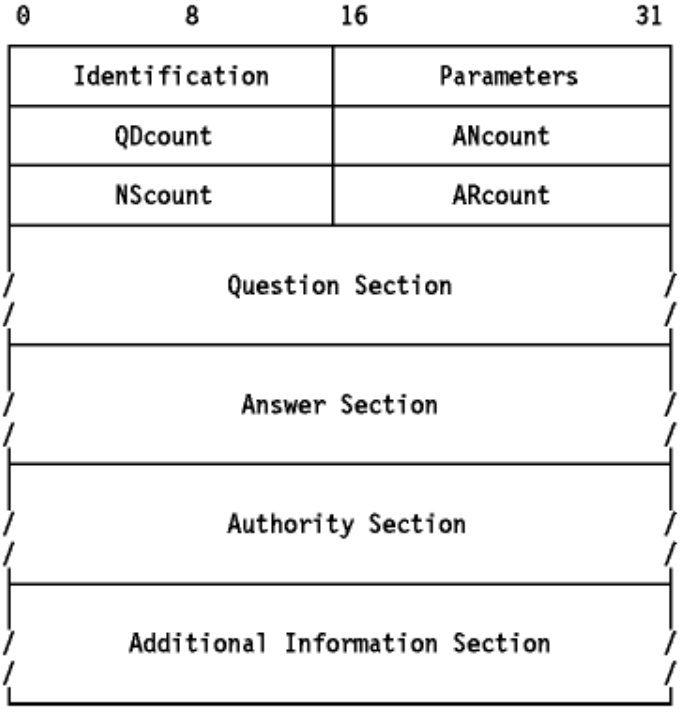
Length of the data part.

RDATA

Record data. Depending on the type and class of the record, the data varies. For example, if this is type A and the class IN, the data will be four bytes indicating an IP address.

2.4. DNS Messages

All DNS messages use a single format:



The resolver sends a frame to the name server. Only the header and the question section are used for the query. The query responses or re-transmissions use the same frame but fill out more sections (the "answer/authority/additional sections).

a) Header format

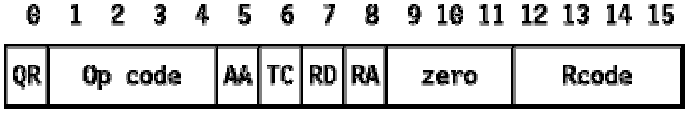
The header section must always appear and has a set length of 12 bytes. The other sections have a variable length.

- ***ID (Identification)***

A 16 bits identifier assigned by the resolver. This identifier is copied in the response corresponding to the name server and can be used to differentiate between responses when multiple queries occur.

- ***Parameters***

A 16 bits field with the following format:



QR

Flag indicating query (0) or response (1).

Op code

4 bits field specifying the type of query:

- 0: standard query (QUERY)
- 1: inverse query (IQUERY)
- 2: request for server status (STATUS)

The rest of the values are reserved for future use.

AA

Authoritive Answer Flag. If this is active in a response, this specifies that the name server responding has authority for the domain name sent in the query.

TC

TrunCation Flag. Active if the message is longer than that permitted in the channel.

RD

Recursion Desired Flag. This bit indicates to the name server that recursive resolution is required. The bit is copied into the response.

RA

Recursion Available Flag. Indicates if the name server supports recursive resolution.

zero

3 bits reserved for future use. This must be zero.

Rcode

4 bits response code. The possible values are:

- 0: No error condition.
- 1: Format error. The name server was unable to interpret the message.
- 2: Server Failure. The message was not processed due to a problem with the name server.
- 3: Name Error. The domain name in the query does not exist. This is only valid if the AA bit is active in the response.
- 4: Not Implemented. The name server does not support the requested kind of query.
- 5: Refused. The name server refuses to respond due to policy reasons.

The rest of the values are reserved for future use.

- ***QDcount***

  An unsigned 16 bit integer specifying the number of entries in the question section.

- ***ANcount***

  An unsigned 16 bit integer specifying the number of RRs in the answer section.

- ***NScount***

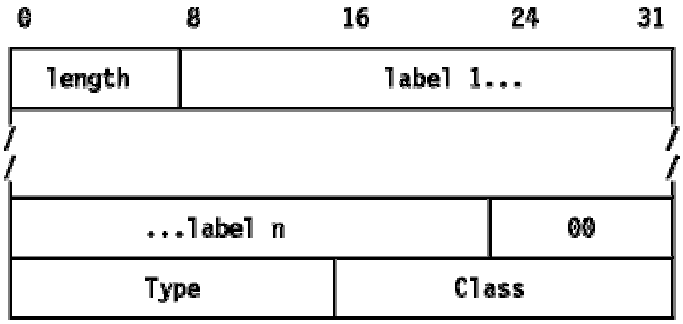
  An unsigned 16 bit integer specifying the number of RRs in the authority section.

- ***ARcount***

  An unsigned 16 bit integer specifying the number of RRs in the additional records section.

b) "Question" Section

The following section contains the queries to the name server. This contains Qdcount (usually 1) entries, each one with the following format:



All the fields are aligned by bytes. The alignment of the "Type" field at 4 bytes is an example and is not mandatory in the format.

- ***length***

  One byte indicating the length of the next label.

- ***label***

  A domain name element. The domain name is stored as a series of labels with variable lengths, each preceded by a "length" field.

- ***00***

  A 00 value indicates the end of the domain and represents the null label of the root domain.

- ***Type***

  2 bytes specifying the type of query. For address queries, the value 'A' (1) is used.

- ***Class***

  2 bytes specifying the class of query. For Internet queries, the 'IN' (1) value is used.

c) "Answer", "Authority" and "Additional Resource" Sections

These three sections contain a variable number of resource records. The number is specified in the field corresponding to the header. The resources records format is further discussed in section 2.3.

d) Message compression

With the aim of reducing the size of the message, a compression scheme is used to eliminate the repetition of the domain names in the various RRs. Any duplicated domain or list of labels is replaced with a pointer in the previous occurrence. The pointer takes the form of a two byte field:

The first 2 bits distinguish the pointer from a normal label, which restricts the latter to a length of 63 bytes plus the length byte.

The 'offset' field specifies an offset from the start of the message. A zero 'offset' specifies the first byte of the header ID field.

e) ***Transport***

The DNS messages are transmitted as datagrams (UDP) or over a channel (TCP). In both cases, port 53 (server source port) is used as the DNS petitions destination port.

A DNS resolver or server sending a query that does not suppose a zone transfer *must* first send a UDP query. If the response 'answer' section is split and the requested supports TCP, you should try again using TCP. It is preferable to use UDP instead of TCP for queries because UDP has a lower overhead factor and its use is essential for a heavily loaded server. Truncating messages is not usually a problem given that the actual contents of the DNS database as 15 records can be typically sent in a datagram. However this could change when adding new types of records to the DNS.

TCP must be used for zone transfer activities due to the fact that UDP is restricted to 512 bytes and this will always be inadequate for a zone transfer.

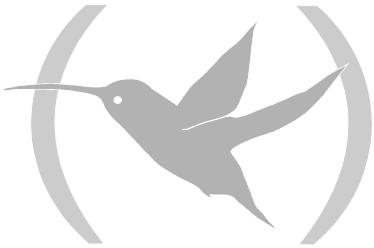
The name servers must support both types of transport.

3. References

RFC 1034
DOMAIN NAMES – CONCEPTS AND FACILITIES, P. Mockapetris, November 1987

RFC 1035
DOMAIN NAMES – IMPLEMENTATION AND SPECIFICATION, P. Mockapetris, November 1987

Chapter 2
Configuring DNS



1. Configuring DNS

To access the DNS client parameters configuration menu, enter the **FEATURE DNS** command from the configuration menu.

```
*P 4
User Configuration
Config>FEATURE DNS
-- DNS resolver user configuration --
DNS config>
```

The options for this configuration menu are as follows:

```
DNS config>?
LIST                    Displays the DNS configuration
N-RETRANSMISSIONS       Maximum number of DNS petition transmissions
NO
SERVER                  DNS name server to which the petitions are carried out
SOURCE-PORT             UDP port used as source in the DNS petitions
T-RETRANSMISSIONS       Time between DNS petition retransmissions
EXIT
```

1.1. LIST

Displays the DNS configuration.

```
DNS config>LIST ?
ALL                     Displays the whole of the DNS configuration
N-RETRANSMISSIONS       Displays the maximum number of retransmissions
SERVERS                 Displays the IP addresses for the configured DNS servers
SOURCE-PORT             Displays the UDP port used as source
T-RETRANSMISSIONS       Displays the time between DNS petition retransmissions
DNS config>
```

a) LIST ALL

Displays the whole of the DNS configuration.

```
DNS config>LIST ALL
Source port: 2658
Number of retransmissions: 5
Time between retransmissions: 1 sec
Name servers:
               172.24.0.6
               172.24.0.13
DNS config>
```

"Source port", UDP port used as source in the DNS petitions.

"Number of retransmissions", maximum number of transmissions for a DNS petition.

"Time between retransmissions", time between DNS petition retransmissions.

"Name servers", IP addresses of the configured DNS servers.

b) LIST N-RETRANSMISSIONS

Displays the maximum number of transmissions for a DNS petition.

```
DNS config>LIST N-RETRANSMISSIONS
Number of retransmissions: 5
DNS config>
```

c) LIST SERVERS

Displays the IP addresses for the configured DNS servers.

```
DNS config>LIST SERVERS
Name servers:
               172.24.0.6
               172.24.0.13
DNS config>
```

d) LIST SOURCE-PORT

Displays the UDP port used as source in the DNS petitions.

```
DNS config>LIST SOURCE-PORT
Source port: 2658
DNS config>
```

e) LIST T-RETRANSMISSIONS

Displays the time between DNS petition retransmissions.

```
DNS config>LIST T-RETRANSMISSIONS
Time between retransmissions: 1 sec
DNS config>
```

1.2. N-RETRANSMISSIONS

Configures the maximum number of DNS petition transmissions.

```
DNS config>N-RETRANSMISSIONS
Maximum number of retransmissions (1-10)[5]? 3
DNS config>
```

1.3. NO

a) NO SERVER

Deletes a configured DNS name server.

```
DNS config>NO SERVER
Name server ip address to delete [0.0.0.0]? 192.68.63.56
DNS config>NO SERVER
Name server ip address to delete [0.0.0.0]? 1.2.3.4
Name server not found
DNS config>
```

1.4. SERVER

Adds a DNS name server to which the petitions are carried out. In cases where the maximum number of possible servers have already been configured (currently 3), an error message is produced.

```
DNS config>SERVER
Name server ip address [0.0.0.0]? 192.68.63.197
DNS config>
DNS config>SERVER
Maximum number of name servers already configured
DNS config>
```

1.5. SOURCE-PORT

Configures the UDP port used as source in the DNS petitions.

```
DNS config>SOURCE-PORT
Source port[2658]? 2345
DNS config>
```

1.6. T-RETRANSMISSIONS

Configures the time between DNS petition retransmissions.

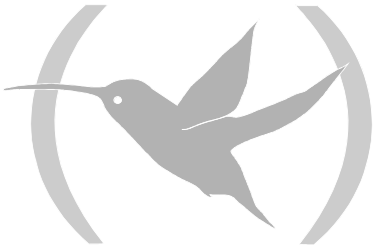
```
DNS config>T-RETRANSMISSIONS
Time between retransmissions (sec) (1-5 sg)[1]? 5
DNS config>
```

1.7. EXIT

Exits the DNS configuration menu.

```
DNS config>EXIT
Config>
```

Chapter 3
Monitoring DNS



1. Monitoring DNS

To access the DNS client parameters monitoring menu, enter the **FEATURE DNS** command from the global monitoring menu.

```
*P 3

+FEATURE DNS
-- DNS resolver user monitoring --
DNS>
```

The options of this monitoring menu are as follows:

```
DNS>?
LIST
LOOKUP
EXIT
DNS>
```

1.1. LIST

Displays the distinct DNS operating parameters.

```
DNS>LIST ?
MEMORY-USED
LOOKUP-RESULTS
DNS>
```

a) LIST MEMORY-USED

Displays the memory resources in use for the DNS client.

```
DNS>LIST MEMORY-USED

Memory in use: 0
DNS>
```

b) LIST LOOKUP-RESULTS

Displays the results of the last 10 DNS petitions carried out from monitoring (using the LOOKUP command).

```
DNS>LIST LOOKUP-RESULTS

Last DNS Lookup Queries
-----------------------
www.elmundo.es: IP addresses
    212.80.177.133
www.microsoft.com: Maximum number of retries reached
DNS>
```

1.2. LOOKUP

Carries out a DNS petition for a specified name. When the address has been resolved, this appears on the screen. While the query is being resolved, the console is blocked. If you wish to stop the DNS query before it has been resolved, press Ctrl+C.

```
DNS>LOOKUP
Name  []? www.teldat.es
Press Ctrl+C to stop the query

172.24.0.56
DNS>
```

In cases where the petition does not successfully complete, this is indicated with a message signifying the type of error that has been produced.

```
DNS>LOOKUP
Name  []? www.microsoft.com
Press Ctrl+C to stop the query


DNS Error: Maximum number of retries reached
DNS>
```

1.3. EXIT

Exits the DNS client monitoring menu.

```
DNS>EXIT
+
```