

CS 305 Lab Tutorial

Lab 5 DNS

ipconfig /displaydns
/flashdns

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

cashe) forward

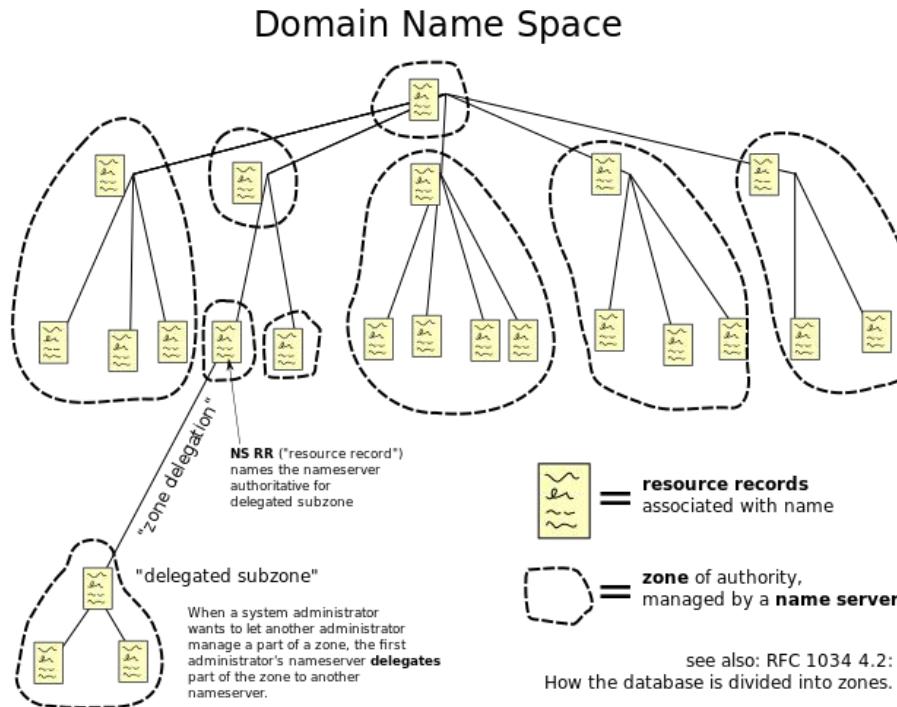
Topic

- 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**.

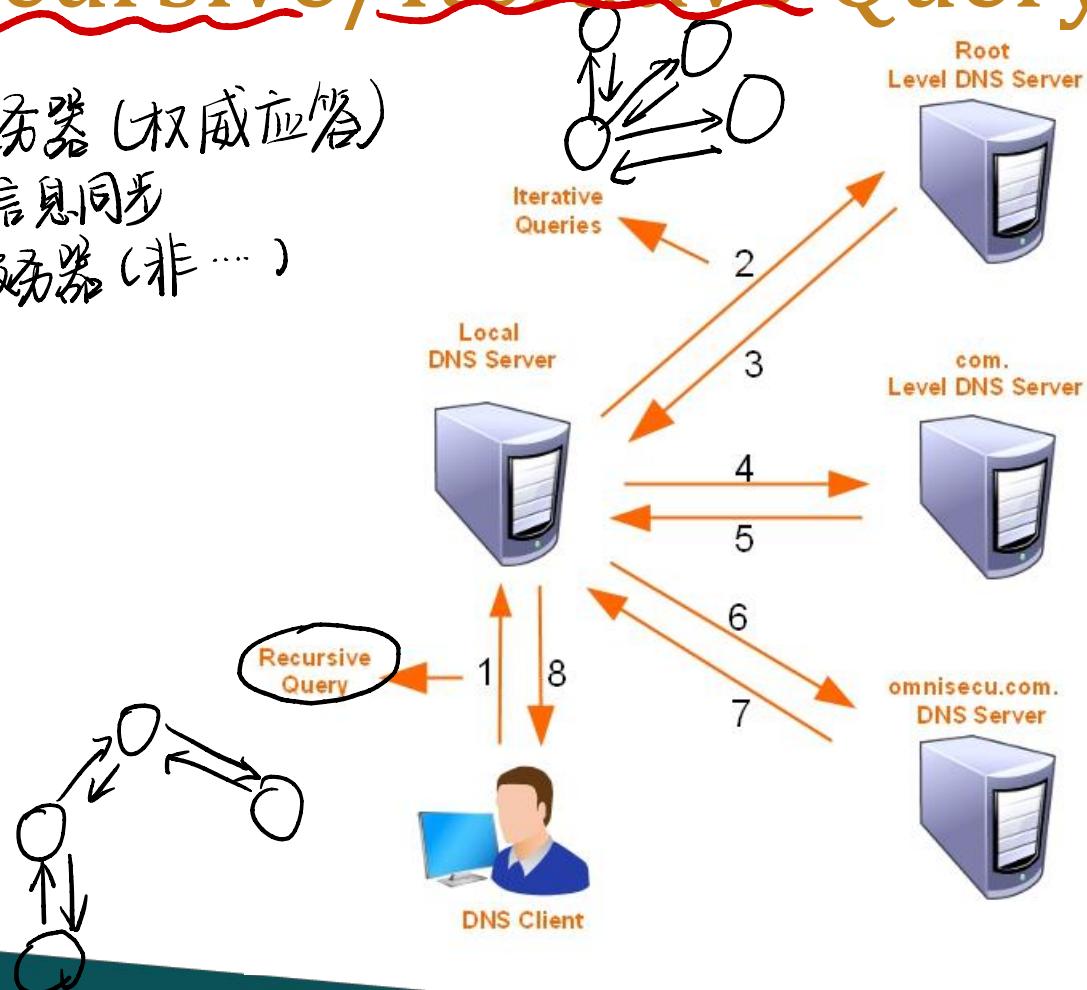


Recursive/Iterative Query

主域名服务器 (权威应答)

信息同步

从域名服务器 (非 ...)



Part A.2 **udp port 53** 捕获过滤器

DNS Message Structure

① 查询 DNS 服务器的域名 (type=PTR)

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



A query message of DNS

nslookup www.baidu.com

The screenshot shows a Wireshark capture of a DNS query message. The packet details pane shows two entries:

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 response

The selected packet (83) is expanded in the tree view:

- 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:00:00)
- 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)
- Domain Name System (query)
 - Transaction ID: 0x0006
 - Flags: 0x0100 Standard query
 - 0... = Response: Message is a query ← QR; 0 - query, 1 - response
 - .000 0... = Opcode: Standard query (0) ← 0 标准查询；1 反向查询；2 服务器状态请求
 -0. = Truncated: Message is not truncated
 - 1 = Recursion desired: Do query recursively ← RD
 -0. = Z: reserved (0)
 - 0 = Non-authenticated data: Unacceptable
 - Questions: 1
 - just 1 question with no answer 表示响应超过512字节，并被截断
 - Answers: 0
 - Authorities: 0
 - Additional: 0
- Queries
 - 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]

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

A response message of DNS

nslookup www.baidu.com

dnsqry.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)
▼ Domain Name System (response)
 Transaction ID: 0x0006
 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 server
 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]

AA

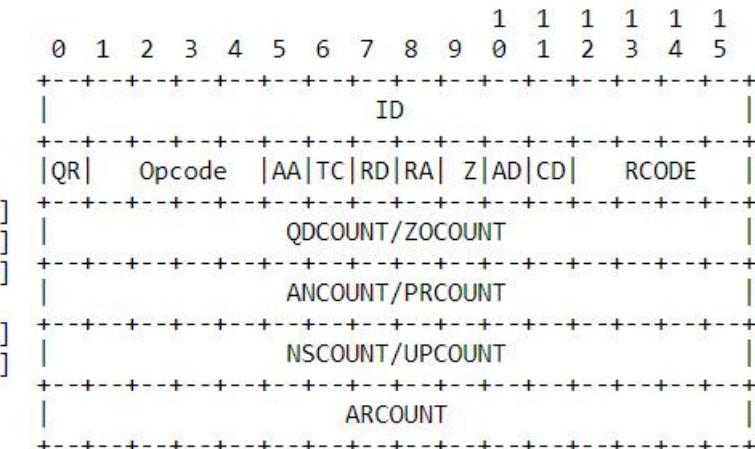
该域名的权威名解析器列表

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	

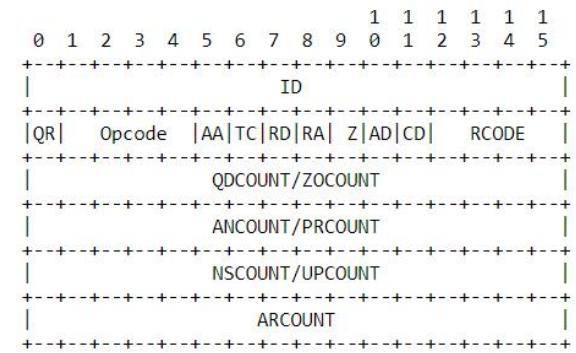


Example Structure Code in C:

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

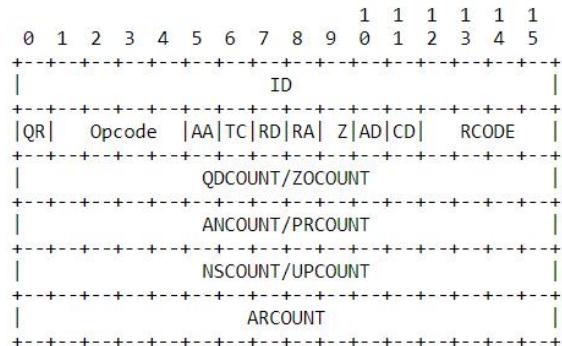
    unsigned char qr :1; // query/response flag
    unsigned char opcode :4; // purpose of message
    unsigned char aa :1; // authoritative answer
    unsigned char tc :1; // truncated message
    unsigned char rd :1; // recursion desired
    unsigned char ra :1; // recursion available
    unsigned char 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
};
```



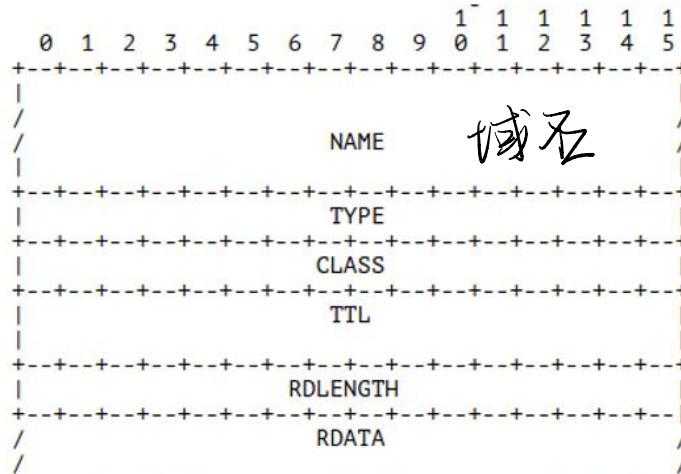
Decode Message Header in Python

```
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_))
```



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

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

查詢規範五。
— 查詢規範五對應的IP

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 the value of rdata depend on the type
  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]
```

a.shifen.com
权威的NS

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)
<           >
<           > 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
<           > Additional records
<           > 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)
<           > 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 id: 83]
[Time: 0.001829000 seconds]
```

获取权威命名服务器的IP

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>

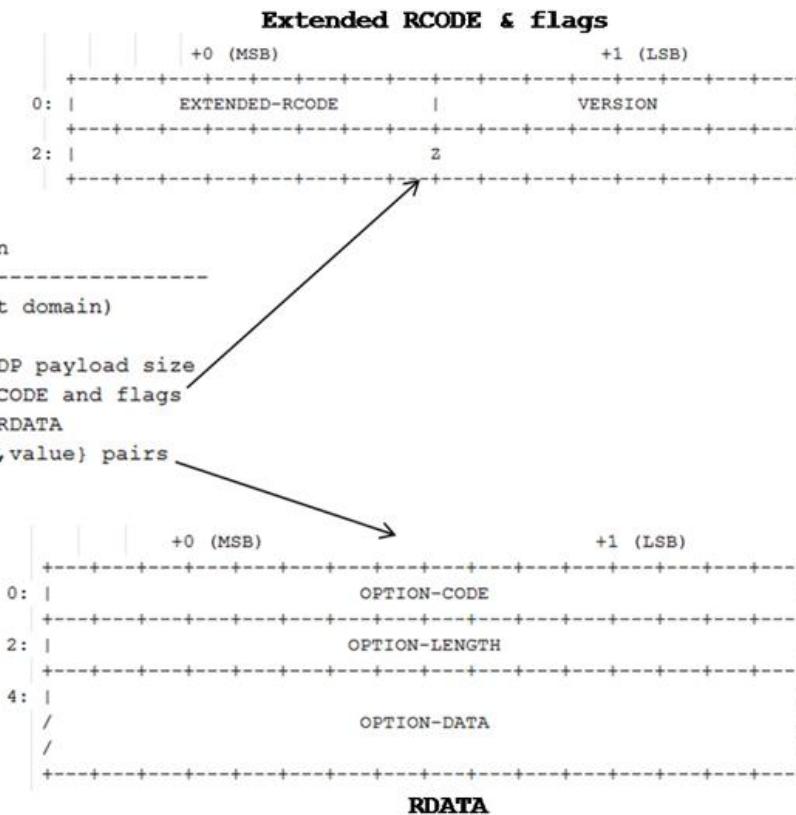
EDNS 在遵循已有 DNS 消息格式基础上增加一些字段.

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.

伪资源记录

Field Name	Field Type	Description
<hr/>		
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



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

Additional records

> <Root>: type OPT

Name: <Root>

Type: OPT (41) 伪查询

UDP payload size: 4096

Higher bits in extended RCODE: 0x00

EDNS0 version: 0

> 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

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

< Root: type OPT

Name: <Root>

Type: OPT (41)

UDP payload size: 4096

Higher bits in extended RCODE: 0x00

EDNS0 version: 0

Z: 0x0000

0... = DO bit: Cannot handle DNSSEC security RRs

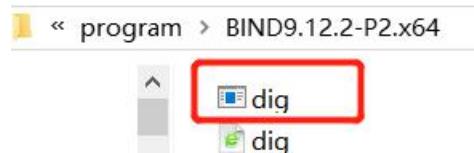
.000 0000 0000 0000 = Reserved: 0x0000

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

dig(1)

- **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(2)

A typical invocation of dig looks like:

dig @server name type

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 Files\bind\bin>dig @ns2.sustech.edu.cn www.baidu.com a +short
www.a.shifen.com.
163.177.151.109
163.177.151.110

D:\Program Files\bind\bin>dig @ns2.sustech.edu.cn www.baidu.com cname +short
www.a.shifen.com.

D:\Program Files\bind\bin>dig @ns2.sustech.edu.cn www.baidu.com mx +short
www.a.shifen.com.
```

lots of useful options like: +tcp, +noedns, +bufsize,+trace,.etc. have a try!

Practise 5.1

dig +trace @nameServer name -b ip#port
udp. port=port ← 过滤 指定本机端口

- Make the query of “www.sina.com.cn” by using “dig” with option “+trace”
- Screenshot on the command and its output . answer the questions by analysis the packets:
 - what's the value of 'rd' field in query? what's the value of 'ra' field in response from the local DNS server?
 - how many queries are sent form the local host? Do they share the same “transaction id”?
 - which server sent 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 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
 - 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? *distributed one, easy the load.*

Practise 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?
172.18.1.92 53
 - what's the name, type and class of this query
www.bilibili.com A In
 - what's the opcode of these query, what does it mean? (0-standard query)
 - Is there any additional RR, what's the type of the RR? *OPT(4)*
 - what is the content of the response message
 - Is there any answers, what's life time of each answer?*78S 93S ...*
 - Is there any authority RRs, what's the type of each RR?
13 NS

Practise 5.3

implement a DNS client

- Function:
 - invoke DNS query.
 - Support common query types: A, AAAA, CNAME, NS, MX
 - EDNS implementation is not required.
 - implement both 'RD' of query is set and not set.
 - Check out the response.
 - display the answer, who send the answer, if the answer is from authority Name Server.
 - Do the Iterative query while 'RD' is not set 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.

Tips: 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:\Users\wini>pip install dnspython ←
Collecting dnspython
  Downloading https://files.pythonhosted.org/packages/a6/72/209e18bdfef78c6994e9ec96981624a5ad7738524dd474237268422cb
/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 info 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
>>>
```

Tips: 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 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: UDP socket programming

```
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)
```

```
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_s.py
The server is ready to receive
```

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

Tips: UDP socket programming

```
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)
```

```
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_s.py
The server is ready to receive
```

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