

Research on Data Synchronization Mechanism for Distributed System

LeGe^{1,a}, YixinSu^{2,b}, Shijia Cheng^{2,c}

School of Automation, Wuhan University of Technology, Wuhan 430070, China;

^agele_1005@163.com, ^bsuyixin@whut.edu.cn, ^ccheng_sj@whut.edu.cn

Keyword: UDS; data Synchronization; distributed system; network congestion

Abstract. In the field of data sync technology, compared with other protocols, because of high efficiency and speed UDP transmission has been widely applied, but it can't provide a reliable data transmission service which is connection-oriented, as a result of which decreases the utilization of data transmission. Based on this, combined bulk retransmission and acknowledgement mechanism, this paper presents a new protocol UDS (Period ACK Reliable UDP) to improve the traditional UDP, merged resend task into the time slice of single circle sending to achieve virtual parallel processing for sending task and resend task. The experimental results indicate that the UDS keep the advantages of traditional UDP transmission, at same time, improve the reliability and reduce the influence of retransmission on transmission efficiency and network bandwidth.

I. Introduction

In the aspect of data sync, there's often a tradeoff between speed and reliability [1-2]; The UDP protocol has advantages of high efficiency and speed, but it also exists disadvantage that it can't provide a reliable transmission service. So improvements and optimization of the UDP protocol has become the focus of the research of distributed data sync mechanism.

Document [3] has designed a kind of reliable RUDP. Every time the sender sends a data message, it enters timeout waiting state and will not jump out until ACK message is received from the receiver. Then a new turn begins. Undoubtedly, too much time delay affects data sync. Based on the RUDP, the document [4] proposed a kind of BARUDP (Bulk ACK Reliable UDP) with the function of Bulk ACK or timing confirmation. As the receiver will send ACK on the condition that a certain amount of data packets have been received. Compared with the RUDP, the mechanism has saved lots of time and resources; But there is only one sender that once retransmission is occurred, the sender has to suspend the current sending task and turn to retransmission which reduces the data sync efficiency. Document [5] has proposed a enhanced reliable UDP protocol named ERUDP. It adds a sender specialized in resending those packets that are losing or error, as a result of which, avoiding the impacts on other senders. However, the retransmission mechanism is established on the condition that data messages are sequential which against the basic laws of UDP protocol; Moreover, that adding a sender may cause network congestion with the increase of network nodes.

Through the comparative documents above, this paper presents a kind of UDS (UDP-based Data Synchronization Protocol), the data sync periods will be divided into several time slots by which the sender will execute sending task and resending task with Round-Robin. An end task data packet will be appended to the end of the task. The receiver will periodically send retransmission ACK to the sender to realize the toleration of losing and out of order of packets during the data sync. It increases the reliability of data sync and reduces the impacts caused by retransmission on sender and network bandwidth.

II. UDS Implementation Mechanism

The connectionless-oriented feature of UDP has prompted the inevitability of losing packets and it is usually to introduce the ACK mechanism. The UDS improves the traditional ACK mechanism that sending ACK toward a single period instead of each packet to reduce bandwidth consumption and the impacts on sending packets.

The basic idea of UDS is to divide each data sync period into N time slots that both sending task

and resending task will get one segment from them to perform rotatably in turn. The UDS protocol model is shown in Fig.1

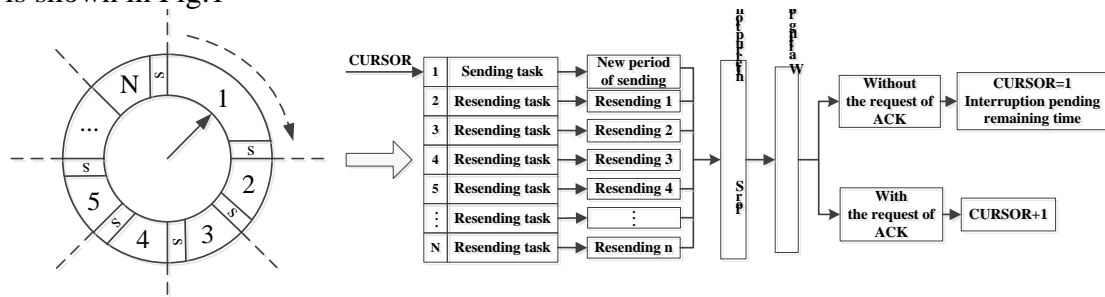


Fig.1 UDS protocol model

The pointer of time wheel rotates a circle stands for one data sync period. The sender maintains two kinds of tasks: normal sending task and retransmission task. When each single period begin, the new period sending task will be executed firstly. Then the sender will be interrupted to hang for S time interval during which it will wait to receive ACK asynchronously from the receiver; After back from the interruption, the ACK packets information will be parsed, if there has packets loss, the first retransmission executes. The second hanging for S time interval will carry on as soon as resending is done and also waiting for ACK and so on; If there hasn't packets loss, sender will keep hanging until the time pointer rotates to the next new period. Assume that the distributed data sync period is T, time interval of sending is $t_n(n=0,1,2,...N)$. T satisfy the following formulas:

$$T = \begin{cases} \sum_{n=1}^N t_n + NS & \text{(Request of ACK in Nth time)} \\ \sum_{n=1}^M t_n + MS + S_{remain} & \text{(No request of ACK in (M+1)th time)} \end{cases} \quad (1)$$

In the industrial control network, the demand for real-time in distributed data sync is usually second or millisecond which means that there have plenty of time for single data transmission. That the remaining time are divided equally between retransmission tasks when new period sending task is done actually parallel virtually normal sending task and retransmission task. The sender doesn't need to suspend current normal sending task to resend the packets that is lost in the previous task which improves data transmission efficiency and also reliability is guaranteed.

III.UDS Strategy Design

The sender will send out data that need to synchronize in current period within the first sending task time slot. A end task data packet will be appended to the end to remind the receiver of the ending of normal sending task this period and to start checking packets loss situation to resend ACK. The end task data packet shares the same structures with the normal data packets. The only difference is that the actual data length is 0. After adding the end task data packet, the PARUDP mechanism no longer relies on packets' arriving in turn which reduce the cling to UDP layer. The structure of data packets is shown in Table 1.

Table 1 Structure of data packets

Packet type	Source node ID	Period serial number	Period total data packets	Data packet serial number	Sending times	Data
-------------	----------------	----------------------	---------------------------	---------------------------	---------------	------

Packet type: including Sync data packets, end task data packets and ACK packets.

Source nod ID: network node ID that the packet belongs to.

Period serial number: period of transmission

Period total data packets: total data packets that need to be sent out in current period.

Data packet serial number: the serial number of current data packet in total data packets.

Sending times: sending times of data packet in current period.

Data: the contents of data packet

The receiver maintains a receiving registration form which will record data packets reception. When a sync data packet is received, receiver will analyses packet type. If it is a new period data packet, the receiving buffer will be cleared and stores the packet, if it is the packet in current period, the packet will be stored in the buffer directly. The registration form will tag for the packet each time the data packet is received. Once the end task data packet is received, the receiver will check the registration form. If data packets have been received integrity, then data will be updated into data memory block in local node. If there have packets loss, the receiver will make ACK packets according to the registration form and send ACK to the corresponding sender asking for retransmission. The working process of the receiver is shown in Fig.2

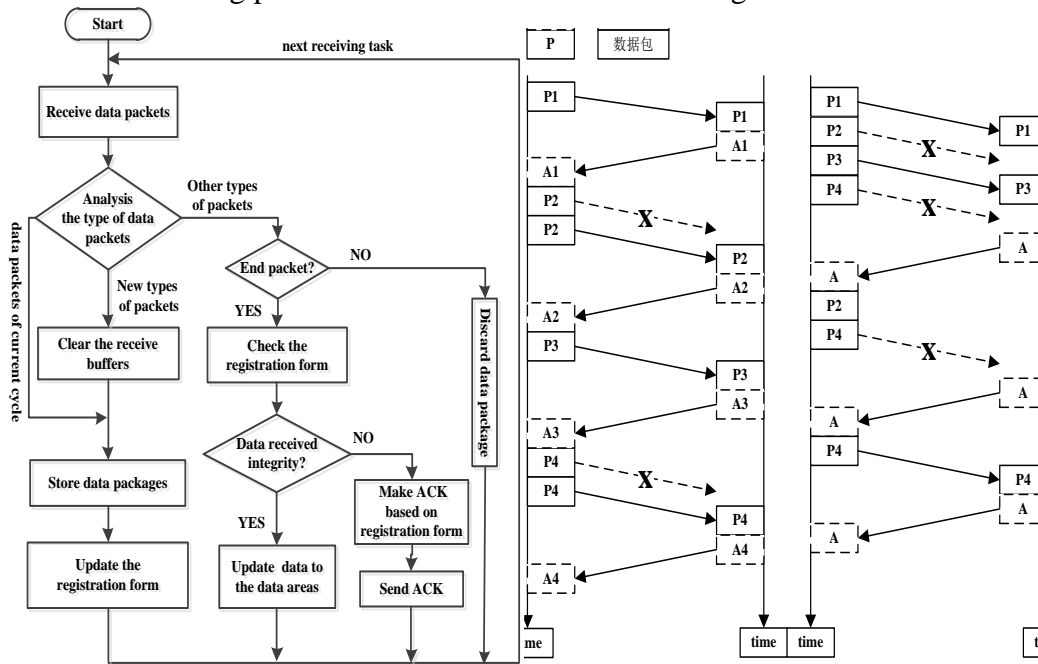


Fig. 2 Working process of the receiver

Fig.3 Comparison of ACK mechanism

VI. Control of Network Congestion

Distributed network often has dozens or even hundreds of nodes, as a result of which produce a large amount of data exchanging. Because store-and-forward resources are limited, if no effective control measures of network load have been carried out, it will lead to data losing, delay increasing, dropping in throughput, even can cause an avalanche slumping of transmission network. The received data packets are confirmed or request a resend one by one in traditional ACK mechanism, on the contrary, UDS aimed at each of the data sync period to send an ACK, thus decreasing the influence of ACK on network load. As shown in Fig.3 for details.

We set K is the number of network node, P is the packet loss of network, L is bandwidth consumption, synchronous data is Q in period T (ms), the size of synchronization packet is U , V is the size of ACK packet. Using UDP protocol to broadcast for each packet, as long as a package is lost, it have to resend this set of data; For traditional ACK mechanism, each received a packet in the network node needs to send an ACK for $K-1$ times, while UDS mechanism needs to send an ACK for n times (n is the number of retransmission times), so retransmission probability of each group of data is showed as follows: $P_k = 1 - (1 - P)^k$

During the period T , expectation of data packet loss is showed as follows:

If using traditional ACK mechanism,

$$L_1 = \frac{1000}{T} \left[Q + RU + \left(\frac{Q}{U} + R \right) (K - 1)V \right] \quad (2)$$

If using improved ACK mechanism,

$$R = \frac{Q}{U} \sum_{n=1}^{\infty} P_k^n$$

$$L_1 = \frac{1000}{T} [Q + RU + n(K-1)V] \quad (3)$$

$$S_1 = \frac{Q}{U} + R$$

$$= \frac{Q}{U} [1 + P_k + P_k^2 + \dots + P_k^n]$$

$$= \frac{Q}{U} \left[\frac{1 - P_k^n}{1 - P_k} \right]$$

$$S_2 = n$$

$$\text{When: } P_k = 1 - (1 - P)^k$$

$$\lim_{k \rightarrow \infty} S_1 = \frac{Q}{U} \frac{[1 - (1 - P)^k]^n - 1}{(1 - P)^k}$$

$$\approx \frac{Qn (1 - P)^k}{U (1 - P)^k} = \frac{Qn}{U}$$

$$\therefore \frac{S_2}{S_1} \approx \frac{Q}{U}$$

Considering the large amount of synchronous data (at least megabytes orders of magnitude) in the distributed storage system of industrial field, however which based on UDP protocol in Ethernet, the maximum length of data transmission is 65535 bytes, while allows the MTU (the size of maximum transmission unit) is only 1500 bytes at the hardware level, so:

$$S_2 \ll S_1 \quad \therefore L_2 \ll L_1,$$

It is obvious that UDS mechanism has better effects on the control of network load which reduces the probability of network congestion

V. Reliability Analysis

To test packet loss based on UDS protocol in gigabit Ethernet in the Linux system. The period T is 100ms, the sync data size is 1M bytes and time slots is 3.

The packet loss is directly related to the size of the packet. Therefore, the minimum unit of data packing is regarded as test parameter to compare the impact of the traditional ACK mechanism with the improved ACK mechanism on the packet loss. Using two computers as test nodes, while a node sends 1M data, another node will receive from it. The interval of sending task chooses 0.5ms. After 50 tests for 102400 packages, the result of all the sample means is shown in Fig. 4.

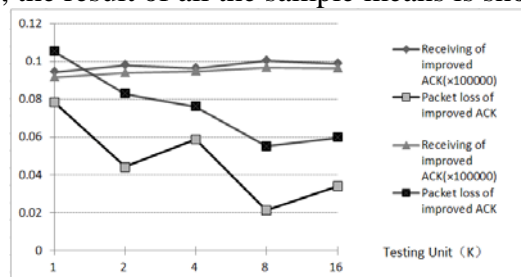


Fig. 4 Effects of traditional and improved ACK mechanism on packet loss

Test results show that the number of received packets by UDS mechanism are more than the number of received packets by traditional ACK mechanism when both of them receive 102400 packets, the packet loss is reduced and accurate of data transmission has reached the allowable error range, which can ensure the reliability of data sync.

VI. Conclusion

Aim at unreliability of data transmission in distributed data sync based on traditional UDP protocol, this paper analyses the existing problems that the improved UDP proposed by the current documents may have impacts on transmission efficiency and network bandwidth and presents a kind of UDS protocol based on traditional UDP which have function of the toleration of losing and out of order of packets, ensuring the efficiency and reliability of data sync. By comparing the network bandwidth under the traditional UDP protocol and the UDS protocol, the optimized effect of UDS protocol on the control of network load is proved, and the experimental results verify the conclusion above.

References

- [1] Zheng H, Boyce J. An improved UDP protocol for video transmission over Internet-to-wireless networks [J].IEEE Transactions on Multimedia,2001,3(3):356 - 365.
- [2] Lam P K, Liew S C. UDP-Liter: an improved UDP protocol for real-time multimedia applications over wireless links[C]Wireless Communication Systems,2004,1st International Symposium on. IEEE, 2004:314 - 318.
- [3] LiGuo-dong, Zhang Lin-lin, Liu Chang-an.Research on Video Transmission Technique Based on RUDP [J].Computer Engineering, 2006, 32(24):226-228.
- [4] Ben Eckart,He Xu-bin,Wu Qi-shi.Performance adaptive UDP for high-speed bulk data transfer over dedicated links[C].Proceedings of the 2008 IEEE InternationalSymposium on Parallel and Distributed Processing,2008,1-10.
- [5] Jin Hai-li, LiJun.Implementation of Enhanced Reliable UDP with Supplement Mechanism[J].Journal of Chinese Computer Systems,2010,31(5): 904-907.