CS 305 Lab Tutorial
Lecture 5 DNS

Dept. Computer Science and Engineering

Southern University of Science and Technology

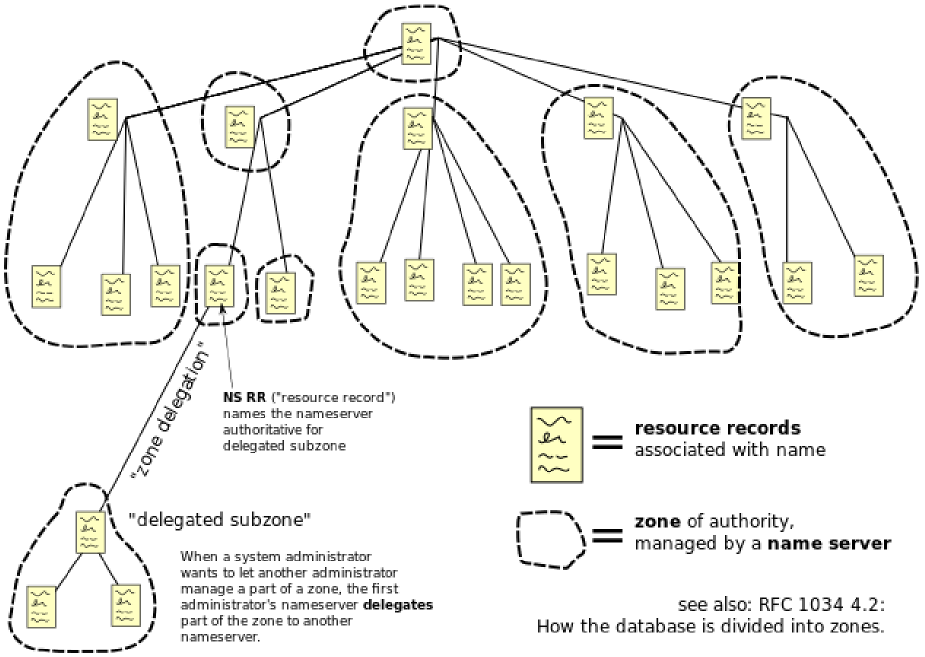
Topic

- DNS
  - DNS Message Structure
  - DNS Message head
  - RR in DNS
- EDNS (aka. Extension mechanisms for DNS)
  - DNSSEC
- DNS Resolver

Part A.1
Domain Name System

- DNS is a distributed database.

Domain Name Space



NS RR ("resource record") names the nameserver authoritative for delegated subzone

"delegated subzone"

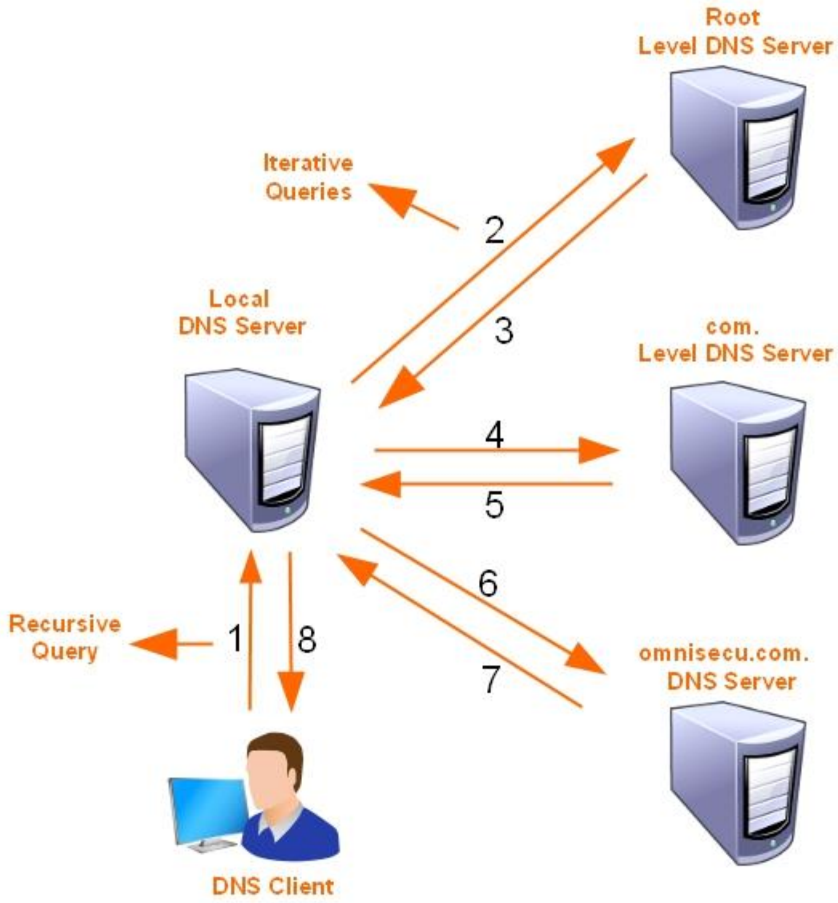
When a system administrator wants to let another administrator manage a part of a zone, the first administrator's nameserver **delegates** part of the zone to another nameserver.

= **resource records** associated with name

= **zone** of authority, managed by a **name server**

see also: RFC 1034 4.2:
How the database is divided into zones.

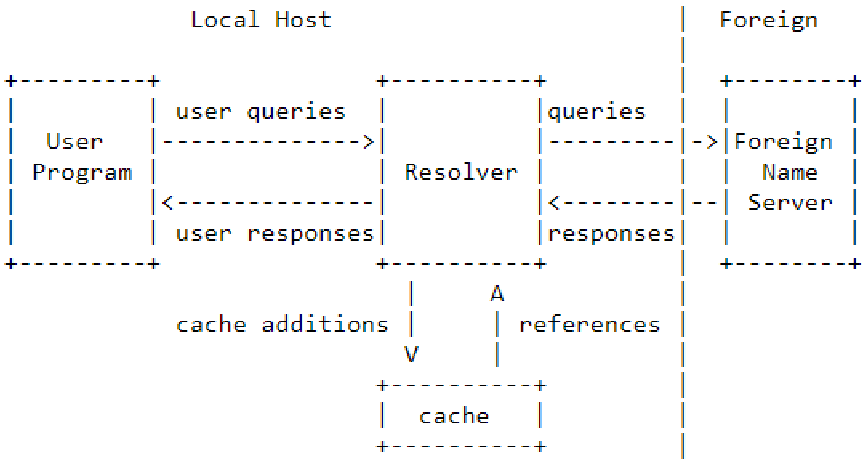
Recursive/Iterative Query



RFC 1035 Local Resolver

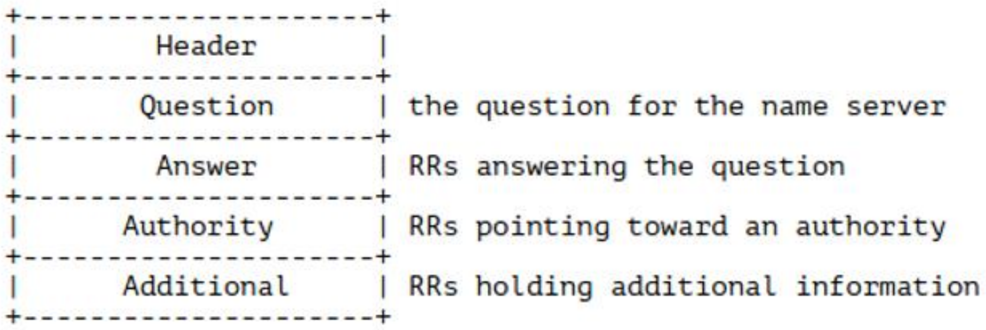
Domain Names - Implementation And Specification

- Most machine has a local resolver which handles request of domain name and maintain a cache of query result.

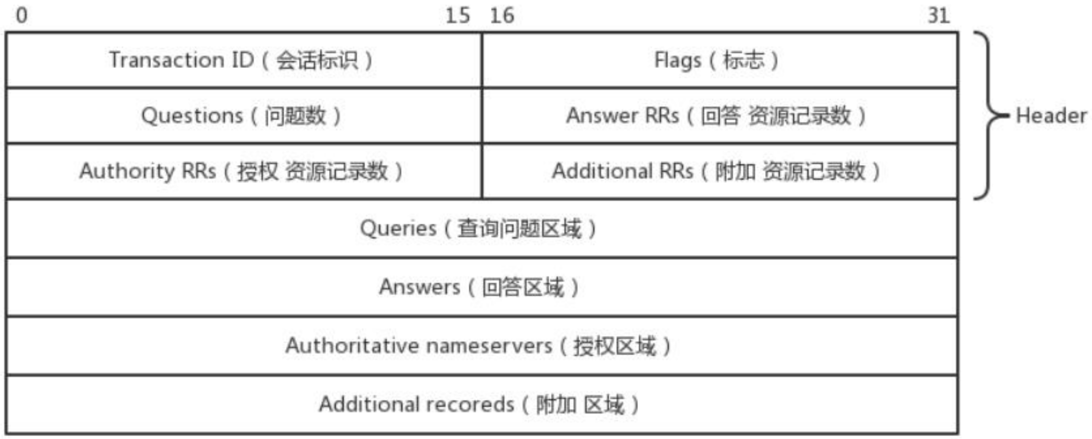


```
             Local Host                        |  Foreign
                                               |
+---------+               +----------+         |  +--------+
|         | user queries  |          |queries  |  |        |
|  User   |-------------->|          |---------|->|Foreign |
| Program |               | Resolver |         |  |  Name  |
|         |<--------------|          |<--------|--| Server |
|         | user responses|          |responses|  |        |
+---------+               +----------+         |  +--------+
                            |     A            |
            cache additions |     | references |
                            V     |            |
                          +----------+         |
                          |  cache   |         |
                          +----------+         |
```

Part A.2
DNS Message Structure



```
+---------------------+
|       Header        |
+---------------------+
|      Question       | the question for the name server
+---------------------+
|       Answer        | RRs answering the question
+---------------------+
|      Authority      | RRs pointing toward an authority
+---------------------+
|      Additional     | RRs holding additional information
+---------------------+
```



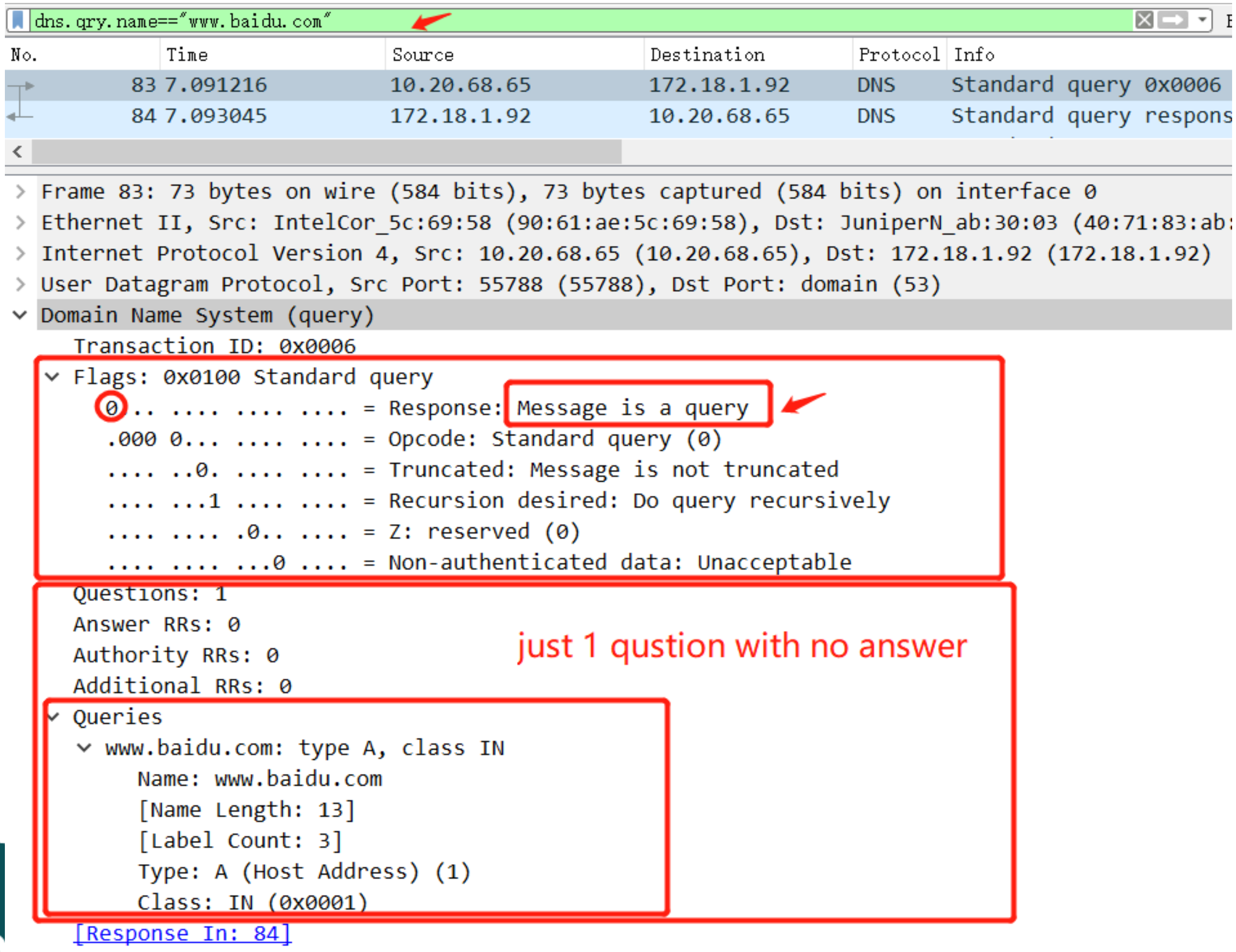
0 15	16 31	
Transaction ID（会话标识）	Flags（标志）	Header
Questions（问题数）	Answer RRs（回答 资源记录数）	
Authority RRs（授权 资源记录数）	Additional RRs（附加 资源记录数）	
Queries（查询问题区域）		
Answers（回答区域）		
Authoritative nameservers（授权区域）		
Additional recoreds（附加 区域）		

DNS协议报文格式

https://www.nslookuptool.com/chs/

A query message of DNS

nslookup   www.baidu.com



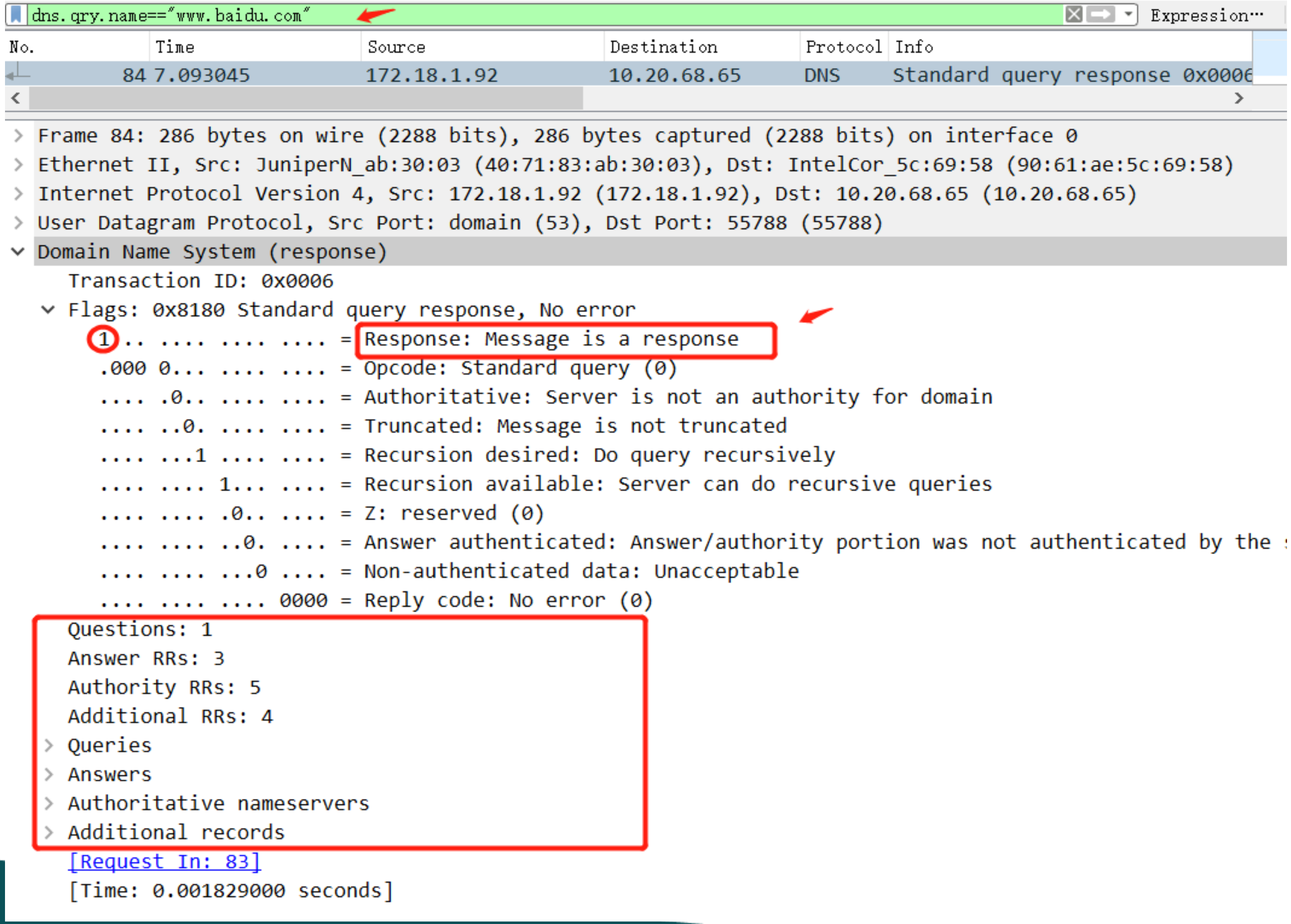
```
dns.qry.name=="www.baidu.com"

No.     Time        Source         Destination    Protocol Info
     83 7.091216    10.20.68.65    172.18.1.92    DNS      Standard query 0x0006
     84 7.093045    172.18.1.92    10.20.68.65    DNS      Standard query respons

> Frame 83: 73 bytes on wire (584 bits), 73 bytes captured (584 bits) on interface 0
> Ethernet II, Src: IntelCor_5c:69:58 (90:61:ae:5c:69:58), Dst: JuniperN_ab:30:03 (40:71:83:ab:
> Internet Protocol Version 4, Src: 10.20.68.65 (10.20.68.65), Dst: 172.18.1.92 (172.18.1.92)
> User Datagram Protocol, Src Port: 55788 (55788), Dst Port: domain (53)
v Domain Name System (query)
    Transaction ID: 0x0006
  v Flags: 0x0100 Standard query
      0... .... .... .... = Response: Message is a query
      .000 0... .... .... = Opcode: Standard query (0)
      .... ..0. .... .... = Truncated: Message is not truncated
      .... ...1 .... .... = Recursion desired: Do query recursively
      .... .... .0.. .... = Z: reserved (0)
      .... .... ...0 .... = Non-authenticated data: Unacceptable
    Questions: 1
    Answer RRs: 0
    Authority RRs: 0
    Additional RRs: 0
  v Queries
    v www.baidu.com: type A, class IN
        Name: www.baidu.com
        [Name Length: 13]
        [Label Count: 3]
        Type: A (Host Address) (1)
        Class: IN (0x0001)
    [Response In: 84]
```

just 1 qustion with no answer

“udp port 53” can be used as a capture filter

A response message of DNS

Nslookup   www.baidu.com



```
dns.qry.name=="www.baidu.com"
No.        Time         Source          Destination     Protocol Info
     84 7.093045        172.18.1.92     10.20.68.65     DNS      Standard query response 0x0006

> Frame 84: 286 bytes on wire (2288 bits), 286 bytes captured (2288 bits) on interface 0
> Ethernet II, Src: JuniperN_ab:30:03 (40:71:83:ab:30:03), Dst: IntelCor_5c:69:58 (90:61:ae:5c:69:58)
> Internet Protocol Version 4, Src: 172.18.1.92 (172.18.1.92), Dst: 10.20.68.65 (10.20.68.65)
> User Datagram Protocol, Src Port: domain (53), Dst Port: 55788 (55788)
v Domain Name System (response)
    Transaction ID: 0x0006
  v Flags: 0x8180 Standard query response, No error
      1... .... .... .... = Response: Message is a response
      .000 0... .... .... = Opcode: Standard query (0)
      .... .0.. .... .... = Authoritative: Server is not an authority for domain
      .... ..0. .... .... = Truncated: Message is not truncated
      .... ...1 .... .... = Recursion desired: Do query recursively
      .... .... 1... .... = Recursion available: Server can do recursive queries
      .... .... .0.. .... = Z: reserved (0)
      .... .... ..0. .... = Answer authenticated: Answer/authority portion was not authenticated by the
      .... .... ...0 .... = Non-authenticated data: Unacceptable
      .... .... .... 0000 = Reply code: No error (0)
    Questions: 1
    Answer RRs: 3
    Authority RRs: 5
    Additional RRs: 4
  > Queries
  > Answers
  > Authoritative nameservers
  > Additional records
    [Request In: 83]
    [Time: 0.001829000 seconds]
```

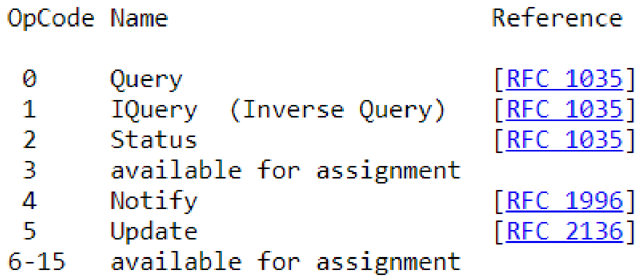
"udp port 53" can be used as a capture filter

IANA Reference: https://www.iana.org/assignments/dns-parameters/dns-parameters.xhtml

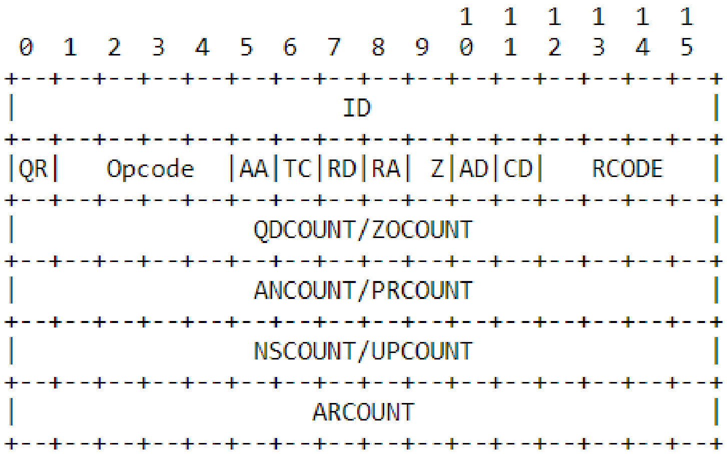
RFC 2929 DNS Message Headers

Domain Name System (DNS) IANA Considerations

- Set QR bit to 0 indicates the header is a query, otherwise is a response.
- OpCode 0 indicates this is a standard query.
- AA, TC, RD, RA, AD, CD stands for Authoritative Answer, Truncated, Recursion Desired, Recursion Available, Checking Disabled.
- Z is a reserved flag.



OpCode	Name	Reference
0	Query	[RFC 1035]
1	IQuery (Inverse Query)	[RFC 1035]
2	Status	[RFC 1035]
3	available for assignment	
4	Notify	[RFC 1996]
5	Update	[RFC 2136]
6-15	available for assignment	



```
                                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|AD|CD|   RCODE   |
+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
|                QDCOUNT/ZOCOUNT                |
+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
|                ANCOUNT/PRCOUNT                |
+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
|                NSCOUNT/UPCOUNT                |
+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
|                    ARCOUNT                    |
+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
```

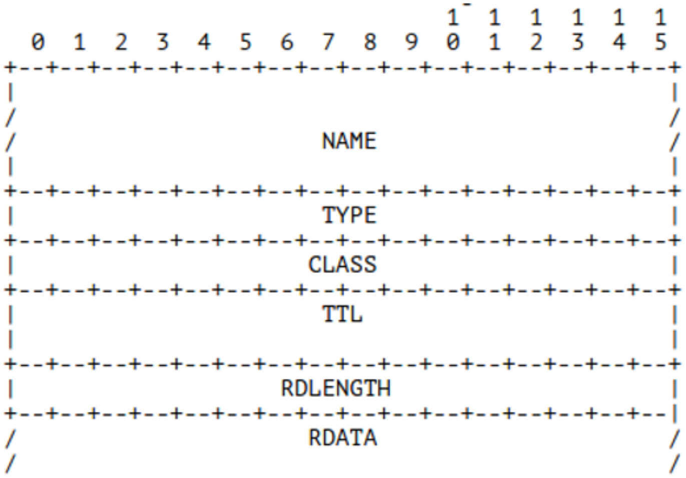
Example Structure Code in C:

```c
//DNS header structure
struct DNS_HEADER {
    unsigned short id;         // identification number

    unsigned char rd :1;       // recursion desired
    unsigned char tc :1;       // truncated message
    unsigned char aa :1;       // authoritive answer
    unsigned char opcode :4;   // purpose of message
    unsigned char qr :1;       // query/response flag

    unsigned char rcode :4;    // response code
    unsigned char cd :1;       // checking disabled
    unsigned char ad :1;       // authenticated data
    unsigned char z :1;        // its z! reserved
    unsigned char ra :1;       // recursion available

    unsigned short q_count;    // number of question entries
    unsigned short ans_count;  // number of answer entries
    unsigned short auth_count; // number of authority entries
    unsigned short add_count;  // number of resource entries
};
```

Decode Message Header in Python

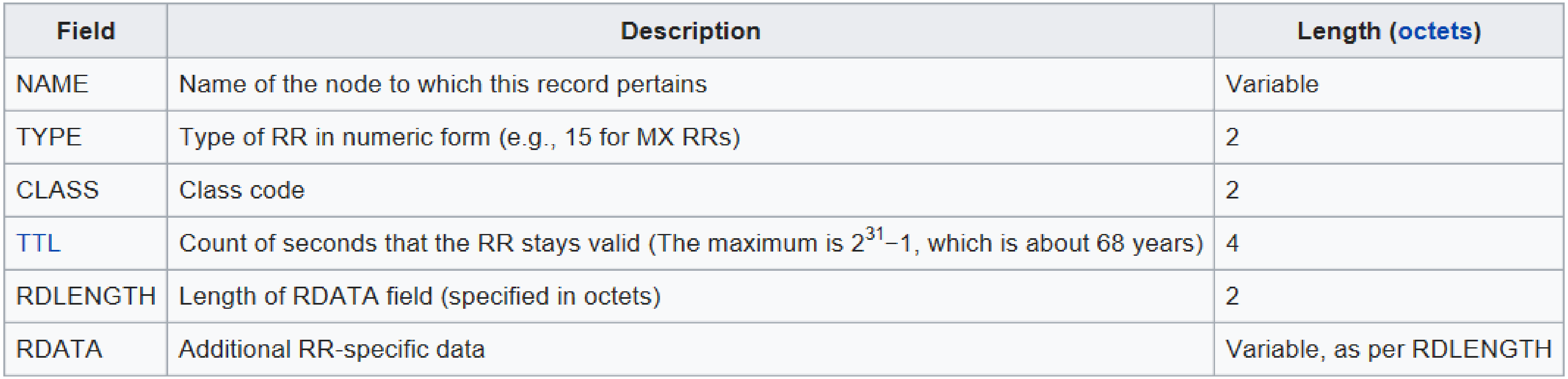
```
class DNSHeader:
    Struct = struct.Struct('!6H’)

    def __init__(self):
        self.__dict__ = {
            field: None
            for field in ('ID', 'QR', 'OpCode’, 'AA', 'TC', 'RD', 'RA', 'Z’,
            'RCode', 'QDCount', 'ANCount', 'NSCount', 'ARCount’)}

    def parse_header(self, data):
        self.ID, misc, self.QDCount, self.ANcount, \
        self.NScount, self.NScount = DNSHeader.Struct.unpack_from(data)
        self.QR = (misc & 0x8000) != 0
        self.OpCode = (misc & 0x7800) >> 11
        self.AA = (misc & 0x0400) != 0
        self.TC = (misc & 0x200) != 0
        self.RD = (misc & 0x100) != 0
        self.RA = (misc & 0x80) != 0
        self.Z = (misc & 0x70) >> 4 # Never used
        self.RCode = misc & 0xF

    def __str__(self):
        return '<DNSHeader {}>'.format(str(self.__dict__))
```

Part A.3
RR in DNS



```
                                    1  1  1  1  1  1
      0  1  2  3  4  5  6  7  8  9  0  1  2  3  4  5
    +--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
    |                                               |
    /                                               /
    /                      NAME                     /
    |                                               |
    +--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
    |                      TYPE                     |
    +--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
    |                     CLASS                     |
    +--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
    |                      TTL                      |
    |                                               |
    +--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
    |                   RDLENGTH                    |
    +--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--|
    /                     RDATA                     /
    /                                               /
```

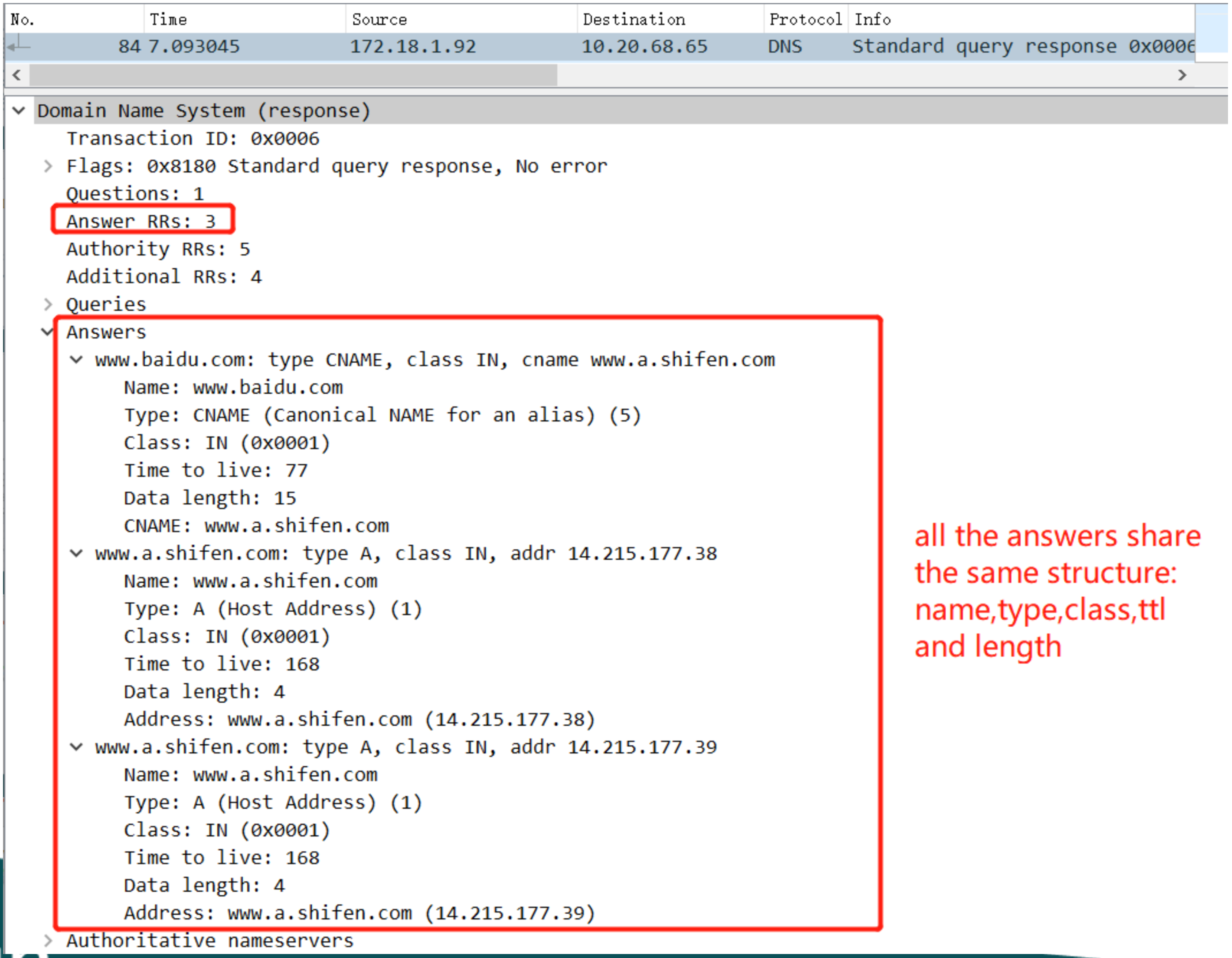
Resource record (RR) fields



Field	Description	Length (octets)
NAME	Name of the node to which this record pertains	Variable
TYPE	Type of RR in numeric form (e.g., 15 for MX RRs)	2
CLASS	Class code	2
TTL	Count of seconds that the RR stays valid (The maximum is $2^{31}-1$, which is about 68 years)	4
RDLENGTH	Length of RDATA field (specified in octets)	2
RDATA	Additional RR-specific data	Variable, as per RDLENGTH

RRs of Answers

nslookup   www.baidu.com

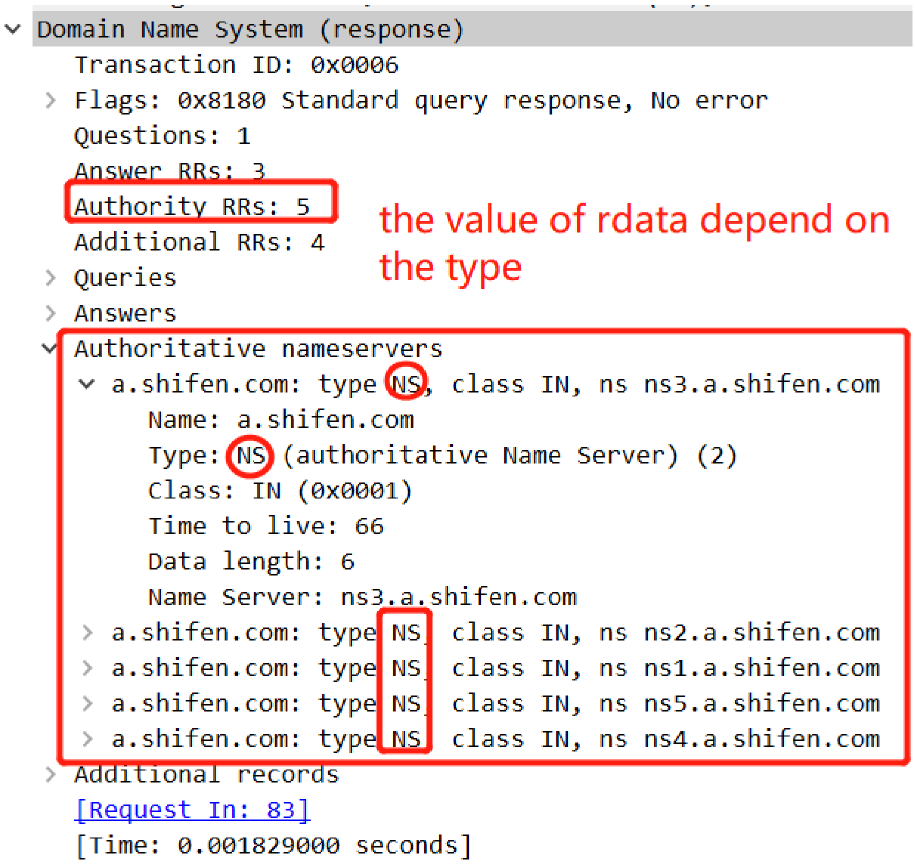


```
No.        Time          Source          Destination    Protocol Info
    84 7.093045     172.18.1.92     10.20.68.65    DNS      Standard query response 0x0006

∨ Domain Name System (response)
     Transaction ID: 0x0006
   > Flags: 0x8180 Standard query response, No error
     Questions: 1
     Answer RRs: 3
     Authority RRs: 5
     Additional RRs: 4
   > Queries
   ∨ Answers
     ∨ www.baidu.com: type CNAME, class IN, cname www.a.shifen.com
          Name: www.baidu.com
          Type: CNAME (Canonical NAME for an alias) (5)
          Class: IN (0x0001)
          Time to live: 77
          Data length: 15
          CNAME: www.a.shifen.com
     ∨ www.a.shifen.com: type A, class IN, addr 14.215.177.38
          Name: www.a.shifen.com
          Type: A (Host Address) (1)
          Class: IN (0x0001)
          Time to live: 168
          Data length: 4
          Address: www.a.shifen.com (14.215.177.38)
     ∨ www.a.shifen.com: type A, class IN, addr 14.215.177.39
          Name: www.a.shifen.com
          Type: A (Host Address) (1)
          Class: IN (0x0001)
          Time to live: 168
          Data length: 4
          Address: www.a.shifen.com (14.215.177.39)
   > Authoritative nameservers
```

all the answers share the same structure: name,type,class,ttl and length

RRs of authoritative name servers

nslookup www.baidu.com

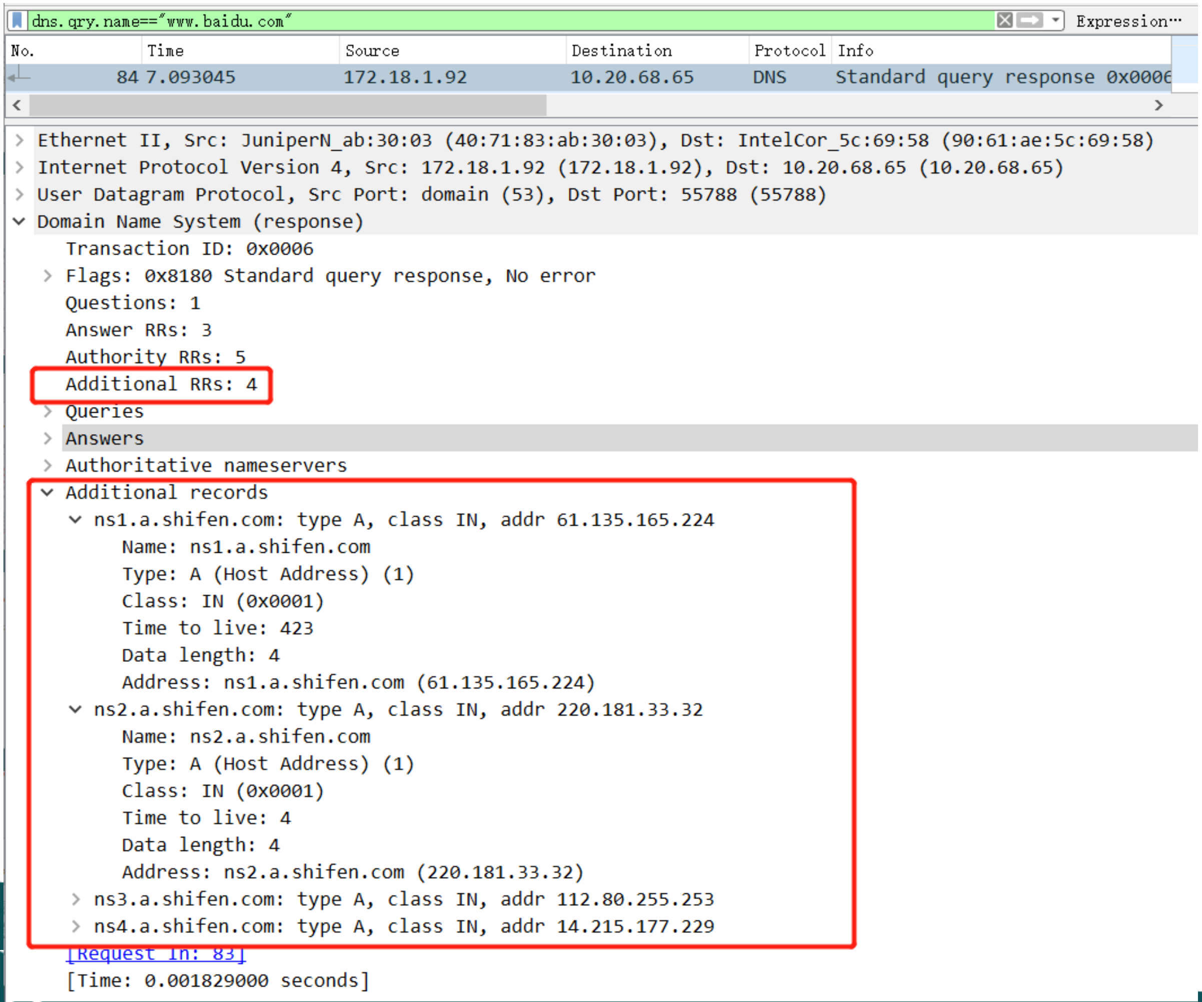


```
∨ Domain Name System (response)
    Transaction ID: 0x0006
  > Flags: 0x8180 Standard query response, No error
    Questions: 1
    Answer RRs: 3
    Authority RRs: 5
    Additional RRs: 4
  > Queries
  > Answers
  ∨ Authoritative nameservers
    ∨ a.shifen.com: type NS, class IN, ns ns3.a.shifen.com
        Name: a.shifen.com
        Type: NS (authoritative Name Server) (2)
        Class: IN (0x0001)
        Time to live: 66
        Data length: 6
        Name Server: ns3.a.shifen.com
    > a.shifen.com: type NS, class IN, ns ns2.a.shifen.com
    > a.shifen.com: type NS, class IN, ns ns1.a.shifen.com
    > a.shifen.com: type NS, class IN, ns ns5.a.shifen.com
    > a.shifen.com: type NS, class IN, ns ns4.a.shifen.com
  > Additional records
    [Request In: 83]
    [Time: 0.001829000 seconds]
```

the value of rdata depend on the type

RRs of Additional records

nslookup   www.baidu.com



```
dns.qry.name=="www.baidu.com"

No.       Time        Source         Destination    Protocol  Info
       84 7.093045    172.18.1.92    10.20.68.65    DNS       Standard query response 0x0006

> Ethernet II, Src: JuniperN_ab:30:03 (40:71:83:ab:30:03), Dst: IntelCor_5c:69:58 (90:61:ae:5c:69:58)
> Internet Protocol Version 4, Src: 172.18.1.92 (172.18.1.92), Dst: 10.20.68.65 (10.20.68.65)
> User Datagram Protocol, Src Port: domain (53), Dst Port: 55788 (55788)
v Domain Name System (response)
    Transaction ID: 0x0006
  > Flags: 0x8180 Standard query response, No error
    Questions: 1
    Answer RRs: 3
    Authority RRs: 5
    Additional RRs: 4
  > Queries
  > Answers
  > Authoritative nameservers
  v Additional records
    v ns1.a.shifen.com: type A, class IN, addr 61.135.165.224
        Name: ns1.a.shifen.com
        Type: A (Host Address) (1)
        Class: IN (0x0001)
        Time to live: 423
        Data length: 4
        Address: ns1.a.shifen.com (61.135.165.224)
    v ns2.a.shifen.com: type A, class IN, addr 220.181.33.32
        Name: ns2.a.shifen.com
        Type: A (Host Address) (1)
        Class: IN (0x0001)
        Time to live: 4
        Data length: 4
        Address: ns2.a.shifen.com (220.181.33.32)
    > ns3.a.shifen.com: type A, class IN, addr 112.80.255.253
    > ns4.a.shifen.com: type A, class IN, addr 14.215.177.229
    [Request In: 83]
    [Time: 0.001829000 seconds]
```

Part B.1
EDNS (aka. Extension mechanisms for DNS)

EDNS: a backward compatible mechanisms for allowing the DNS protocol to grow.

- The Domain Name System's wire protocol includes a number of fixed fields whose range has been or soon will be exhausted and does not allow clients to advertise their capabilities to servers

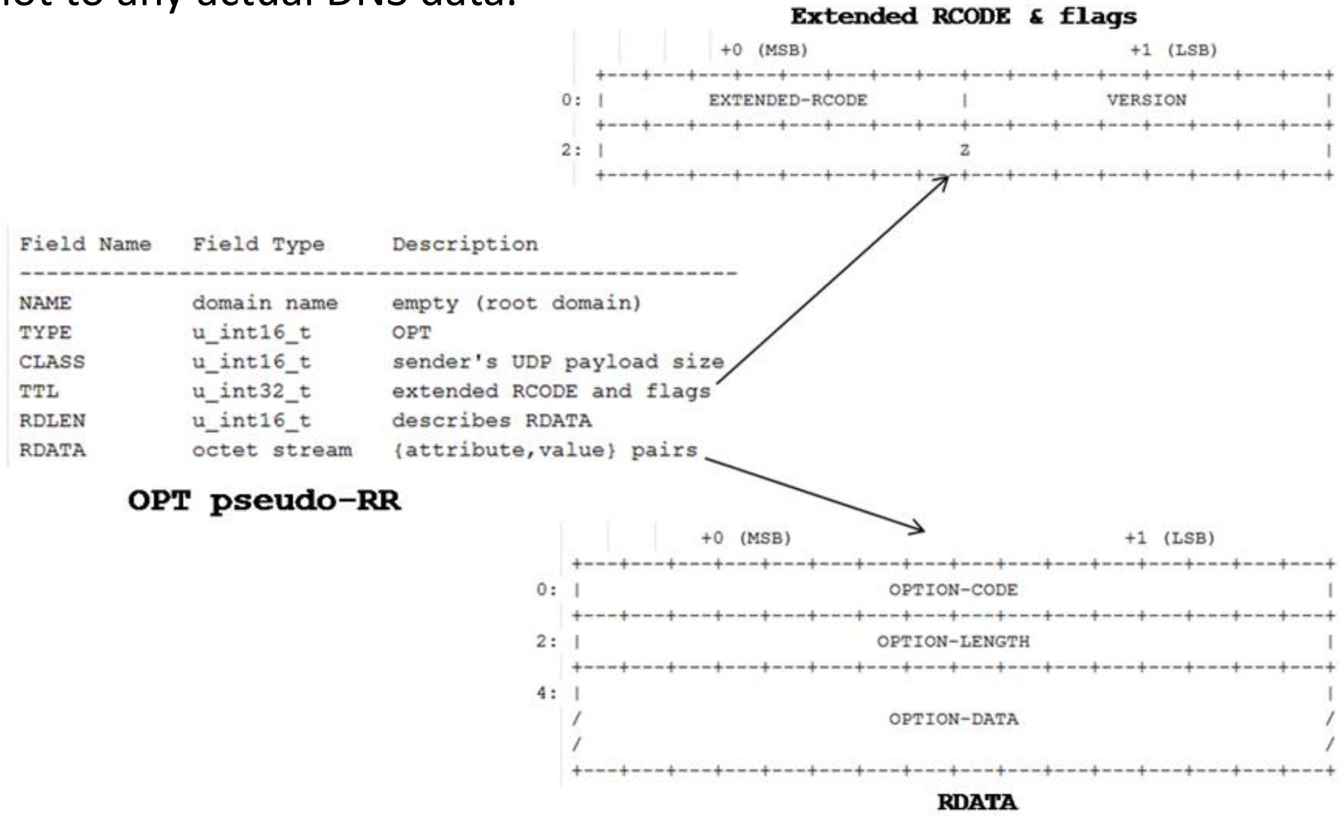
- DNS (see [RFC1035]) specifies a Message Format and within such messages there are standard formats for encoding options, errors, and name compression. The maximum allowable size of a DNS Message is fixed.

- Many of DNS's protocol limits are too small for uses which are or which are desired to become common. There is no way for implementations to advertise their capabilities.

https://tools.ietf.org/html/rfc2671

EDNS

One OPT pseudo-RR can be added to the additional data section of either a request or a response. An OPT is called a pseudo-RR because it pertains to a particular transport level message and not to any actual DNS data.



Extended RCODE & flags

```
                 +0 (MSB)                            +1 (LSB)
      +---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
   0: |         EXTENDED-RCODE        |            VERSION            |
      +---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
   2: |                               Z                               |
      +---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
```

Field Name	Field Type	Description
NAME	domain name	empty (root domain)
TYPE	u_int16_t	OPT
CLASS	u_int16_t	sender's UDP payload size
TTL	u_int32_t	extended RCODE and flags
RDLEN	u_int16_t	describes RDATA
RDATA	octet stream	{attribute,value} pairs

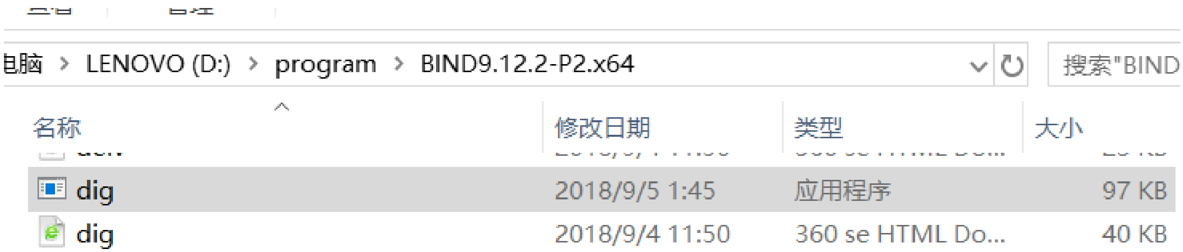
OPT pseudo-RR

```
                 +0 (MSB)                            +1 (LSB)
      +---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
   0: |                          OPTION-CODE                          |
      +---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
   2: |                         OPTION-LENGTH                         |
      +---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
   4: |                                                               |
      /                          OPTION-DATA                          /
      /                                                               /
      +---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
```

RDATA

Using dig to test EDNS

- **dig** is a flexible tool for interrogating DNS name servers.
  - It performs DNS lookups and displays the answers that are returned from the name server(s) that were queried.
  - Most DNS administrators use **dig** to troubleshoot DNS problems because of its flexibility, ease of use and clarity of output.



电脑 › LENOVO (D:) › program › BIND9.12.2-P2.x64

名称	修改日期	类型	大小
dig	2018/9/5 1:45	应用程序	97 KB
dig	2018/9/4 11:50	360 se HTML Do...	40 KB

Bind is a Toolset which includes dig as a component
Bind could be get from http://www.isc.org/downloads/

Using dig

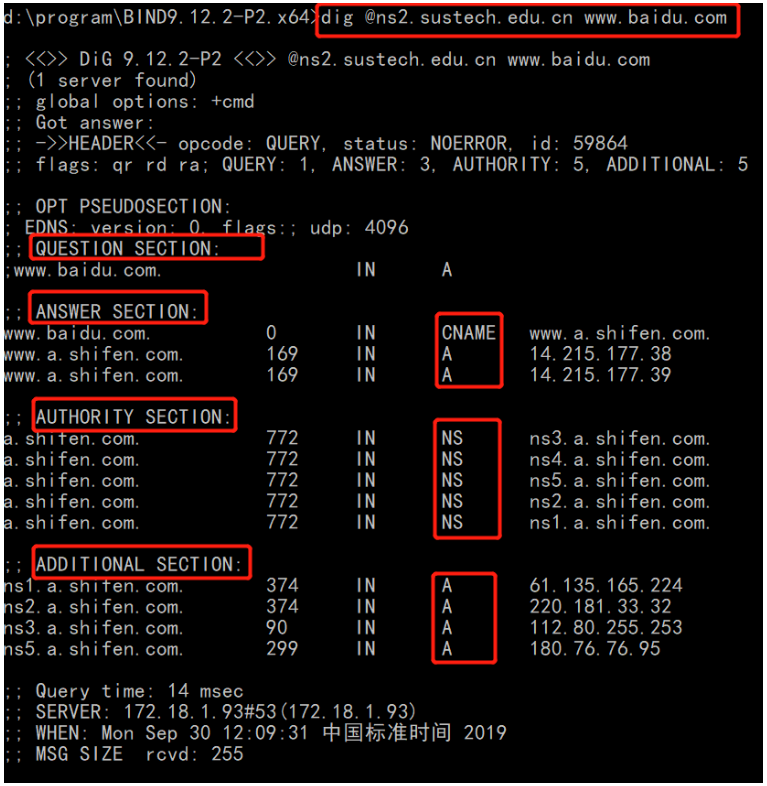
A typical invocation of dig looks like:

dig @server name type
where:

server
is **the name or IP address of the name server to query**. This can be an IPv4 address in dotted-decimal notation or an IPv6 address in colon-delimited notation. When the supplied server argument is a hostname, dig resolves that name before querying that name server.

name
is the name of the resource record that is to be looked up.

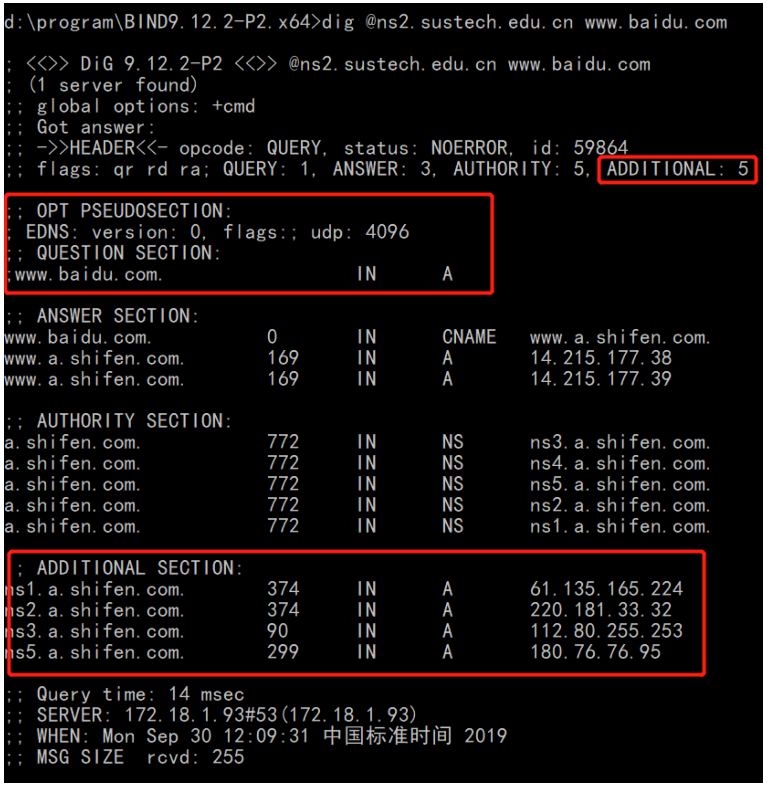
type
indicates **what type of query is required — ANY, A, MX, SIG**, etc. type can be any valid query type. If no type argument is supplied, dig will perform a lookup for an A record.



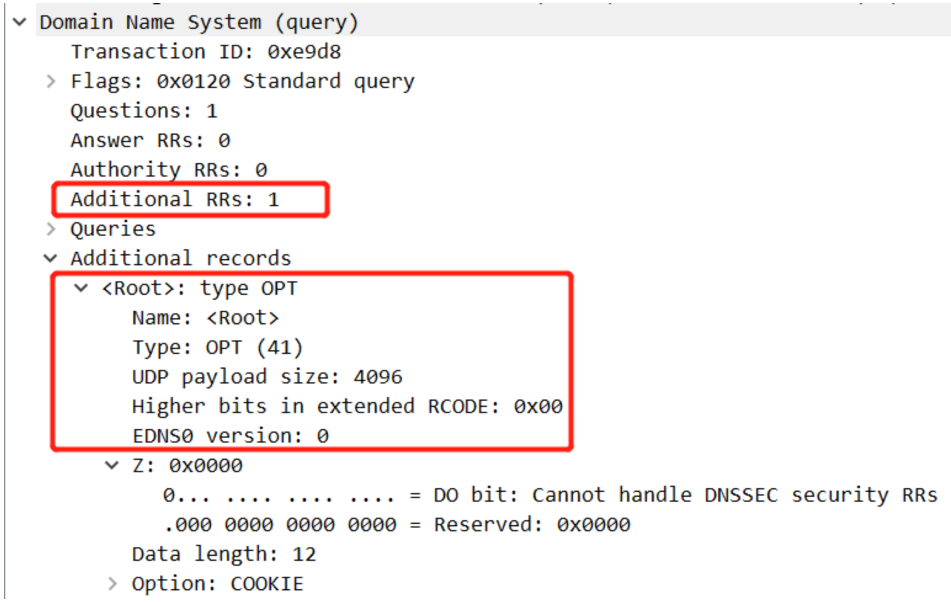
```
d:\program\BIND9.12.2-P2.x64>dig @ns2.sustech.edu.cn www.baidu.com

; <<>> DiG 9.12.2-P2 <<>> @ns2.sustech.edu.cn www.baidu.com
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 59864
;; flags: qr rd ra; QUERY: 1, ANSWER: 3, AUTHORITY: 5, ADDITIONAL: 5

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
;; QUESTION SECTION:
;www.baidu.com.                 IN      A

;; ANSWER SECTION:
www.baidu.com.          0       IN      CNAME   www.a.shifen.com.
www.a.shifen.com.       169     IN      A       14.215.177.38
www.a.shifen.com.       169     IN      A       14.215.177.39

;; AUTHORITY SECTION:
a.shifen.com.           772     IN      NS      ns3.a.shifen.com.
a.shifen.com.           772     IN      NS      ns4.a.shifen.com.
a.shifen.com.           772     IN      NS      ns5.a.shifen.com.
a.shifen.com.           772     IN      NS      ns2.a.shifen.com.
a.shifen.com.           772     IN      NS      ns1.a.shifen.com.

;; ADDITIONAL SECTION:
ns1.a.shifen.com.       374     IN      A       61.135.165.224
ns2.a.shifen.com.       374     IN      A       220.181.33.32
ns3.a.shifen.com.       90      IN      A       112.80.255.253
ns5.a.shifen.com.       299     IN      A       180.76.76.95

;; Query time: 14 msec
;; SERVER: 172.18.1.93#53(172.18.1.93)
;; WHEN: Mon Sep 30 12:09:31 中国标准时间 2019
;; MSG SIZE  rcvd: 255
```

SUSTech
Southern University
of Science and Technology

Using dig to test EDNS



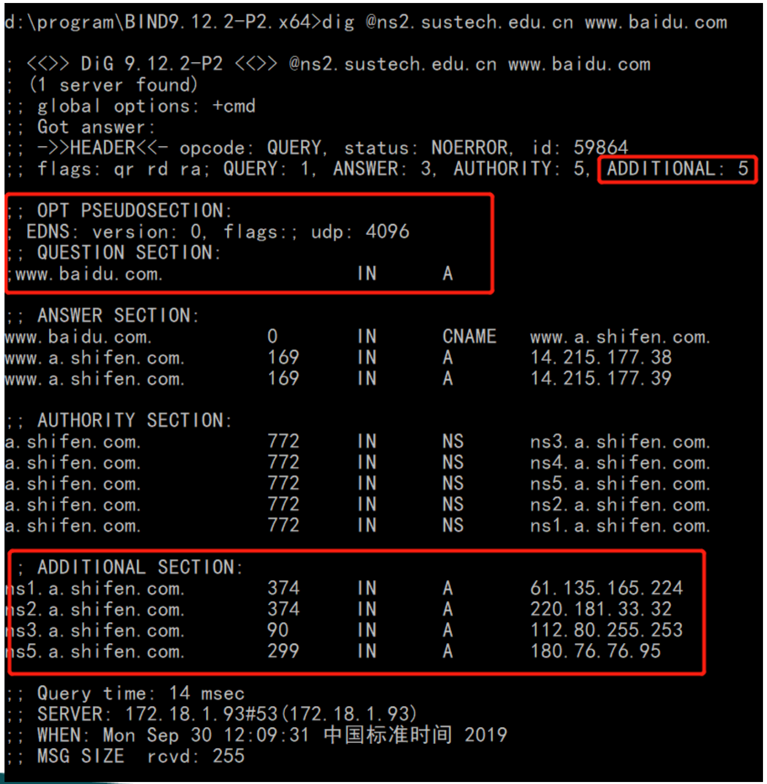
```
d:\program\BIND9.12.2-P2.x64>dig @ns2.sustech.edu.cn www.baidu.com

; <<>> DiG 9.12.2-P2 <<>> @ns2.sustech.edu.cn www.baidu.com
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 59864
;; flags: qr rd ra; QUERY: 1, ANSWER: 3, AUTHORITY: 5, ADDITIONAL: 5

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
;; QUESTION SECTION:
;www.baidu.com.                 IN      A

;; ANSWER SECTION:
www.baidu.com.          0       IN      CNAME   www.a.shifen.com.
www.a.shifen.com.       169     IN      A       14.215.177.38
www.a.shifen.com.       169     IN      A       14.215.177.39

;; AUTHORITY SECTION:
a.shifen.com.           772     IN      NS      ns3.a.shifen.com.
a.shifen.com.           772     IN      NS      ns4.a.shifen.com.
a.shifen.com.           772     IN      NS      ns5.a.shifen.com.
a.shifen.com.           772     IN      NS      ns2.a.shifen.com.
a.shifen.com.           772     IN      NS      ns1.a.shifen.com.

;; ADDITIONAL SECTION:
ns1.a.shifen.com.       374     IN      A       61.135.165.224
ns2.a.shifen.com.       374     IN      A       220.181.33.32
ns3.a.shifen.com.       90      IN      A       112.80.255.253
ns5.a.shifen.com.       299     IN      A       180.76.76.95

;; Query time: 14 msec
;; SERVER: 172.18.1.93#53(172.18.1.93)
;; WHEN: Mon Sep 30 12:09:31 中国标准时间 2019
;; MSG SIZE  rcvd: 255
```



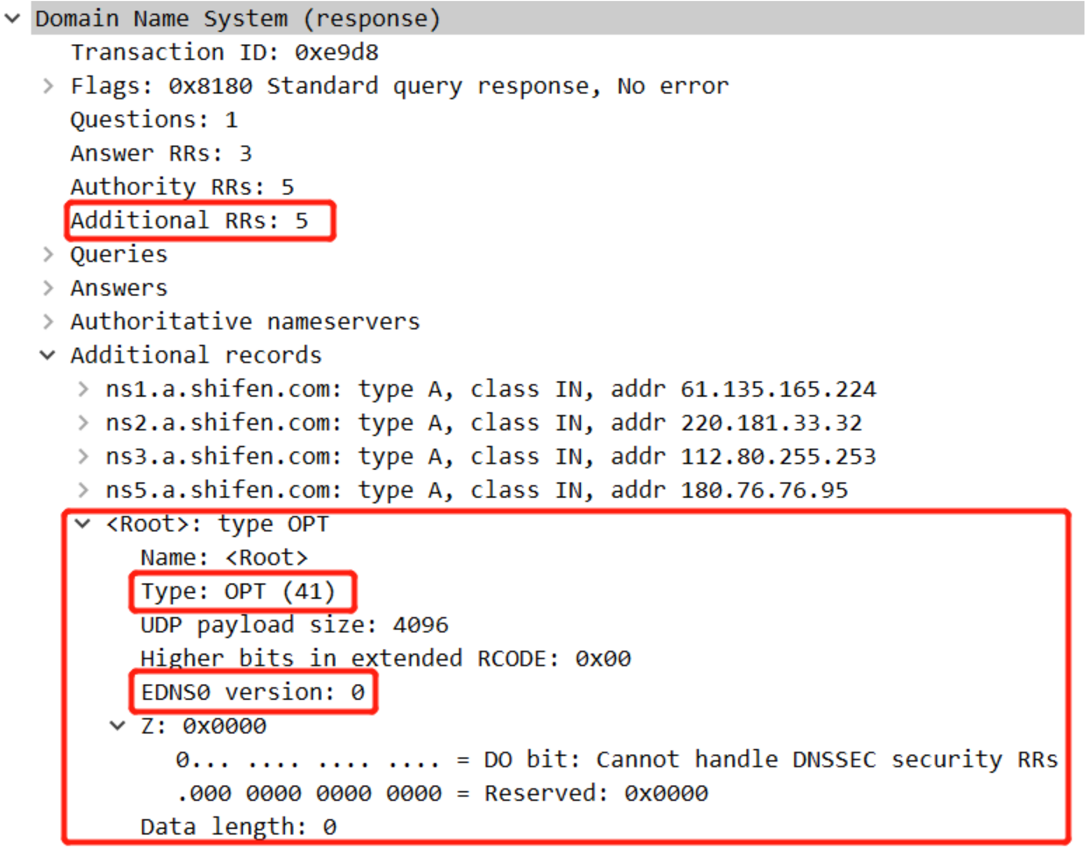
```
v Domain Name System (query)
    Transaction ID: 0xe9d8
  > Flags: 0x0120 Standard query
    Questions: 1
    Answer RRs: 0
    Authority RRs: 0
    Additional RRs: 1
  > Queries
  v Additional records
    v <Root>: type OPT
        Name: <Root>
        Type: OPT (41)
        UDP payload size: 4096
        Higher bits in extended RCODE: 0x00
        EDNS0 version: 0
      v Z: 0x0000
          0... .... .... .... = DO bit: Cannot handle DNSSEC security RRs
          .000 0000 0000 0000 = Reserved: 0x0000
        Data length: 12
      > Option: COOKIE
```

SUSTech
Southern University
of Science and Technology

Using dig to test EDNS



```
d:\program\BIND9.12.2-P2.x64>dig @ns2.sustech.edu.cn www.baidu.com

; <<>> DiG 9.12.2-P2 <<>> @ns2.sustech.edu.cn www.baidu.com
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 59864
;; flags: qr rd ra; QUERY: 1, ANSWER: 3, AUTHORITY: 5, ADDITIONAL: 5

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
;; QUESTION SECTION:
;www.baidu.com.                 IN      A

;; ANSWER SECTION:
www.baidu.com.          0       IN      CNAME   www.a.shifen.com.
www.a.shifen.com.       169     IN      A       14.215.177.38
www.a.shifen.com.       169     IN      A       14.215.177.39

;; AUTHORITY SECTION:
a.shifen.com.           772     IN      NS      ns3.a.shifen.com.
a.shifen.com.           772     IN      NS      ns4.a.shifen.com.
a.shifen.com.           772     IN      NS      ns5.a.shifen.com.
a.shifen.com.           772     IN      NS      ns2.a.shifen.com.
a.shifen.com.           772     IN      NS      ns1.a.shifen.com.

;; ADDITIONAL SECTION:
ns1.a.shifen.com.       374     IN      A       61.135.165.224
ns2.a.shifen.com.       374     IN      A       220.181.33.32
ns3.a.shifen.com.       90      IN      A       112.80.255.253
ns5.a.shifen.com.       299     IN      A       180.76.76.95

;; Query time: 14 msec
;; SERVER: 172.18.1.93#53(172.18.1.93)
;; WHEN: Mon Sep 30 12:09:31 中国标准时间 2019
;; MSG SIZE  rcvd: 255
```



```
v Domain Name System (response)
    Transaction ID: 0xe9d8
  > Flags: 0x8180 Standard query response, No error
    Questions: 1
    Answer RRs: 3
    Authority RRs: 5
    Additional RRs: 5
  > Queries
  > Answers
  > Authoritative nameservers
  v Additional records
    > ns1.a.shifen.com: type A, class IN, addr 61.135.165.224
    > ns2.a.shifen.com: type A, class IN, addr 220.181.33.32
    > ns3.a.shifen.com: type A, class IN, addr 112.80.255.253
    > ns5.a.shifen.com: type A, class IN, addr 180.76.76.95
    v <Root>: type OPT
        Name: <Root>
        Type: OPT (41)
        UDP payload size: 4096
        Higher bits in extended RCODE: 0x00
        EDNS0 version: 0
      v Z: 0x0000
          0... .... .... .... = DO bit: Cannot handle DNSSEC security RRs
          .000 0000 0000 0000 = Reserved: 0x0000
        Data length: 0
```

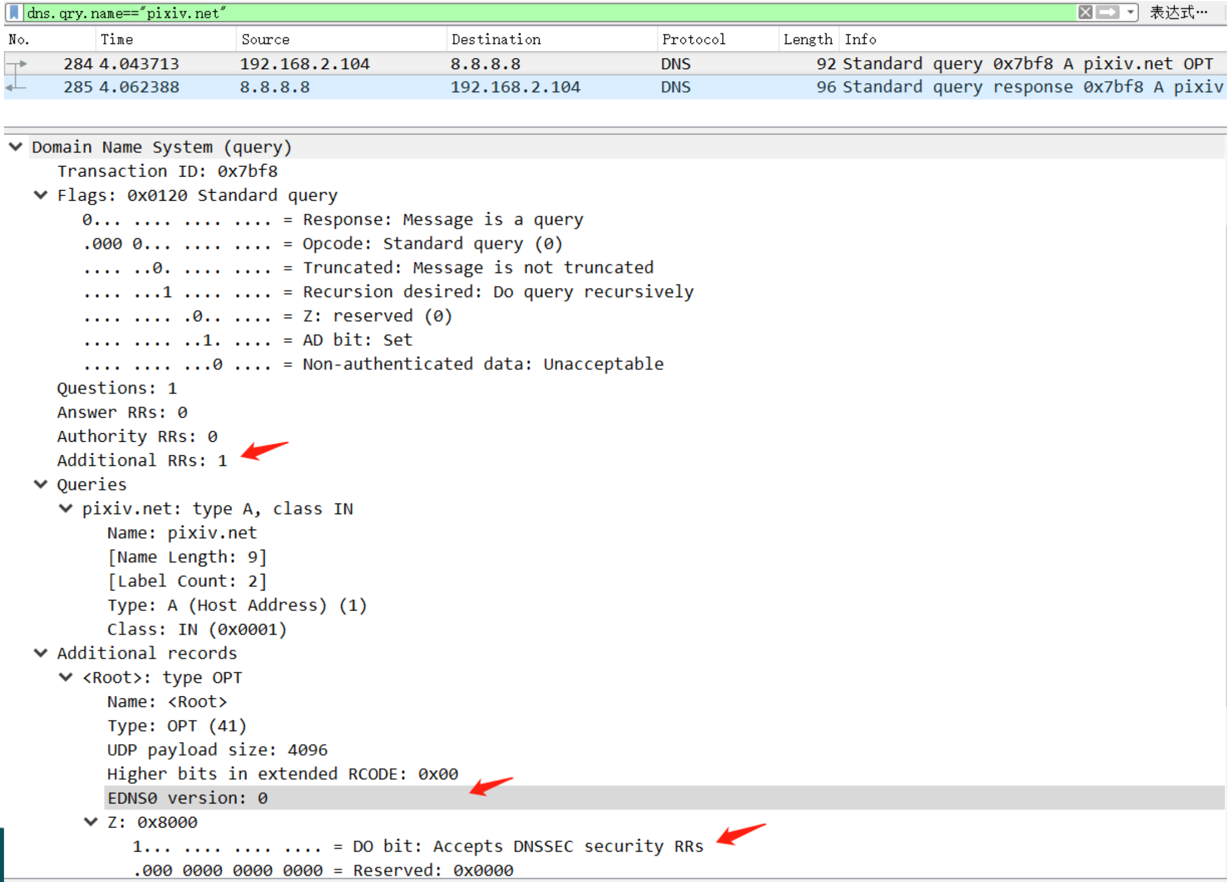
Part B.2
DNSSEC

Domain Name System Security Extensions

- a security mechanism designed to solve DNS spoofing and cache pollution.
- By using cryptography, the DNS resolver can verify whether the reply it receives comes from the real server or is tampered with during transmission.

DNSSEC using EDNS（1）

dig @8.8.8.8 pixiv.net +dnssec



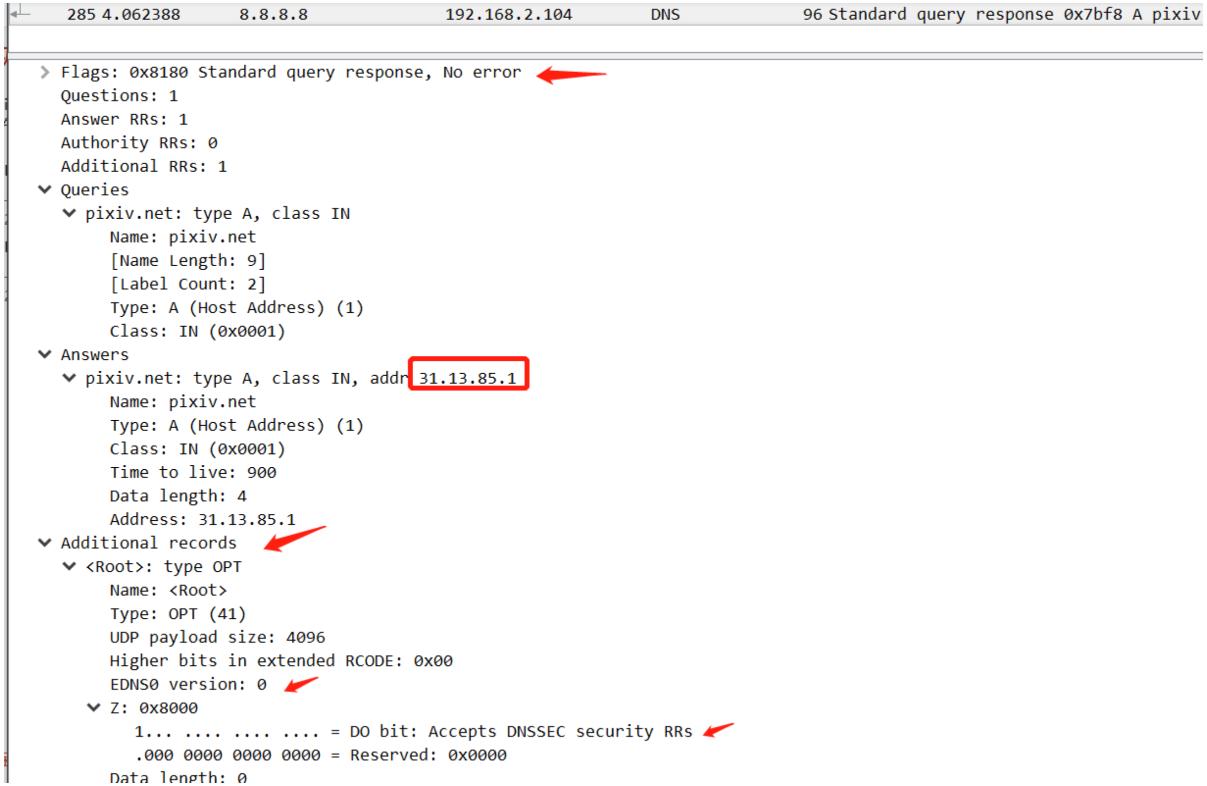
```
dns.qry.name=="pixiv.net"                                        表达式…
```

```
No.   Time         Source          Destination     Protocol  Length Info
  284 4.043713     192.168.2.104   8.8.8.8         DNS           92 Standard query 0x7bf8 A pixiv.net OPT
  285 4.062388     8.8.8.8         192.168.2.104   DNS           96 Standard query response 0x7bf8 A pixiv
```

```
Domain Name System (query)
    Transaction ID: 0x7bf8
  Flags: 0x0120 Standard query
        0... .... .... .... = Response: Message is a query
        .000 0... .... .... = Opcode: Standard query (0)
        .... ..0. .... .... = Truncated: Message is not truncated
        .... ...1 .... .... = Recursion desired: Do query recursively
        .... .... .0.. .... = Z: reserved (0)
        .... .... ..1. .... = AD bit: Set
        .... .... ...0 .... = Non-authenticated data: Unacceptable
    Questions: 1
    Answer RRs: 0
    Authority RRs: 0
    Additional RRs: 1
  Queries
    pixiv.net: type A, class IN
        Name: pixiv.net
        [Name Length: 9]
        [Label Count: 2]
        Type: A (Host Address) (1)
        Class: IN (0x0001)
  Additional records
    <Root>: type OPT
        Name: <Root>
        Type: OPT (41)
        UDP payload size: 4096
        Higher bits in extended RCODE: 0x00
        EDNS0 version: 0
      Z: 0x8000
          1... .... .... .... = DO bit: Accepts DNSSEC security RRs
          .000 0000 0000 0000 = Reserved: 0x0000
```

DNSSEC using EDNS（2）

dig @8.8.8.8 pixiv.net +dnssec



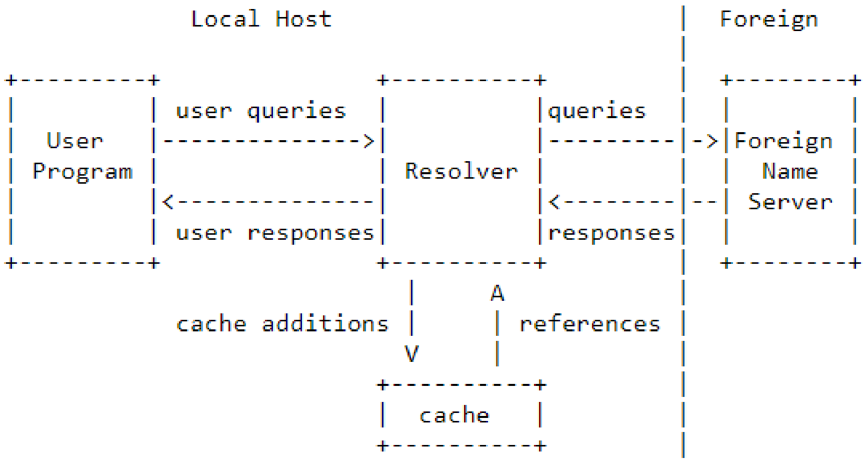
```
  285 4.062388      8.8.8.8             192.168.2.104       DNS           96 Standard query response 0x7bf8 A pixiv

  > Flags: 0x8180 Standard query response, No error
    Questions: 1
    Answer RRs: 1
    Authority RRs: 0
    Additional RRs: 1
  v Queries
    v pixiv.net: type A, class IN
        Name: pixiv.net
        [Name Length: 9]
        [Label Count: 2]
        Type: A (Host Address) (1)
        Class: IN (0x0001)
  v Answers
    v pixiv.net: type A, class IN, addr 31.13.85.1
        Name: pixiv.net
        Type: A (Host Address) (1)
        Class: IN (0x0001)
        Time to live: 900
        Data length: 4
        Address: 31.13.85.1
  v Additional records
    v <Root>: type OPT
        Name: <Root>
        Type: OPT (41)
        UDP payload size: 4096
        Higher bits in extended RCODE: 0x00
        EDNS0 version: 0
      v Z: 0x8000
          1... .... .... .... = DO bit: Accepts DNSSEC security RRs
          .000 0000 0000 0000 = Reserved: 0x0000
        Data length: 0
```

Part C DNS resolver

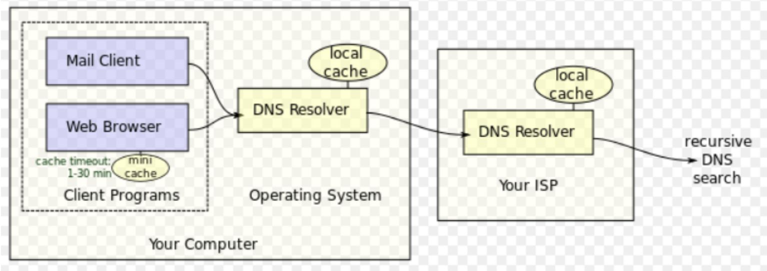
RFC 1035 Local Resolver

Domain Names - Implementation And Specification

- Most machine has a local resolver which handles request of domain name and maintain a cache of query result.



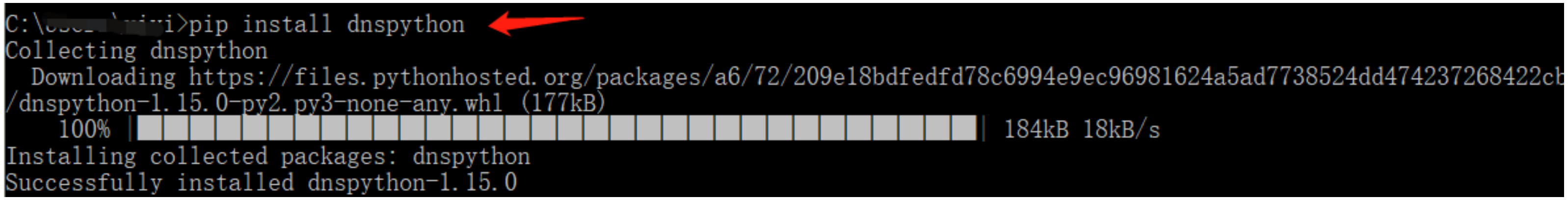
```
                 Local Host                        |  Foreign
                                                   |
    +---------+               +----------+         |  +--------+
    |         | user queries  |          |queries  |  |        |
    |  User   |-------------->|          |---------|->|Foreign |
    | Program |               | Resolver |         |  |  Name  |
    |         |<--------------|          |<--------|--| Server |
    |         | user responses|          |responses|  |        |
    +---------+               +----------+         |  +--------+
                                |     A            |
                cache additions |     | references |
                                V     |            |
                              +----------+         |
                              |  cache   |         |
                              +----------+         |
```



Using dns.resolver of python

Using pip to install dnspython

- pip is the package installer for Python. You can use pip to install packages from the Python Package Index and other indexes.

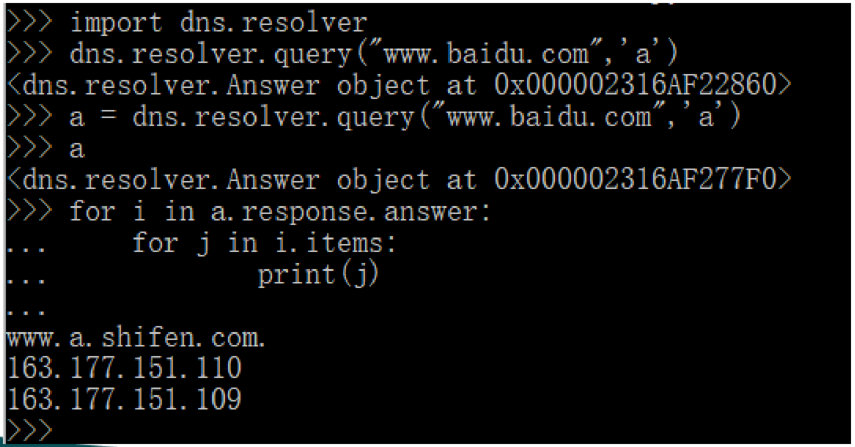


```
C:\[redacted]i>pip install dnspython
Collecting dnspython
  Downloading https://files.pythonhosted.org/packages/a6/72/209e18bdfedfd78c6994e9ec96981624a5ad7738524dd474237268422cb
/dnspython-1.15.0-py2.py3-none-any.whl (177kB)
    100% |████████████████████████████████| 184kB 18kB/s
Installing collected packages: dnspython
Successfully installed dnspython-1.15.0
```

- A demo of using query of dns.resolver

If ‘pip’ is not installed on your computer, get it from https://pypi.org/project/pip/

Get more infor about dnspython, get it from https://pypi.org/project/dnspython/



```
>>> import dns.resolver
>>> dns.resolver.query("www.baidu.com",'a')
<dns.resolver.Answer object at 0x000002316AF22860>
>>> a = dns.resolver.query("www.baidu.com",'a')
>>> a
<dns.resolver.Answer object at 0x000002316AF277F0>
>>> for i in a.response.answer:
...     for j in i.items:
...         print(j)
...
www.a.shifen.com.
163.177.151.110
163.177.151.109
>>>
```

lab 5

- Please finish the lab according to this file
  - submit the **report** of lab 5 based on the lab report template.
  - submit your source code in zip file. (**5.3.zip**)
    - comments is MUST
    - DO NOT copy paste any existing source code of DNS resolver

lab 5.1

- make an DNS query which will invoke the EDNS0
  - Screenshot on this command and its output
- capture the packages using Wireshark
  - what is the content of this query message
    - Find the name, type and class of this query
    - How can you tell this DNS query is based on EDNS0
    - From this query massage , can it handle DNSSEC security RRs or not
  - what is the content of this response message
    - Is there any answers, what's the ttl of each answer
    - Is there any authority RRs, what's the type of each RR
    - Is there any special additional RRs with OPT type, what does its 'Do bit' say: Does it accept DNSSEC security RRs or not

lab 5.2

- Make the query by using query method of "dns resolver"(a python package)
  - To query the type A value of www.sina.com.cn based on TCP and UDP stream respectively
- capture the related TCP stream and UDP stream using Wireshark
  - Screenshot on this two commands .

    what's the default transport lay protocol while invoke DNS query
  - Screenshot on the TCP stream of query by TCP.

    how many TCP packets are captured in this stream, Which port is used?
  - Screenshot on the UDP stream of query by UDP.

    how many UDP packets are captured in this stream, Which port is used?
  - Is there any difference on DNS query and response message while using TCP and UDP respectively

lab 5.3
implement a local resolver

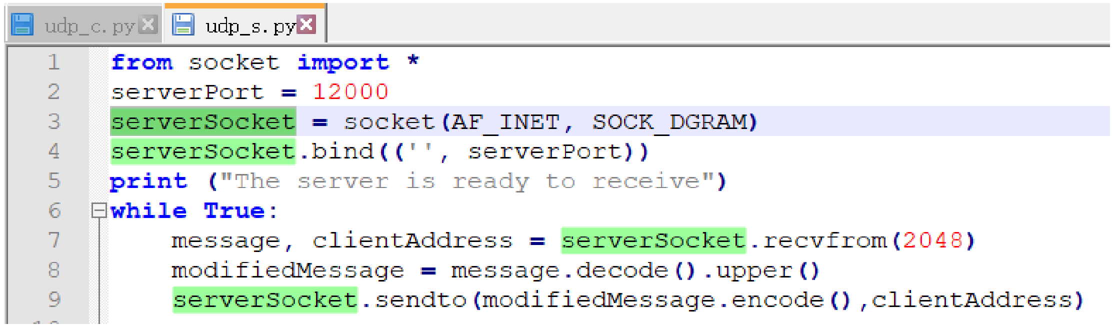
- Function:
  - Listen and accept DNS queries.
    - Support common query types:
      A, AAAA, CNAME, TXT, NS, MX
    - EDNS implementation is not required.
  - Forward query to a upstream DNS resolver (or a public DNS server).
  - Check out the response and send response to your clients.
  - Maintain a cache of DNS query-response of all results.
- Test method:
  - using dig sending query to your resolver
- *comments is MUST
- *DO NOT copy paste any existing source code of DNS resolver.

Tips for assignment 5.2

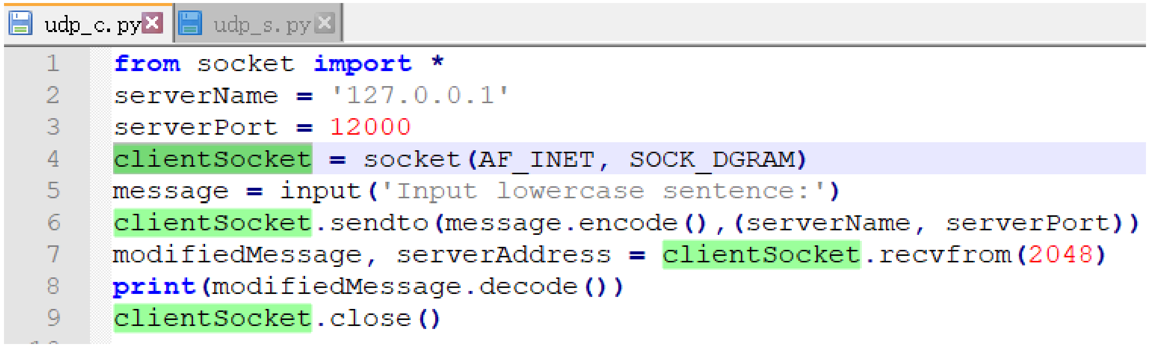
query in dns.resolver of python

- query(self, qname, rdtype=1, rdclass=1, tcp=False, source=None, raise_on_no_answer=True, source_port=0)
  - Query nameservers to find the answer to the question.
  - The qname, rdtype, and rdclass parameters may be objects of the appropriate type, or strings that can be converted into objects of the appropriate type. E.g. For rdtype the integer 2 and the the string 'NS' both mean to query for records with DNS rdata type NS.
- Parameters:
  - qname (dns.name.Name object or string) - the query name
  - rdtype (int or string) - the query type
  - rdclass (int or string) - the query class
  - tcp (bool) - use TCP to make the query (default is False).
  - source (IP address in dotted quad notation) - bind to this IP address (defaults to machine default IP).
  - raise_on_no_answer (bool) - raise NoAnswer if there's no answer (defaults is True).
  - source_port (int) - The port from which to send the message. The default is 0.

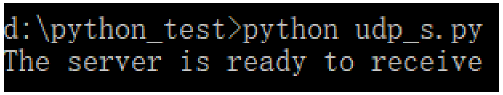
Tips for assignment 5.3



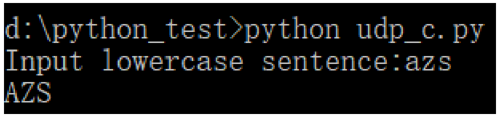
```python
from socket import *
serverPort = 12000
serverSocket = socket(AF_INET, SOCK_DGRAM)
serverSocket.bind(('', serverPort))
print ("The server is ready to receive")
while True:
    message, clientAddress = serverSocket.recvfrom(2048)
    modifiedMessage = message.decode().upper()
    serverSocket.sendto(modifiedMessage.encode(),clientAddress)
```



```python
from socket import *
serverName = '127.0.0.1'
serverPort = 12000
clientSocket = socket(AF_INET, SOCK_DGRAM)
message = input('Input lowercase sentence:')
clientSocket.sendto(message.encode(),(serverName, serverPort))
modifiedMessage, serverAddress = clientSocket.recvfrom(2048)
print(modifiedMessage.decode())
clientSocket.close()
```



```
d:\python_test>python udp_s.py
The server is ready to receive
```



```
d:\python_test>python udp_c.py
Input lowercase sentence:azs
AZS
```