EDITOR: Yong Cui, cuiyong@ tsinghua.edu.cn

DEPARTMENT: STANDARDS

Multipath Deadline-Aware Transport Proxy for Space Network

Hang Shi, Lei Zhang, Xutong Zuo, Qian Wu, Hewu Li, and Yong Cui, *Tsinghua University, China*

Space travel is becoming more accessible due to the low-cost spaceflight. One way to maintain the communication between the space station and the ground station is to use several relay satellites. However, those relay links suffer from high loss rate, limited bandwidth, and long round trip time (RTT). Meanwhile applications running over these links, such as real-time communication, usually have deadline requirements for their data transfer. Recently, QUIC version 1 is released as RFC9000 (by Iyengar and Thomson in 2021), which provides great opportunity to improve transport services. Based on QUIC, we develop multipath deadline-aware transport proxy (MPDTP) to provide the deadline-aware transmission service for those applications. To avoid the loss caused by congestion control of each application, the proxy aggregates the data transmission of applications and transmits them along multiple paths. MPDTP uses deadline-aware scheduling and adaptive redundancy to deal with the link fluctuation and packet loss. Based on our implementation experience, we give some guidance on how to standardize MPDTP.

SpaceX and Blue Origin significantly reduce the cost of spaceflight, making the space travel more accessible. To maintain the 24×7 transmission between the space station and the ground station, multiple geosynchronous satellites can be used to relay data packets. Those relay links have the following issues:

1) high loss rate due to the long distance wireless nature;
2) limited bandwidth;
3) long RTT.

Meanwhile, many applications running on the space station have a deadline requirement for their data transmission. For example, the end-to-end latency of the video conferencing system needs to be below 1 s to enable smooth interaction. The remote control command needs to be executed before a certain time. How to deliver the data before its deadline over the relay links is the key to the application experience.

Existing transport services cannot provide a deadline-aware service to the application and the congestion control for the relay link is not well studied either. Recently, QUIC version 1 is released as RFC9000.[1] As a user space transport protocol, QUIC provides great opportunity to improve transport services.[2] However, if each application builds its own transmission scheme over QUIC and competes with each other for the limited bandwidth resource, there will be a lot of bandwidth wastage due to the congestion loss and redundant engineering efforts for each application to implement deadline-aware delivery. To provide better transmission service and reduce the application development overhead, we propose multipath deadline-aware transport proxy (MPDTP) for the space network. Applications submit their data and transmission requirements to the proxy. MPDTP manages the transmission process over multiple links to deliver the data before its deadline. MPDTP uses a deadline-aware block scheduler and a multipath scheduler to allocate the bandwidth of two paths to each application based on their deadlines and priorities. MPDTP also uses redundancy coding to recover from packet loss of

1089-7801 © 2021 IEEE
Digital Object Identifier 10.1109/MIC.2021.3112804
Date of current version 6 December 2021.

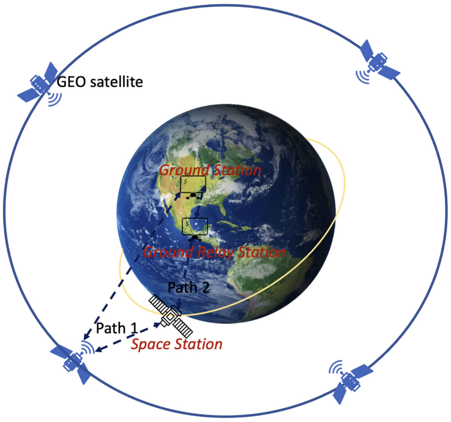

FIGURE 1. Communicate with space station using multiple links.

each path. We implement the MPDTP as an extension to QUIC. Based on our implementation experience, we give some guidance on how to standardize MPDTP making minimal modification on existing standards.

BACKGROUND AND MOTIVATION

Recently, low-Earth-orbit space travel is becoming increasingly accessible thanks to the low-cost spaceflight developed by SpaceX and Blue Origin. To maintain the communication between the space station and the ground station, geosynchronous satellites are used to relay the communication between ground station and the space station, as shown in Figure 1. This approach is similar to the Tracking Data Relay Satellite system run by NASA[3] for the International Space Station. China also launched four satellites to communicate with Tiangong space station[4] while NASA launched 13. Those satellites are dedicated for the communication between the space station and the ground.

The link created by the relay satellites has its own limitations:

1) *High loss rate:*[5] Because of the long transmission distance and the high possibility of interference, the bit error rate is generally high. At the same time, weather conditions may also cause changes in the bit error rate. The overall bit error rate cannot be ignored.
2) *Limited bandwidth of satellite link:* Due to the limitation of the signal transceiver, the overall capacity of the satellite link may be limited, which will make the link more prone to congestion. The latest European Data Relay Satellite System[6] can only provide 300 Mb/s downlink to the ground station.
3) *Long RTT:* The RTT of the relay link can be as high as 600 ms due to the height of geosynchronous satellites.

We can use the ground station to create another link to the space station.[7] The RTT of the link is smaller than the one of the relay link. Combining the direct link and relay link can provide better network service to the space station.

Many applications' data flow between the space station and ground station has deadline requirement for its data transfer, such as real-time video communications, data collection, and text message. The real-time video communication and text message need to be delivered in time for smooth interaction. The data collection also has a deadline to be processed. We summarize the common transport requirements of those applications as follows.

1) *Block-based data transmission:* Those applications all generate and process the data in a block fashion and each block has a clear deadline requirement. The block is defined as the minimal unit of data for applications. A partially delivered block is useless for those applications. For example, video/audio encoders produce the encoded streams as a series of blocks (Macro blocks, I, B, P frame, or GOP, depending on the fine granularity of application control). The remote control command is packed inside a message.
2) *Deadline requirement:* The meaningful deadline for an application is the block completion time, i.e., the time between when the block is generated at the sender and when it is submitted to the application at the receiver. If a block cannot arrive before the deadline, then the QoE will be affected and the whole block is useless (obsoleted by newer blocks or no longer needed). In video conferencing, a video frame from each participant is a block and the deadline is the expected play time minus decoding time. For remote control, each control command is a message to be transmitted and executed by the receiver.
3) *Block dependency:* For a video stream, there are dependency between I/P frame in H264 and the base/enhance layer in SVC codec. If the depended block cannot arrive in time, then there is no point for the depending block to arrive. Therefore, the dependency should be respected when sending blocks.

In short, those applications need a transport service that can deliver their blocks of data before the

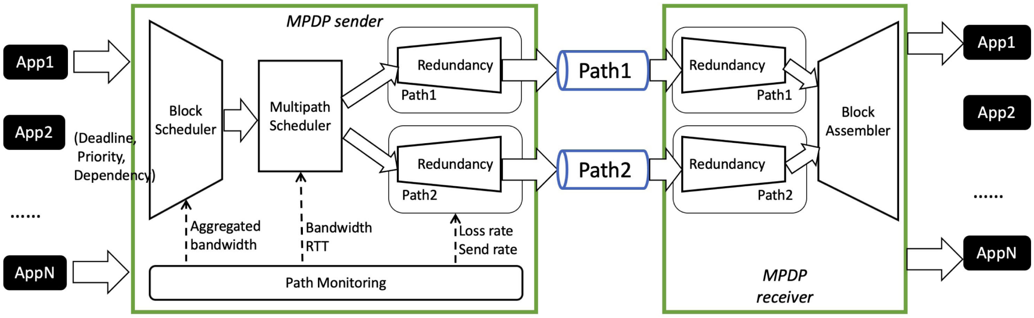

FIGURE 2. Architecture of MPDTP.

deadline. Each block has its own property such as deadline and dependency. Existing transport protocol such as TCP, UDP, QUIC, and RTP[8] cannot provide such a service, especially the deadline requirement. Without proper transport layer support, applications are forced to build their own wheels, usually on top of UDP. Each application has to handle the prioritization, sending, and acknowledging of data, etc., which are tedious and complex. A typical example is WebRTC,[9] which incorporates everything from encoding to transport, resulting in millions lines of code, making it really hard to be adopted by other applications.

Another problem of this approach is that the competition between different applications using congestion control will result in unnecessary loss and bandwidth wastage. In the terrestrial Internet, different applications' flows share different bottleneck. Congestion control is used to maintain the fairness between different applications in a distributed way. But applications mentioned above share a common bottleneck (relay links are dedicated for those applications). Applications are also controlled by a single entity (ground/space station). There is no need to use congestion control for each applications.

ARCHITECTURE OF MPDTP

We propose MPDTP, instead of letting each application run its own congestion control algorithm to compete with each other and result in lots of packet loss and low utilization of the link capacity. The proxy aggregates multiple application flows into one connection. Then, the proxy will distribute the data into multiple paths. The proxy will allocate the bandwidth to different applications based on their priority and urgency. For example, the priority of a text message is higher than the video calling.

The architecture of MPDTP is shown in Figure 2. Each *data block* and its metadata are first stored in a separate sending queue. Then, the *scheduler* module determines the sending order of blocks and discards the expired blocks. Next, the multipath scheduler allocates the data packets of the block to different paths. Then, they are divided into redundancy groups in the redundancy module to generate the corresponding redundant packets. These redundant data packets and the original data packets will be sent to the network. Their ACKs are collected and the packet loss is detected in a similar way to that in QUIC or TCP.

Lost packets are placed at the top of the corresponding sending queue and will be sent first when rescheduling. The *path monitor* module monitors the network status and provides estimates of bandwidth, RTT, and loss rate to the scheduler and redundancy module to assist in scheduling and redundancy decisions. Since the link is dedicated to be used by MPDTP, path monitoring can be done easily from signal strength. On the receiving side, the transport layer will receive the data and reassemble each block. This process is symmetrical to the sender.

DEADLINE-AWARE SCHEDULER

The goal of the scheduler is to use the existing network resources to complete the transmission of as many high-priority data blocks as possible before the deadline. This problem can be modeled as a classical task scheduling problem—schedule m blocks over n network paths. Each block has its own deadline and priority. There may exist dependencies between different data blocks. How can m data blocks be allocated to n paths so that the highest number of high-priority blocks can arrive before the deadline. This problem can be solved by integer programming but the complexity is too high.

To reduce the complexity of scheduling, MPDTP divides this scheduler into two parts: a block scheduler and a multipath scheduler. The block scheduler is responsible for determining the order in which blocks are sent, and the multipath scheduler is responsible for distributing the current block data on multiple

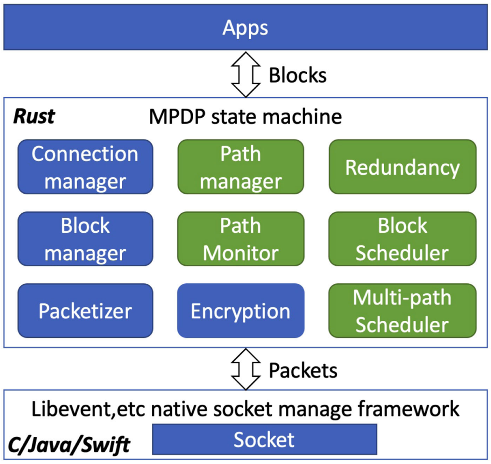

FIGURE 3. Implement MPDTP as an extension to QUIC.

paths. From the block scheduler's point of view, the multipath scheduler aggregates multiple heterogeneous paths into one path. By fixing the multipath scheduler, the completion time of a block can be estimated. The scheduler runs when any path can send packets. It first determines the data block to send, and then selects the first data packet to send from the data block.

There are multiple conflicting factors involved when the block scheduler determines the sending order. The first factor is the application-specified priority. Blocks with a higher priority should be sent first since it is more important to the user experience. The second factor is how close the block is to the deadline. MPDTP estimates the block completion time using the bandwidth and one-way delay estimation provided by the path monitoring module. Consider that the deadline will force the scheduler to iterate over all blocks every time. The overhead can be reduced by only recalculating when the bandwidth is changing and only considering some blocks with high priority or close to deadline. The bandwidth is the aggregated bandwidth of two paths. Then, MPDTP compares the estimated block completion time to the deadline. The third factor is the $unsent\_ratio$, which is defined as the ratio of the remaining size. If a block is almost completed, then we would better finish its remaining part so that we do not waste the network resource.

DEADLINE-AWARE ADAPTIVE REDUNDANCY

The design of multipath redundant module has two options: the redundancy for different paths is coupled or not. If the redundancy packets are sent over multiple paths (coupled), then the redundancy module immediately follows the block scheduler. After the block scheduler schedules a block, the redundancy module will generate redundancy packets. Then, the original packet and the redundant packet are distributed to multiple paths through the multipath scheduler. This can make full use of all available resources of multiple paths, but the cost is that the recovery time of redundant packets is more difficult to estimate, which is related to the specific multipath scheduling algorithm and the degree of heterogeneity and dynamics of the path. On the contrary, we design the redundant modules of each path to be independent of each other, and deal with packet loss and recovery on their respective paths. In this way, the multipath scheduler can more easily aggregate the resources of multiple paths into a virtual path for the block scheduler to use.

After the multipath scheduler distributes packets to each path, the redundancy module will generate the redundancy packets for the packet stream. To reduce the computation and transmission overhead, the redundancy module is activated only for tail packets. MPDTP use block-based forward error correction (FEC) to generate n redundancy packets for m original packets. m is defined as the largest value satisfying the deadline requirements of each packet: $\frac{m\times \mathrm{MSS}}{\mathrm{bandwidth}} + 2\times \mathrm{RTT} < \mathrm{deadline}$. n needs to be big enough to cover the loss rate: $n > m * \mathrm{loss\_rate}$.

STANDARDIZING MPDTP

We implement the MPDTP as an extension to QUIC. Based on the implementation experience, we give some guidance on how to standardize MPDTP.

First, MPDTP should leverage existing standards as much as possible. The extension to QUIC is shown in the green blocks shown in Figure 3. For path manager and handshake, the multipath extension proposed by Lui *et al.*[10] can be utilized. However, QOE_CONTROL_SINGALS frame is not needed because MPDTP does not use the QoE-based scheduling. For redundancy module, the process and frame definition from the work of Cui *et al.*[11] is used.

Second, modification to existing standards should be minimal too. To implement the block-based transfer, one block can be mapped to one QUIC stream so that the QUIC stream management can be reused to manage the block. When a block is dropped by sender, the RESET_STREAM frame is sent to cancel the corresponding stream. The triggering of the block scheduler and multipath scheduler remains the same as QUIC,

TABLE 1. Applications setup.

Application	Block	Deadline	Priority	Dependency
Text message	Message	800 ms	1 (High)	No
Audio calling	Audio frame	1 s	2 (Mid)	No
Video streaming	I/P frame	3 s	3 (Low)	P depends on I

i.e., when the application pushes new block or ACK is received. First, the block scheduler update the weighted priority value of each block. Then, the data of the block is distributed to each path by the multipath scheduler. The congestion control module can be replaced with the path monitoring module. The module has the following three functions.

1) Analyze the ACK to derive the path RTT, which is used to estimate the one-way delay.
2) Monitor the link signal strength to derive the bandwidth of each path for the calculation of the block remaining time.
3) Monitor the bit error rate of each link to derive the packet loss rate for the redundancy rate.

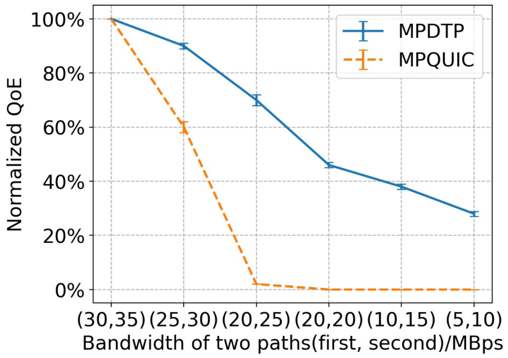

FIGURE 4. QoE under different bandwidth.

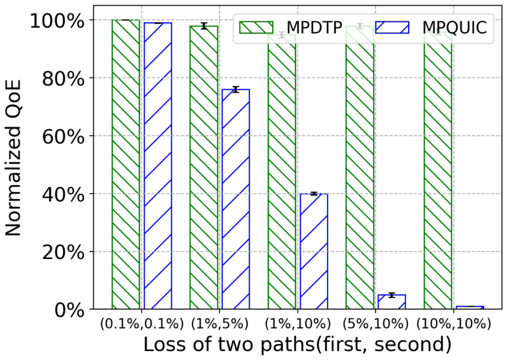

FIGURE 5. QoE under different loss rate.

Third, the implementation should be cross-platform. To accomplish that, the implementation can be separated into two parts: core state machine, which is platform agnostic, and the I/O process written in platform specific code, as shown in Figure 3. The API between the application and the state machine should be similar to the BSD socket API for backward compatibility.

EVALUATION

We use two PCs running Ubuntu 20.04 to simulate the ground station and the space station. The height of the simulated space station is 40 km and the orbital period is 90 min, which is similar to the Internet Space Station. They are connected by an OpenWrt router. The client has two interfaces and the server has one interface. We run TC[12] in the OpenWrt to simulate the relay links. The RTTs of path are set to 600 and 100 ms. We vary the bandwidth and loss rate of these two links to test MPDTP. When varying the bandwidth, the loss rate is set to 0.01% for each path. When varying the loss rate, the bandwidth is set to 40 MB/s for each path.

We run three applications' traces simultaneously between client and server. The setup of applications is shown in Table 1. We define the QoE of three applications as the total number of blocks arrive before the deadline weighted by its priority. For comparison, we run three applications over three MPQUIC connections. MPQUIC uses a decoupled New Reno congestion control algorithm for each path.

The result is shown in Figures 4 and 5. When bandwidth is enough, both MPQUIC and MPDTP can deliver all blocks before the deadline. When the bandwidth is smaller, there will be congestion on the link. Using congestion control to compete for limited bandwidth will waste limited bandwidth. Sending obsolete data is also a waste of bandwidth. MPDTP can allocate the limited bandwidth resource to high-priority blocks and thus can accomplish more blocks.

When the loss rate is high, MPDTP performs way better than MPQUIC. This is caused by two factors. The first factor is the New Reno congestion control algorithm used by the MPQUIC. When loss rate is high, the throughput of MPQUIC is very low. The second one is the redundancy module used by MPDTP, which can recover from the lost without retransmission.

CONCLUSION

In this article, we presented MPDTP for Space Network. MPDTP uses deadline-aware scheduling and redundancy to provide deliver-before-deadline service for applications in the space network. Our simulation showed that MPDTP can improve the deadline delivery performance by 5x.

ACKNOWLEDGMENTS

The authors would like to thank Dr. Kai Zheng and Dr. Zeqi Lai for their helpful suggestions and comments. This work was supported in part by the National Key RD Program of China under Grant 2018YFB1800303 and in part by National Natural Science Foundation of China Project under Grant 6213000078.

REFERENCES

1. J. Iyengar and M. Thomson, “QUIC: A UDP-based multiplexed and secure transport,” RFC 9000, RFC Editor, May 2021.
2. Y. Cui, T. Li, C. Liu, X. Wang, and M. Kühlewind, “Innovating transport with QUIC: Design approaches and research challenges,” *IEEE Internet Comput.*, vol. 21, no. 2, pp. 72–76, Mar./Apr. 2017, doi: 10.1109/MIC.2017.44.
3. Tracking and Data Relay Satellite (TDRS) |NASA, 2021. [Online]. Available: https://www.nasa.gov/directorates/heo/scan/services/networks/tdrs_main/
4. Tiangong space station: Wikipedia, 2021. [Online]. Available: https://en.wikipedia.org/wiki/Tiangong_space_station
5. M. Dong, Q. Li, D. Zarchy, P. B. Godfrey, and M. Schapira, “{PCC}: Re-architecting congestion control for consistent high performance,” in *Proc. 12th {USENIX} Symp. Networked Syst. Des. Implementation*, 2015, pp. 395–408, doi: 10.5555/2789770.2789798.
6. European Data Relay Satellite System, 2021. [Online]. Available: https://artes.esa.int/european-data-relay-satellite-system-edrs-overview
7. D. Vasisht, J. Shenoy, and R. Chandra, “L2D2: Low latency distributed downlink for LEO satellites,” in *Proc. ACM SIGCOMM Conf.*, 2021, pp. 151–164, doi: 10.1145/3452296.3472932.
8. H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson, “RTP: A transport protocol for real-time applications,” STD 64, RFC Editor, Jul. 2003.
9. Web real time communication, 2018. [Online]. Available: https://webrtc.org
10. Y. Liu, Y. Ma, C. Huitema, Q. An, and Z. Li, “Multipath extension for QUIC,” Internet-draft draft-liu-multipath-quic-03, IETF Secretariat, Mar. 2021. [Online]. Available: https://www.ietf.org/archive/id/draft-liu-multipath-quic-03.txt
11. Y. Cui, Z. Liu, H. Shi, J. Zhang, K. Zheng, and W. Wang, “Deadline-aware transport protocol,” Internet-draft draft-shi-quic-dtp-04, IETF Secretariat, Jul. 2021. [Online]. Available: https://www.ietf.org/archive/id/draft-shi-quic-dtp-04.txt
12. TC: Linux advanced routing and traffic control, 2021. [Online]. Available: http://lartc.org/lartc.html

HANG SHI is currently a Ph.D. student with the Department of Computer Science and Technology, Tsinghua University, Beijing, China. His research interests include transport layer optimization and protocol design. He received the B.E. degree in material science and engineering from Tsinghua University in 2015. Contact him at shi-h15@mails.tsinghua.edu.cn.

LEI ZHANG is currently working toward the Ph.D. degree with the Department of Computer Science and Technology, Tsinghua University, Beijing, China. Contact her at leizhang16@mails.tsinghua.edu.cn.

XUTONG ZUO is currently a Ph.D. student with the Department of Computer Science and Technology, Tsinghua University, Beijing, China. Her research interests include video streaming and multipath scheduling. She received the B.E. degree in computer science and technology from the Beijing University of Posts and Telecommunications, Beijing. Contact her at zuoxt18@mails.tsinghua.edu.cn.

QIAN WU is currently an Associate Professor with the Institute for Network Sciences and Cyberspace, Tsinghua University, Beijing, China. Her research interests include the next-generation Internet architecture and protocols, integrated space-terrestrial networks (ISTN), mobile and wireless networks, multipath transfer, and mobile multicast. She received the M.S. and Ph.D. degrees in computer science from Tsinghua University in 2002 and 2006, respectively. Contact her at wuqian@cernet.edu.cn.

HEWU LI is currently an Associate Professor with the Institute for Network Sciences and Cyberspace, Tsinghua University, Beijing, China, and the Director of the Wireless and Mobile Network Research Laboratory, Network Research Center. His research interests include mobile wireless network architecture, integrated spaceterrestrial networks (ISTN), testbed, broadband wireless access technology, and mobility architecture in next-generation networks. He received the M.S. and Ph.D. degrees in computer science from Tsinghua University in 2001 and 2004, respectively. Contact him at lihewu@cernet.edu.cn.

YONG CUI is currently a Full Professor with the Department of Computer Science and Technology, Tsinghua University. His research interests include computer network architecture and mobile computing. Cui received the Ph.D. degree in computer science from Tsinghua University. He is the corresponding author of this article. Contact him at cuiyong@tsinghua.edu.cn.