

CS305 Programming Assignment1 [中文]

代码解读

基本设计

DNS解析

[最困难部分] DNS编码

DNS服务器

基本设计

Answer段仅CNAME情况处理

测试结果

姓名：黄政东

SID: 12212230

代码解读

基本设计

本项目主要基于所提供的Python模板进行具体功能的实现，主要分为DNS数据处理与DNS服务器两大部分，同时在DNS数据处理逻辑中抽取了可复用的代码作为py文件下的静态函数

DNS解析

DNS解析部分主要实现了DNSHeader、DNSQuestion、DNSRR、DNSMessage四个类下的类方法from_wire，其中其中前三个类的from_wire方法将读取字节流data与当前数据起始位置的偏移量idx（默认idx为0以供测试），随后该函数将会解析数据并返回存有数据的对象。DNSMessage的from_wire方法将调用前三个类的from_wire方法逐步进行DNS消息的解析并创建完整的消息体对象

[最困难部分] DNS编码

为实现后续DNS服务器发送特定查询的需求，本项目同时扩展了DNS数据处理的相关模块，使其支持将DNS消息对象重新编码为字节流，同时对于域名采用了压缩格式以降低字节流大小。该功能的实现为整个项目中最为困难的部分。

为实现DNS编码功能，本项目为DNSHeader、DNSQuestion、DNSRR、DNSMessage四个类添加了较为统一的encode_bytes方法，外部通过调用该方法即可将对象内数据编码为字节流。

DNSMessage类的encode_bytes方法实现如下

```
▼ DNSMessage Python |
1 def encode_bytes(self) -> bytes:
2     res = self.header.encode_bytes()
3     domain_dict = {}
4     seg, domain_dict = self.question.encode_bytes(len(res), domain_dict)
5     res += seg
6     for rr in self.answer + self.authority + self.additional:
7         seg, domain_dict = rr.encode_bytes(len(res), domain_dict)
8         res += seg
9     return res
```

该方法首先调用self.header.encode_bytes()方法将DNSHeader部分转化为字节流，随后创建了一个domain_dict = {}对象用于存储域名段与其第一次出现的位置，随后逐次调用DNSQuestion对象和DNSRR对象的encode_bytes方法，该方法将接收两个参数，第一个参数表示 当前数据段起始位置位于最终字节流中所处位置的偏移量，第二个参数表示 先于该数据段中所出现的域名与偏移量，随后该方法将会返回 该数据段的对应字节流 与 更新后的域名-偏移量字典。通过逐步拼接字节流段与更新域名-偏移量字典，该方法将最终获得并返回DNS消息体的字节流

DNSHeader类的encode_bytes方法实现如下，该方法的实现逻辑较为简单，通过将属性转化成字节流，并依据DNS首部格式逐步拼接，该方法最终获得并返回数据段对应的字节流

```
▼ DNSHeader Python |
1 def encoder_bytes(self) -> bytes:
2     id_bytes = self.id.to_bytes(2, byteorder='big')
3     qdcount_bytes = self.qdcount.to_bytes(2, byteorder='big')
4     ancourt_bytes = self.ancourt.to_bytes(2, byteorder='big')
5     nscount_bytes = self.nscount.to_bytes(2, byteorder='big')
6     arcount_bytes = self.arcount.to_bytes(2, byteorder='big')
7     return (id_bytes + self.flag + qdcount_bytes +
8           ancourt_bytes + nscount_bytes + arcount_bytes)
```

DNSRR与DNSQuestion的encode_bytes方法实现较为相似，此处展示DNSQuestion的encode_bytes方法

```
1 def encode_bytes(self, previous_len: int, previous_dict: Dict[str, int]) \
2 -> Tuple[bytes, Dict[str, int]]:
3     qname_bytes, _, previous_dict = encode_dns_name(self.qname, previous_le
4     n,
5                                     previous_dict)
6     qtype_bytes = self.qtype.to_bytes(2, byteorder='big')
7     qclass_bytes = self.qclass.to_bytes(2, byteorder='big')
8     return qname_bytes + qtype_bytes + qclass_bytes, previous_dict
```

该方法同样依据格式逐步解析并最终拼接得对应字节流，其中在域名解析部分调用了`encode_dns_name`静态方法进行解析，该方法是实现域名压缩编码的重要实现，其代码如下

```

1  def encode_dns_name(name: str, prev_len: int, prev_dict: Dict[str, int]) -
    > Tuple[bytes, int, Dict[str, int]]:
2      if name == 'root':
3          return b'\x00', 1, prev_dict
4      name = '.' + name
5      res = b''
6      i = 0
7      while i < len(name):
8          if name[i] != '.':
9              i += 1
10             continue
11         key = name[i + 1:]
12         if key in prev_dict: # 指针模式
13             val = prev_dict[key] + (192 << 8)
14             res += val.to_bytes(2, byteorder='big')
15             return res, len(res), prev_dict
16             # 不用在指针后加全0chunk
17         # 普通模式
18         i += 1
19         begin = i
20         while i < len(name) and name[i] != '.':
21             # 找到chunk的len,注意这个chunk找到i停留在.上,不用++
22             i += 1
23         seg = name[begin:i]
24         a = len(res)
25         res += len(seg).to_bytes(1, byteorder='big')
26         res += seg.encode('utf-8')
27         # assert a + len(seg) + 1 == len(res)
28         # 添加name[begin:]到prev_dict中,其位置为begin-1+prev_len
29         prev_dict[name[begin:]] = begin - 1 + prev_len
30     return res + b'\x00', len(res) + 1, prev_dict

```

由于域名str存在三种情况：第一种为root域名，此时需要将域名编码为b'\x00'，同时由于指针类型的域名编码占用字节数大于根服务器域名编码，因此当root域名重复出现时，不需要使用指针类型编码降低字节流长度，因此不更新prev_dict。

随后代码遍历域名并以.符号作为分隔，对于当前循环所需编码的一段域名（如 www.baidu.com 中，包含有 www.baidu.com, baidu.com, .com 三段域名段），在prev_dict中查找是否先前出现过该段域名，如果存在则在域名字节流中添加指针类型编码指向字典中所存储的偏移量并返回最终结果。如果不存在，则编码一节域名（如 www.baidu.com 编码www.节后剩余域名段为 baidu.com），将该域名段与字节流起始位置偏移量添加至字典中，并继续循环编码剩余的域名段直至完成编码。

上述代码实现了域名编码的压缩过程，有效降低了最终DNS消息的字节流长度。

在DNS解析与编码的部分，正确实现特殊数据的解析过程，如域名解析与编码，ipv6缩写与全写的转换等功能不仅十分重要，且由于字节流的不易读性存在一定的难度，本项目同时使用unittest对相关接口功能进行了测试以保障实现的正确性与可靠性，该模块的部分代码如下

```
Python |  
  
1 class TestStaticFunc(unittest.TestCase):  
2     def test_encode_and_decode(self):  
3         b, i, d = encode_dns_name('baidu.com', 0, {})  
4         b2, i2, d = encode_dns_name('www.baidu.com', i, d)  
5         b += b2  
6         str, _ = parse_dns_name(b, i)  
7         assert str == 'www.baidu.com'  
8  
9     def test_ipv6_shorten(self):  
10        assert shorten_ipv6("0:0:0:0:0:0:0:0") == '::'  
11        assert shorten_ipv6("0:0:0:0:1:0:0:1") == '::1:0:0:1'  
12        assert shorten_ipv6("0:0:0:0:1:0:0:0") == '::1:0:0:0'  
13        assert shorten_ipv6("1:0:0:0:1:0:0:0") == '1::1:0:0:0' or shorten_  
    ipv6("1:0:0:0:1:0:0:0") == '1:0:0:0:1::  
14        assert shorten_ipv6("1:0:0:0:1:1:1:0") == '1::1:1:1:0'  
15        assert shorten_ipv6("1:1:1:1:1:0:0:0") == '1:1:1:1:1::  
16        assert shorten_ipv6("1:0:0:0:1:0:0:114") == '1::1:0:0:114'  
17  
18     def test_ipv6_extend(self):  
19        assert extend_ipv6('1::1') == '1:0:0:0:0:0:0:1'  
20        assert extend_ipv6('::') == '0:0:0:0:0:0:0:0'  
21        assert extend_ipv6('1::') == '1:0:0:0:0:0:0:0'  
22        assert extend_ipv6('::1') == '0:0:0:0:0:0:0:1'  
23        assert extend_ipv6('1::1:0:0:0') == '1:0:0:0:1:0:0:0'  
24        assert extend_ipv6('1:0:0:0:1::') == '1:0:0:0:1:0:0:0'  
25        assert extend_ipv6('1:1:1:1:1::') == '1:1:1:1:1:0:0:0'  
26        assert extend_ipv6('1::1:0:0:114') == '1:0:0:0:1:0:0:114'  
27  
28     def test_message(self):  
29        dns_response_bytes = read_bytes_from_file('./raw_packet/dns_respon  
    se.raw')  
30        msg = DNSMessage.from_wire(dns_response_bytes)  
31        b = msg.encode_bytes()  
32        msg2 = DNSMessage.from_wire(b)  
33        assert msg == msg2
```

DNS服务器

基本设计

DNS服务器部分实现了执行迭代查询的LocalDNS服务器，同时实现了Answer段仅CNAME，查询无Answer与Additional段的特殊情况处理。

本项目主要设计并实现了iterative_query方法，该方法接收一个DNS请求与请求IP地址，其内部将执行迭代查询并保证最终返回一个在Answer字段存在所请求域名对应A类型RR的DNS响应，从而实现了迭代查询的功能。

Answer段仅CNAME情况处理

若响应代码中只存在CNAME类型的RR，则表明需要进一步对别名域名的ip地址进行查询。该功能的代码实现如下

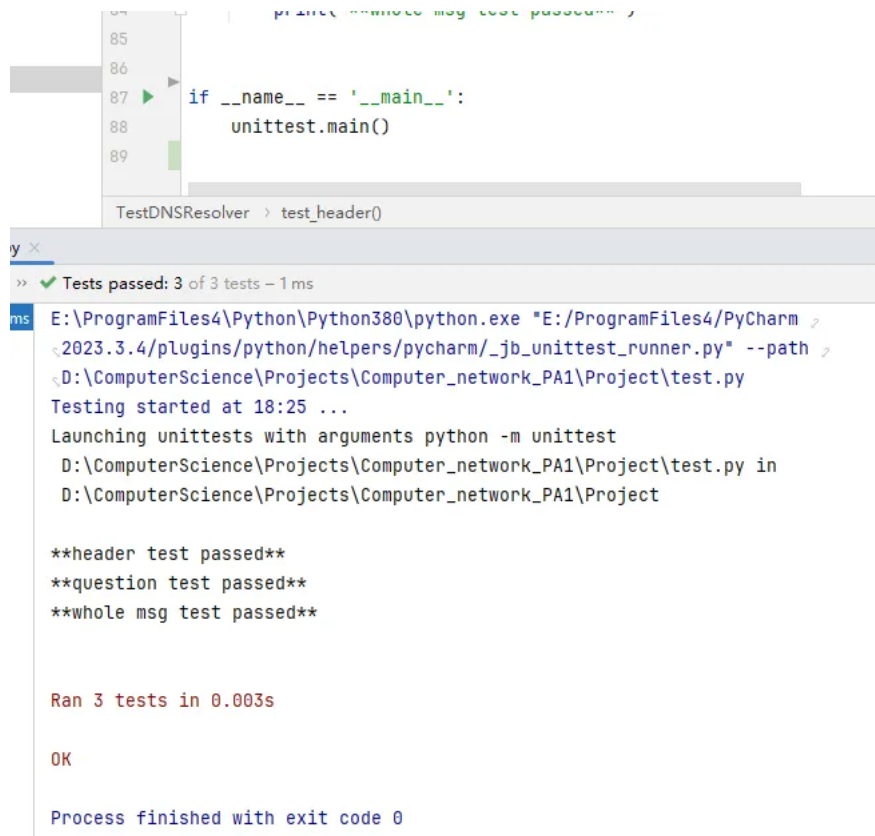
```
Python |  
  
1 if len(resp_message.answer) > 0: # got answer  
2     cname = None  
3     for rr in resp_message.answer:  
4         if rr.type == TYPE_A:  
5             return response_data  
6         elif rr.type == TYPE_CNAME:  
7             cname = rr.rdata  
8     # only cname  
9     if cname: # send another request to cname field and get question  
10         cname_data = self.iterative_query(self.construct_query(cname).encode_bytes(), self.root_dns_address)  
11         cname_msg = DNSMessage.from_wire(cname_data)  
12         add_an_len = len(cname_msg.answer)  
13         resp_message.header.ancount += add_an_len  
14         resp_message.answer.extend(cname_msg.answer)  
15         return resp_message.encode_bytes()  
16     else:  
17         raise 'error! no cname for final select'
```

在该代码中，代码将查看响应中是否存在A类型RR，如有则直接终止并返回响应字节流。如仅存在CNAME类型的RR，则再次调用iterative_query方法自根节点查询CNAME域名的IP地址。由于iterative_query保证返回一个包含A类型查询结果的字节流，因此可以直接将结果添加至先前DNS包的Answer段，并调用encode_bytes重新编码并返回。

由于先前已实现了将DNS对象编码为字节流的功能，因此可以较为方便的实现construct_query方法构造一个对指定域名的查询请求并编码，且可以通过解析-添加Answer段-重新编码的方式添加查询结果，因此该项目实现了该特殊情况的处理过程。

测试结果

1. 通过了test.py的测试



```
85
86
87 if __name__ == '__main__':
88     unittest.main()
89
```

TestDNSResolver > test_header()

Tests passed: 3 of 3 tests - 1 ms

E:\ProgramFiles4\Python\Python380\python.exe "E:/ProgramFiles4/PyCharm/2023.3.4/plugins/python/helpers/pycharm/_jb_unittest_runner.py" --path D:\ComputerScience\Projects\Computer_network_PA1\Project\test.py

Testing started at 18:25 ...

Launching unittests with arguments python -m unittest D:\ComputerScience\Projects\Computer_network_PA1\Project\test.py in D:\ComputerScience\Projects\Computer_network_PA1\Project

header test passed

question test passed

whole msg test passed

Ran 3 tests in 0.003s

OK

Process finished with exit code 0

2. wireshark抓包结果

www.sustech.edu.cn结果

正在捕获 Adapter for loopback traffic capture

文件(F) 编辑(E) 视图(V) 跳转(G) 捕获(C) 分析(A) 统计(S) 电话(Y) 无线(W) 工具(T) 帮助(H)

dns

No.	Time	Source	Destination	Protocol	Length	Info
93	5.310411	127.0.0.1	127.0.0.1	DNS	91	Standard query 0xe239 A www.sustech.edu.cn OPT
144	8.778975	127.0.0.1	127.0.0.1	DNS	233	Standard query response 0xe239 A www.sustech.edu.cn CNAME sustech.edu.cn A 172.18.1.3 NS n

> Frame 93: 91 bytes on wire (728 bits), 91 bytes captured on interface Null/Loopback

> Internet Protocol Version 4, Src: 127.0.0.1, Dst: 127.0.0.1

> User Datagram Protocol, Src Port: 56156, Dst Port: 9999

> Domain Name System (query)

Transaction ID: 0xe239

Flags: 0x0120 Standard query

0... .. = Response: Message is a query

.000 0... .. = Opcode: Standard query (0)

... .. = Truncated: Message is not truncated

... ..1... .. = Recursion desired: Do query recursively

... ..0... .. = Z: reserved (0)

... ..1... .. = AD bit: Set

... ..0... .. = Non-authenticated data: No

Questions: 1

Answer RRs: 0

Authority RRs: 0

Additional RRs: 1

> Queries

> www.sustech.edu.cn: type A, class IN

```

C:\WINDOWS\system32\cmd. x + v
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 57913
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 2, AUTHORITY: 2, ADDITIONAL: 5

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
;; QUESTION SECTION:
;www.sustech.edu.cn.      IN      A

;; ANSWER SECTION:
www.sustech.edu.cn.      3600    IN      CNAME   sustech.edu.cn.
sustech.edu.cn.          3600    IN      A       172.18.1.3

;; AUTHORITY SECTION:
sustech.edu.cn.          3600    IN      NS       ns2.sustech.edu.cn.
sustech.edu.cn.          3600    IN      NS       ns1.sustech.edu.cn.

;; ADDITIONAL SECTION:
ns1.sustech.edu.cn.      3600    IN      A        172.18.1.92
ns1.sustech.edu.cn.      3600    IN      AAAA     2001:da8:201d::42:92
ns2.sustech.edu.cn.      3600    IN      A        172.18.1.93
ns2.sustech.edu.cn.      3600    IN      AAAA     2001:da8:201d::42:93

;; Query time: 3468 msec
;; SERVER: 127.0.0.1#9999(127.0.0.1)
;; WHEN: Wed Mar 27 18:28:54 ;; MSG SIZE rcvd: 201

```

93	5.310411	127.0.0.1	127.0.0.1	DNS	91	Standard query 0xe239 A www.sustech.edu.cn OPT
144	8.778975	127.0.0.1	127.0.0.1	DNS	233	Standard query response 0xe239 A www.sustech.edu.cn CNAME sustech

Transaction ID: 0xe239

Flags: 0x8580 Standard query response, No error

1... .. = Response: Message is a response

.000 0... .. = Opcode: Standard query (0)

... ..1... .. = Authoritative: Server is 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: 2

Authority RRs: 2

Additional RRs: 5

> Queries

> www.sustech.edu.cn: type A, class IN

> Answers

> www.sustech.edu.cn: type CNAME, class IN, cname sustech.edu.cn

> sustech.edu.cn: type A, class IN, addr 172.18.1.3

> Authoritative nameservers

> sustech.edu.cn: type NS, class IN, ns ns2.sustech.edu.cn

> sustech.edu.cn: type NS, class IN, ns ns1.sustech.edu.cn

> Additional records

> ns1.sustech.edu.cn: type A, class IN, addr 172.18.1.92

> ns1.sustech.edu.cn: type AAAA, class IN, addr 2001:da8:201d::42:92

> ns2.sustech.edu.cn: type A, class IN, addr 172.18.1.93

> ns2.sustech.edu.cn: type AAAA, class IN, addr 2001:da8:201d::42:93

> <Root>: type OPT

[Request In: 93]

[Time: 3.468564000 seconds]

0000 02 00 00 00 45 00 00 e5 7d
0010 7f 00 00 01 7f 00 00 01 27
0020 e2 39 85 80 00 01 00 02 00
0030 07 73 75 73 74 65 63 68 03
0040 00 01 00 01 c0 0c 00 05 00
0050 c0 10 c0 10 00 01 00 01 00
0060 01 03 c0 10 00 02 00 01 00
0070 73 32 c0 10 c0 10 00 02 00
0080 03 6e 73 31 c0 10 c0 60 00
0090 00 04 ac 12 01 5c c0 60 00
00a0 00 10 20 01 0d a8 20 1d 00
00b0 00 92 c0 4e 00 01 00 01 00
00c0 01 5d c0 4e 00 1c 00 01 00
00d0 0d a8 20 1d 00 00 00 00 00
00e0 29 10 00 00 00 00 00 00 00

www.baidu.com结果

No.	Time	Source	Destination	Protocol	Length	Info
295	13.325374	127.0.0.1	127.0.0.1	DNS	86	Standard query 0x4daf A www.baidu.com OPT
298	13.781011	127.0.0.1	127.0.0.1	DNS	161	Standard query response 0x4daf A www.baidu.com CNAME www.a.shifen.com A 182.61.200.7

```

C:\WINDOWS\system32\cmd. X + v

C:\Users\86136> dig @127.0.0.1 www.baidu.com a -p 9999

; <<>> DiG 9.16.48 <<>> @127.0.0.1 www.baidu.com a -p 9999
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->HEADER<- opcode: QUERY, status: NOERROR, id: 19887
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 3, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 1232
; COOKIE: a738141ace538e37010000006603f5b140400d5be23dae50 (good)
;; QUESTION SECTION:
;www.baidu.com.                IN      A

;; ANSWER SECTION:
www.baidu.com.                1200    IN      CNAME   www.a.shifen.com.
www.a.shifen.com.            300     IN      A        182.61.200.7
www.a.shifen.com.            300     IN      A        182.61.200.6

;; Query time: 455 msec
;; SERVER: 127.0.0.1#9999(127.0.0.1)
;; WHEN: Wed Mar 27 18:32:16 ;; MSG SIZE rcvd: 129

C:\Users\86136>

```

> www.baidu.com: type CNAME, class IN
 > www.a.shifen.com: type A, class IN
 > www.a.shifen.com: type A, class IN
 Additional records
 > <Root>: type OPT
 Name: <Root>
 Type: OPT (41)
 UDP payload size: 1232
 Higher bits in extended RCODE: 0
 EDNS version: 0
 Z: 0x0000
 Data length: 28
 Option: COOKIE
 Option Code: COOKIE (10)
 Option Length: 24
 Option Data: a738141ace538e37010000006603f5b140400d5be23dae50
 Client Cookie: a738141ace538e37010000006603f5b140400d5be23dae50

www.example.com结果

No.	Time	Source	Destination	Protocol	Length	Info
220	7.337610	127.0.0.1	127.0.0.1	DNS	86	Standard query 0x4daf A www.example.com OPT
273	8.045138	127.0.0.1	127.0.0.1	DNS	161	Standard query response 0x4daf A www.example.com A 93.184.216.34

```

C:\WINDOWS\system32\cmd. X + v

C:\Users\86136> dig @127.0.0.1 www.example.com a -p 9999

; <<>> DiG 9.16.48 <<>> @127.0.0.1 www.example.com a -p 9999
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->HEADER<- opcode: QUERY, status: NOERROR, id: 52849
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

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

;; ANSWER SECTION:
www.example.com.              86400   IN      A        93.184.216.34

;; Query time: 708 msec
;; SERVER: 127.0.0.1#9999(127.0.0.1)
;; WHEN: Wed Mar 27 18:33:50 ;; MSG SIZE rcvd: 60

C:\Users\86136>

```

> www.example.com: type A, class IN
 Name: www.example.com
 [Name Length: 15]
 [Label Count: 3]
 Type: A (1) (Host Address)
 Class: IN (0x0001)
 Time to live: 86400 (1 day)
 Data length: 4
 Address: 93.184.216.34
 Additional records
 > <Root>: type OPT
 Name: <Root>
 Type: OPT (41)
 UDP payload size: 1232
 Higher bits in extended RCODE: 0
 EDNS version: 0
 Z: 0x0000
 Data length: 28
 Option: COOKIE
 Option Code: COOKIE (10)
 Option Length: 24
 Option Data: a738141ace538e37010000006603f5b140400d5be23dae50
 Client Cookie: a738141ace538e37010000006603f5b140400d5be23dae50