Lab 3: A DNS Client Using TCP and TCP/UDP Performance Comparison

Author : Michelle(Jie) Gao
Instructor: EJ Jung

Experiment 1: Response time

I write java code in DNS Client to count the response time of TCP and UDP's connection.
The timer begins before the socketing building, and ends after receiving packets.
So, TCP's response time is assumed to be longer than UDP's since there is 3-way handshake time included.
From the data in the following spreadsheet, TCP's response time is longer than UDP's response time.
Then I create two forms for UDP and TCP's response time under different DNS server: 8.8.8.8, and 8.8.4.4

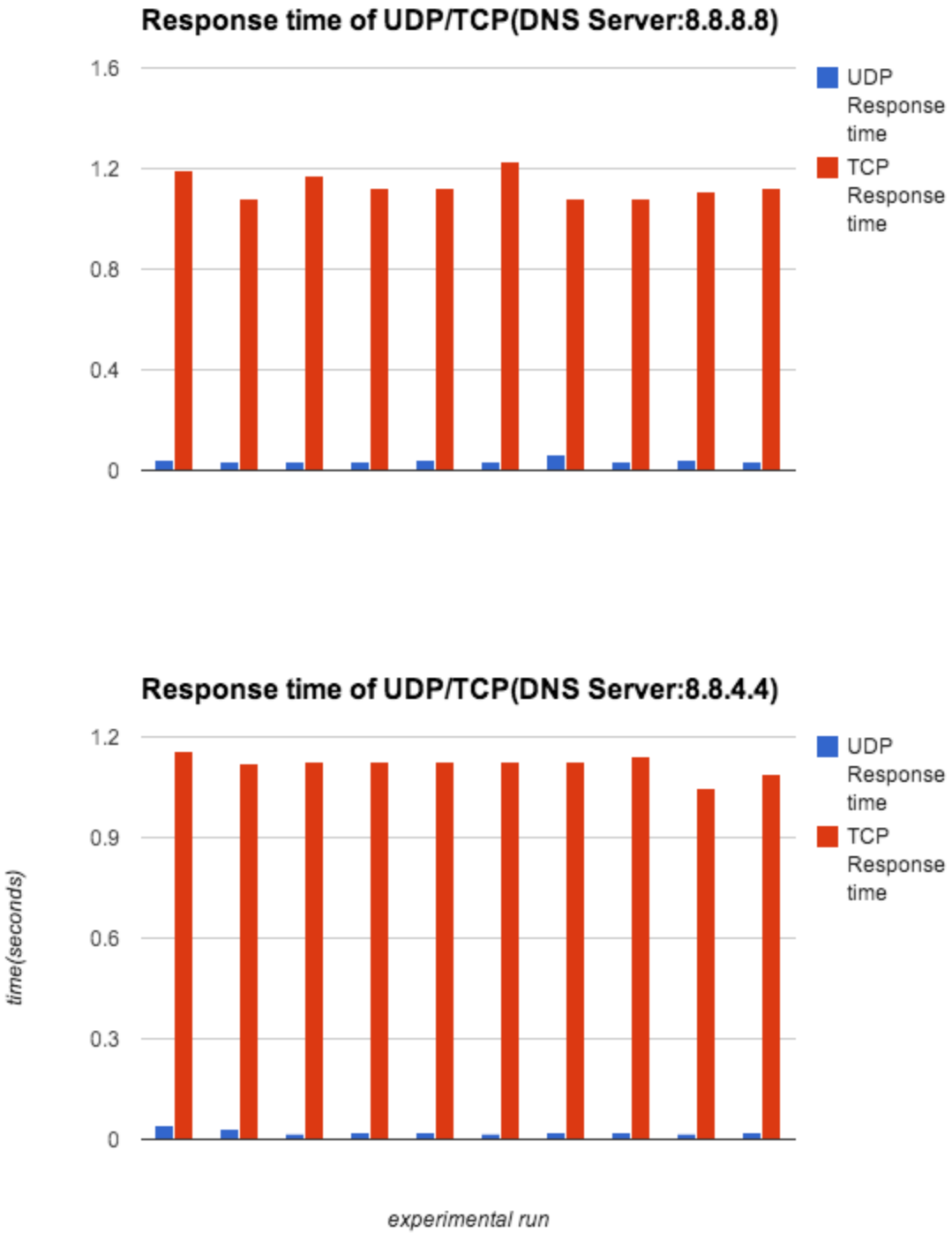
Compare the response time using different dns server:

DNS Server:8.8.8.8	1th	2th	3th	4th	5th	6th	7th	8th	9th	10th	min time	average time	max time	average(20 times of Exp)
UDP Response time	0.04	0.036	0.036	0.037	0.04	0.037	0.062	0.037	0.04	0.037	0.036	0.040	0.062	0.039
TCP Response time	1.195	1.078	1.171	1.123	1.125	1.226	1.08	1.082	1.112	1.124	1.078	1.132	1.226	1.175

DNS Server:8.8.4.4	1th	2th	3th	4th	5th	6th	7th	8th	9th	10th	min time	average time	max time	average(20 times of Exp)
UDP Response time	0.04	0.029	0.018	0.02	0.019	0.018	0.021	0.019	0.018	0.019	0.018	0.022	0.04	0.021
TCP Response time	1.156	1.123	1.125	1.124	1.124	1.125	1.125	1.14	1.045	1.088	1.045	1.1175	1.156	1.126

From the data above, DNS Server:8.8.4.4 's average response time is shorter than 8.8.8.8

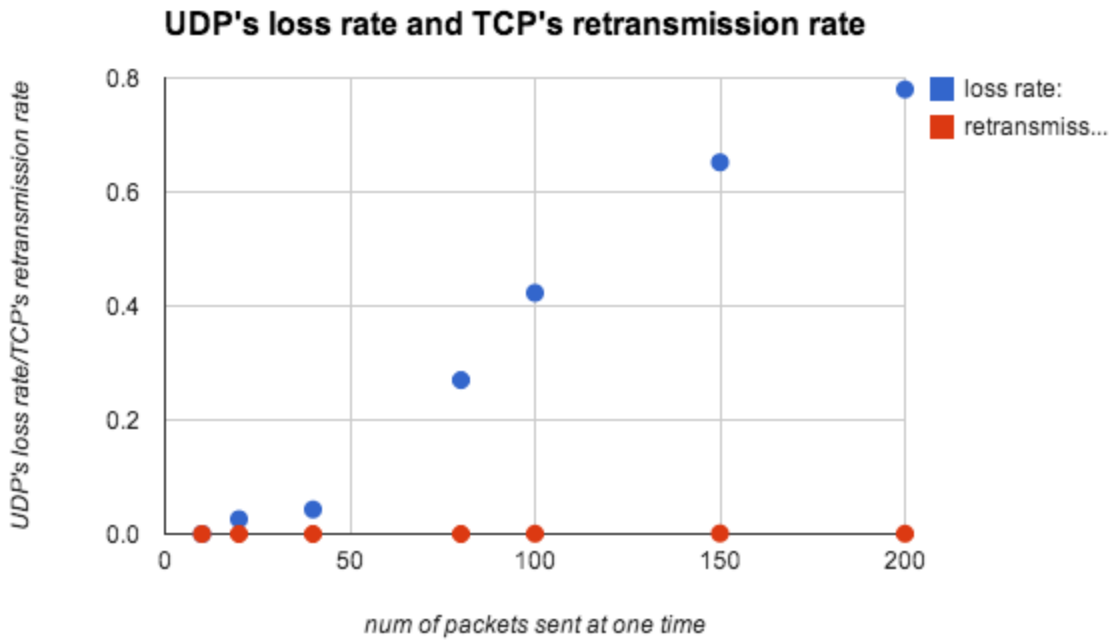
Here is the graph:



Experiment 2:

Comparison of the TCP's retransmission rate and UDP's loss rate:

I modify the packets I sent at one time to see the changing of TCP's retransmission rate and the changing of UDP's loss rate. Then I create a spreadsheet to store those datas to draw a graph.



Parameter and formula in experiment:

Loss Rate = (Potential packets sent and received - packet captured in wireshark)/Potential packets sent and received.

Retransmission Rate = TCP retransmission packets/ Potential packets sent and received

Potential packets sent and received(in 10 threads): (1+k)*10*2 (k is the num of Packets sent in one thread at one time)

UDP's loss rate data :

k(num Of packets sent in one thread at one time)	10	20	40	80	100	150	200
packet captured in wireshark	220	409	785	1181	1165	1051	884
loss rate:	0%	2.6%	4.3%	27%	42.3%	65.2%	78%

TCP's retransmission rate data:

k(num Of packets sent in one thread at one time)	10	20	40	80	100	150	200	400	600	800	1000	200
TCP retransmission packets:	0	0	0		1	3	2	2	12	24	45	98
retransmission rate:	0	0	0	0	0.05%	0.099 %	0.0498 %	0.0125 %	0.0749%	0.1498 %	0.2246 %	0.408%

Analyze TCP's retransmission rate using Link Conditioner and output of Java Program:

I use Network Link Conditioner to modify network.

When I write a script to run muiltithreaded program(10 DNSClient at the same time), each DNSClient handle 1000's DNS request, then I will get 23 retransmission out of 200609 packets,using tcp.analysis.retransmission filters in wireshark.

Since I can not see the difference when I change the packet drop rate through Network Link Conditioner,I guess the reason is that the Network Link Conditioner is just a virtua l machine which can not change the real network condition, so I can not see any difference of retransmission rate in wireshark by changing the packet drop rate.

But I can get the output from java program, and when I change packet drop rate to 90%, run the program:./run.sh > /dev/null there is only 2 lines of reponse written in the output_tcp file, which means the retransmission rate is (10-2)/10 = 80%.

Analyze and Compare the TCP's retransmission rate and UDP's loss rate:

When k(k is the number of packets I sent one time in one thread) gets bigger, the TCP's retransmission rate will gets bigger, so does the UDP's loss rate.

UDP's loss rate is higher than TCP's retransmission rate.

The reason is no matter what the width is, when sending more and more request at one time, network condition can not handle as well

as the requests are fewer, so the retransmission rate and loss rate will get higher.