DNS 协议报文(RFC1035)

一、域名和资源记录的定义

1、Name space definitions

2、资源记录定义(RR definitions)

2.1 格式

后面分析报文的时候详细解释。

2.2 类型值(TYPE values)

类型主要用在资源记录中，注意下面的值是 QTYPE 的一个子集。

类型	值和含义
A	1 a host address
NS	2 an authoritative name server
MD	3 a mail destination (Obsolete - use MX)
MF	4 a mail forwarder (Obsolete - use MX)
CNAME	5 the canonical name for an alias
SOA	6 marks the start of a zone of authority
MB	7 a mailbox domain name (EXPERIMENTAL)
MG	8 a mail group member (EXPERIMENTAL)
MR	9 a mail rename domain name (EXPERIMENTAL)
NULL	10 a null RR (EXPERIMENTAL)
WKS	11 a well known service description
PTR	12 a domain name pointer
HINFO	13 host information
MINFO	14 mailbox or mail list information
MX	15 mail exchange
TXT	16 text strings

2.3 查询类型(QTYPE values)

查询类型出现在问题字段中，查询类型是类型的一个超集，所有的类型都是可用的查询类型，其他查询类型如下：

AXFR	252 A request for a transfer of an entire zone
MAILB	253 A request for mailbox-related records (MB, MG or MR)
MAILA	254 A request for mail agent RRs (Obsolete - see MX)
*	255 A request for all records

2.4 类(CLASS values)

IN	1 the Internet

CS 2 the CSNET class (Obsolete - used only for examples in some obsolete RFCs)

CH 3 the CHAOS class

HS 4 Hesiod [Dyer 87]

2.5 查询类(QCLASS values)

查询类是类的一个超集

* 255 any class

3、Standard RRs

3.1 CNAME RDATA format

3.2 HINFO RDATA format

3.3 MB RDATA format (EXPERIMENTAL)

3.4 MD RDATA format (Obsolete)

3.5 MF RDATA format (Obsolete)

3.6 MG RDATA format (EXPERIMENTAL)

3.7 MINFO RDATA format (EXPERIMENTAL)

3.8 MR RDATA format (EXPERIMENTAL)

3.9 MX RDATA format

3.10 NULL RDATA format (EXPERIMENTAL)

3.11 NS RDATA format

3.12 PTR RDATA format

3.13 SOA RDATA format

3.14 TXT RDATA format

4、ARPA Internet specific RRs

4.1 A RDATA format

4.2 WKS RDATA format

5、IN-ADDR.ARPA domain

6、Defining new types, classes, and special namespaces

二、报文

1、报文格式(Format)

dns 请求和应答都是用相同的报文格式，分成 5 个段（有的报文段在不同的情况下可能为空），如下：

```
+---------------------+
|     Header     | 报文头
+---------------------+
```

```
|     Question     | 查询的问题
+------------------+
|     Answer       | 应答
+------------------+
|    Authority     | 授权应答
+------------------+
|    Additional    | 附加信息
+------------------+
```

Header 段是必须存在的，它定义了报文是请求还是应答，也定义了其他段是否需要存在，以及是标准查询还是其他。

Question 段描述了查询的问题，包括查询类型(QTYPE)，查询类(QCLASS)，以及查询的域名(QNAME)。剩下的 3 个段包含相同的格式:一系列可能为空的资源记录(RRs)。Answer 段包含回答问题的 RRs；授权段包含授权域名服务器的 RRs；附加段包含和请求相关的，但是不是必须回答的 RRs。

1.1 Header 的格式

报文头包含如下字段：

```
                    1 1 1 1 1 1
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5
+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
|              ID                 |
+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
|QR|  Opcode  |AA|TC|RD|RA|   Z    |   RCODE   |
+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
|            QDCOUNT              |
+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
|            ANCOUNT              |
+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
|            NSCOUNT              |
+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
|            ARCOUNT              |
+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
```

各字段分别解释如下：

ID　请求客户端设置的 16 位标示，服务器给出应答的时候会带相同的标示字段回来，这样请求客户端就可以区分不同的请求应答了。

QR　1 个比特位用来区分是请求（0）还是应答（1）。

OPCODE 4 个比特位用来设置查询的种类，应答的时候会带相同值，可用的值如下：

0　　标准查询 (QUERY)

1　　反向查询 (IQUERY)

2　　服务器状态查询 (STATUS)

3-15　　保留值，暂时未使用

AA　授权应答(Authoritative Answer) - 这个比特位在应答的时候才有意义，指出给出应答的服务器是查询域名的授权解析服务器。

注意因为别名的存在，应答可能存在多个主域名，这个 AA 位对应请求名，或者应答中的第一个主域名。

TC　截断(TrunCation) - 用来指出报文比允许的长度还要长，导致被截断。

RD　期望递归(Recursion Desired) - 这个比特位被请求设置，应答的时候使用的相同的值返回。如果设置了 RD，就建议域名服务器进行递归解析，递归查询的支持是可选的。

RA　支持递归(Recursion Available) - 这个比特位在应答中设置或取消，用来代表服务器是否支持递归查询。

Z　保留值，暂时未使用。在所有的请求和应答报文中必须置为 0。

RCODE　应答码(Response code) - 这 4 个比特位在应答报文中设置，代表的含义如下：

0　　没有错误。

1　　报文格式错误(Format error) - 服务器不能理解请求的报文。

2　　服务器失败(Server failure) - 因为服务器的原因导致没办法处理这个请求。

3　　名字错误(Name Error) - 只有对授权域名解析服务器有意义，指出解析的域名不存在。

4　　没有实现(Not Implemented) - 域名服务器不支持查询类型。

5　　拒绝(Refused) - 服务器由于设置的策略拒绝给出应答。比如，服务器不希望对某些请求者给出应答，或者服务器不希望进行某些操作（比如区域传送 zone transfer）。

6-15　　保留值，暂时未使用。

QDCOUNT 无符号 16 位整数表示报文请求段中的问题记录数。

ANCOUNT 无符号 16 位整数表示报文回答段中的回答记录数。

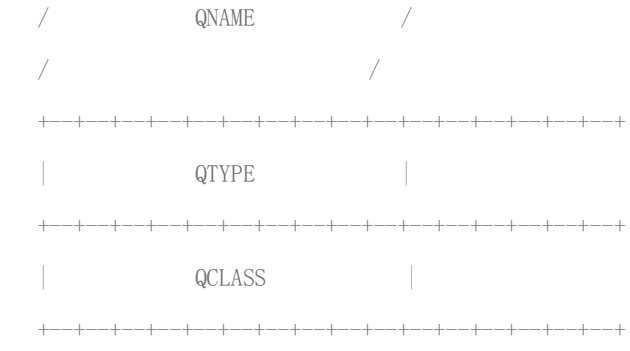
NSCOUNT 无符号 16 位整数表示报文授权段中的授权记录数。

ARCOUNT 无符号 16 位整数表示报文附加段中的附加记录数。

1.2 Question 的格式

在大多数查询中，Question 段包含着问题(question)，比如，指定问什么。这个段包含 QDCOUNT (usually 1)个问题，每个问题为下面的格式：

```
                                1  1  1  1  1  1
  0  1  2  3  4  5  6  7  8  9  0  1  2  3  4  5
+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
|                               |
```



字段含义如下

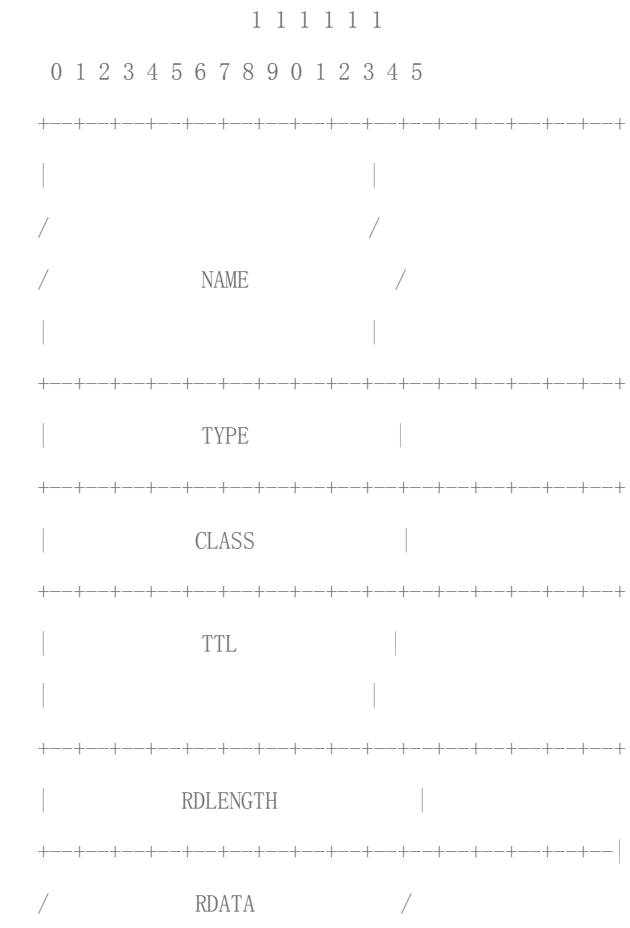
QNAME 域名被编码为一些 labels 序列，每个 labels 包含一个字节表示后续字符串长度，以及这个字符串，以 0 长度和空字符串来表示域名结束。注意这个字段可能为奇数字节，不需要进行边界填充对齐。

QTYPE 2 个字节表示查询类型，.取值可以为任何可用的类型值，以及通配码来表示所有的资源记录。

QCLASS 2 个字节表示查询的协议类，比如，IN 代表 Internet。

1.3 资源记录格式(Resource record)

应答，授权，附加段都共用相同的格式：多个资源记录，资源记录的个数由报文头段中对应的几个数值确定，每个资源记录格式如下：



/ /

+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+

各字段含义如下：

NAME 资源记录包含的域名

TYPE 2 个字节表示资源记录的类型，指出 RDATA 数据的含义

CLASS 2 个字节表示 RDATA 的类

TTL 4 字节无符号整数表示资源记录可以缓存的时间。0 代表只能被传输，但是不能被缓存。

RDLENGTH 2 个字节无符号整数表示 RDATA 的长度

RDATA 不定长字符串来表示记录，格式根 TYPE 和 CLASS 有关。比如，TYPE 是 A，CLASS 是 IN，那么 RDATA 就是一个 4 个字节的 ARPA 网络地址。

1.4 报文压缩

为了减小报文，域名系统使用一种压缩方法来消除报文中域名的重复。使用这种方法，后面重复出现的域名或者 labels 被替换为指向之前出现位置的指针。

指针占用 2 个字节，格式如下：

0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5

+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+

| 1 1| OFFSET |

+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+

前两个比特位都为 1。因为 lablels 限制为不多于 63 个字节，所以 label 的前两位一定为 0，这样就可以让指针与 label 进行区分。(10 和 01 组合保留，以便日后使用) 。偏移值(OFFSET)表示从报文开始的字节指针。偏移量为 0 表示 ID 字段的第一个字节。

压缩方法让报文中的域名成为：

- 以 0 结尾的 labels 序列
- 一个指针
- 指针结尾的 labels 序列

指针只能在域名不是特殊格式的时候使用，否则域名服务器或解析器需要知道资源记录的格式。目前还没有这种情况，但是以后可能会出现。

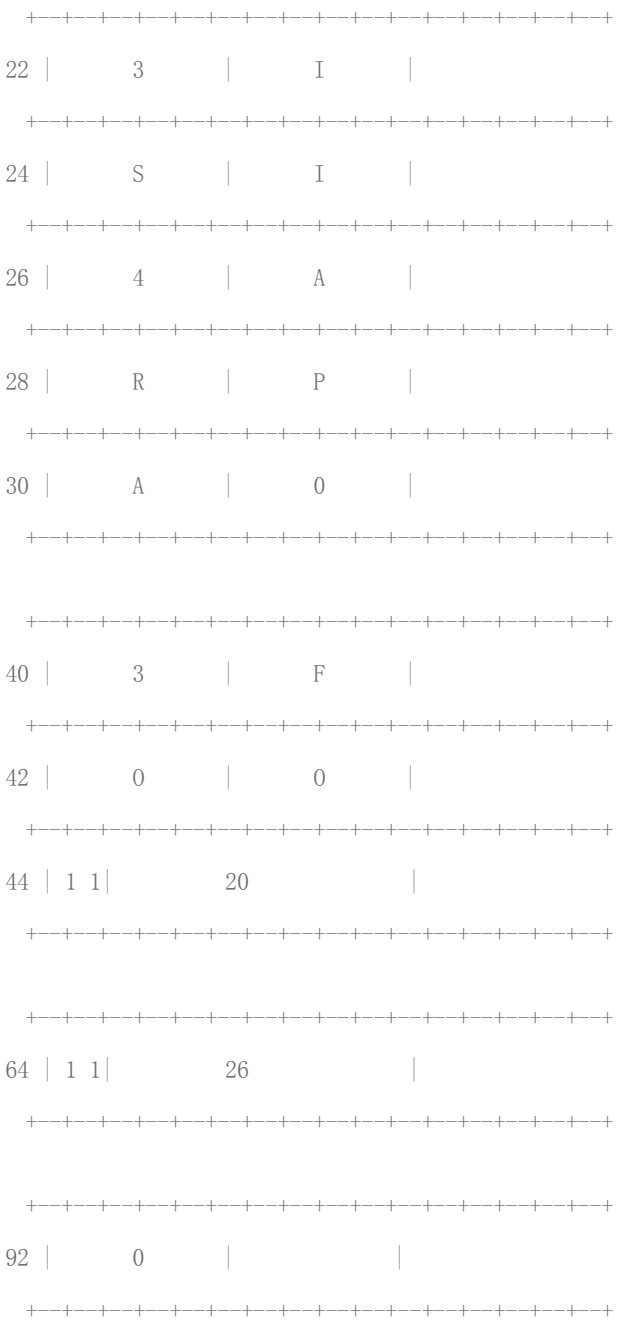
如果报文中的域名需要计算长度，并且使用了压缩算法，那么应该使用压缩后的长度，而不是压缩前的长度。

程序可以自由选择是否使用指针，虽然这回降低报文的容量，而且很容易产生截断。不过所有的程序都应该能够理解收到的报文中包含的指针。

比如，一个报文需要使用域名 F.ISI.ARPA，FOO.F.ISI.ARPA，ARPA，以及根。忽略报文中的其他字段，应该编码为：

+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+

20 | 1 | F |



偏移 20 的是域名 F.ISI.ARPA。域名 FOO.F.ISI.ARPA 偏移 40; 这样表示 FOO 的 label 后面跟着一个指向之前 F.ISI.ARPA 的指针。域名 ARPA 偏移 64，使用一个指针指向 F.ISI.ARPA 的 ARPA。注意可以用这个指针是因为 ARPA 是从偏移位置 20 开始的 labels 序列中的最后一个 label。 根域名在位置 92 定义为一个 0，没有 labels。

2、传输(Transport)

DNS 假设报文以数据报，或者从虚链路上以字节流进行传输。虚链路可以用来任何的 DNS 的传输，数据报可以减少代价提高传输性能。区域刷新必须使用虚链路，因为需要一个可靠的传输。

因特网中 DNS 支持端口 53 的 TCP[RFC-793]和端口 53 的 UDP [RFC-768]传输。

2.1 使用 UDP

消息通过 UDP 的 53 端口进行传输。

UDP 传输的消息严格要求限制在 512 字节内(不包括 IP 和 UDP 头)。长报文被截断，同时置报文头的 TC 标志位。

UDP 不能用于区域传输，主要用在标准的域名查询。报文通过 UDP 可能会丢失，所以重传机制是需要的，请求和应答可能在网络中或者服务器处理的时候被重新排序，所以解析客户端不能依赖请求的发送顺序。

UDP 的最优重传策略会因为网络的性能，客户的需要而不同，但是下面是推荐的：

- 客户端在对一台固定的服务器重试之前，尝试一下其他的服务器。

- 如果可能的话，重传的时间间隔需要建立在统计分析数据的基础上，太快的重试可能因为量太大导致服务器响应慢。建议的重试时间为 2-5 秒。

2.2 使用 TCP

通过 TCP 发送的报文使用 53 端口，报文的前面有个字节表示后面报文的长度，长度不包括自己占用的 2 个字节，这个长度使得底层收取完整的报文后在交给上层处理。

很多连接管理策略如下:

- 服务器不能阻塞其他传输 TCP 数据的请求。

- 服务器需要支持多连接

- 服务器要等客户端主动关闭连接，除非所有的数据都已经传输完了。

- 如果服务器想关闭没有通讯的连接来释放资源，那么需要等待大约 2 分钟的时间。特别是要等 SOA 和 AXFR(刷新操作中)在一个连接上传输完。服务器关闭连接的时候可以单方面的关闭，或者直接 reset 掉连接。

三、实例

1、请求解析 www.baidu.com.

在 linux 下使用 tcpdump port 53 抓包，同时使用 dig 进行解析测试，得到结果如下：

```
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 1169
;; flags: qr rd ra; QUERY: 1, ANSWER: 3, AUTHORITY: 4, ADDITIONAL: 0

;; QUESTION SECTION:
;www.baidu.com.    IN A

;; ANSWER SECTION:
www.baidu.com.   1200 IN CNAME www.a.shifen.com.
```

www.a.shifen.com. 600 IN A 121.14.88.76

www.a.shifen.com. 600 IN A 121.14.89.10

;; AUTHORITY SECTION:

a.shifen.com. 86411 IN NS ns5.a.shifen.com.

a.shifen.com. 86411 IN NS ns6.a.shifen.com.

a.shifen.com. 86411 IN NS ns1.a.shifen.com.

a.shifen.com. 86411 IN NS ns3.a.shifen.com.

1.1 请求报文

```
0x0000: 4500 003b f8cf 0000 4011 f9ae xxxx xxxx E..;....@.....r
0x0010:  xxxx xxxx 92b8 0035 0027 23ed 0491 0100 ...q...5.'#.....
0x0020: 0001 0000 0000 0000 0377 7777 0562 6169 .........www.bai
0x0030: 6475 0363 6f6d 0000 0100 01          du.com.....
```

0491：报文 ID，也就是十进制的 1169

0100：标志，置了 RD 字段，也就是期望递归的请求

0001 0000 0000 0000：分别为问题数，应答数，授权记录数，附加记录数，也就是 1 个问题

0377 7777 0562 6169 6475 0363 6f6d 00：也就是 www.baidu.com 的编码

00 0100 01：查询类型和查询类都为 1，也就是 internet 的 A 记录查询

1.2 应答报文

```
0x0000: 4500 00be 0016 4000 4011 b1e5 xxxx xxxx E.....@.@.....q
0x0010:  xxxx xxxx 0035 92b8 00aa 33e1 0491 8180 ...r.5....3.....
0x0020: 0001 0003 0004 0000 0377 7777 0562 6169 .........www.bai
0x0030: 6475 0363 6f6d 0000 0100 01c0 0c00 0500 du.com..........
0x0040: 0100 0004 b000 0f03 7777 7701 6106 7368 ........www.a.sh
0x0050: 6966 656e c016 c02b 0001 0001 0000 0258 ifen...+.......X
0x0060: 0004 790e 584c c02b 0001 0001 0000 0258 ..y.XL.+.......X
0x0070: 0004 790e 590a c02f 0002 0001 0001 518b ..y.Y../......Q.
0x0080: 0006 036e 7335 c02f c02f 0002 0001 0001 ...ns5././......
0x0090: 518b 0006 036e 7336 c02f c02f 0002 0001 Q....ns6././....
0x00a0: 0001 518b 0006 036e 7331 c02f c02f 0002 ..Q....ns1././..
0x00b0: 0001 0001 518b 0006 036e 7333 c02f      ....Q....ns3./
```

注意 8180，也就是二进制的 1 0000 0 0 1 1 000 0000 ，说明是应答，置了 RD 和 RA 位

黄色背景为压缩编码，比如 c016 就代表第 22 个字节，也就是 com。