CS 305 Lab Tutorial
Lab 5 DNS

Dept. Computer Science and Engineering
Southern University of Science and Technology

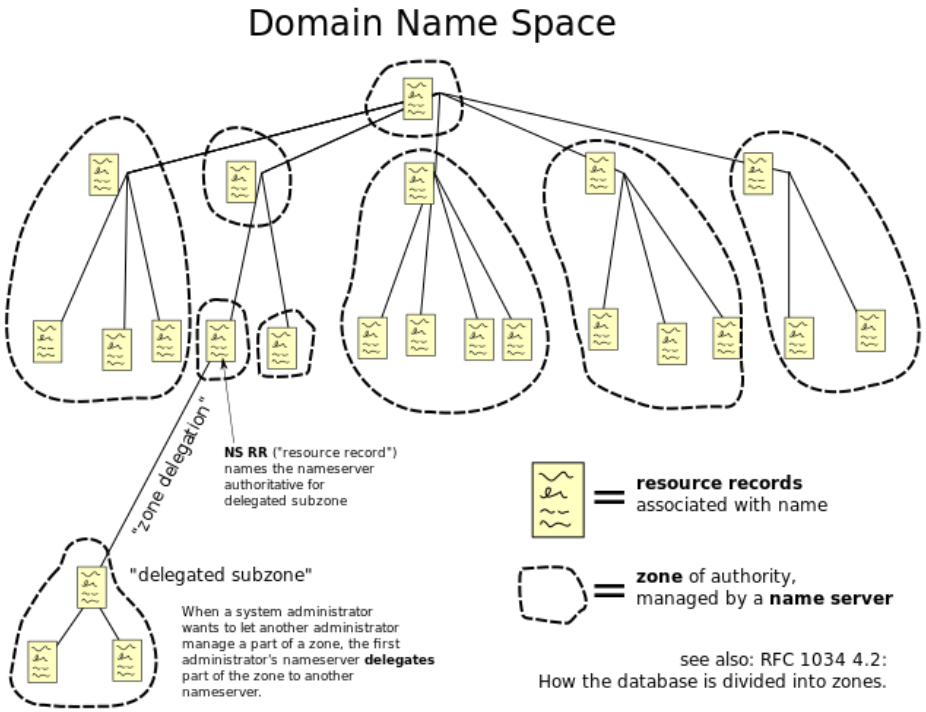
SUSTech
Southern University
of Science and Technology

Topic

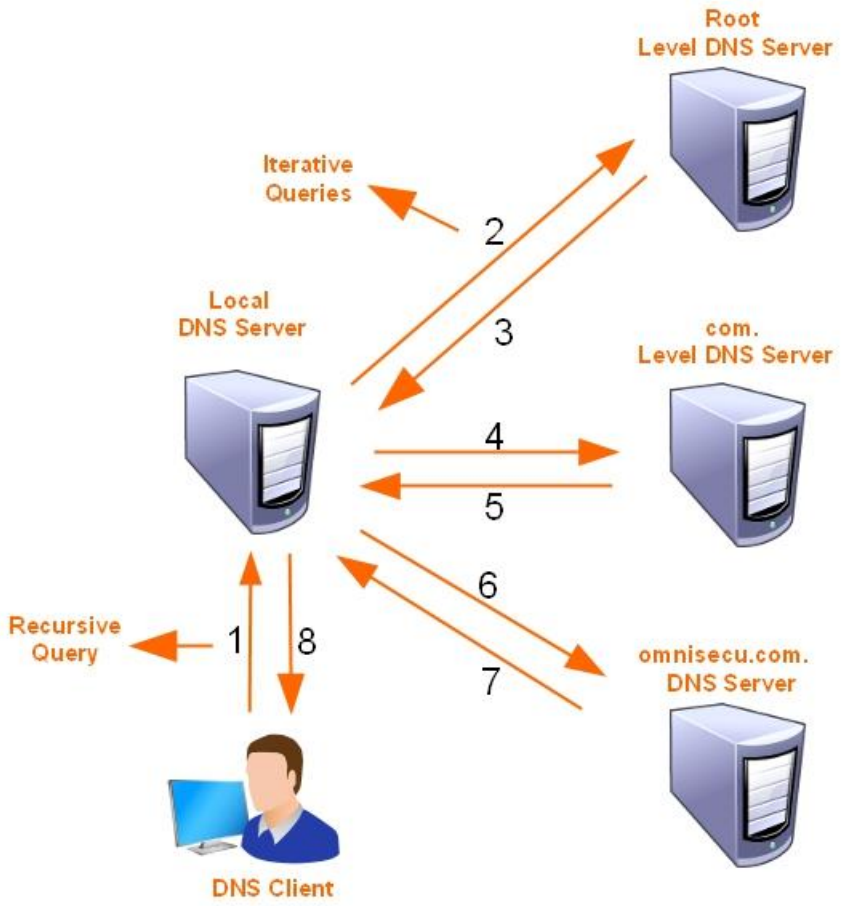
- DNS
  - DNS Message Structure
  - DNS Message head
  - RR in DNS
- EDNS (aka. Extension mechanisms for DNS)
  - DNSSEC
- Tool : dig

Part A.1
Domain Name System

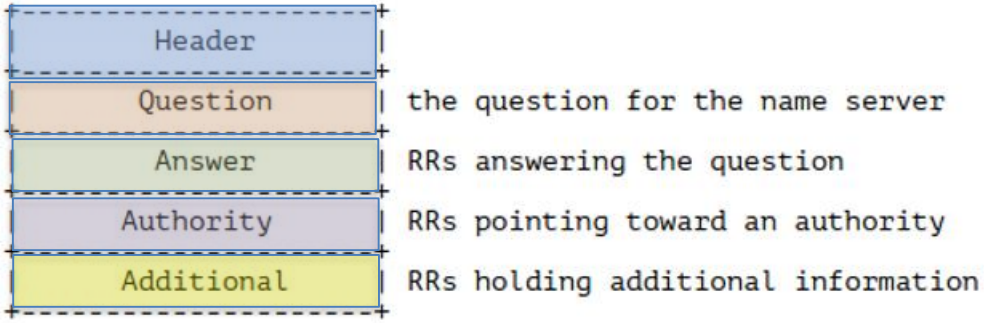
- DNS is a distributed database.

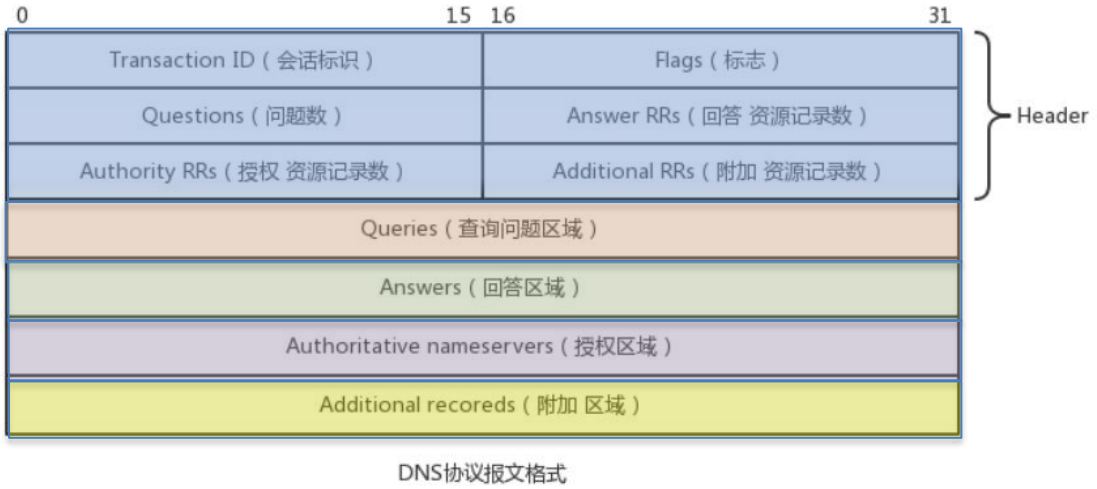


Recursive/Iterative Query



Part A.2
DNS Message Structure





DNS协议报文格式

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

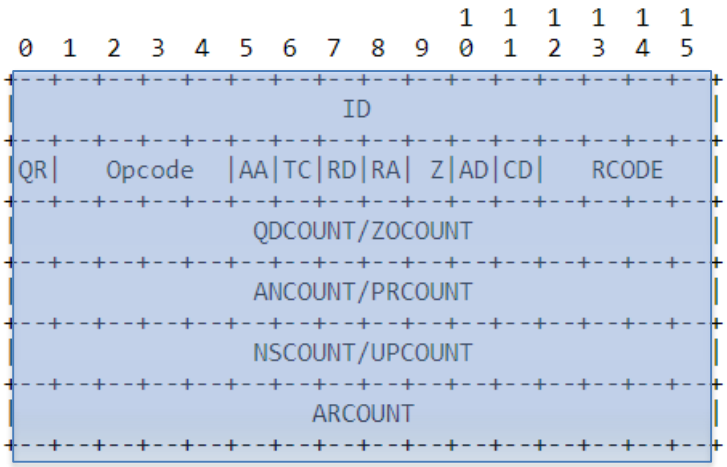
IANA Reference: https://www.iana.org/assignments/dns-parameters/dns-parameters.xhtml

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, Authentic Data, Checking Disabled.
- Z is a reserved flag.

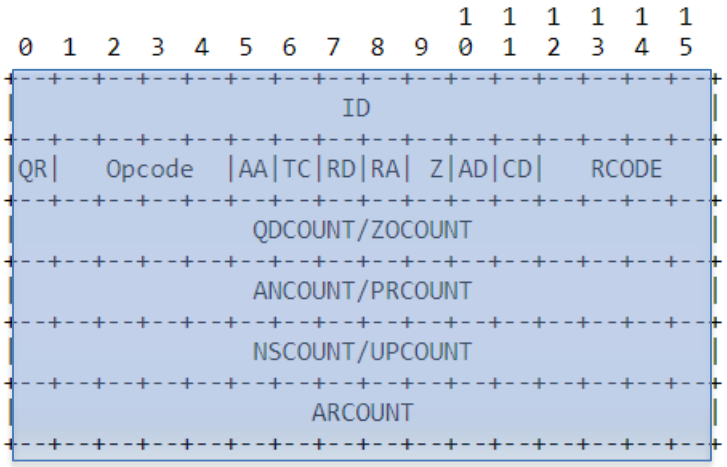
OpCode	Name	Reference
0	Query	[RFC 1035]
1	IQuery (Inverse Query)	[RFC 1035]
2	Status	[RFC 1035]
3	available for assignment	
4	Notify	[RFC 1996]
5	Update	[RFC 2136]
6-15	available for assignment	



SUSTech
Southern University of Science and Technology

Example Structure Code in C:

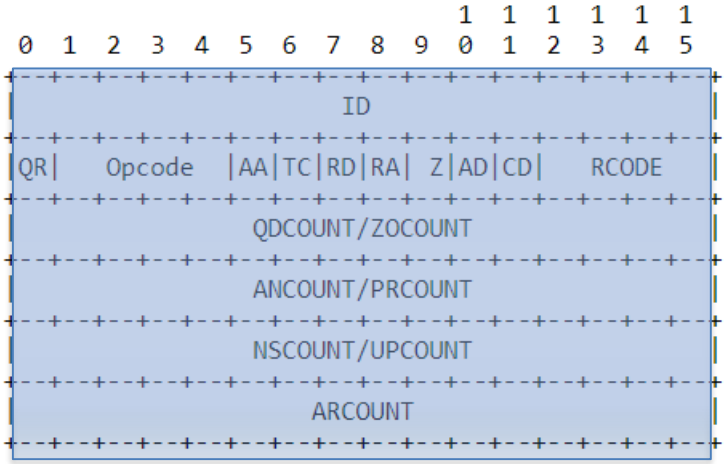
```
struct DNS_HEADER {     //DNS header structure
   unsigned short id;       // identification number

   unsigned char qr :1;      // query/response flag
   unsigned char opcode :4;   // purpose of message
   unsigned char aa :1;      // authoritive answer
   unsigned char tc :1;      // truncated message
   unsigned char rd :1;      // recursion desired
   unsigned char ra :1;      // recursion available
   unsigned char z :1;       // its z reserved
   unsigned char ad :1;      // authenticated data
   unsigned char cd :1;      // checking disabled
   unsigned char rcode :4;    // response code

   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
};
```



SUSTech
Southern University of Science and Technology

Decode Message Header in Python

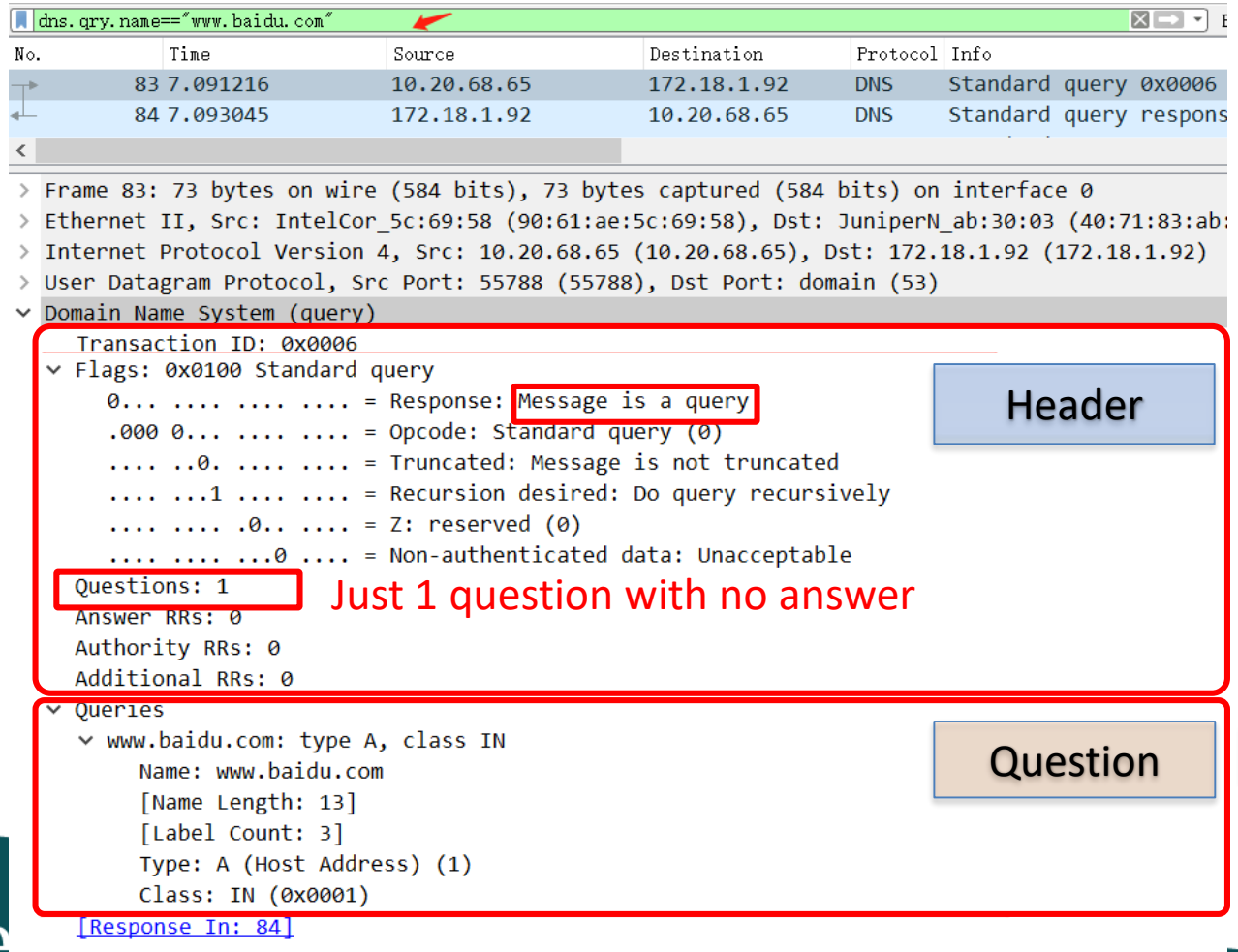
```
class DNSHeader:
    Struct = struct.Struct('!6H')

    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.ARcount = 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__))
```



SUSTech
Southern University of Science and Technology

A query message of DNS

nslookup www.baidu.com

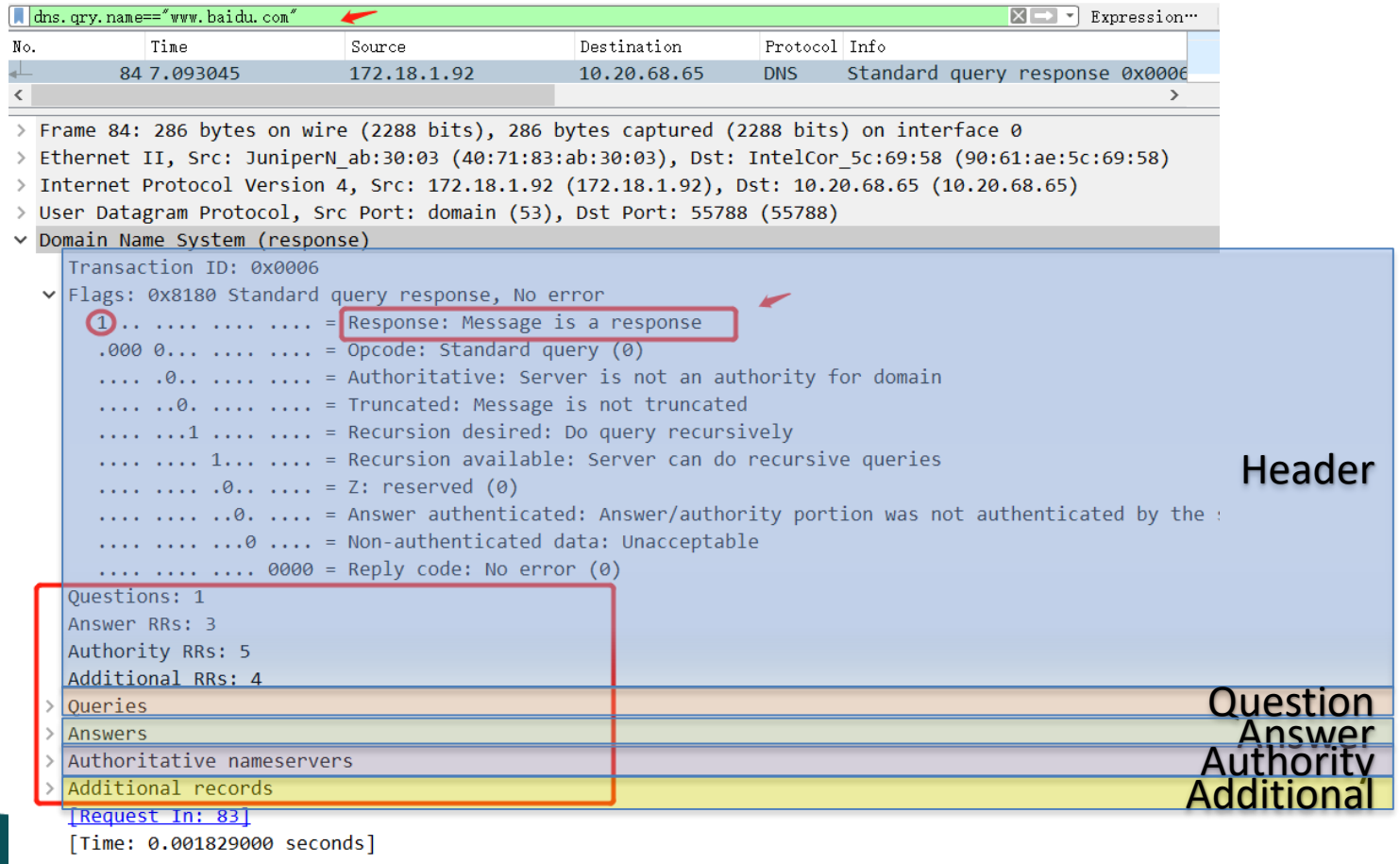


SUSTech
Southern University of Science and Technology

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

A response message of DNS

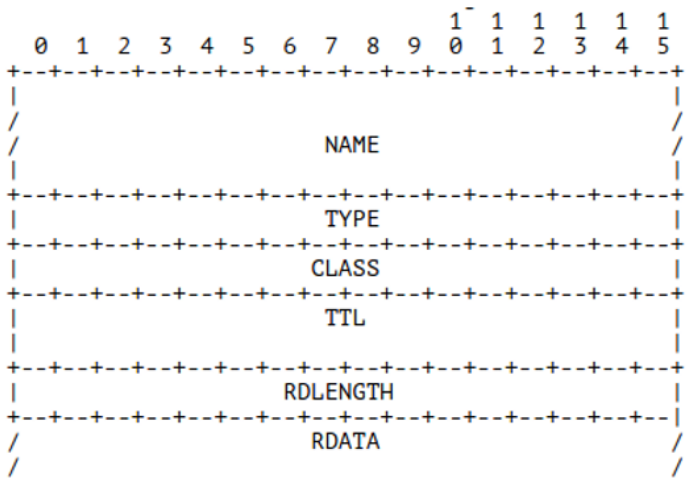
nslookup www.baidu.com



SUSTech
Southern University of Science and Technology

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

Part A.3
RR in DNS

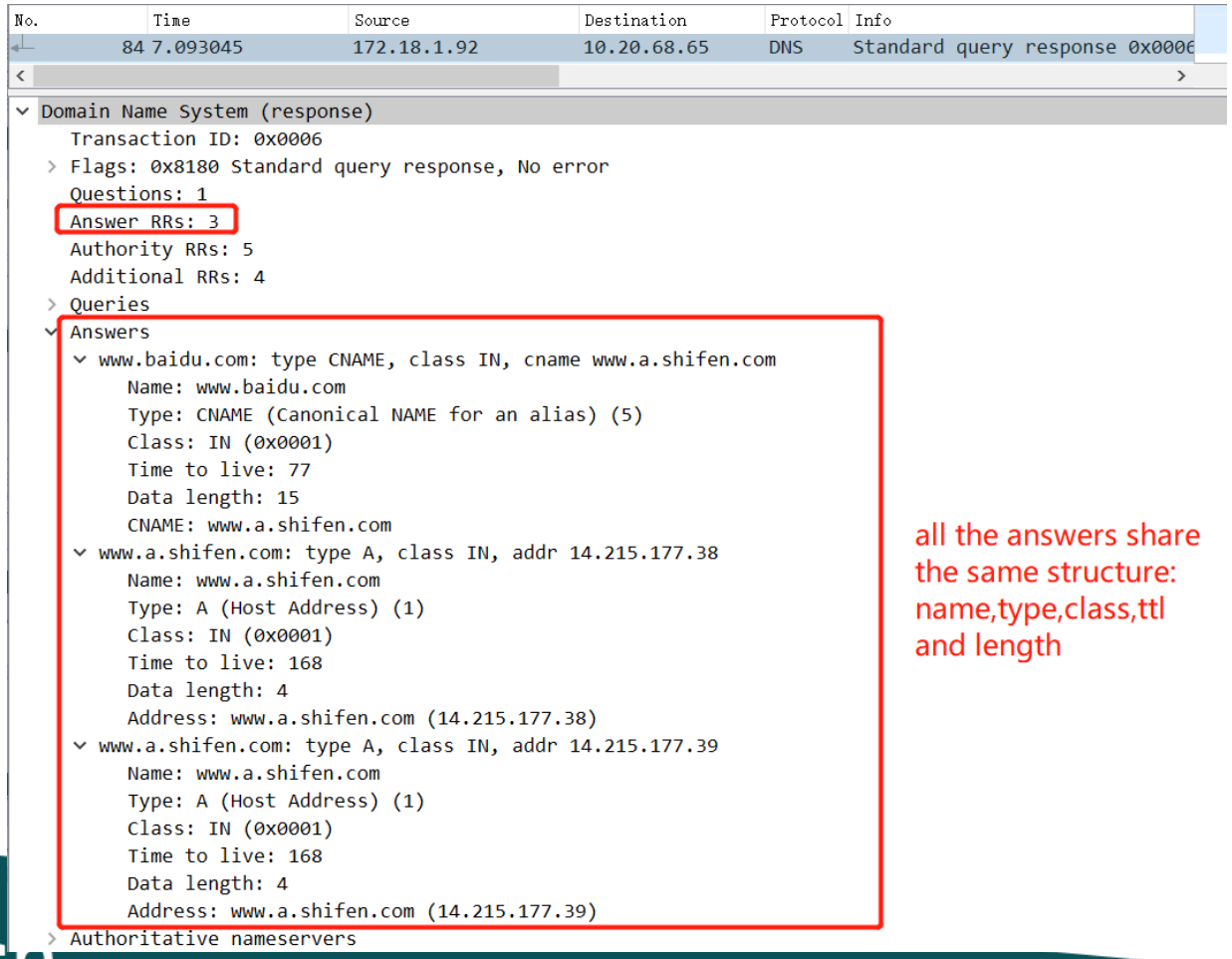


Resource record (RR) fields

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

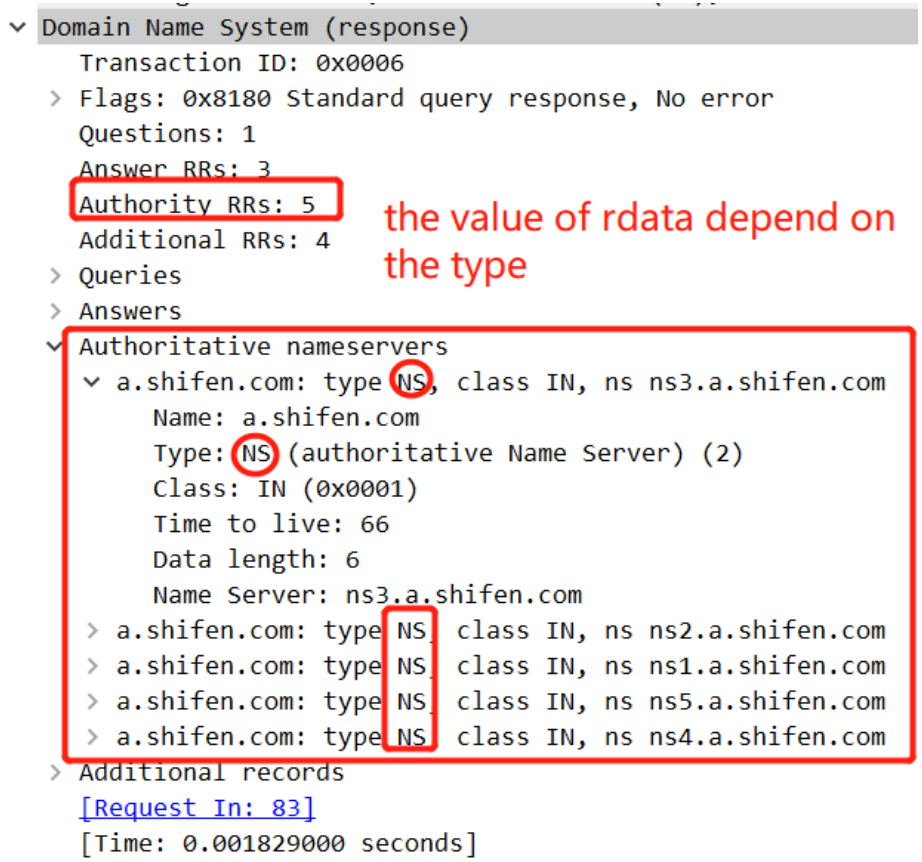
nslookup www.baidu.com



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

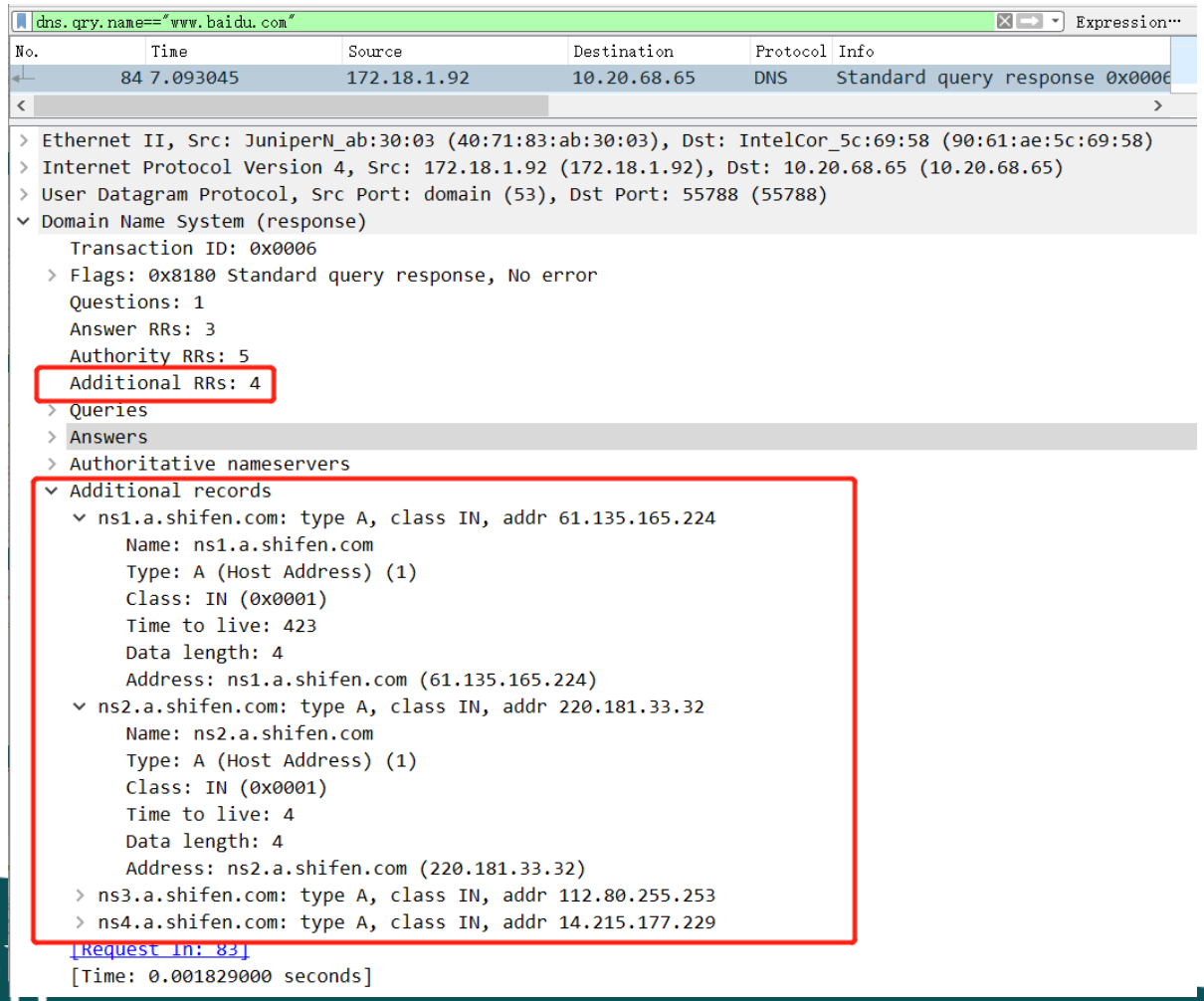
RRs of authoritative name servers

nslookup www.baidu.com



RRs of Additional records

nslookup www.baidu.com



Part B
EDNS (aka. Extension mechanisms for DNS)

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

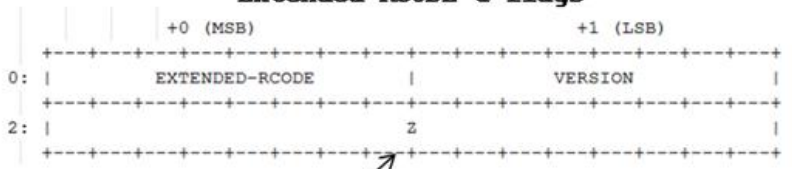
- The Domain Name System's wire protocol includes a number of fixed fields whose range has been or soon will be exhausted and does not allow clients to advertise their capabilities to servers.
- DNS (see [RFC1035]) specifies a Message Format and within such messages there are standard formats for encoding 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

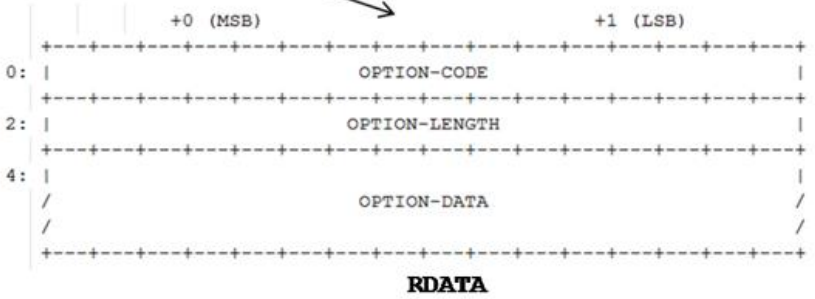
One **OPT pseudo-RR** can be added to the **additional data section** of either a request or a response. An OPT is called a pseudo-RR because it pertains to a particular transport level message and not to any actual DNS data.

Extended RCODE & flags



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



RDATA

EDNS query

```
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
```

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

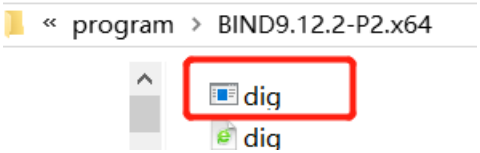
EDNS response

```
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
```

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

Part C dig

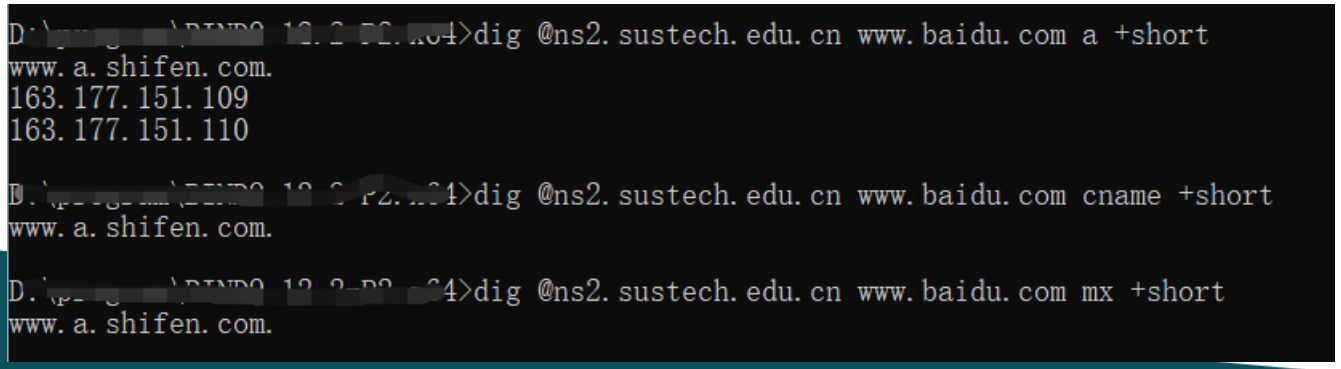
- **dig** is a flexible tool for interrogating DNS name servers.
  - It performs DNS lookups and displays the answers that are returned from the name server(s) that were queried.
  - Most DNS administrators use **dig** to troubleshoot DNS problems because of its flexibility, ease of use and clarity of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



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

Using dig

- A typical invocation of dig looks like: dig @server name type
  - **server**: the name or IP address of the name server to query. This can be an IPv4 address or an IPv6 address. When the supplied server argument is a hostname, dig resolves that name before querying that name server.
  - **name:** 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. and A record is default when using dig.
  - Some useful options: -h, +tcp, +noedns, +bufsize,+trace, .etc.



SUSTech
Southern University
of Science and Technology

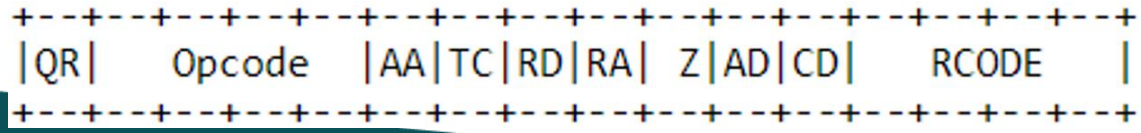
Tip 1

- Use nslookup command to check your local DNS server.

- Some common public DNS server
  - 114 DNS Preferred address: 114.114.114.114, alternative address: 114.114.115.115
  - Baidu DNS IPv4 address:180.76.76.76, IPv6 address: 2400:da00::6666
  - Google Public DNS IPv4 address: 8.8.8.8, DNS IPv6: 2001:4860:4860::8888

Practice 5.1

- Make the query of "www.sina.com.cn" by using "dig" with option "+trace" .
- Capture screenshots on the command and its output, answer the questions by analyzing the packets:
  - How many queries are sent form the local host? Do they share the same " transaction id"?
  - How many responses are received by local host?
  - what's the value of 'RD' field in query? what's the value of 'RA' field in response from the local DNS server?
  - which server sent the the last response, is it the local DNS server of the local host or is it the Authoritative DNS server?
    - List the name, IP address and port number of the server which sent the last response to the local host.
    - Is there any answer in the response? What's the value of 'AA' field in this response ?
  - try the same query again
    - At this time, is the last response from the same server as the last response in the previous query? If they are different, what is the reason for this? will it bring any benefits?



Tip 2

- Use “ipconfig /displaydns” to list DNS buffer of local host.

```
C:\Users\wq>ipconfig /displaydns

Windows IP 配置

    ping.pinyin.sogou.com
    ----------------------------------------
    记录名称. . . . . : ping.pinyin.sogou.com
    记录类型. . . . . : 5
    生存时间. . . . . : 36
    数据长度. . . . . : 8
    部分. . . . . : 答案
    CNAME 记录 . . . . . : ping.sogou.com
```

```
sakai.sustech.edu.cn
----------------------------------------
记录名称. . . . . : sakai.sustech.edu.cn
记录类型. . . . . : 1
生存时间. . . . . : 89
数据长度. . . . . : 4
部分. . . . . : 答案
A (主机)记录 . . . . : 116.7.234.94
```

- Use “ipconfig /flushdns” to clear DNS buffer.

```
C:\Users\wq>ipconfig /flushdns

Windows IP 配置

已成功刷新 DNS 解析缓存。
```

Practice 5.2

- Make an DNS query for “www.bilibili.com” with 'EDNS0' option.
- Capture the packages using Wireshark and answer the following questions:
  - What is the content of the query message
    - what's the destination IP address and destination port of the query?
    - what's the name, type and class of this query?
    - what's the opcode of these query, what does it mean?
    - Is there any additional RR, what’s the type of the RR?
  - What is the content of the response message
    - Is there any answers, what’s life time of each answer?
    - Is there any authority RRs, what’s the type of each RR?

Practice 5.3
Implement a DNS client

- Function:
  - Invoke DNS queries.
    - Support common query types: A, AAAA, CNAME, NS, MX
    - EDNS implementation is not required.
    - Implement both queries with 'RD' equals to 0 and 1.
  - Check out the response.
    - Display the answer, check who sends the answer? Whether the answer is from authority Name Server or not? How to judge that?
  - Do the iterative query while 'RD' equals to 0 to get the desired answer which is identified in the query.
- Tips:
  - using “dig” with “+trace” option to invoke the iterative query, using Wireshark to capture the packets of the iterative query.
  - while using “dig” with “+trace” option, the desired answer may not be got. for example: the desired answer is A type record in query while “dig” ends the iterative query while it got the 1st answer which maybe the CNAME type.

Tip 3: Using dns.resolver of python(1)

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 information 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
>>>
```

SUSTech
Southern University of Science and Technology

Tip 3: Using dns.resolver of python(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.

Tip 4: UDP socket progamming

UDP Server

udp_c.py udp_s.py

```
1  from socket import *
2  serverPort = 12000
3  serverSocket = socket(AF_INET, SOCK_DGRAM)
4  serverSocket.bind(('', serverPort))
5  print ("The server is ready to receive")
6  while True:
7      message, clientAddress = serverSocket.recvfrom(2048)
8      modifiedMessage = message.decode().upper()
9      serverSocket.sendto(modifiedMessage.encode(),clientAddress)
```

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

UDP Client

udp_c.py udp_s.py

```
1  from socket import *
2  serverName = '127.0.0.1'
3  serverPort = 12000
4  clientSocket = socket(AF_INET, SOCK_DGRAM)
5  message = input('Input lowercase sentence:')
6  clientSocket.sendto(message.encode(),(serverName, serverPort))
7  modifiedMessage, serverAddress = clientSocket.recvfrom(2048)
8  print(modifiedMessage.decode())
9  clientSocket.close()
```

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