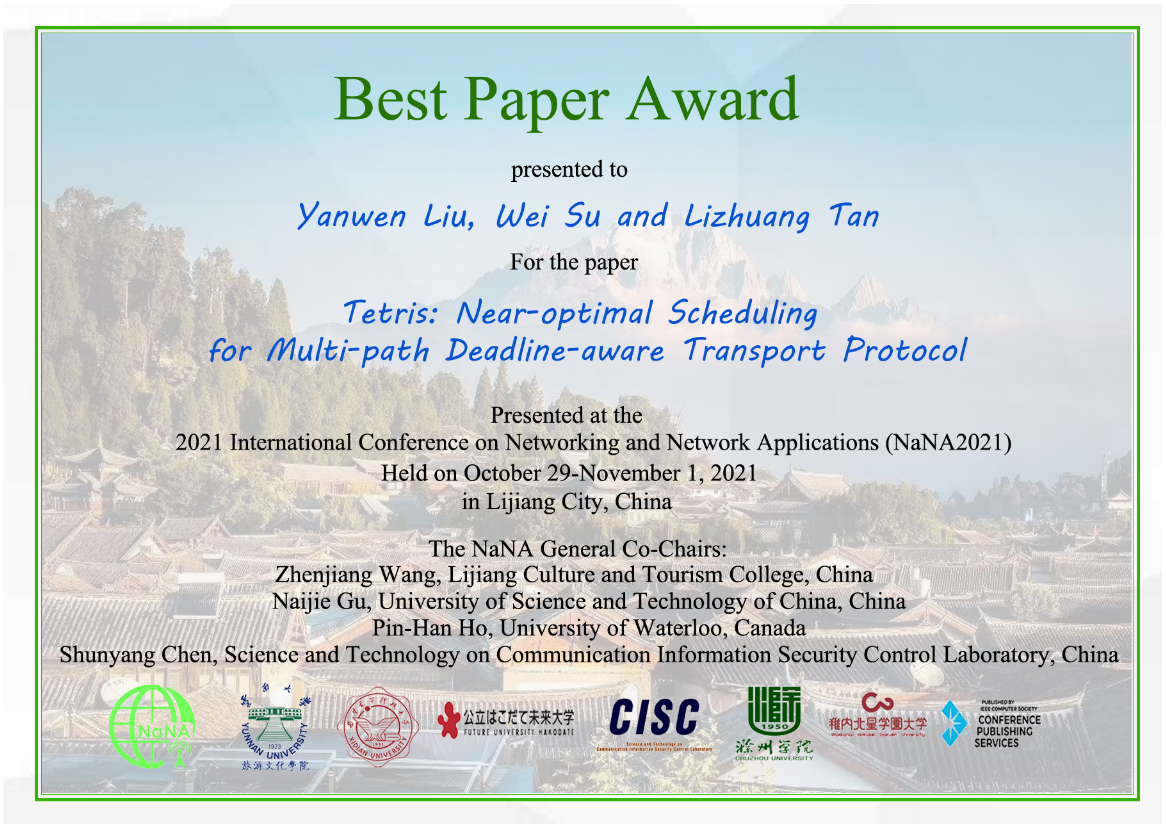


Best Paper Award

presented to

Yanwen Liu, Wei Su and Lizhuang Tan

For the paper

Tetris: Near-optimal Scheduling for Multi-path Deadline-aware Transport Protocol

Presented at the
2021 International Conference on Networking and Network Applications (NaNA2021)
Held on October 29-November 1, 2021
in Lijiang City, China

The NaNA General Co-Chairs:
Zhenjiang Wang, Lijiang Culture and Tourism College, China
Naijie Gu, University of Science and Technology of China, China
Pin-Han Ho, University of Waterloo, Canada
Shunyang Chen, Science and Technology on Communication Information Security Control Laboratory, China

2021 International Conference on Networking and Network Applications (NaNA) | 978-1-6654-4158-2/21/$31.00 ©2021 IEEE | DOI: 10.1109/NaNA53684.2021.00014

Tetris: Near-optimal Scheduling for Multi-path Deadline-aware Transport Protocol

Yanwen Liu, Wei Su*, Lizhuang Tan
School of Electronics and Information Engineering, Beijing Jiaotong University,
Beijing, PR China
19120081, wsu, lzhtan@bjtu.edu.cn

***Abstract*—Deadline-aware Transport Protocol (DTP) is a new QUIC-based transmission protocol that provides deliver-before-deadline service. Single-path DTP is not conducive to flow fairness and does not make full use of bandwidth. Compared with single-path DTP, the decision space and the solving difficulty of multi-path DTP (MPDTP) scheduling are larger. In this paper, we propose a near-optimal scheduling algorithm Tetris for MPDTP. Tetris is a block scheduler based on stream characteristics at the transmission layer. We have verified its feasibility on the simulator under the deployed heterogeneous path of different network environment. The results show that our scheduling algorithm allows data blocks to be delivered before the delivery time as much as possible. The transmission completion time has been increased by 19.53% on average, and the transmission delay of all blocks have been reduced by 11.27%.**

***Index Terms*—QUIC, Multi-path Scheduling, DTP, Transport Protocol**

I. Introduction

Nowadays web services need to be able to sense business and have strict deadline guarantees [1]. The delivery of many applications is delay-sensitive, such as video calls, online classrooms, multiplayer online games, and VR etc. We make the entire communication process as the on-time delivery of several blocks with different priority from the sender to the receiver, so DTP [2] came into being.

The DTP protocol first takes the concept of deadline in the protocol, and ensures the on-time delivery of blocks by adjusting the sending rate and other strategies. The deployment of DTP based on the QUIC [3] [4] protocol provides a secure, reliable, and low-latency communication link for HTTP, and can execute online scheduling strategies based on the characteristics of block and path. As shown in Fig.1 below, we use webpagetest to capture the time to load different types of data when accessing youtube.com. In terms of delivery time, different types of blocks such as html, css, js, image and flash have different delivery time requirements, so the introduction of deadlines is very necessary.

However, DTP is a single path transmission protocol, which does not make full use of the improvement of transmission efficiency brought by multi-path transmission. In multi-path, different paths have path attributes such as Round-Trip Time(RTT) and packet loss whch make DTP scheduling more complicated. Multi-path QUIC(MPQUIC) [5] gives an example of multi-path in the QUIC scenario, but the combination of the DTP transmission protocol and the MPQUIC [6]

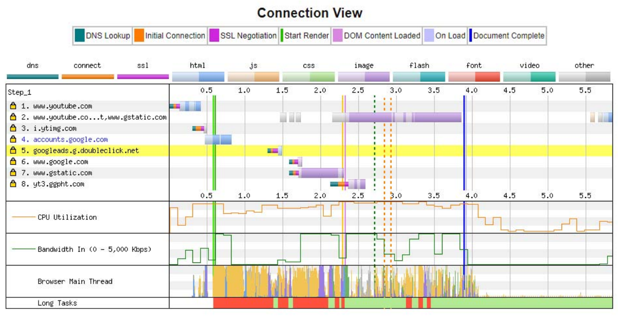


Fig. 1. Loading time of different modules to access youtube captured by webpagetest

transmission protocol has not been paid attention and is more difficult.

Therefore, in this article, we propose a new idea of MPDTP and propose a near-optimal scheduling algorithm based on delivery time, which is Tetris. For different data blocks, we divide them into different tasks according to tasks. In Tetris, we define the comprehensive priority for scheduling, including the priority, deadline and left size of the block. In the end, we conducted experimental verification through the designed scheduler and found that the proportion of data blocks transmitted before the deadline approaches 76.1%, and the block transmission time is decreased by 19.53%.

The organization of the remaining chapters of the paper has sex sections as follows: Section II introduces the related works, Section III describes the existing problems, Section IV describes the design principle of Tetris Scheduler in detail, Section V discusses the implementation, and finally, Section VI summarizes our work in this paper.

II. Related Works

QUIC is a new communication protocol proposed by Google in 2013 to improve the network communication performance of HTTP. It is a transmission protocol located at the application layer, which effectively replaces the traditional TLS+HTTP/2+TCP on the client.

Since QUIC establishes the connection quickly and does not head line blocking, Carlucci et al. [7] found that QUIC has better bandwidth utilization and lower link delay than other communication protocols. Cook et al. [4] found that

QUIC fluctuates more in the network state. In general, we have better performance advantages when accessing YouTube through QUIC, which allows more efficient video transmission and web browsing.

QUIC [3] supports multiple streams in a connection, ensuring that lost UDP packets only affect part of the streams that carry data in the packets. The endpoints in the QUIC protocol need to decide how to allocate the available bandwidth [8]. However, among multiple streams, it only depends on the priority of the HTTP/2 stream for scheduling. Since many data have deadlines during the transmission process, Cui et al. [2] proposed DTP based on task delivery deadlines. DTP extends the QUIC protocol to support block-based delivery, and maps the blocks to the QUIC streams one by one. DTP is different from the QUIC protocol based on priority scheduling, but weighs the priority of the stream and the delivery deadline, calculates the actual priority of the stream, and sends it in the order of priority. Although the DTP protocol considers the task delivery time, it does not consider the use of multiple path parallel transmissions.

Multi-path parallel transmission based on QUIC realizes the effect of switching and aggregating multi-path bandwidth [9] The MPQUIC [5] prototype adopts the lowest transmission rate first, and selects the link with the lowest round-trip RTT for transmission. Most of the existing multi- path scheduling methods are considered comprehensively, and are mainly divided into two categories: traffic scheduling schemes based on dynamic heuristics and stream scheduling methods based on reinforcement learning. Among them, the dynamic heuristic stream scheduling scheme is widely used in the scheduling design of MPQUIC.

The SRPT-ECF [10] algorithm is a stream-aware multipath scheduling algorithm. Using this method to minimize the completion time of the message, and the path allocation can be changed at any time. However, the algorithm has some problems: priority is given to reducing the completion time of smaller streams without significant impact on larger streams, making most of the messages scheduled to a main path, not fully utilizing all links which resulting in poor link bandwidth utilization. Wang et al. [11] proposed an uplink and downlink scheduler with a scheduling tree structure based on MPQUIC, which ensures that the transmission time used on multiple links is consistent and effectively improves the page loading time, but it assumes that the server knows the dependency tree of the web page in advance. Shi et al. [12] comprehensively considered priority and stream size, and dispatched a stream to a single path without making full use of multi-path aggregation to allocate bandwidth reasonably.

III. Problem Description

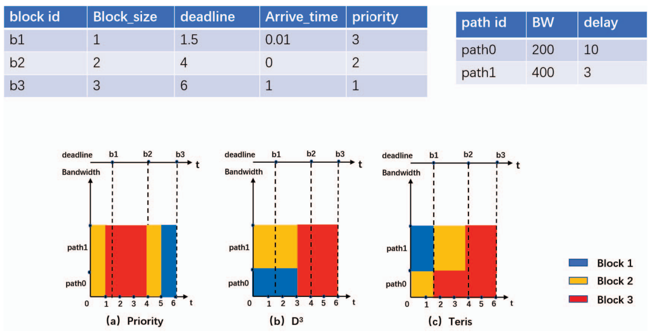
A. overview

In the context of network communication, we establish a stream by establishing a handshake connection. On the stream, we divide it into several logically different types of blocks according to different data on the application layer. There are three types of blocks: video, audio, and control. Each block includes several data packets. Different application tasks produce different types of blocks, and the transmission can be interleaved. But on the same path, the data blocks in the task need to be strictly transmitted to avoid data disorder.

To present a near-optimal scheduling strategy considering the size, arrival time, deadline and priority of the data block and effectively balance the quality of service objectives, which includes maximizing the effective utilization of the link and maximizing the number of data blocks effectively transmitted, we propose a mathematical programming model based on block scheduling algorithm.

B. Applicable scene

Traditional scheduling algorithm are mostly based on one of the parameter indicators. In Fig.2 below, the smaller the value of priority means the block needs to be sent earlier. As shown in Fig.2(a), it is delivered according to the strict stream priority strategy. Whenever a new block is generated, it rearranges the data blocks to be sent in the sender's buffer, and selects the block with the highest priority in each round and uses it to transmit as many link resources as possible. This algorithm is completely sorted by priority and ignoring the deadline of low-priority blocks. As a result, the data blocks b1 and b3 that can be sent instinctively are blocked because of their relatively low priority and cannot be completed before the delivery time.



block id	Block_size	deadline	Arrive_time	priority
b1	1	1.5	0.01	3
b2	2	4	0	2
b3	3	6	1	1

path id	BW	delay
path0	200	10
path1	400	3

Fig. 2. Transmission comparison of different algorithms

Considering that the block delivery time seriously affects the quality of network data transmission, D_3 scheduling is proposed in the central network. The algorithm is to design a network data algorithm to calculate the minimum length of different data blocks $r = \frac{s}{d}$, s is the size of the block, and d is the completion time of the block. D_3 gives priority to solving the smallest bandwidth r of the data block before the last time, as shown in Fig.2(b) because we receive the data block b2, here we do not proceed with D_3 and adopt the greedy possible first-served time. Having said that, the preemption starts immediately after the block is allocated which results in the data block b_1 cannot be completed before the time.

The above two typical algorithms only consider a key indicator of the data block in the link, and cannot maximize the use of link bandwidth to allow more blocks to complete scheduled transmission. Here, we propose a near-optimal scheduling Tetris based on multi-path deadline-aware transmission protocol, which considers block priority, arrival time, minimum completion time and deadline. Without affecting the overall transmission effect, the block trigger update strategy is adopted as shown in Fig.2(c). By sacrificing the transmission time of some high-priority blocks, the blocks with lower priority but earlier deadlines are sent first to achieve the goal of completing the transmission block and maximizing the remaining completion time.

IV. Tetris Design

In complex application scenarios, due to the content delivery of many applications is delay-sensitive, the corresponding stream needs to be effectively transmitted before the deadline to ensure the timeliness and effectiveness of application data transmission, so we propose a fine-grained data disassembly. Firstly, the data stream sent by the sender in batches is disassembled into control, audio, and video according to the stream type. Different types of streams are defined as logical sub-streams. Several blocks are on the sub-stream, and the block may contain one or more data packets. For example, the control block contains 1 data packet, the audio contains about 3 data packets, and the video contains about 20 data packets. Our scheduling strategy is to schedule data blocks on three sub-streams.

The goals of our algorithm are:

(1)Ensure that more blocks are delivered before the deadline, and improve the transmission completion rate.

(2)When the status of different paths changes dynamically, the scheduler can adapt to the network conditions and reasonably improve the QoE of the network.

A. overview

In the process of link communication, it is invalid for information transmission to transmit only part of the bytes in the data block and the transmission of all contents cannot be completed normally. Therefore, we designed a scheduling strategy, where the block is not transmitted according to the priority preemption mode, nor is it sent according to the deadline, nor is it according to the first-come, first-served polling strategy. Our scheduler transmits streams in units of finer-grained blocks and performs block path allocation with reference to the overall priority of the block and the actual status of the link. Our algorithm can mainly adopt the following scenarios:

(1) For blocks with low priority but some bytes are already in the transmission process

(2) Some blocks with close deadlines and low priority and blocks with low deadlines and high priority exist at the same time.

Based on the above scenario, our scheduler is consists of two parts: a multi-path block scheduler and a single-path block manager. We will introduce them in detail later. The block scheduler is responsible for distributing different types of streams to different links according to the fine-grained block. Each link has a block manager to maintain the block scheduling, and perform unified scheduling according to the priority of the block on the same link are transfer in order. The overall scheduler design has the following 3 principles:

(1) Each time a new message is received, the scheduler updates the block priority first and then allocates it.

(2) The blocks with higher comprehensive priority are sent first, and the blocks with lower priority waits for the higher priority to be sent before sending. Streams with the same priority are sent according to the principle of fairness.

(3) The blocks scheduled to the same path in the block manager are sent according to the actual low priority, and the blocks with the same priority are sent first.

We establish the communication between the client and the server first.After completing the connection communication of the QUIC protocol, there is a unique connection identifier CID for identification. At the sending end of the client, the block allocation is completed through our block scheduler and distributed to each sub-path. The sub-path transmits data blocks to the receiving server in sequence through the block manager. When all the blocks have been transmitted, close the scheduler. The structure diagram of the scheduler is as follows:

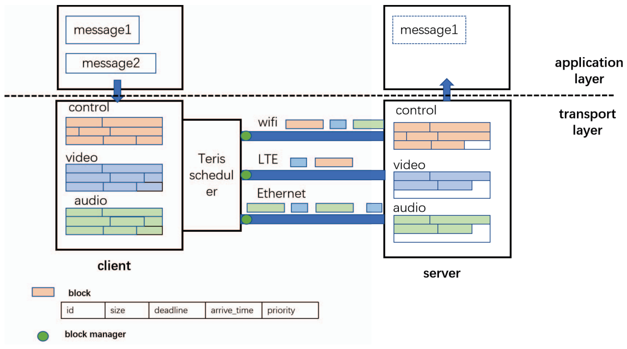


Fig. 3. The structure of Tetris design

B. Scheduling mechanism

Since the network status and the data stream to be sent are dynamically changing, our algorithm is an online scheduling algorithm. In each round, when a new data content is generated at the sending end or a block has been send completed, the scheduler will be triggered to perform two steps: 1. Update the link and data block status 2. Perform a mathematical programming distribution scheduling model.

The first step is to calculate P_i^{send}, the comprehensive priority of all data blocks to be sent, and the second step is to allocate the blocks to the path. When there is no available bandwidth on the link or the congestion buffer is full, we switch to the next path and continue to allocate until all the blocks to be sent have been allocated on the link. After the general scheduler completes the multi-path block scheduling allocation, the single-path block scheduler sends the blocks in order of priority from high to low on the same path, and blocks with the same priority are sent according to the urgency of the block deadline.

C. structure

The scheduler updates all existing data blocks when a new data block arrives. We assume that the current round of triggering the scheduler is c ($c = 1, 2, 3...n$), the subsequent design of the scheduling strategy is the parameters of this round, and a prompt will be given if it is not in this round.. Tetris mainly consists of the following two steps: block-based scheduling priority recalculation and block allocation on the path. We assume that:

(1) The date, size, priority, delivery deadline and other information of the data block are known when the stream is created at the sending end.

(2) Using the smooth RTT and the congestion window provided by QUIC to estimate the current round-trip delay and bandwidth of the path. Assuming that the path state remains unchanged in the current scheduling round, we define all the parameters in TABLE 1 and the variables in TABLE 2. In order to obtain a solution that is as optimal as possible, a linear programming model is established to reduce the maximum transmission time as much as possible. The model ensures that the amount of data allocated each time is equal to the amount of data that needs to be sent. In order to allocate link bandwidth resources based on priority, a binary variable is introduced to determine whether a link sends the corresponding block, so as to match the corresponding bandwidth resource according to the priority share of the block that needs to be sent on each link.

TABLE I
PARAMETER DEFINITION

Parameters	**Meaning**
d_i	The data block id
n	The numbers of path$j$
j	The path j
$D_{dl,i}$	The deadline of the data block i
P_i	The primary priority of data block i
P_i^{send}	The comprehensive priority of data block i
d_i^{size}	The size of data block i
$d_{i,c-1}^{send_size}$	The sent size of data block i at round $c-1$
$d_i^{left_size}$	The left size of data block i
rtt_j	The Round-Trip Time of path j
$\overline{BW_j}$	The average bandwidth of path j
BW_j	The bandwidth of path j
S_{rub}	The threshold for data blocks to enter the collection
P_j^{sum}	The total priority of data blocks on path j
$t_{j,i}$	Completion time of data block i on path j
T_j	The total completion time of the data block on path j
T_{cur}	The current time
$BW_{j,i}$	Bandwidth of data block i allocated to path j
$S_{j,\lim}$	Threshold of path j

TABLE II
VARIABLE DEFINITION

Variable	Meaning	Type
$d_{j,i}^{size}$	Total amount of data block i allocated to path j	Integer
$\lambda_{j,i}$	Whether path j sends data block i	Boolen

1) step1 Block-based scheduling priority reordering: As mentioned before,we proposed a block-based scheduler . Each stream is divided into blocks with a finer granularity of time. The deadline-priority judgment is performed on the block, and the actual priority of the block is given. Through statistics of a large amount of data, we found that in most application scenarios, the priority of control signaling block, audio block and video block for network communication, network quality and user experience in different business scenarios decreases in sequence. Therefore, we set the corresponding priority rating weights as $P_i = 1, 2, 3$ respectively. Both P_i^{send} and P_i are sent in order of priority, the lower the priority is, the earlier the block is sent.

In the actual transmission, we comprehensively design a priority evaluation method, including the length of the completed data block in the previous round of the link and the remaining length of the data block. The specific equation is as follows:

$$BW_{j,i} = \frac{P_i^{send}}{\sum_i \lambda_{j,i}^{size} \times P_i^{send}} \times BW_j \tag{1}$$

$$t_{j,i} = \frac{d_{j,i}^{size}}{BW_{j,i}} + rtt_j \times \lambda_{j,i} \tag{2}$$

Considering the existing data already in the transmission process and the part waiting to be transmitted as shown in Equation (3) , in the same transmitted data block i, the smaller the bandwidth of the transmission arrangement of the part that has been transmitted in the previous round, the slower the rate, and the closer the transmitted part is to the deadline, the more possibility the block is transmitted. Among them, $\overline{BW_j}$ as shown in Equation (4) is the average bandwidth of this round of multi-path, which can be imagined as a constant and only affected by changes in the network status for different rounds. Considering that the P_i^{send} of some blocks is relatively high during the transmission process, while some of the other blocks may already be in the transmission process, for this type of blocks increasing the restriction conditions to adjust P_i^{send} in Equation (5)

$$P_i^{send} = P_i * \left\{ \frac{d_{i,c-1}^{send_size}}{\overline{BW}} + \left[D_{dl} - \left(\frac{d_i^{left_size}}{\overline{BW}} + \frac{RTT}{2} \right) \right] \right\} \tag{3}$$

Part of the transmission is meaningless to the network, but in most application scenarios, the data block arriving later will not have a serious impact on the network quality. The biggest impact is that subsequent content loading needs to be delivered successfully, therefore we set the threshold S_{rub} to judge whether the data block in the network that exceeds the deadline should be transmitted. The judgment condition is as shown in Equation (6)

$$\overline{BW} = \frac{\sum_j BW_j}{n} \tag{4}$$

The threshold value is set to be $S_{rub} = 2ms$. During the execution, real-time monitoring is performed on all data blocks to be sent. If the mutation is 0, the content of the remaining data block is directly discarded and the scheduling collector is triggered, and a notification frame RESET STREAM is sent to inform the receiving buffer that the block has been recycled. The subsequent sending end can be scheduled according to the actual priority.

The block-based scheduling algorithm comprehensively considers the deadline and priority of the block to prevent the data block whose sub-stream is approaching the delivery deadline from missing the delivery time because of the low block priority.

$$\begin{cases} P_{isend} = \min P_i^{send} & d_i^{left_s ize} \leq \frac{1}{3} \times d_i^{size} \\ P_{isend} & ELSE \end{cases} \tag{5}$$

$$\begin{cases} d_i^{left_s ize} = 0 & \frac{d_i^{left_s ize}}{\overline{BW}} + \frac{RTT}{2} \geq S_{rub} \\ d_i^{left_s ize} & ELSE \end{cases} \tag{6}$$

2) step2 Block allocation: The distribution of blocks on the path is divided into two categories: one is the blocks with the same priority are sent according to the principle of fairness and the urgency of the block, the other is blocks with different priorities are sent according to the principle of comprehensive priority. The block distribution on the path is constructed as a linear optimization mathematical model, and the size of the data blocks distributed on different paths is obtained by modeling and solving. The constraint condition of data block allocation is shown in Equation (7):

$$\sum_j d_{j,i}^{size} = d_i^{size} \tag{7}$$

For a data block, the sum of the data block sent by all paths is equal to its size.

If all the blocks to be sent have the same priority, we assume that each link j has its own threshold $S_{j,\lim}$ is transmitted in a block-indifferent way. When the transmitted block size is less than $S_{j,\lim}$, we only select the link with the smallest delay for transmission, because at this time, even if other links are used to transmit sub-blocks together, the block will not be transmitted faster. When the block transmission size is greater than the threshold , we use multiple links to parallel for the block. We use Equation (8) to define decision variables to determine whether a path sends a block.

$$0 \leq d_{j,i}^{size} + (1 - \lambda_{j,i}^{size}) * M \leq M \tag{8}$$

If path j send data block i, then $d_{j,i}^{size}$>0 and $\lambda_{j,i} = 1$, otherwise both are 0.

Each path has its own carrying capacity, combined with the sub-stream congestion control algorithm and traffic distribution rate, the split ratio on each path is allocated according to its network state quality. We assume that the link state remains unchanged in each round, and maintain a smooth estimate of the bandwidth and round-trip delay of each link. We define $t_{j,i}$ as the completion time of each block, and we construct a mathematical programming model to minimize the maximum transmission completion time on the link, as shown in Equation (9):

$$Obj = \min \max t_{j,i} \quad \forall i \in I \quad j \in J \tag{9}$$

I and J are the data block set and path set of the current round respectively.

We integrate the scheduling algorithm of the scheduler Tetris in two steps, and give the pseudo code of Tetris:

Algorithm 1 Tetris Scheduler

```
Initialize:
$d_i^{left_s ize} = d_i^{size}$,  $P_i^{send}$
$MIN = \min P_i^{send}$,  $d_{i,c-1}^{send_s ize} = 0$
while True do
  while There is a block in sender do  Give id to data block as $d_i$;
  Calculate $\overline{BW}$,  $P_i^{send}$,  $d_i^{left_s ize}$
    if $d_i^{left_s ize} = 0$ then
    send RESET STREAM to Receive Buffer
    $P_i^{send} = \infty$
    $i = i + 1$
      if $d_{i,c-1}^{send_s ize}$>0 and $d_i^{left_s ize} \leq \frac{1}{3} d_i^{send}$ then
      $P_i^{send} = MIN - 1$
      end if
  Order $d_i \leftarrow$ sort $d_i^{left_s ize}$ by ascending order of $P_i^{send}$
    Calculate $D$ according to the mathematical model
    for $d_{i,j}$ in $D$ do
      if $d_{j,i}$>0 then
        $d_{i,c-1}^{send_s ize} \leftarrow d_{j,i}$
        $d_i^{left_s ize} \leftarrow d_i^{left_s ize} - d_{j,i}$
        $P_j^{sum} \leftarrow P_j^{sum} + P_i^{send}$
        $j \leftarrow j + 1$
      else
        $j \leftarrow j + 1$
        Path $j$ keep sending data and Break
      end if
    end for
    if There is no block to be send then  Break;
end procedure
```

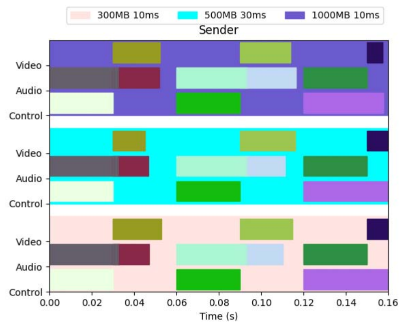
V. Evaluation Results

A. Comparison of Tetris scheduler under different link configurations

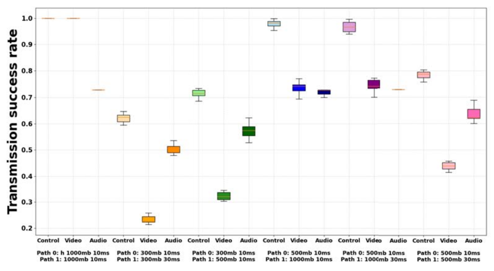
TABLE III
Link set

group	Path0(Bandwidth,RTT)	Path1(Bandwidth,RTT)
a	1000Mbps, 10ms	1000Mbps, 10ms
b	500Mbps, 10ms	500Mbps, 30ms
c	300Mbps, 10ms	300Mbps, 30ms
d	300Mbps, 10ms	500Mbps, 10ms
e	500Mbps, 10ms	1000Mbps, 10ms
f	500Mbps, 10ms	1000Mbps, 30ms

To verify the effectiveness of Tetris, a comparative experiment that combines 6 groups with different networks states

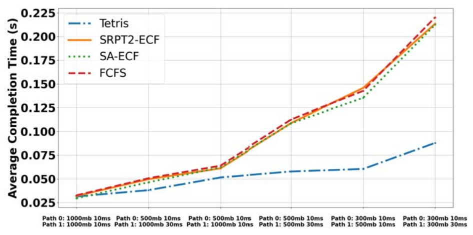


(a) Sender

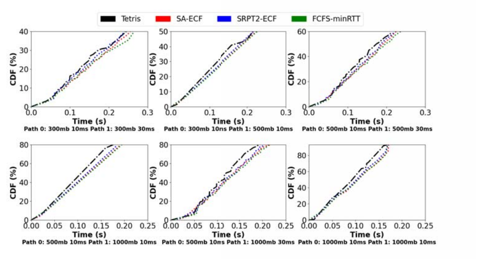


(b) Tetris Transmission success rate in different linkway

Fig. 4. comparison of the scheduler Tetris



(a) Average transmission completion time



(b) CDF

Fig. 5. Comparison between route schedulers

is completed. The delay and bandwidth are set as follows respectively in Table3.

To display the specific scheduling result of Tetris, Fig.4(a) shows the data block sending situation. The blocks arrive in batches, and different colors represent different data blocks. Tetris splits the blocks into video, audio and control according to the block type,and they are sent in multi channels concurrently. The algorithm on each path takes into account the remaining block size in history, and can effectively deliver as many data blocks as possible before the deadline. The sending and receiving order of data blocks in Tetris are basically the same, which avoids the problem of data reorganization caused by data blocks splitting.

We calculate the block completion time under the heterogeneous network link, use the cut-off time as the card line to calculate the average completion success rate of the block, and test each link setting 10 times to obtain the Fig.4(b), and find that the Tetris algorithm has following features:
(1) The block transmission success rate is different under different link settings. The smaller the link difference, the higher the block transmission success rate.
(2) The success rate of block transmission under the same link is related to the block priority. The higher the priority, the higher the transmission success rate of the block, such as the control type block. We found that the average value of block transmission under several groups of different settings is 76.5 %, and the control type block with the highest priority has the highest overall transmission success rate.

B. Comparison between route schedulers

We set up 6 different network links as shown in table 3, and each scheduling algorithm was repeatedly tested 10 times under the above environment to obtain the average result of the link configuration. We evaluated the performance of the proposed Tetris scheduler, including the success rate of block transmission before the deadline, the cumulative function of successful packet transmission, with the default QUIC-based multi-path scheduler FCFS, SRPT2-ECF and SA-ECF for comparison.

We found that under the same settings, FCFS sends small data blocks based on the shortest transmission delay preferentially, SA-ECF combined comprehensive weighted round-robin scheduling and the earliest completion priority strategy, weighted routing is sent after the estimated message completion time. To satisfy the complex balance as much as possible, SRPT2-ECF first considers the priority provided by the application, but it can reduce the completion time of smaller streams while ignoring large streams. As shown in Fig.5(b), the average completion time of the flexibly allocated Tetris in different link scenarios ranges from 0.031356ms to 0.008ms, but is relatively less affected by the link, and the transmission delay can be increased by 3.75% to 82.7%.

We calculated the cumulative distribution function of each algorithm under different links as shown in Fig.5(b). Tetris is relatively stable and its completion time can be less than other in link a when cumulative probability of all reach algorithms 90%. The algorithm is improved by about 19.53%. In the network, path delay in link c is higher, the performance is worse and the cumulative probability reaches 40%. Tetris is about 2.1% higher than SA-ECF and about 5.1% higher than FCFS, which is basically the same as SRPT2-ECF. The statistical conclusion we draw from the data analysis is that in the same state, Tetris can reasonably schedule the resources of multiple path and improve the overall transmission efficiency, which are more obvious than SRPT2-ECF, SA-ECF, and FCFS.

We found that the comprehensive comparison of the Tetris algorithm has significant advantages in data transmission efficiency and effectiveness, and can significantly improve the user experience.

VI. Conclusion

In the complex and changeable heterogeneous environment, the use of multi-path transmission protocol for data packet scheduling can effectively improve the communication quality, and the proposal and deployment of the QUIC protocol paved the way for improving user experience. Combining with the heterogeneity of paths based on QUIC, this paper proposes an algorithm for near-optimal scheduling of multi-path deadline-aware transmission protocols, which is Tetris. We propose Tetris in order to take the characteristics of the link into account more effectively and make fully use of link resources to ensure that the block can be transmitted before the deadline. Tetris is designed considering the deadline, block priority, create time and block size to allocate the bandwidth. In this way, the block completion time on each path is as consistent and minimized as possible. Experimental results show that compared with other multi-path schedulers, Tetris has obvious advantages in link resource allocation and utilization which can significantly reduce page load time and improve the success rate of data block transmission. Tetris is protocol-independent, which can still be reused in other communication protocols in the future and has strong practical significance.

Acknowledgment

This work was supported in part by the National Key R&D Program of China under Grant No.2018YFB1800305.

References

[1] HTTPArchive. [Online]. Available: https://httparchive.org/. https://httparchive.org/.

[2] Hang Shi, Yong Cui, Feng Qian, and Yuming Hu. Dtp: Deadline-aware transport protocol. In *Proceedings of the 3rd Asia-Pacific Workshop on Networking 2019*, pages 1–7, 2019.

[3] Adam Langley, Alistair Riddoch, Alyssa Wilk, Antonio Vicente, Charles Krasic, Dan Zhang, Fan Yang, Fedor Kouranov, Ian Swett, Janardhan Iyengar, et al. The quic transport protocol: Design and internet-scale deployment. In *Proceedings of the conference of the ACM special interest group on data communication*, pages 183–196, 2017.

[4] Sarah Cook, Bertrand Mathieu, Patrick Truong, and Isabelle Hamchaoui. Quic: Better for what and for whom? In *2017 IEEE International Conference on Communications (ICC)*, pages 1–6. IEEE, 2017.

[5] Quentin De Coninck and Olivier Bonaventure. Multipath quic: Design and evaluation. In *Proceedings of the 13th international conference on emerging networking experiments and technologies*, pages 160–166, 2017.

[6] Tobias Viernickel, Alexander Froemmgen, Amr Rizk, Boris Koldehofe, and Ralf Steinmetz. Multipath quic: A deployable multipath transport protocol. In *2018 IEEE International Conference on Communications (ICC)*, pages 1–7. IEEE, 2018.

[7] Gaetano Carlucci, Luca De Cicco, and Saverio Mascolo. Http over udp: an experimental investigation of quic. In *Proceedings of the 30th Annual ACM Symposium on Applied Computing*, pages 609–614, 2015.

[8] Federico Chiariotti, Anay Ajit Deshpande, Marco Giordani, Kostantinos Antonakoglou, Andrea Zanella, and Toktam Mahmoodi. Quic-est: A transmission scheme to maximize voi of multi-stream correlated data flows. *arXiv preprint arXiv:2010.03831*, 2020.

[9] Christoph Paasch, Simone Ferlin, Ozgu Alay, and Olivier Bonaventure. Experimental evaluation of multipath tcp schedulers. In *Proceedings of the 2014 ACM SIGCOMM workshop on Capacity sharing workshop*, pages 27–32, 2014.

[10] Baptiste Jonglez, Martin Heusse, and Bruno Gaujal. Srpt-ecf: challenging round-robin for stream-aware multipath scheduling. In *2020 IFIP Networking Conference (Networking)*, pages 719–724. IEEE, 2020.

[11] Jing Wang, Yunfeng Gao, and Chenren Xu. A multipath quic scheduler for mobile http/2. In *Proceedings of the 3rd Asia-Pacific Workshop on Networking 2019*, APNet '19, page 4349, New York, NY, USA, 2019. Association for Computing Machinery.

[12] Xiang Shi, Lin Wang, Fa Zhang, and Zhiyong Liu. Fstream: Flexible stream scheduling and prioritizing in multipath-quic. In *2019 IEEE 25th International Conference on Parallel and Distributed Systems (ICPADS)*, pages 921–924. IEEE, 2019.