

Project report

Design and implementation of DNS Client and Server

Course Title: Internet Application

Name: Zhiwei Ren (2015213096)

Chunhao Li (2015213094)

Date: 2018/6/24

1. Overview

This project requires us follow the DNS protocol and write our C program through Linux system to implement the domain name resolution function.

2. Requirements Analysis

The requirement of the project is analyzed here, including,

- Development Environment:

Our Group is based on the Linux system and usage of C language for project development.

The test environment for the project's results is Ubuntu system.

The running environment of the project is Linux command line terminal.

- Functional requirements in details:

1.Implement Chinese domain name resolution 2. support four top-level domains Support Resource Record type 3.Support iterative and recursive parsing 4.Support cache check, and can print query trace records 4.Client and Local server connection using TCP 5.Local server and DNS server connection using UDP 6.DNS packets can be correctly parsed by Wireshark.

3. Preliminary Design

Preliminary design includes

- Decomposition of functional modules

Client is responsible for reading user input and encapsulate the packet to send to local server then displaying the returned result.

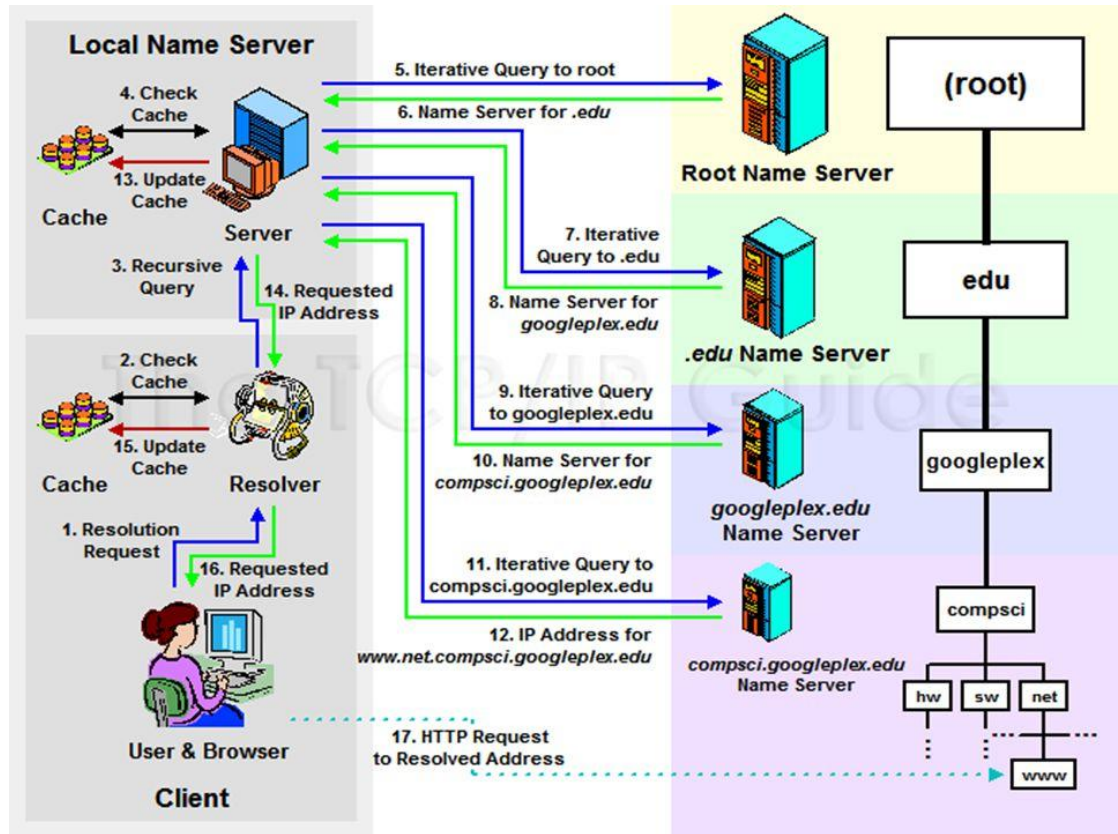
Local Server is responsible for resolving domain names, and query to domain name server if must, then response to the client

Domain name server received the query and response to local server

- Relationship and interface between the modules

Client sends a DNS request packet to the server based on the user's input. The local server returns the DNS packet to the client. Local server send query to domain name server. Domain name server response to local server.

- Overall flow chart



- Design of data structures

DNS Message Format

0	15	16						31
ID	QR	OPCODE	AA	TC	RD	RA	Z	Rcode
Question count	Answer count							
Authority count	Additional count							
<p style="text-align: center;">Question Section (variable number of questions)</p>								
<p style="text-align: center;">Answer Section (variable number of RRs)</p>								
<p style="text-align: center;">Authority Section (variable number of RRs)</p>								
<p style="text-align: center;">Additional Section (variable number of RRs)</p>								

0	15 16	31
QUERY DOMAIN NAME (variable number of domain names)		
QUERY TYPE		QUERY CLASS

4. Detailed Design

▪ Decomposition of functional modules

Client is responsible for reading user input and encapsulate the packet to send to local server then displaying the returned result.

Local Server is responsible for resolving domain names, and query to domain name server if must, then response to the client

Domain name server received the query and response to local server

▪ Relationship and interface between the modules

Client sends a DNS request packet to the server based on the user's input, there are three types of the request: A, MX and CNAME. The local server returns the DNS packet to the client. Local server send query to domain name server. Domain name server response to local server.

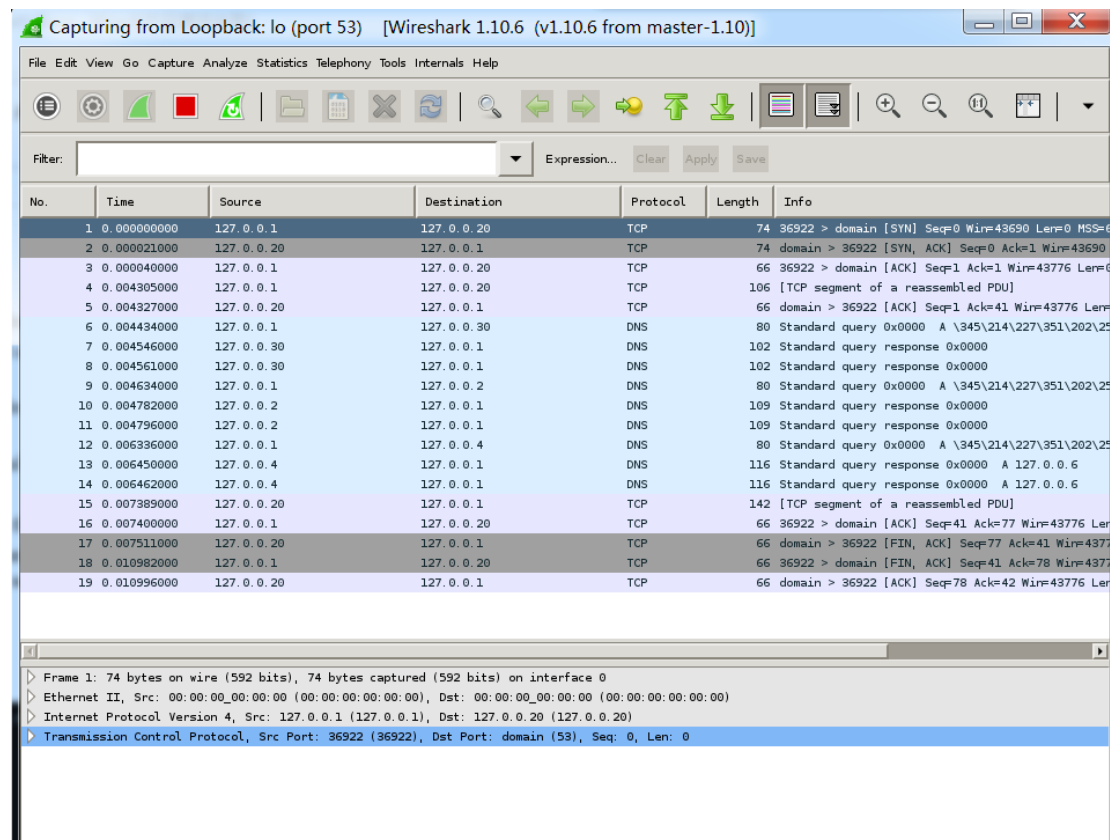
5. Results :

A:

```
student@BUPTIA:~/DNS$ sudo ./client A 北邮.教育.中国
连接到本地DNS服务器 127.0.0.20@53...
【准备发送报文】
<---- DNS 报文开始 ---->
  id:0  status:NO ERROR  报文头信息
Query Section: 1
  北邮.教育.中国      A      IN
Answer Section: 0
Authority Section: 0
Additional Section: 0
<---- DNS 报文结束 ---->

【接受报文】
<---- DNS 报文开始 ---->
  id:0  status:NO ERROR  报文头信息 qr aa
Query Section: 1
  北邮.教育.中国      A      IN
Answer Section: 1
  北邮.教育.中国      A      IN      86400      数据长度 = 4      127.0.0.6
Authority Section: 0
Additional Section: 0
<---- DNS 报文结束 ---->

【请求用时】: 10.305000 ms
Wed Jun 20 06:03:51 2018
```



Trace:

```

添加数据库资料      .,86400,IN,A,127.0.0.30
本地DNS开始运行 127.0.0.20:53
开始DNS解析 127.0.0.1:36922
<---- DNS 报文开始 ---->
  id:0  status:NO ERROR  报文头信息
Query Section: 1
  北邮.教育.中国      A      IN
Answer Section: 0
Authority Section: 0
Additional Section: 0
<---- DNS 报文结束 ---->
准备访问 127.0.0.30@53
Authority Section
  中国  A      IN      86400      数据长度 = 4      127.0.0.2

Authority Section
  教育.中国  A      IN      86400      数据长度 = 4      127.0.0.4

Answer Section
{0}  北邮.教育.中国  A      IN      86400      数据长度 = 4      127.0.0.6
  
```

MX:

```

student@BUPTIA:~/DNS$ sudo ./client MX 北邮.教育.中国
连接到本地DNS服务器 127.0.0.20@53...
【准备发送报文】
<----- DNS 报文开始 ----->
  id:0  status:NO ERROR  报文头信息
Query Section: 1
  北邮.教育.中国      MX      IN
Answer Section: 0
Authority Section: 0
Additional Section: 0
<----- DNS 报文结束 ----->

【接受报文】
<----- DNS 报文开始 ----->
  id:0  status:NO ERROR  报文头信息 aa
Query Section: 1
  北邮.教育.中国      A      IN
Answer Section: 1
  北邮.教育.中国      A      IN      86400      数据长度 = 4      127.0.0.1
Authority Section: 0
Additional Section: 0
<----- DNS 报文结束 ----->

【请求用时】: 9.639000 ms
Wed Jun 20 06:07:01 2018

student@BUPTIA:~/DNS$

```

Capturing from Loopback: lo (port 53) [Wireshark 1.10.6 (v1.10.6 from master-1.10)]

File Edit View Go Capture Analyze Statistics Telephony Tools Internals Help

Filter: Expression... Clear Apply Save

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	127.0.0.1	127.0.0.20	TCP	74	36924 > domain [SYN] Seq=0 Win=43690 Len=0 MSS=6
2	0.000025000	127.0.0.20	127.0.0.1	TCP	74	domain > 36924 [SYN, ACK] Seq=0 Ack=1 Win=43690
3	0.000048000	127.0.0.1	127.0.0.20	TCP	66	36924 > domain [ACK] Seq=1 Ack=1 Win=43776 Len=0
4	0.004946000	127.0.0.1	127.0.0.20	TCP	106	[TCP segment of a reassembled PDU]
5	0.004978000	127.0.0.20	127.0.0.1	TCP	66	domain > 36924 [ACK] Seq=1 Ack=41 Win=43776 Len=0
6	0.005088000	127.0.0.20	127.0.0.1	TCP	142	[TCP segment of a reassembled PDU]
7	0.005094000	127.0.0.1	127.0.0.20	TCP	66	36924 > domain [ACK] Seq=41 Ack=77 Win=43776 Len=0
8	0.005156000	127.0.0.20	127.0.0.1	TCP	66	domain > 36924 [FIN, ACK] Seq=77 Ack=41 Win=43776 Len=0
9	0.009702000	127.0.0.1	127.0.0.20	TCP	66	36924 > domain [FIN, ACK] Seq=41 Ack=78 Win=43776 Len=0
10	0.009723000	127.0.0.20	127.0.0.1	TCP	66	domain > 36924 [ACK] Seq=78 Ack=42 Win=43776 Len=0

Frame 1: 74 bytes on wire (592 bits), 74 bytes captured (592 bits) on interface 0

Ethernet II, Src: 00:00:00:00:00:00 (00:00:00:00:00:00), Dst: 00:00:00:00:00:00 (00:00:00:00:00:00)

Internet Protocol Version 4, Src: 127.0.0.1 (127.0.0.1), Dst: 127.0.0.20 (127.0.0.20)

Transmission Control Protocol, Src Port: 36924 (36924), Dst Port: domain (53), Seq: 0, Len: 0

```

0000  00 00 00 00 00 00 00 00 00 00 08 00 45 00  ....E.
0010  00 3c b7 29 40 00 40 06 85 7d 7f 00 00 01 7f 00  <.)@.}.....
0020  00 14 90 3c 00 35 8f cd c1 64 00 00 00 a0 02  ...<.S.....d.....
0030  aa aa fe 43 00 00 02 04 ff d7 04 02 08 0a 00 63  ...C.....c
0040  76 4d 00 00 00 00 01 03 03 07                  vM.....

```

CNAME:

```
student@BUPTIA:~/DNS$ sudo ./client CNAME 主页.政府.美国
连接到本地DNS服务器 127.0.0.20@53...
【准备发送报文】
<---- DNS 报文开始 ---->
  id:0  status:NO ERROR  报文头信息
Query Section: 1
  主页.政府.美国          CNAME  IN
Answer Section: 0
Authority Section: 0
Additional Section: 0
<---- DNS 报文结束 ---->

【接受报文】
<---- DNS 报文开始 ---->
  id:0  status:NO ERROR  报文头信息 qr aa
Query Section: 1
  主页.政府.美国          CNAME  IN
Answer Section: 1
  主页.政府.美国          CNAME  IN      86400    数据长度 = 15    政府.美国
Authority Section: 0
Additional Section: 0
<---- DNS 报文结束 ---->

【请求用时】: 7.813000 ms
Wed Jun 20 06:10:29 2018
```

The image shows a Wireshark packet capture window titled "Capturing from Loopback: lo (port 53) [Wireshark 1.10.6 (v1.10.6 from master-1.10)]". The interface includes a menu bar, a toolbar, a filter field, and a packet list pane. The packet list shows 16 packets. Packet 14 is a DNS query for "domain" with type CNAME. Packet 15 is the corresponding DNS response, which includes an answer for "domain" with type CNAME and value "政府.美国". The packet details pane for packet 15 shows the "Standard query response" structure, including the "Answer" section with the CNAME record. The packet bytes pane at the bottom shows the raw data of the response packet.

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	127.0.0.1	127.0.0.20	TCP	74	36928 > domain [SYN] Seq=0 Win=43690 Len=0 MSS=65535
2	0.000018000	127.0.0.20	127.0.0.1	TCP	74	domain > 36928 [SYN, ACK] Seq=0 Ack=1 Win=43690 Len=0
3	0.000034000	127.0.0.1	127.0.0.20	TCP	66	36928 > domain [ACK] Seq=1 Ack=1 Win=43776 Len=0
4	0.004309000	127.0.0.1	127.0.0.20	TCP	106	[TCP segment of a reassembled PDU]
5	0.004328000	127.0.0.20	127.0.0.1	TCP	66	domain > 36928 [ACK] Seq=1 Ack=41 Win=43776 Len=0
6	0.004478000	127.0.0.1	127.0.0.30	DNS	80	Standard query 0x0000 CNAME \344\270\273\351\242
7	0.004576000	127.0.0.30	127.0.0.1	DNS	102	Standard query response 0x0000
8	0.004587000	127.0.0.30	127.0.0.1	DNS	102	Standard query response 0x0000
9	0.005308000	127.0.0.1	127.0.0.3	DNS	80	Standard query 0x0000 CNAME \344\270\273\351\242
10	0.005582000	127.0.0.3	127.0.0.1	DNS	127	Standard query response 0x0000 CNAME \346\224\225
11	0.005596000	127.0.0.3	127.0.0.1	DNS	127	Standard query response 0x0000 CNAME \346\224\225
12	0.006586000	127.0.0.20	127.0.0.1	TCP	133	[TCP segment of a reassembled PDU]
13	0.006594000	127.0.0.1	127.0.0.20	TCP	66	36928 > domain [ACK] Seq=41 Ack=88 Win=43776 Len=0
14	0.006644000	127.0.0.20	127.0.0.1	TCP	66	domain > 36928 [FIN, ACK] Seq=88 Ack=41 Win=43776 Len=0
15	0.007797000	127.0.0.1	127.0.0.20	TCP	66	36928 > domain [FIN, ACK] Seq=41 Ack=89 Win=43776 Len=0
16	0.007808000	127.0.0.20	127.0.0.1	TCP	66	domain > 36928 [ACK] Seq=89 Ack=42 Win=43776 Len=0

Frame 1: 74 bytes on wire (592 bits), 74 bytes captured (592 bits) on interface 0
Ethernet II, Src: 00:00:00:00:00:00 (00:00:00:00:00:00), Dst: 00:00:00:00:00:00 (00:00:00:00:00:00)
Internet Protocol Version 4, Src: 127.0.0.1 (127.0.0.1), Dst: 127.0.0.20 (127.0.0.20)
Transmission Control Protocol, Src Port: 36928 (36928), Dst Port: domain (53), Seq: 0, Len: 0

0000 00 00 00 00 00 00 00 00 00 00 00 08 00 45 00E.
0010 00 3c 16 95 40 00 40 06 26 12 7f 00 00 01 7f 00 <...@.@.....
0020 00 14 90 40 00 35 2c b2 57 37 00 00 00 00 a0 02 ...@.5...W7.....
0030 aa aa fe 43 00 00 02 04 ff d7 04 02 08 0a 00 64 ...C.....d
0040 41 a2 00 00 00 00 01 03 05 07 A.....

6. Summary and Conclusion

Through this project we have a comprehensive understanding of DNS, and the usage of C languages to program this project makes us have a profound understanding of socket program.