

A Data Scheduling Scheme Based on Double Queue and Send-window

Hongcheng Huang

Department of Software Center
Chongqing University of Posts and Telecommunications
Chongqing, china
Huanghc@cqupt.edu.cn

Hongmei Zhang

Department of Software Center
Chongqing University of Posts and Telecommunications
Chongqing, china
zhanghongmei_cqupt@163.com

Abstract—Multipath Transmission Control Protocol (MPTCP) can provide multipath communication end to end. It improve the utilization of network resources and network transmission reliability. In this paper, the receive buffer blocking problem[1], which is the basic problem in the MPTCP, is presented. A data scheduling scheme, DQS, is proposed to deal with this problem. We divide the sender packet queue into two queues and then reasonably dispatch data packets to the sub-path, in combination with the number of send window and the sub-path performance parameters. Analytical results show that the DQS scheme reduces the receive buffer blocking, effectively improving the throughput and the average network bandwidth utilization rate.

Keywords—MPTCP; receive buffer blocking; data schedule scheme

I. INTRODUCTION

The user demand for bandwidth is growing with the development of the Internet services. However, the Internet resources, especially the limited bandwidth resources, cannot be effectively used because the existing Internet uses single-path transmission. At the same time, along with the reduction of the cost of access equipment and the diversification of network access technology, we can access Internet through ADSL, LAN, WIFI, GPRS, WIMAX, 3G, LTE, etc, namely, more and more devices can be equipped with multiple network address. Under this premise, using multiple network interface of communication terminal for data transmission because, which can lead to higher throughput and robustness, is becoming a hot spot of academic research.

MPTCP[2], which can provide multipath communication end to end, with the characteristics of high fault tolerance, provides a solution for the above problem[4]. According to RFC6182 suggestion, MPTCP usually use Round Robin scheduling scheme (RR) [3] to dispatch data. It dispatches the packets, which come from application layer, to different path orderly. The RR scheme is the most simple and direct packet scheduling method. However, this scheme hardly takes the diversity of sub-path performance parameters into account, which may make the packets cannot reach the receive buffer in order and cause disorder phenomena. Moreover, the loss of packets may leads the packets which are sent later arrive the receiver earlier, resulting in more serious disorder problem[4]. As MPTCP is a reliable transport mechanism, the receiver can only submit data for the upper layer in order. When disorder phenomena occur, the packets will be

accumulated in the receiver. When large differences exist in performance between sub-path, the receive buffer will cause serious obstruction, greatly reducing the transmission efficiency and the throughput of MPTCP[5].

In order to deal with the problem of receive buffer blocking, we propose a data scheduling scheme using double queue and send-window (DQS), aims to reasonably dispatch the data stream to each sub-path, solving disorder and receive buffer blocking issues.

II. THE DQS DATA SCHEDULING SCHEME

A. Description

In this paper, we shall first briefly introduce some formulas and related concepts.

1) RTT_i and FT_i

RTT_i is the packet round-trip delay of path i ; the amount of time it takes for a sender to transmit a data segment to the receiver and for the receiver to transmit an ACK to the sender.

FT_i is forward transmission delay of path i , namely the time between packet leaves the send buffer and the packets is received by the receiver. We rough estimate FT_i equals half of RTT_i .

$$FT_i = RTT_i / 2 \quad (1)$$

When the sender of path i receives ACK, it will estimate the current value of RTT_i . For estimate RTT more accurately, we need to collect the value of multiple samples. So RTT_i is calculated as: (β is a ratio factor. Based on a series of experiments, we choose $\beta=0.75$.)

$$RTT_i \leftarrow \beta RTT_i + (1 - \beta) rtt_i \quad (2)$$

All of the RTT and FT in this paper are average RTT and FT .

2) Average Successful Transmission Time(T_i)

Taking into account the actual environment state of each sub-path changing, we calculate T_i based on transmission delay and packet loss rate of each sub-path.

Assuming the packet loss rate of sub-path S_i is l_i , the forward transmission delay is T_i , and the timeout delay is

RTO_i . Assuming packet loss events are independent. Then the time that a packet once successfully reached the receiver is:

$$R_0 = FT_i \quad (3)$$

The time that a packet reach the receiver successfully, experienced one retransmission, is:

$$R_1 = RTO_i + FT_i \quad (4)$$

The time that a packet reach the receiver successfully, experienced two retransmissions, is:

$$R_2 = 2RTO_i + FT_i \quad (5)$$

The time that a packet reach the receiver successfully, experienced m times of retransmissions, is:

$$R_m = m \times RTO_i + FT_i \quad (6)$$

Thus, the average time of a packet successfully reaches the receiver is:

$$T_i = (1-l_i)R_0 + l_i(1-l_i)R_1 + \dots + l_i^m(1-l_i)R_m \quad (7)$$

Into the above formulas, we have

$$T_i = (1-l_i) \sum_{k=0}^m k \times RTO_i + FT_i \quad (8)$$

In the actual network environment, three successive retransmissions timeout occurs in MPTCP is almost impossible. Therefore, we make $m=3$ into the formula, we have:

$$T_i = (1-l^4)FT_i + (l+l^2+l^3-3l^4)RTO_i \quad (9)$$

B. The Implementation of the DQS

DQS uses double send queue, the data queue and sub-path queue. As shown in Fig.1, the data queue is responsible for grouping and sequentially numbering the data stream from the application layer. The sub-path queue is responsible for processing the data, which comes from the data queue, in the sub-path buffer.

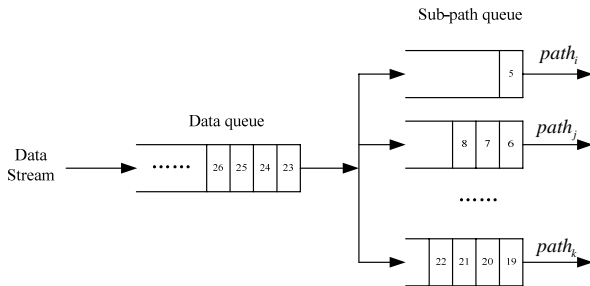


Figure 1. Double Queue.

The implementation steps of DQS scheduling scheme are as follow:

- The data stream is divided into several data packets with the same size. Then number the packets in order;

- According to the values from small to large order T_i , we dispatch data packets to each sub-path. The number of the dispatched data packets is equal to the number of receive window. That is to say, $n_i = w_i$, $n_j = w_j$, ..., $n_k = w_k$;
- After the sub-path queue receives data packets, we put the data packets into the sub-path send buffer, waiting to be sent;
- The Re-Scheduling is used when one sub-path becomes invalid or timeout. The Re-Scheduling puts the data packets, which are waiting to be sent in the sub-path send queue, into data queue. Then the data packets will be dispatched to other sub-path.

III. SIMULATION

We use NS3[6] to demonstrate the performance of the scheme proposed. In this section, the numerical results based on the analytical expressions obtained in Section II are presented.

A. Set up Simulation Scenario

We use the network topology shown in Fig.2 and set up the simulation environment without bottleneck link to evaluate the performance of DQS scheme. We set the sender buffer to 128KB and the receiver buffer to 64 KB, use DQS data scheduling scheme, use RTT_Compensator Congestion control scheme, use D_SACK receiver data reconstruction scheme.

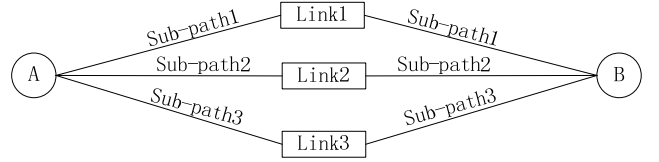


Figure 2. Network Topology

There are four sub-paths between the sending and receiving ends. The parameters of sub-paths are shown in TABLE I.

TABLE I. JUDGMENT MATRIX

sub-path number	Sub-flow0	Subflow1	Subflow2	Subflow3
Bandwidth (Mbps)	5	5	2	2
RTT(ms)	20	(20+n*5)	(20+n*10)	(20+n*20)
Packet loss rate(%)	0.2	0.1	0.3	0.2
n=0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10				

B. Simulation Results and Analysis

In this section, we compared the performances of the two scheduling schemes: DQS scheme and RR scheme. The simulation results are shown in Fig.3 and Fig.4.

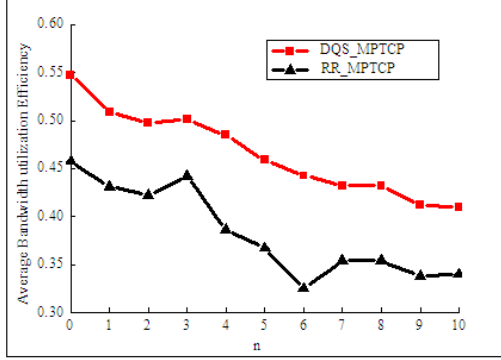
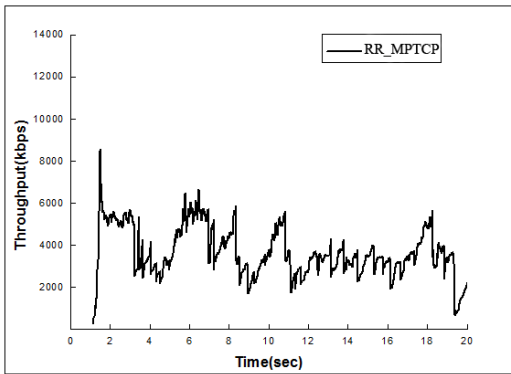


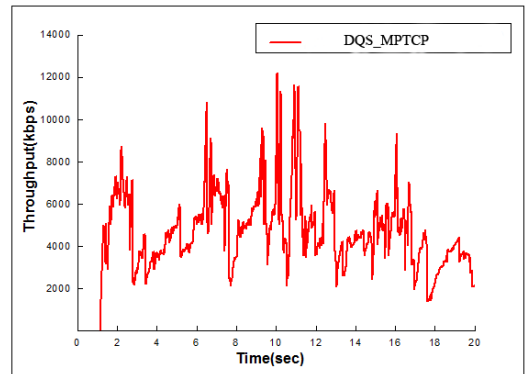
Figure 3. the average bandwidth utilization rate for RR_MPTCP and DQS

Fig.3 shows the average bandwidth utilization rate for two settings when n ranges from 0 to 10. Under the situation, the bandwidth utilization rate of RR_MPTCP is less than that of DQS_MPTCP. As n increases, the bandwidth utilization rate of RR-MPTCP decreases unsteadily because of the unstable variation of the congestion window, which is caused by multiple packet loss or timeout. But in DQS_MPTCP mechanism, the average bandwidth utilization decreases steadily. DQS_MPTCP uses double send queue, as far as possible to ensure packet number in the sub-path sender meets the current number of send window, without resulting in the data accumulation of the sender, avoiding congestion problems. Thus effectively avoid the emergence of out-of-order packets, making the transmission more stable and achieving load balancing.

Fig.4 shows the throughput comparison between the DQS_MPTCP and RR_MPTCP. Suppose that $n=0$ and the four sub-paths have the same RTT and different bandwidth.



(a)



(b)

Figure 4. the throughput of DQS_MPTCP and RR_MPTCP

From the Fig.4 (a), we can see that the throughput of RR_MPTCP are less than 1000kbps. In the Fig.4 (b), we find that the throughput of DQS_MPTCP is larger than 1200kbps, which is greater than the throughput of

RR_MPTCP. This is because DQS_MPTCP can reduce the number of disorder packets and improve the efficiency of transferring data packets.

IV. CONCLUSION

In this paper, in order to deal with the problem of receive buffer blocking, the DQS data scheduling scheme is studied. We divide the sender packet queue into two queues and then reasonable dispatch data packets to the sub-path, in combination with the number of send window and the sub-path performance parameters. The numerical results show that DQS scheme outperforms the round-robin scheme. Moreover, relieving the receive buffer blocking, DQS scheme further improve the average network bandwidth utilization rate and throughput.

REFERENCES

- [1] Yi Wang, Xiaojun Liao, Zeyou Pan. Multipath Transmission Control Protocol Technology Review. Information and Electronic Engineering. 1672-2892(2011)01-0007-5, Vol.9, No.1, Feb., 2011.
- [2] A. Ford C. Raiciu, M. Handley, S. Barre, J. Iyengar. Architectural Guidelines for Multipath TCP Development. IETF, RFC6182, March 2011.
- [3] M. Bagnulo. Threat Analysis for TCP Extensions for Multipath Operation with Multipath Addresses. IETF, RFC6181, March 2011.
- [4] Fei Song, Bo Wang, Hongke Zhang. Multi-path Parallel Transmission Blocking Problems of The Receive Buffer. Electronic Journal. Vol.38, No.3, Mar. 2010.
- [5] M. Scharf, A. Ford. MPTCP Application Interface Considerations. IETF, August 2012.
- [6] B.Chihani, D.Collange. A Multipath TCP model for ns-3 simulator. Cornell University Library arXiv:1112.1932, 2011.