

DNS server part

1.Introduction

In DNS section, I implemented a basic DNS server based on RFC 1035.

This tiny DNS server only resolves request which have an address of “video.pku.edu.cn” and return the packet contains one possible ip address. If address requested is not “video.pku.edu.cn”, the server will response a packet contains RCODE_3.

Three ip-choosing algorithm is implemented in this server: (1)Simply return the first available ip address. (2)Round-Robin algorithm to balance the load. (3) ”Minimum geographic distance” algorithm using link state advertisement.

A tiny resolution library which sends, receives and resolves DNS packet is also implemented.

2.Basic Communication Test

This test is performed on my laptop, Ubuntu 16.04 LTS.

Since my implementation is base on RFC 1035, the packet which generated from my resolution library should pass the test of a packet sniffer. Therefore, I wrote a additional test program to test the correctness of the communication packets [See test2.c]. This small test program calls “resolve” function which the library provides, and send two request of both “video.pku.edu.cn” and “pku.edu.cn”. Then I use wireshark to capture the packets during the communication.

Command:

```
sudo ./nameserver -r logs/log.txt 127.0.0.3 53 topos/topo2/topo2.servers topos/topo2/topo2.lsa
sudo ./test2 127.0.0.1 127.0.0.3 53 test2.out
```

Nameserver opens on 127.0.0.3:53 (DNS service port), and test program runs on 127.0.0.1

Nameserver output on shell

```
aposta@aposta-YOGA:~/projects/network/lab3/bitrates-project-starter$ sudo ./nameserver -r logs/log.txt 127.0.0.3 53 topos/topo2/topo2.servers topos/topo2/topo2.lsa
2 127.0.0.1 video.pku.edu.cn 6.0.0.1
2 127.0.0.1 pku.edu.cn RCODE_3,NO SERVER FOUND!
```

Test program output on shell

```
aposta@aposta-YOGA:~/projects/network/lab3/bitrates-project-starter$ sudo ./test2 127.0.0.1 127.0.0.3 53 test2.out
dns_ip (127.0.0.3:53)
cli_ip (127.0.0.1:53)
1 ok
2 ok
```

Wireshark result

dns						
No.	Source	Destination	Protocol	Length	Time	Info
1	127.0.0.1	127.0.0.3	DNS		760.000000000	Standard query 0x0001 A video.pku.edu.cn
2	127.0.0.3	127.0.0.1	DNS		860.000061282	Standard query response 0x0001 A 6.0.0.1
3	127.0.0.1	127.0.0.3	DNS		700.000147696	Standard query 0x0002 A pku.edu.cn
4	127.0.0.3	127.0.0.1	DNS		760.000162298	Standard query response 0x0002 No such name A

Packets generated from my implementation successfully passed the test of the wireshark, which indicates the content conforms to RFC 1035.

3.Algorithm Test

This test is performed on the virtual box of this lab.

To test the correctness of my ip-choosing algorithm, I still use test2.c to generate the request and then look into the log file of the nameserver.

I didn't test the simple algorithm without load balancing

Test command

```
./test2 1.0.0.1 7.0.0.1 1234 test.out; \
sleep 3; \
./test2 2.0.0.1 7.0.0.1 1234 test.out; \
./test2 3.0.0.1 7.0.0.1 1234 test.out; \
sleep 5; \
./test2 3.0.0.1 7.0.0.1 1234 test.out; \
./test1 1.0.0.1 7.0.0.1 1234 test.out; \
sleep 2; \
./test2 2.0.0.1 7.0.0.1 1234 test.out;
```

3.1 Round-Robin load balancing

Command to start server

```
./nameserver -r logs/log.txt 7.0.0.1 1234 topos/topo2/topo2.servers
topos/topo2/topo2.lsa
```

Log file

```
1 1 1.0.0.1 video.pku.edu.cn 6.0.0.1
2 1 1.0.0.1 pku.edu.cn RCODE_3,NO SERVER FOUND!
3 4 2.0.0.1 video.pku.edu.cn 5.0.0.1
4 4 2.0.0.1 pku.edu.cn RCODE_3,NO SERVER FOUND!
5 4 3.0.0.1 video.pku.edu.cn 4.0.0.1
6 4 3.0.0.1 pku.edu.cn RCODE_3,NO SERVER FOUND!
7 9 3.0.0.1 video.pku.edu.cn 6.0.0.1
8 9 3.0.0.1 pku.edu.cn RCODE_3,NO SERVER FOUND!
9 9 1.0.0.1 video.pku.edu.cn 5.0.0.1
10 9 1.0.0.1 pku.edu.cn RCODE_3,NO SERVER FOUND!
11 11 2.0.0.1 video.pku.edu.cn 4.0.0.1
12 11 2.0.0.1 pku.edu.cn RCODE_3,NO SERVER FOUND!
```

Log file shows that the ip address returned from server conform to the round-robin algorithm.

3.2 Minimum geographic distance load balancing

Command to start server

```
./nameserver logs/log.txt 7.0.0.1 1234 topos/topo2/topo2.servers  
topos/topo2/topo2.lsa
```

Log File

```
1 3 1.0.0.1 video.pku.edu.cn 4.0.0.1  
2 3 1.0.0.1 pku.edu.cn RCODE_3,NO SERVER FOUND!  
3 6 2.0.0.1 video.pku.edu.cn 5.0.0.1  
4 6 2.0.0.1 pku.edu.cn RCODE_3,NO SERVER FOUND!  
5 6 3.0.0.1 video.pku.edu.cn 6.0.0.1  
6 6 3.0.0.1 pku.edu.cn RCODE_3,NO SERVER FOUND!  
7 11 3.0.0.1 video.pku.edu.cn 6.0.0.1  
8 11 3.0.0.1 pku.edu.cn RCODE_3,NO SERVER FOUND!  
9 11 1.0.0.1 video.pku.edu.cn 4.0.0.1  
10 11 1.0.0.1 pku.edu.cn RCODE_3,NO SERVER FOUND!  
11 13 2.0.0.1 video.pku.edu.cn 5.0.0.1  
12 13 2.0.0.1 pku.edu.cn RCODE_3,NO SERVER FOUND!
```

I use dijkstra to find the minimum distance, the correctness is shown in the log file.