Sender-Side Data Allocation Scheme for Mitigating Reordering Degree in CMT (Postprint)

Authors: Wang Zhenchao, Li Haixiao, Hou Huanhuan

Date: 2018-12-13T00:00:00+00:00

Abstract

To mitigate the impact of receiver-side packet reordering on system transmission performance in concurrent multipath transfer (CMT) systems, a novel sender-side data allocation scheme is proposed. The scheme predicts packet forward transmission delay based on path bandwidth, round-trip time, and congestion window, and employs this metric as a factor for determining path transmission priorities within the system. The sender allocates to each path, according to path transmission priority and send buffer status, data packets from the transmission queue that will not cause receiver-side reordering. Simulation results demonstrate that, compared with round-robin and ATLB algorithms, the proposed sender-side data allocation scheme can effectively reduce the number of reordered packets at the receiver.

Full Text

Preamble

Vol. 37 No. 2
Application Research of Computers
ChinaXiv Partner Journal

A Sender-Side Data Allocation Scheme for Reducing Disorder Degree in CMT

Wang Zhenchao[1,2], *Li Haixiao*[1], *Hou Huanhuan*[1]
(1. College of Electronic & Information Engineering, Hebei University, Baoding Hebei 071002, China;
2. Beijing Jiaotong University Haibin College, Cangzhou Hebei 061100, China)

Abstract: To mitigate the impact of packet disorder at the receiver on transmission performance in Concurrent Multipath Transfer (CMT) systems, this paper proposes a novel sender-side data allocation scheme. The scheme predicts the forward transmission delay of packets based on path bandwidth, round-trip transmission delay, and congestion window size, using this predicted delay as a metric for prioritizing path transmission. The sender allocates packets from the transmission queue to each path according to path transmission priority and send buffer status, ensuring that allocated packets will not cause receiver-side disorder. Simulation results demonstrate that compared with round-robin and ATLB algorithms, the proposed sender-side data allocation scheme effectively reduces the number of out-of-order packets at the receiver.

Keywords: concurrent multipath transfer; data disorder; data allocation; forward transmission delay; throughput

Classification: TP393
doi: 10.19734/j.issn.1001-3695.2018.07.0551

0 Introduction

The continuous development of Radio Access Technologies (RAT) and the increasing number of interfaces available on end devices have made it possible for terminals to transmit data concurrently through multiple access methods [1,2]. To meet the requirements of multi-interface technology, the IETF working group proposed the Multipath TCP (MPTCP) protocol in 2009 [3,4], which provides parallel data transmission services for end users while maintaining TCP compatibility, effectively improving the transmission capacity and stability of CMT systems. However, differences in delay, bandwidth, and packet loss rate among multiple paths cause data packets transmitted via different paths to arrive out of order at the receiver, leading to receive buffer blocking and degraded system throughput.

Existing sender-side data allocation schemes for reducing receiver-side packet disorder primarily adopt two packet allocation granularities: batch allocation and single-packet allocation. Xue et al. [5] proposed DPSAF (forward prediction based dynamic packet scheduling and adjusting with feedback), a scheduling algorithm that considers path packet loss rate and transmission time deviation to calculate the number of packets that can be allocated to each path without causing receiver-side disorder, using batch allocation to assign consecutive DSN-numbered packets to each path. She Dongping et al. [6] proposed a sender-side allocation scheme that evaluates path transmission capacity by real-time throughput estimation, using single-packet granularity to assign packets to the path with the strongest transmission capability, thereby reducing receiver-side disorder.

While batch allocation granularity [5] can effectively improve sender-side allocation efficiency, it exacerbates receiver-side disorder in environments with significantly different path qualities. Conversely, single-packet allocation granularity [6] reduces allocation speed but provides higher allocation precision than batch allocation. These differences in allocation speed and precision cause varying impacts on receiver-side disorder across different CMT system environments. Therefore, this paper analyzes the advantages and disadvantages of these two allocation granularities to select an appropriate sender-side data allocation granularity.

After selecting the allocation granularity, the next research focus of our sender-side data allocation scheme is how to allocate packets to various paths in the CMT system to reduce receiver-side disorder. Numerous scholars have proposed solutions to address out-of-order arrival via different paths. Hasegawa et al. [7] proposed ATLB (arrival-time matching load-balancing), which predicts packet forward transmission delay on each path and defines the predicted delay as each path’ s score, allocating packets to the path with the minimum score. However, this algorithm ignores packet transmission delay when predicting forward transmission delay, resulting in inaccurate delay predictions. Chen et al. [8] proposed a sender-side allocation scheme that considers packet transmission delay and receive buffer size limitations when allocating packets to each path, partially alleviating receiver-side disorder. However, both algorithms neglect congestion window size limitations during allocation, causing heavily loaded low-delay paths while other paths remain idle in environments with large path delay differences.

Du et al. [9] considered path delay, bandwidth, and receive buffer size, proposing a differentiated CMT data allocation algorithm based on disorder feedback that establishes a path performance evaluation model to dynamically adjust packet transmission counts per path. Huang Hui [10] proposed a weight-based scheduling algorithm that defines path quality based on delay and packet loss rate, with the sender selecting the best-quality path for transmission. Both [9,10] reduce the impact of poor-performance paths on overall system performance by increasing usage of better paths while decreasing or suppressing usage of poorer paths, but this approach reduces system resource utilization and wastes available bandwidth.

To fully utilize CMT system bandwidth, achieve load balancing, and reduce receiver-side disorder, this paper proposes a novel sender-side data allocation scheme building upon existing algorithms. The scheme considers both path forward transmission delay and available send window space. First, based on the delay prediction methods in [7,8], we incorporate congestion window effects to predict packet forward transmission delay on each path. The predicted delay values then serve as metrics for path transmission priority. Based on path transmission priority and available send window space, the sender calculates appropriate DSN numbers for packets that can be allocated to each path without causing receiver-side disorder, ensuring in-order packet arrival while fully utilizing system bandwidth to improve resource utilization.

---

1 CMT System Transmission Model

Multipath data 分流 (shunting) can be implemented at multiple protocol layers (application, transport, network, and link layers) [11]. This paper focuses on transport-layer multipath data 分流, establishing the CMT system transmission model shown in [Figure 1: see original paper].

After data streams from the application layer arrive at the transport layer, the transport layer segments the data stream into packets of size *Datasize* according to certain load segmentation granularity. Each packet header includes a Data Sequence Number (DSN) for receiver-side data reassembly and a Subflow Sequence Number (SSN) to avoid unnecessary fast retransmissions.

In existing CMT systems, all subpaths share a single send buffer, making it difficult to fully utilize high-quality paths' transmission capacity and potentially causing transmission blocking due to limited buffer space [12]. Therefore, this paper allocates independent send buffers for each TCP subpath. If a path' s send buffer has available space in its send window, allocated packets can be transmitted immediately; otherwise, packets must wait until successfully transmitted packets are acknowledged and window space becomes available. The send window size equals the path' s congestion window value and changes dynamically with congestion control strategies. Independent send buffers enable better-quality paths to handle more data transmission tasks, thereby improving CMT system data transmission rates.

Based on this model, we address the following problem: The sender determines that packet k (with DSN number k) should be allocated to path i to ensure in-order arrival of transmitted packets at the receiver, but path i' s send window is full of unacknowledged packets while other paths have available window space. Existing sender-side allocation algorithms [13-18] typically adopt two approaches: (a) To avoid receiver-side disorder, they wait for path i' s send window to become available before transmitting packet k, despite available bandwidth on other paths. While this reduces receiver-side disorder, it wastes system bandwidth and reduces transmission speed. (b) They directly allocate packet k to other paths, which utilizes available bandwidth but may cause severe receiver-side disorder in environments with large path delay differences.

Our proposed sender-side allocation scheme modifies the traditional allocation pattern, as shown in [Figure 2: see original paper]. Instead of waiting for packet k to be sent before transmitting packets with DSN numbers greater than k, the scheme calculates based on path transmission priority and available send window space, allocating packet from the sender to path x with available send window space, thereby fully utilizing system bandwidth while avoiding receiver-side disorder.

2.1 Path Transmission Priority Classification

Our sender-side allocation scheme classifies path transmission priorities by predicting packet forward transmission delay on each path. The forward transmission delay for a packet on path i, denoted as T_i, is calculated as:

$$T_i = \begin{cases} \frac{Q_i}{B_i} + \frac{\text{Datasize}}{B_i} + RTT_i & \text{if } Q_i < CWND_i \\ \frac{Q_i}{B_i} + \frac{\text{Datasize}}{B_i} + RTT_i + RTT_i & \text{if } Q_i \geq CWND_i \end{cases}$$

where Q_i represents the existing data length in path i' s send buffer, B_i is path i' s bandwidth, *Datasize* is the sender' s packet size, RTT_i is path i' s round-trip transmission delay, and $CWND_i$ is path i' s congestion window value.

When $Q_i < CWND_i$, path i' s send window has available space. After packet allocation, the packet only needs to wait for existing buffer data to be transmitted. The first term represents the time required to transmit existing data out of the buffer, the second term represents the predicted time to transmit the new packet out of the buffer, and the third term represents the predicted time for the packet to reach the receiver after leaving the buffer. When $Q_i \geq CWND_i$, the send window is full of unackmitted packets. After allocation, the packet must wait for acknowledgments before transmission. Therefore, the first three terms have the same meaning, while the fourth term represents the time required for the receiver to return an ACK.

Path transmission priority is defined based on predicted delay values: larger forward transmission delay indicates longer transmission time to the receiver and lower path transmission priority. Therefore, path i' s transmission priority is defined as:

$$P_i = \frac{1}{T_i}$$

2.2 Determination of Sender Packet Allocation Granularity

After classifying path transmission priorities, the sender must determine packet allocation granularity. Existing sender-side allocation algorithms primarily use two granularities [5,6,11-15]: batch allocation and single-packet allocation.

Batch allocation granularity refers to the sender allocating consecutive DSN-numbered packets to fill a selected path's send buffer. While this significantly improves allocation efficiency, it imposes strict requirements on path transmission performance. In environments with substantially different path performance, it causes severe receiver-side disorder. As shown in [Figure 3: see original paper], when using batch allocation, to prevent receiver-side disorder, the first packet on path j must arrive later than the n-th packet on path i, requiring

$RTT_j + \frac{\text{Datasize}}{B_j} < RTT_i + n \cdot \frac{\text{Datasize}}{B_i}$. This demonstrates the strict performance requirements of batch allocation.

Single-packet allocation, while less efficient than batch allocation, provides higher allocation precision. As shown in [Figure 4: see original paper], after allocating one packet to path i, the sender updates path parameters and re-evaluates path performance before selecting the next allocation path j. Single-packet allocation imposes less strict performance requirements and is suitable for CMT systems with various path performance characteristics. Therefore, this paper adopts single-packet allocation granularity.

2.3 Determination of Packet DSN Numbers for Each Path

After determining path transmission priority and allocation granularity, our scheme proceeds as illustrated in [Figure 2: see original paper]. CMT system paths are sorted by transmission priority as $\{P_1, P_2, ..., P_n\}$.

If $Q_1 < CWND_1$, the highest-priority path has available send window space. The sender allocates the smallest DSN-numbered packet k to this path, then updates path 1’ s transmission priority factor and reselects the highest-priority path.

If $Q_1 \geq CWND_1$, the highest-priority path’ s send window is full, preventing immediate transmission. However, other paths with available window space have transmission capability. The sender identifies path x in descending priority order that satisfies $Q_x < CWND_x$ and $1 \leq x \leq n$.

Path x’ s transmission priority is P_x. The sender allocates the $(k + N_x)$-th packet after the current smallest DSN k to path x. The calculation process is as follows:

Let T^i_{trans} represent the time when the packet allocated to path i arrives at the receiver, and T^i_{out} represent the time when the packet currently being sent on path i leaves the sender. The relationship is:

$$T^i_{trans} = T^i_{out} + T_i$$

To ensure in-order arrival at the receiver, the sender must guarantee:

$$T^x_{trans} < T^1_{trans}$$

From this, we derive:

$$T^x_{out} + T_x < T^1_{out} + T_1$$

Since $T_{out}^{x} = T_{out}^{1} + \sum_{i=1}^{x-1} d_i$, where d_i is the transmission time of the packet allocated to path i, we obtain:

$$\sum_{i=1}^{x-1} d_i < T_1 - T_x$$

Let N_x represent the number of data packets that can be allocated to path x without causing receiver-side disorder. Then:

$$N_x = \arg\max_{N_x} \left(\sum_{i=1}^{x-1} d_i < T_1 - T_x \right)$$

After allocation, the path' s transmission priority factor is updated, priorities are re-sorted, and the process repeats until no packets remain for allocation.

2.4 Data Allocation Flow Chart

Our sender-side allocation scheme allocates appropriately DSN-numbered packets to paths with available send window space, ensuring in-order arrival while fully utilizing system bandwidth to improve throughput and data transmission rates. The allocation flow is shown in [Figure 5: see original paper].

3.1 Simulation Environment and Parameter Settings

We validate the proposed algorithm using the NS-3 network simulator [20] under Linux. The simulation employs the MPTCP protocol supporting concurrent multipath transfer. Packet size is 1500 Bytes, retransmission strategy is RTX_{LOSSRATE}, and receive buffer size is 2.55 MB [19]. The FTP protocol transfers a sufficiently large file (maintaining continuous FTP data flow throughout the simulation) to model actual transmission scenarios.

The network topology is constructed as shown in [Figure 6: see original paper], with three heterogeneous parallel transmission paths between sender and receiver. The simulation compares our scheme with round-robin and ATLB algorithms under the following scenario: Path 1, 2, and 3 bandwidths are 1 Mbps, 1.5 Mbps, and 2 Mbps, respectively; round-trip transmission delays are 50 ms, 50 ms, and 500 ms.

3.2 Simulation Results Analysis

[Figure 7: see original paper] shows receiver-side packet disorder when using round-robin, ATLB, and our scheme for allocation. All three schemes exhibit varying degrees of disorder. With round-robin, the receive buffer typically contains 16-18 out-of-order packets. With ATLB, this reduces to 9-11 packets. Our scheme further reduces this to 4-6 out-of-order packets. Our scheme significantly decreases out-of-order packets compared to round-robin and ATLB. Round-robin is heavily affected by poor-performance paths, causing substantial disorder. While ATLB performs better by allocating packets based on predicted delay, inaccurate delay predictions still lead to out-of-order arrival. Our scheme accurately calculates appropriate DSN numbers for each path based on path performance, ensuring in-order arrival and smooth delivery to the application layer.

Varying Path 3's round-trip delay demonstrates receiver buffer occupancy under different path delay differences, as shown in [Figure 8: see original paper]. Our scheme consistently yields fewer average out-of-order packets than round-robin and ATLB. Round-robin ignores path quality, and increasing delay differences cause more disorder. ATLB allocates packets to the minimum-delay path, but as delay differences increase, low-delay paths become overloaded while high-delay paths remain idle. ATLB' s delay prediction ignores both packet transmission delay and ACK return time, causing inaccurate allocation and increased disorder. Our scheme effectively allocates packets across varying path performance conditions, reducing receiver-side disorder and improving system throughput.

Simulation analysis confirms that our sender-side allocation scheme effectively reduces receiver-side out-of-order packets, thereby improving data transmission rates and system throughput.

---

4 Conclusion

This paper first analyzes the packet transmission process in CMT systems through to upper-layer application delivery, establishing a transmission model. Based on this model, we predict packet forward transmission delay on each path and use these predictions as metrics for path transmission priority. The sender then allocates appropriately DSN-numbered packets to each path using single-packet granularity according to transmission priority and available send window space, ensuring efficient in-order arrival. NS-3 simulation validation demonstrates that our sender-side allocation scheme effectively reduces receiver-side out-of-order packets, improving system data transmission rates and throughput.

References

[1] Verma L P, Kumar M. An adaptive data chunk scheduling for concurrent multipath transfer [J]. *Computer Standards & Interfaces*, 2017, 52(33).

[2] Huang C J, Guan C T, Chen Hengming, et al. A self-adaptive joint bandwidth allocation scheme for heterogeneous wireless networks [J]. *Applied Soft Computing*, 2015, 37 (2015): 156-165.

[3] Oh B H, Lee J Y. Constraint-based proactive scheduling for MPTCP in wireless networks [J]. *Computer Networks*, 2015, 91 (2015): 548-563.

[4] Li Xiaokai, Zhang Xiangli. A concurrent multi-path transmission of data allocation algorithm based on heterogeneous networks[C]//Proc of the 3rd International Conference on Systems and Informatics. 2016.

[5] Xue Kaiping, Han Jiangping, Ni Dan, et al. DPSAF: forward prediction based dynamic packet scheduling and adjusting with feedback for multipath TCP in lossy heterogeneous networks [J]. *IEEE Trans on Vehicular Technology*, 2018, 67(2): 1521-1534.

[6] She Dongping, Wang Cong, Zhang Jianfeng, et al. Research on data scheduling mechanism in concurrent multipath transfer [J]. *Journal of Military Communications Technology*, 2015, 36(3): 18-22.

[7] Hasegawa Y, Yamaguchi, Hama T, et al. Improved data distribution for multipath communication[C]//Proc IEEE Global Telecommunications Conference. 2005: 271-275.

[8] Chen Rui, Wang Wennai, Zhu Jingling, et al. A round-trip-time based concurrent transmission scheduling for MPTCP[C]//Proc of the 6th International Conference on Wireless Communications and Signal Processing. 2014: 179-183.

[9] Du Wenfeng, Wu Zhen. Data distribution algorithm with out-of-order feedback for CMT over diversity network [J]. *Computer Science*, 2015, 42(3): 60-64.

[10] Huang Hui. Weight-based Data Scheduling Algorithm Designing for MPTCP [J]. *Computer Engineering & Software*, 2016, 37(2): 77-80.

[11] Nakasan C, Ichikawa K, Uthayopas P. Performance evaluation of MPTCP over open flow network [J]. *Information Processing Society of Japan SIG Notes*, 2014, 2014 (30): 1-6.

[12] Zhou Dizhi, Song Wei, Wang Ping, et al. Multipath TCP for user cooperation in LTE networks [J]. *IEEE Network Magazine*, 2015, 29 (1): 20-26.

[13] Wang Zhenchao, Hou Huanhuan, Li Haixiao. Path optimization scheme based on quality of service in concurrent multipath transmission [J]. *Journal of Chinese Computer Systems*, 2018, 39(3): 503-507.

[14] Xu Changqiao, Liu Tianjiao, Guan Jianfeng, et al. CMT-QA: quality-aware adaptive concurrent multipath data transfer in heterogeneous wireless networks

[J]. *IEEE Trans on Mobile Computing*, 2013, 12(11): 2193-2205.

[15] Prabhavat S, Ansari N, Kato N, et al. Effective delay-controlled load distribution over multipath networks [J]. *IEEE Trans on Parallel and Distributed Systems*, 2011, 22(10): 1730-1741.

[16] Oh B H, Lee J Y. Constraint-based proactive scheduling for MPTCP in wireless networks [J]. *Computer Network*, 2015, 91(14): 549-563.

[17] Gong Weibing, Yang Xiaolong, Zhang Min, et al. An adaptive traffic distribution scheme for CMT based on Lotka-Volterra model in multihomed networks [J]. *China Communications*, 2017, 14(2): 79-89.

[18] Xu Changqiao, Li Zhuofeng, Zhong Lujie, et al. CMT-NC: improving the concurrent multipath transfer using network coding in wireless networks [J]. *IEEE Trans on Vehicular Technology*, 2016, 65(3): 1730-1741.

[19] Chen Sengyang, Yuan Zhenhui, Muntean G M. An energy-aware multipath TCP-based content delivery scheme in heterogeneous wireless network[C]//Proc of IEEE Wireless Communication and Networking Conference. 2013: 1291-1296.

[20] MPTCP implementation for NS-3[EB/OL]. http://code.google.com/p/mptcp-ns3.

Note: Figure translations are in progress. See original paper for figures.

Source: ChinaXiv —Machine translation. Verify with original.