Research and implementation on high-speed transmission performance optimization technology of ten gigabit firewall UDP multicast

Fan Yang, Xinlei Yang, Wei Wang, Peng Du, Jifeng Shi, Feng Li and Hong Hu*

Beijing Space Information Relay and Transmission Technology Research Center, China

Keywords: Ten Gigabit firewall, High speed data transmission, Kernel scheduling, Network card characteristics.

Abstract. Under the development trend of high throughput, large capacity and strong time effective data security transmission of the second generation data relay system in China, the high-performance processing requirements for the firewall of the backbone network security protection equipment are constantly improving. Focusing on the characteristics of the second generation data relay system, the key technologies of the transmission processing of the 10 Gigabit firewall based on the x86 architecture for different data flows are researched, the optimization design of the CPU allocation and scheduling mode, the function judgment mechanism of the network card characteristics and the kernel state write operation function of the 10 Gigabit firewall are completed. The test results show that the optimized 10 Gigabit firewall gives full play to its maximum transmission performance, and provides a better security transmission service guarantee for the integrated data relay users.

1 Introduction

The second generation data relay satellite ground application system firewall is the core equipment to realize the security protection of the computer network architecture. It uses the white list control strategy to access the backbone transmission network to filter the data flow through the network boundary[1], prevent the unauthorized access of external users, and realize the security protection of the internal infrastructure and data transmission business of the system.

The transmission network of data relay satellite system is oriented to the space laboratory, space station, spacecraft and other manned spacecraft, as well as the integrated users of satellites, launch vehicles, etc[2]. It carries a variety of application data streams, such as measurement and control messages, video voice, image data, experimental data, file data, website data, etc[3]. Among them, the measurement and control message, the video voice of

* Corresponding author: yx1199207@126.com

heaven and earth, and some experimental data are not only the key business, but also the data communication based on UDP multicast in the transmission layer, which puts forward higher requirements for the real-time high-speed processing ability of UDP multicast in the backbone network firewall.

The existing firewalls are divided into a variety of hardware technology architectures, such as x86 architecture [4], MIPS Architecture [5], ASIC Architecture [6]. The 10 Gigabit firewall deployed in the backbone transmission network of data relay satellite system is based on the mainstream x86 hardware architecture, which has the advantages of mature technology, good functionality and scalability. However, through the test of UDP multicast transmission performance under the condition of different packet length and different data stream number, the actual transmission capacity does not reach 10Gbps. When 24 UDP protocol data flows (each packet length is 800byte) are generated on the backbone network by the Spirent 10 Gigabit tester, the UDP multicast transmission performance of the firewall is only 6.5Gbps. According to the statistics of the messages sent and received by the network port of the firewall, there are a lot of packet losses in the process of data transmission, and the CPU occupation of the firewall has reached 100%. If the IP address of the test data stream is changed, the transmission throughput performance of the firewall will fluctuate between 6.5Gbps and 9Gbps, which can not meet the real-time and high-speed data transmission requirements of the data relay satellite system.

Based on the characteristics of data relay satellite system's real high-speed data transmission service, the research on the scheduling algorithm and key technologies of the backbone network firewall for different data flows are completed, the kernel allocation scheduling method based on "flow oriented filtering rules", and function switch judgment mechanism of the network card characteristics is proposed, the write operation function of the kernel state of network card is optimized. The test and verification in the actual backbone network environment shows that the optimized 10 Gigabit firewall has given full play to its maximum processing performance on the basis of maintaining the existing functionality and reliability, and the safe transmission capacity of the backbone network of the data relay satellite system is further improved.

2 Analysis of Firewall scheduling algorithm and processing capacity for data flow

In order to match the real-time and high-speed processing requirements of the transmission network of data relay satellite system for key services such as video and voice, the firewall needs to have the function of keeping the order of the processed data flow, that is, to keep the same data flow through the firewall without changing the transmission order of all data packets. For example, if the high-speed data flow of UDP protocol is out of order, the real-time processing ability cannot be guaranteed even if the application software corrects the out of order data packets. Therefore, the 10 Gigabit firewall of backbone network adopts single stream and single core processing mode for different data streams.

At the same time, the number of internal CPUs corresponding to the network interface card interconnected between the existing 10 Gigabit firewall and the routing and switching equipment is 8. If the data flow through the firewall can not be evenly distributed to the 8-core CPU, some CPUs will process multiple data flows while other CPUs will be idle, resulting in the problem of uneven CPU allocation in the firewall. The overload of some CPUs directly affects the UDP multicast in the firewall processing performance.

2.1 Data flow scheduling algorithm of hash

When the network interface card of 10 Gigabit firewall receives the first packet of a data stream, it will first check a series of security rules such as "two-layer filtering" and "five tuple filtering", and then use HASH algorithm to calculate the queue index[7]. Different queue indexes will correspond to different logical CPUs, that is to say, the firewall will complete the CPU allocation of the data stream through HASH algorithm. The subsequent packets of the data stream will be processed according to the CPU allocation result of the first packet. The key design of hash algorithm is as follows:

Algorithm 1 firewall hash calculation pseudo code

```
1  RSSKEY[10] = A 40 byte key set by the network card ;
2  ComputeHASH(input[]=Packet quintuple , N= The number of bit bits in a five
tuple)
3  {
4  result = 0 ;
5  for( N > 0 ) {
6      b = input[] // Leftmost bit value ;
7      If ( b == 1 )
8          result = result ^ RSSKEY[] // Leftmost 32bit value ;
9      N = N-1 ;
10     RSSKEY[] = RSSKEY[] // Shift 1 bit left ;
11 }
12     Return result ;
13 }
```

It can be seen from the implementation of hash algorithm that the key parameters of calculation include the packet quintuple information of the data stream and the rsskey information of the network card. The rsskey value of each network interface card of the 10 Gigabit firewall is different. Theoretically, different quintuple information of packets and different network card receiving packets will lead to different hash calculation results. The hash calculation results need to be moduled again, and mapped to the corresponding CPU according to the moduling results. Different hash values may be the same after moduling, that is to say, the hash algorithm results have certain randomness, and then different data stream addresses correspond to different allocation results, as shown in section I. The firewall processing performance appears up and down.

2.2 Single core processing capacity of firewall CPU

In addition to achieving the average allocation of CPU resources as far as possible, the processing power of single core CPU is also the key to affect the transmission performance of firewall. In order to evaluate and test the processing capacity of 10 Gigabit firewall single core CPU, the special parameters of network card at the bottom of firewall can be used. we can directly specify a data flow source address to a queue and associate it with a CPU core. By changing the number of data flows and data flow size assigned to the CPU core, we can get the ultimate performance of the CPU single core.

In the actual test, the load of 10 Gigabit firewall single core CPU on different data traffic is shown in Figure 1. Before each test, the default configuration of the network card is reset. For the first time, four 700Mbps and one 200Mbps data stream are bound to the CPU core. The utilization rate of the CPU reaches 100%. A large number of packets are lost when the data stream arrives at the network interface. For the second time, four 700Mbps data streams are bound to the CPU core, and the utilization rate of the CPU is still 100%. The third time, the bound data traffic is reduced to three 700Mbps, and the CPU utilization rate is reduced to 78%. Finally, three 700Mbps and one 400Mbps data stream are bound to the CPU core. The utilization rate of the CPU is 95%, almost reaching its performance limit.

Fig. 1. The load distribution of different data flow processed by a single CPU in 10 Gigabit firewall.

Based on the above evaluation and test, it can be concluded that the single core CPU of the 10 Gigabit firewall can carry 3 pieces of 700Mbps and a rest 400Mbps multicast traffic at most (the remaining few CPUs complete the information processing such as the operation of the equipment system), and when it reaches 4 pieces of 700Mbps multicast traffic, it will exceed the load. The interconnection network interface of the 10 Gigabit firewall can process data flow with 8-core CPU, so it can support data flow processing of 24 treaty 700Mbps at most for high-speed data transmission service.

3 Optimized design of firewall processing mechanism

3.1 Optimization of CPU binding

In the actual data service, the data flow of backbone transmission network can be divided into high-speed flow and low-speed flow. We need to focus on solving the problem of average CPU allocation of high-speed flow to avoid more than three high-speed flows processed by the same core CPU at a certain time. We know that the address of the server generating the high-speed flow is some fixed address range, so we can identify the high-speed flow related to these addresses in the pre-defined configuration of the firewall, and carry out the automatic average allocation of the high-speed flow to the CPU core processing, so as to ensure that each CPU processes three high-speed flows at most, so as to achieve the maximum processing performance of the 10 Gigabit firewall. Those unidentified and undefined data flows are still automatically allocated to a CPU core for processing according to the results of the default hash algorithm.

Specifically, after the network card receives the data flow, the branch of "flow oriented filter rule" is added in the existing processing flow, so that the high-speed flow in the specified address range can be accurately allocated to the logical CPU without hash calculation. The

processing flow of the network card queue calculation is shown in Figure 2. The blue marking flow in the figure is the basic CPU allocation and scheduling process of the unbound data flow, and the red marking flow the automatic average allocation process for a specified address range high-speed stream.

Fig. 2. Optimization diagram of CPU allocation and scheduling mode of 10 Gigabit firewall.

3.2 Optimization of function judgment mechanism of network card

To ensure the correct distribution of the specified high-speed flow address configuration, it is necessary to check the feature function switch of the network card and the current status of the network card[8], and add the following judgment mechanism in the firewall configuration logic (as shown in Figure 3):

1) before issuing the rule each time, judge whether the network card feature function switch is on. If it is on, the start command will not be executed. If it is not on, the

corresponding start command will be executed.

2) when issuing rules, judge whether the network card status is link up. If not, do not issue the specified address rule (that is, the corresponding interface status of the network card is unavailable). When the network card status changes to link up, issue the corresponding rules.

Fig. 3. Flow chart of feature function judgment mechanism of 10 Gigabit firewall network card.

3.3 Optimization of kernel state write operation of network card

Because the specified high-speed stream address configuration needs to be written to the network card control unit [9,10] through the kernel state receive buffer, the structure of the specified stream address configuration contains the header information and IP address information. Each IP address stored in integer form will occupy 4Byte space size. Considering the maximum processing performance of the 10 Gigabit firewall, it can send up to 24 primary IP addresses and 24 standby IP addresses, the IP address needs 192Byte in total. Without calculating the space occupied by the structure header information, the address space requirement has exceeded the actual size of the kernel receive buffer by 128Byte.

By optimizing the kernel state write operation function of the specified stream address configuration of the firewall, the method of circular reading and writing to the control unit one by one is adopted to avoid the overflow of the kernel state receive buffer during the distribution of the firewall function configuration, so as to ensure the high-speed data transmission and reliable and stable operation of the 10 Gigabit firewall. The specific optimization design of kernel state write operation function is as follows:

Algorithm 2 firewall kernel state write operation function pseudo code

```
1  struct rule_node rule;
2  /*.....*/
3  for (i = 0; i<PORT_LIST_MAX && portlist[i][0]; i++)
4  {
5      if ((fd = ethtool_glink(portlist[i])) < 0)
6          Continue;
7      for (j = 0; j < IP_ADDR_MAX && host[j][0]; j++)
8      {
9          Memset(&rule, 0, sizeof(struct rule_node));
10         /*.....*/
11         ret = down_rule(fd, ADD_RULE_CMD, &rule);
12         if (ret < 0)
13             return ret;
14     }
15 }
16 int down_rule(int fd, unsigned long cmd, struct rule_node *rule )
17 {
18     int ret = 0;
19     ret = ioctl(fd, cmd, rule);
20     return ret;
21 }
```

4 Test and verification

The system test environment is shown in Figure 4. For the optimized 10 Gigabit firewall (configuration: CPU, 8-core 2.6GHz; Memory: 256GByte, network interface, 4 full duplex 10 Gigabit), use the Spirent 10 Gigabit tester[11] to construct UDP protocol data flow, and respectively complete the performance test of 10 Gigabit firewall, the specified high-speed flow function effectiveness test, redundant switching function and performance test and the effect of the specified high-speed flow function.

Fig. 4. Network topology of actual system test environment.

Specific test results include:

1) After upgrading the 10Gbps firewall, the transmission throughput performance of packet length above 800Byte is close to 10Gbps, which plays the maximum processing performance of the 10Gbps firewall. The performance test results of different packet lengths are shown in Figure 5.

Total Trials	Number of Passed Trials	Frame Size (bytes)	Intended Load (%)	Offered Load (%)	Throughput (%)	Aggregated Throughput (fps)	Aggregated Theoretical Max (fps)	Aggregated Throughput (Mbps)	Aggregated Theoretical Max (Mbps)
1	1	64	13.375	13.373	13.373	3,979,944.467	29,761,904.762	2,675	20,000
1	1	128	24.203	24.183	24.183	4,084,967.692	16,891,891.892	4,840.625	20,000
1	1	256	45.086	45.05	45.05	4,080,586.917	9,057,971.014	9,017.188	20,000
1	1	512	86.078	85.806	85.806	4,032,258.792	4,699,248.12	17,215.625	20,000
1	1	809	100	99.639	99.639	3,004,807.733	3,015,681.544	20,000	20,000
1	1	1,024	99.227	99.188	99.188	2,375,183.15	2,394,636.015	19,845.312	20,000
1	1	1,280	99.227	99.085	99.085	1,905,487.817	1,923,076.923	19,845.312	20,000
1	1	1,518	100	99.87	99.87	1,623,376.933	1,625,487.646	20,000	20,000

Fig. 5. Test results of throughput performance of 10 Gigabit firewall.

2) By starting the test traffic one by one, check that the CPU load defined by the firewall policy is gradually increasing, and the firewall can automatically realize the average distribution of CPU according to the policy configuration accurately. When two-way 8 streams, two-way 16 streams and two-way 24 streams are started, the results of 0-7 core load utilization of firewall CPU are shown in Figure 6.

Fig. 6. CPU load utilization rate of different data flow of 10 Gigabit firewall.

3) Restarting the firewall carrying data traffic does not affect the normal switching function of the firewall, and the switching time is within 1 second. After the handover, the data flow is still allocated to the CPU according to the predetermined strategy, and after the original firewall status is normal, there is no data flow failback and abnormal transmission, which ensures the redundancy and high availability of the firewall. The results of comparing the redundancy switching performance of firewall after upgrading and before upgrading under the conditions of single link cutting, device cutting and device restarting are shown in Figure 7.

4) When the specified high-speed flow function of the firewall takes effect or fails, a small number of packet loss and disorder may occur to the data flow being transmitted, which is caused by the corresponding switching of the processing flow of the network card and the triggering of CPU reallocation. By controlling the firewall configuration change operation in the gap between the transmission of critical data services, the impact of packet loss and disorder on critical data services can be avoided.

Fig. 7. Test results of redundant switching performance under different switching conditions.

5 Conclusion

Based on the characteristics of high-speed data transmission in data relay satellite system, the key technologies of different data flow forward processing in 10 Gigabit firewall are studied, several optimization methods, such as CPU allocation and scheduling, function judgment mechanism of network card characteristics and kernel state write operation function are proposed. The optimization method has been extended to the forwarding process of 10 Gigabit firewall for TCP protocol data flow, and finally improved the overall transmission throughput performance of the firewall, which can provide a reference for the subsequent use of the backbone network firewall and related system construction work.

References

1. Chen Jingyu. Firewall strategy anomaly detection and handling, University of Electronic Science and technology, 2017, 5:11-15
2. Wiki. Tianlian-2, 01, <https://vip.kuaimen.bid/extdomains/zh.wikipedia.org//zh/Hans/tianlian-2,01>
3. Tao Ji,Fan Yang,and Luyang Pan. The Realization of High Speed Data Transmission and Processing of Space-Based TT&C Network. Proceedings of the 27th Conference of Spacecraft TT&C Technology in China,v323:597-607,2014
4. Rust, Adamson. AMD bans use of Hammer word X86-64,The Inquirer,2003
5. MIPS. MIPS Instruction Reference, <http://www.mrc.uidaho.edu/mrc/people/jff/digital/MIPSir.html>

6. Kunwar Singh, Satish Chandra Tiwari, Maneesha Gupta. A closed-loop ASIC design approach based on logical effort theory and artificial neural networks, *Integration*, 07:10-22,2019
7. Shang Wenli, Lei Yanqing, Wan Ming, Zeng Peng. Self learning method of industrial firewall rules based on hash algorithm, *computer engineering and design* [J], 2016,3:61-65
8. CSDN. Introduction to embedded network card driver framework, <https://blog.csdn.net/w1107101310/article/details/79616286>), 2018
9. Liu Jinming. Research and implementation of high-speed network card data transmission, *University of Electronic Science and technology*, 2013, 3:30-33
10. Wu Yong. Interrupt mitigation mechanism in high performance packet processing, *high performance computing technology* [J], 2004,3:197-201
11. spirit. Test scheme for high performance network security equipment, <https://www.spierent.cn/products/testcenter/platforms/appliances>), 2019.