A Study on Optimization of Network latency and Pocket loss Rate

Sujatha krishanmoorthy 1, Zhangyu Wei 1,Yihuai Zhang1, and Hao Jin1,Hao Dong1

1Assistant professor , Students (1064082,1064213,1063845,1063739) Department of Computer science, Wenzhou-Kean University Wenzhou, China.

Sujatha@wku.edu.cn, weiz@kean.edu,
zhanyihu@kean.edu,jinhao@kean.edu,dongh@kean.edu

corresponding author’s email:sujatha.ssps@gmail.com

Abstract. Network delay refers to the transmission of various data in the network medium through network protocol, such as TCP / IP. If the amount of information is too large and there is no limit at the same time, the excessive network traffic will lead to the slow response of equipment, resulting in network delay. [2] The main causes of network delay are transmission delay, propagation delay, packet switching, queuing, packet drop and processing. [1]Therefore, in order to study the optimization of network delay, the causes of network delay was studied in detail. Some of the methods were deployed to learn the testing network latency, delay measurement algorithm, for example, Cristian’s Algorithm, network analysis by establishing a packet processing model, and TCP BBR algorithm and analyze the optimization of network delay through the simulation of BBR. Finally, by comparing the data before and after optimization, the results of the optimization of the network delay test data by the BBR algorithm are obtained.

1. Introduction

With the rapid increase in the number of Internet users, the scale, and the dependence on the network, people are increasingly demanding high performance, low latency, and mode of the network. However, because the network not only carries huge data services, but also multimedia voice, video, financial, and other real-time services, and due to some issues such as insufficient server performance, system design defects and etc., a large number of network delays have seriously affected the user experience.

Unreasonable server system design and low computing power are an important cause of network delays. In the network, the server provides users with resources (such as CPU, network, data, communication, etc.). If the server performance is poor (there is software on the network that can
detect the server performance), it will take a long time to perform related operations. In other words, the time spent on network transmission is negligible. Therefore, according to the barrel principle, if the bottleneck of network delay is the server, even if the network performance is increased, the delay will not change. At the same time, system design also plays an important role in the overall performance of the server.

The transmission of the application data unit (ADU) on the network is delayed compared to the application layer ADU. After the application layer protocol invokes the services provided by the lower layer protocol, a protocol data unit (PDU) is formed. Since the protocol payload lengths of different layers are different, the data packet size is not necessarily the same. PDU is transmitted on the network as an independent data unit. The main delays at this time are caused by packaging, transmission, propagation, transmission, and processing.

Routing delays include domain name request delay, routing delay, connection establishment delay, and connection release delay. First, due to the special nature of the host address, it is difficult to remember. Now accessing the network is basically accessed through the domain name, so the host address must be resolved before communication. The delay caused by that process called the domain name request delay, which is related to the speed of the network and server load. Secondly, after finding the address, you also need to select the optimal path according to the relevant situation. This process is called routing delay, which includes the processing of the gateway routing table and the address resolution delay. The gateway address resolution is done by ARP at the local machine, so the delay in routing delay is relatively slight. Finally, the process of user connection establishment and connection release is related to network and server load.

According to queuing theory, when the utilization rate of a channel increases, the delay caused by the channel increases rapidly. When there is little traffic on the network, the delay caused by the network is not large. However, in the case of increasing network traffic because the packets need to be queued for processing at the network node (router or node switch), the delay caused by the network will increase. If we let $D_0$ denote the delay when the network is idle and $D$ denote the current delay of the network, then under appropriate assumptions, the following simple formula can be used to express the relationship between $D$, $D_0$ and utilization $U$: $D=D_0/(1-U)$. When the utilization rate of the network reaches 1/2 of its capacity, the delay will be doubled. It is particularly worth noting that when the network utilization is close to the maximum of 1, the network delay tends to infinity. Therefore, we must have the concept that excessive channel or network utilization will cause very large delays.

In this paper, the contribution is to study about a reasonable network delay measurement method. BBR Congestion Control algorithm was implemented to optimize network latency and packet loss rate.

2. Methodology

Our simulation follows a quantitative experimental study method during the whole procedure. Before deciding our topic, we conducted a survey about some usual network knowledge. The questionnaires are conducted to our friends and relatives, without gender, culture or age limits. Questions we designed follow quantitative method and are all open-ended, since we want to collect participants' description and opinion rather than answers such as "Yes", "No" or "how many times". For the question "what do you think of latency", the participants’ feedbacks are 100% negative and that’s why we choose this topic. Then we start from theoretical research.
First of all, we conclude what is latency and what causes latency. Some of the answers come from textbook, and I will generalize them with the help of academic library sources. Network latency can be defined by the time taken for a request to travel from the sender to the receiver and for the receiver to process that request. Though requests and data travel through fiber cable with a speed of light, when the distance goes further, delays must exist since you cannot beat the laws of physics. Maybe the speed of light makes latency paltry when distance is small enough, the repeaters or amplifiers also introduce additional delays. When the requests need to travel through Atlantic submarine cable, latency becomes influential to your life.

Bandwidth determines the maximum rate of data transfer across a given path. Since there are typically many clients per server, the requests to be processed on a server will be greater than each individual client, and sometimes the traffic on such a network exceeds the capacity of calculation and causes congestion. In that situation, a server administrator may control the number of requests a server responds to within a specified period of time. When a server reaches the limit, new requests will be offloaded without responds. This also causes more latency. Other factors such as transmission medium and storage delays also impact your network latency time. Storage delays can occur when a packet is stored or accessed resulting in a delay caused by intermediate devices like switches and bridges.

Hence we get a list of the factors that impact network latency: distance, transition delays, storage delays, bandwidth limitation, transmission medium. Other factors may be added with our study goes deeper. After finishing quantitative description of the factors, the next step we search for methods point at each of the factors listed.

Before testing algorithms, we need tools to measure network latency for an immediate observation. RTT, also known as Round Trip Time, is the length of time it takes for a signal to be sent plus the length of time it takes for an acknowledgement of that signal to be received. It is also known as the ping time for public. RTT has an impact on service response time and it depends on OWD, One-Way Delay, which is the length of time that a packet takes from one point to another across the network. Our analysis of the following tests is all based on the two measurements.

Simulate and analyze methods. Here we choose TCP-BBR Congestion Control Algorithm and Cristian’s Algorithm as our research goal. The BBR algorithm is chosen because the simulation process can be finished on a simple computer, and Linux network names paces provide lightweight network emulation, including processes. Though bandwidths available is low, the Linux’s Token-Bucket Filter (TBF) helps to limit rate and set the buffer size. Traffic from the receivers to the senders is not subject to rate limiting since we only send data from the senders and the returning acknowledgment stream does not exceed the bottleneck bandwidth. Since many users make no sense what do 10ms delay and 100ms delay mean, and their feelings are just “the feedback is quick” or “the waiting time is too long”, we perform qualitative tests which can offer a direct assessment of the new algorithms. In the next part we will talk more about those algorithms.

2.1. Network latency measurement method

This part, it will introduce two network latency measurement methods, they are One-way delay measurements method and Round-trip delay measurement method. In fact, the network performance measurement method is divided into active measurement and passive measurement according to the measurement method. Active measurement is to inject probe packets into the network under test. The probe packets record the transmission in the network. The terminal analyzes the probe packets to
obtain network performance data. The passive measurement method uses monitoring equipment or tools to analyze the network characteristics by capturing the data packets transmitted in the network under test. According to the definition of the network delay metric, the delay of any layer can be attributed to one-way delay or round-trip delay, but the protocol is different. Therefore, the following test methods can be applied to the delay measurement above the network layer.

2.1.1 One-way delay measurement method

Firstly, selecting the protocol used for measurement, such as TCP, UDP, OWDP (One Way Delay Protocol), IPMP (Internet Protocol Measurement Protocol, IP layer measurement protocol), and determine the details of the selected protocol, such as TCP, UDP port number, timeout waiting time and measurement packet size. After determining these parameters, proceed as follows:

(1) Negotiate source and destination synchronization: That is, the clocks between the source and destination must be accurately synchronized. (2) At the source end, determining the IP address of the destination end and fill test packets according to the structure of the selected measurement protocol. The vacant part should be filled with random bits to avoid that the measured delay is lower than the delay after the compression technique is used in the path. Set the time stamp in the test package, and then send the filled test package to the destination host. (3) At the destination, preparing to receive the packet. If the packet arrives within a reasonable time, the estimated one-way delay can be calculated by immediately subtracting the receiving time from the sending time in the packet. This value is only valid under the premise that the source and destination clocks are synchronized. If the packet cannot arrive within a reasonable time, the one-way delay value should be set to undefined.

There are two main methods to solve the clock synchronization problem: software synchronization and hardware synchronization. Synchronization based on GPS belongs to hardware synchronization; synchronization based on Network Time Protocol and synchronization based on algorithm estimation belong to software synchronization. Time synchronization on the Internet is mostly achieved by using an NTP server.

2.1.2. Round-trip delay measurement method

The preparation work is the same as the one-way delay measurement method. Then, the measurement is performed in the following steps:

(1) At the source end, determine the IP address of the destination end and fill the test packet according to the structure of the selected measurement protocol. In order to avoid that the measured delay is lower than the expected result due to the compression technology on the path, the vacant part should be filled with random bits; and the test packet must be set with an ID number so that after receiving the response packet, the source host can judge that it is the response of sending out the test packet by itself. (2) At the source, set the sending timestamp in the test package, and then send the filled test package to the destination host. The timestamp can be placed in the package or outside the package, as long as it contains an appropriate identifier so that the received timestamp can be compared with the sent timestamps. (3) At the destination, prepare to receive test packets. If the test packet reaches the destination host, a response packet should be quickly generated and sent to the source.

(4) At the source, prepare to receive the corresponding response packet. According to the ID number to determine whether it is the response of the test package. If the response packet arrives within the timeout waiting time, immediately subtract the receiving time from the sending time in the response
packet to calculate the round trip delay value. If the response packet cannot arrive within the timeout period, the round trip delay value is set to undefined.

2.2 Cristian’s Algorithm

Delay:
request sent time: $T_1$
reply received time: $T_1$

Then, we can assume network delays are symmetric.

**The Client Time:**

$$T_{\text{new}} = T_{\text{server}} + \frac{T_1 - T_0}{2}$$

Accuracy Time:

minimum message transit time: $T_{\text{min}}$

$$\text{Range} = T_1 - T_0 - 2T_{\text{min}}$$

Therefore, the accuracy time $= \pm \frac{T_1 - T_0}{2} - T_{\text{min}}$
2.2.1 Processing Packet Modeling

From the graph, when the packet arrives, metadata’s behavior will be matched with flow meter. If there is no reasonable flow meter that contain L7 metadata, ->sent into SDN controller -> be inspected by DPI based application-aware module. If malicious behaviors are identified, packet will be discarded.

Therefore, it can prevent malicious programs stepping into the networks. Otherwise, a new flow meter which contains behaviour type and priority of this packet will be distributed into OpenFlow switch.

Then, we can get the Network Latency Analysis that

“n” is network nodes
“K” kinds of new traffic flow
Latency time = \( T_{LTC} \)

\[
T_{LTC}(Lc) = \sum_{i=1}^{n} \left( \frac{N-K}{N} t'_1(i) + \frac{K}{N} t'_3(i) \right)
\]

“m” unidentified packets in new packets
“M” identified packets in total

\[
T_{LTC}(Lc) = \sum_{i=1}^{n} \left( \frac{N-K-M+m}{N} t'_1(i) + \frac{M-m}{N} t'_3(i) + \frac{K-m}{N} t'_3(i) + \frac{m}{N} t'_4(i) \right)
\]

Whereby, \( t'_1(i) = t_1(i) + t_2(i) \), \( t'_3(i) = t_3(i) + t_4(i) \)

3. CP-BBR Congestion Control Algorithm.

BBR is a new TCP congestion control algorithm proposed by Google. It uses round-trip delay and bandwidth to control the congestion window. It no longer uses packet loss as a signal of congestion, but instead detects congestion based on RTT like Vegas. Compared with packet loss, detecting RTT can detect congestion earlier, and then take measures to slow down. In the BBR algorithm, the RTT and bandwidth are periodically detected to discover the network congestion.
The BBR congestion control algorithm has four stages, they are STARTUP, DRAIN, PROBE_BW, PROBE_RTT. Compared with other congestion control algorithms, BBR does not have the traditional slow start phase.

[1]STARTUP can be regarded as equivalent to slow start. In the STARTUP phase, gain will take a relatively large value. The default is \( \frac{2}{\ln 2} \), which is about 2.885. In this phase, the amount of data injected into the network will increase rapidly from a small value until the bandwidth no longer increases.

[2]In the DRAIN phase, a smaller gain value will be set, the default is \( \frac{\ln 2}{2} \), which is about 0.3466. After too many packets injected into the network before this phase are emptied, it will enter the PROBE_BW phase.

[3]Under normal circumstances, the majority of the BBR time will be in the PROBE_BW phase. In this phase, the sender will tentatively inject more data packets into the network to detect whether there is excess bandwidth. The specific method is to increase the gain to 1.25 in the first RTT time to detect whether there is excess bandwidth. In the second RTT, reduce the gain to 0.75 to empty the excess data packets, then set the gain in 6 RTT time is 1, continue to send data.

[4]In this way, every 8 RTTs are used as a period to detect the bandwidth. In general, without data queuing, the measured RTT value is the smallest. BBR will consume 200ms to detect the minimum RTT value every 10 seconds of transmission. This is the PROBE_RTT stage. At this stage, the window is reduced to 4 MSS, and the smallest RTT can be detected, which is also a process of bandwidth politeness, which allows other TCP connections to have the opportunity to occupy bandwidth.

3.1 Components of BBR Congestion Control Algorithm

Calculation of immediate rate

Calculate an instantaneous bandwidth, which is the benchmark for all calculations of BBR. BBR will calculate the pacing rate and cwnd (Congestion Windows) based on the current instantaneous bandwidth and the pipe status it is in. The breakthrough improvement of the bandwidth calculation method is the reason why BBR is simple and efficient. The calculation scheme is calculated according to a scalar quantity, and no longer pays attention to the meaning of the data. During the operation of BBR, the system will track the largest instantaneous bandwidth so far.

RTT tracking

The reason why BBR can obtain a very high bandwidth utilization rate is because it can detect the maximum bandwidth and the minimum value of RTT very safely and boldly. The BDP calculated in this way is the maximum capacity of the TCP pipeline so far. Therefore, we think BBR is to achieve this maximum capacity! This goal ultimately drives the calculation of cwnd. During BBR operation, the system will track the minimum RTT so far.

Maintenance of the BBR pipe state machine

The BBR algorithm specifically defines four states according to the congestion behavior of the Internet, namely STARTUP, DRAIN, PROBE_BW, PROBE_RTT.

Results output -pacing rate and cwnd
First of all, it must be said that the output of BBR is not just a cwnd, but more important is the pacing rate. In the traditional sense, cwnd is the only output of the TCP congestion control algorithm, but it only specifies how much data the current TCP can send at most. It does not specify how to send so much data. In the implementation of Linux, use of other external mechanisms fq, rack.

The reason why BBR can run efficiently and so simple is because many mechanisms are not implemented by itself, but use external existing mechanisms.

3.2 BBR Simulation

The experience use the MiniNet Simulator in VMware, then using the BBR algorithm, the bandwidth setting is 10Mbit/s, the link delay is 250ms, and the amount of data sent in the first 20 seconds.

![Figure 1 BBR Sending Data](image1)

![Figure 2. RRT value of BBR Algorithm](image2)

It can be seen from the figure that the initial data transmission volume grows rapidly, enters a stable state at the 6th second, and there is a possibility of a larger window for multiple detection. At about 11 seconds, the transmission volume decreases rapidly and enters the RTT detection phase. When the amount of data injected into the network is large, the RTT becomes large. When the amount of data is small, the RTT becomes small, but it is not less than the delay of the link. According to above, as a new congestion control algorithm, BBR is mainly reflected in the adjustment of the transmission rate based on delay in TCP transmission. Since BBR is not sensitive to packet loss, it has positive significance on some networks with high bit error rate.

4. Conclusion

Due to server performance and design defects and other reasons, network delay will seriously affect the user's experience. By reducing network delay, that is, reducing the time of network transmission, users can hardly feel the existence of the transmission process, so as to achieve a good experience effect. Aiming at this goal, we optimize two algorithms, TCP and BBR. After testing the network delay before and after the algorithm optimization, we get the conclusion that there is a big difference between before and after the algorithm optimization, and then there is a more obvious delay reduction and stability improvement.


References

[1]. Scholz, D., Jaeger, B., Schwaighofer, L., Raumer, D., Geyer, F., & Carle, G. (2018, May). Towards a deeper understanding of tcp bbr congestion control. In 2018 IFIP Networking Conference (IFIP Networking) and Workshops (pp. 1-9). IEEE.

[2]. Li, G., Dong, M., Ota, K., Wu, J., Li, J., & Ye, T. (2016, December). Deep packet inspection based application-aware traffic control for software defined networks. In 2016 IEEE Global Communications Conference (GLOBECOM) (pp. 1-6). IEEE.

[3]. Fetzer, C., & Cristian, F. (1995, June). An optimal internal clock synchronization algorithm. In COMPASS'95 Proceedings of the Tenth Annual Conference on Computer Assurance Systems Integrity, Software Safety and Process Security' (pp. 187-196). IEEE.

[4]. Cardwell, N., Cheng, Y., Gunn, C. S., Yeganeh, S. H., & Jacobson, V. (2017). BBR: congestion-based congestion control. Communications of the ACM, 60(2), 58-66.

[5]. Tools, P. (2020). Interpreting Latency and Packet Loss | PingPlotter. Retrieved 26 May 2020, from https://www.pingplotter.com/wisdom/article/latency-packet-loss.html

[6]. Yi, X. (2020). What causes network latency? How to reduce (optimize) network latency?. Retrieved 26 May 2020, from https://zhuanlan.zhihu.com/p/142790243

[7]. Singh, Parminder, et al. "Soft-computing-based false alarm reduction for hierarchical data of intrusion detection system." International Journal of Distributed Sensor Networks 15.10 (2019): 1550147719883132.

[8]. Sujatha, Krishnamoorthy. "Automatic epilepsy detection using hybrid decomposition with multi class support vector method." Multimedia Tools and Applications (2019): 1-20.

[9]. Krishnamoorthy, Dr, et al. "A Novel Approach towards Lowest Energy Usage Routing Protocol (LEuRP) in Wireless Mobile Ad-Hoc Network (WMANet)." International Journal of Pure and Applied Mathematics 116.12 (2017): 21-29.