Literature Review

Shirley Shu, Tianfang Chang

1 Multipath Transport-Layer Protocol

MPTCP [1] is a multipath extension on top of TCP, supporting transmission with multiple paths in the transport layer. Thanks to its outstanding features such as throughput aggregation and congestion shift, it has been considerably adopted for commercial use. An example is the use of it since iOS11. Nevertheless, the next-generation networks pose a set of challenges to the protocols built upon the TCP/IP stack, e.g., connection breakage, and Head-of-Line (HoL) blocking issue. Motivated by the success of MPTCP, multipath extension over QUIC (MPQUIC) [2] is more promising to satisfy the demands of future applications. Even though MPQUIC is still under discussion by Internet Engineering Task Force (IETF), there are already some related works around the protocol implementation [3].

1.1 MPQUIC

The multipath QUIC protocol intends to compensate for the missing features in QUIC by utilizing different paths that exist between a client and a server [2]. The layered structure of MPQUIC is illustrated in Fig. 1.

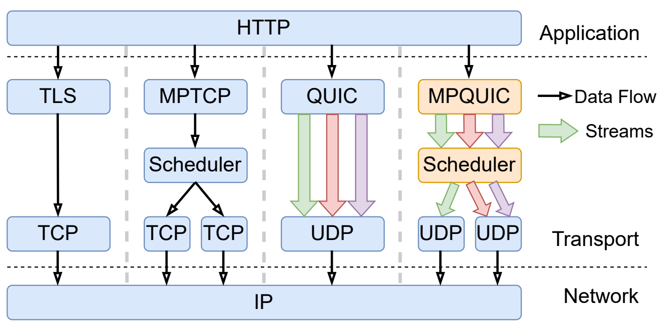

Figure 1: Structure of MPQUIC in comparison with others.

Based on several salient features and improvements in QUIC, the design specifications of MPQUIC [2, 3] are described in the following components.

Path Identification. MPQUIC introduced a path identification in its packet header to determine different paths in use. QUIC uses increasing packet numbers to identify and retransmit the lost packet. However, if all packets share one numbering space in MPQUIC and are sent over different paths, they might arrive out of order, resulting in a misinterpretation of packet loss. To deal with this issue, MPQUIC designed a per-path numbering space, which means packet numbers on different paths are isolated from each other and the sequential feature occurs only within that path.

Path Management. A path manager manages the path creation and removal in MPQUIC. In QUIC packets, the payload is comprised of multiple frames that can store stream data or control information. Benefiting from this frame structure, the extension of QUIC can define some specific types of frames to store the multipath information. To manage multiple paths, new frames, e.g., `ADD_ADDRESS`, `REMOVE_ADDRESS`, and `PATH_ABANDON`, are introduced to help the path establishment and removal.

Packet Scheduling. The path scheduler in MPQUIC is responsible for allocating packets onto different paths. There have been numerous scheduling policies proposed, each with pros and cons depending on the network scenarios and application requirements. The simplest scheduling algorithm is Round-Robin, which schedules packets on different paths sequentially, but may cause high latency when two flows have a large difference in bandwidth and round-trip time (RTT). So, MPQUIC uses Min-RTT by default, which is implemented in the Linux kernel for MPTCP. Provided that the congestion window still has space, the path with the lowest measured round-trip time (RTT) is preferred.

2 Scheduler

Research around the scheduling algorithms are getting popular in recent year. Here we introduce some typical works for packet scheduling algorithms in transport-layer protocols. Short summaries are introduced in the following subsections. Sec. 2.1 describes the default scheduling algorithms including Round-Robin, Min-RTT, and Blest. Some further works based on the default schedulers are illustrated in Sec. 2.2. In addition, combining with machine learning algorithms are popular these years. Sec. 2.3 explains papers working on learning-based scheduler.

2.1 Default Algorithms

The basic scheduling algorithm in multipath transport-layer protocols are Round-Robin, which arranges packets on each paths one-by-one. This idea is easy to implement but causes large gaps when two paths have distinct data rates or delays. Min-RTT is introduced to deal with this problem, which always select the path that have lowest round-trip-time (RTT). However, this algorithm will ignore the slow paths, which may not fully uses all essential paths. Blest[4] aims to deal with the problem of head-of-line blocking when multiple paths have heterogeneous data rates and delay. The main idea of this algorithm is to try to make the packets arriving in sequence with a prediction of the number of packets during RTT. The experiment shows it performs well with bulk send, but is not that good with websites or videos.

2.2 Improved Algorithms

Based on the idea of Blest, ECF[5] not only utilize the information of RTT but also uses other relevant information of paths, like the congestion window, to improve the scheduling algorithm. Fig. 2 shows the idea in detail. [6, 7] designed the stream-aware scheduling with the features of MPQUIC, refering the idea of ECF over MPTCP.

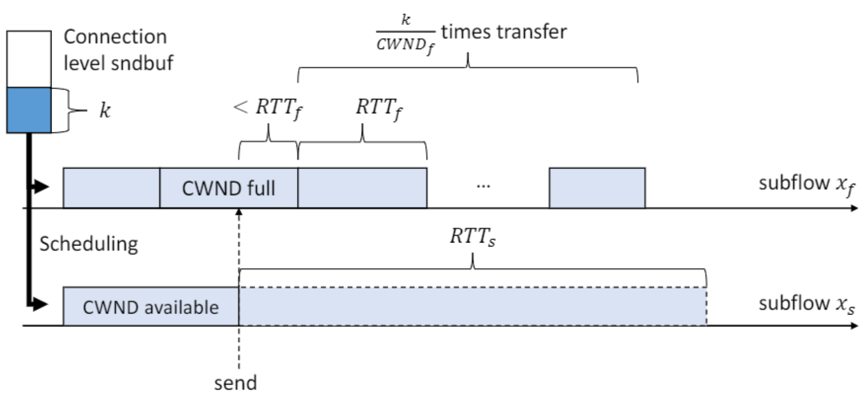

Figure 2: ECF

With the popularity of MPQUIC, the scheduling algorithms need to consider not only arranging packets over different paths but also take the priority streams into account. Pstream [8, 9] finds that scheduling without the recognition of the stream features can aggravate inter-stream blocking when sharing paths. Thus, it proposes Priority-Based Stream Scheduling which has a global scheduler for allocating streams to paths and the stream managers for each path. It uses a job shop scheduling algorithm to distribute streams' data to heterogeneity paths according to their priorities and sizes so that it can decrease the overall transmission time. It also introduces a bandwidth sharing mechanism for streams on the same path.

2.3 Learning-based Algorithms

Since the network in real world is always dynamic and unsatable, some researchers consider to apply online learning algorithms with the scheduling strategy. [10] introduced an approach to apply deep reinforcement learning on MPQUIC scheduler. It uses Q-learning to train the data collected by running MPQUIC srtt scheduler, then apply the DNN model (trained by Q-learning) to the new scheduler. This paper shared a way of utilizing some existing libraries to connect with the MPQUIC scheduler, but the performance is not that significant with higher delay and background traffic. In addition, Peekaboo [11] uses online learning to improve the scheduler. The speed at which the network changes can surpass the learning speed achieved through online learning. Few data, like mobility, may not help finish training and get a good schedule policy. [12] extends Peekaboo in 5G scenarios which has more homogeneous contexts, like average RTT, throughput, and loss rate. It shows that learning based scheduler (Peekaboo) performs a bit worse in a dynamic scenario as it needs frequent retraining. In the worst case, when the retraining is completed, the environment has already changed again. Thus, lifelong learning scheduler (or learning-based scheduler with a small training set) could be considered as our further research.

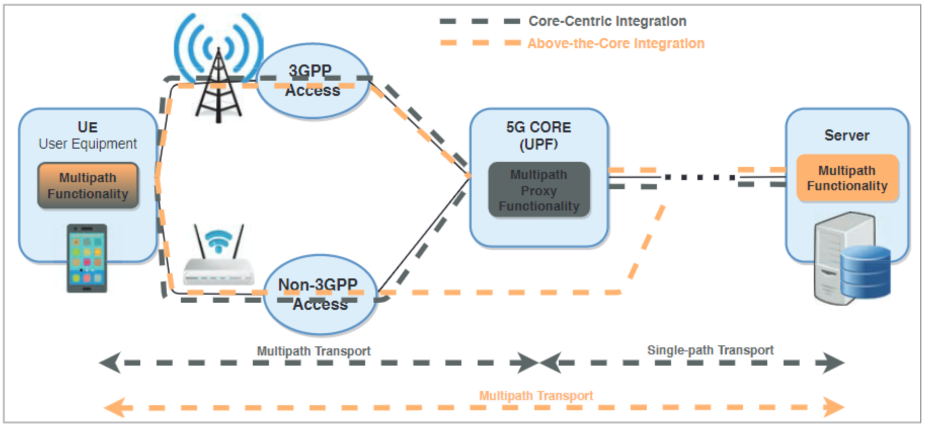

Figure 3: Peekaboo

References

[1] C. Raiciu, S. Barre, C. Pluntke, A. Greenhalgh, D. Wischik, and M. Handley, "Improving datacenter performance and robustness with multipath tcp," in *Proceedings of the ACM SIGCOMM 2011 Conference*, ser. SIGCOMM '11, 2011, p. 266–277.

[2] Q. De Coninck and O. Bonaventure, "Multipath quic: Design and evaluation," in *Proceedings of the 13th International Conference on Emerging Networking EXperiments and Technologies*, ser. CoNEXT '17, 2017, p. 160–166.

[3] T. Viernickel, A. Froemmgen, A. Rizk, B. Koldehofe, and R. Steinmetz, "Multipath quic: A deployable multipath transport protocol," in *Proceedings of the IEEE International Conference on Communications (ICC)*, 2018, pp. 1–7.

[4] S. Ferlin, Ö. Alay, O. Mehani, and R. Boreli, "Blest: Blocking estimation-based mptcp scheduler for heterogeneous networks," in *2016 IFIP Networking Conference (IFIP Networking) and Workshops*. IEEE, 2016, pp. 431–439.

[5] Y.-s. Lim, E. M. Nahum, D. Towsley, and R. J. Gibbens, "Ecf: An mptcp path scheduler to manage heterogeneous paths," in *Proceedings of the 13th International Conference on Emerging Networking EXperiments and Technologies*, ser. CoNEXT '17. New York, NY, USA: Association for Computing Machinery, 2017, p. 147–159. [Online]. Available: https://doi.org/10.1145/3143361.3143376

[6] B. Jonglez, M. Heusse, and B. Gaujal, "Srpt-ecf: challenging round-robin for stream-aware multipath scheduling," in *2020 IFIP Networking Conference (Networking)*, 2020, pp. 719–724.

[7] A. Rabitsch, P. Hurtig, and A. Brunstrom, "A stream-aware multipath quic scheduler for heterogeneous paths," in *Proceedings of the Workshop on the Evolution, Performance, and Interoperability of QUIC*, ser. EPIQ'18. New York, NY, USA: Association for Computing Machinery, 2018, p. 29–35. [Online]. Available: https://doi.org/10.1145/3284850.3284855

[8] X. Shi, F. Zhang, and Z. Liu, "Prioritybucket: A multipath-quic scheduler on accelerating first rendering time in page loading," in *Proceedings of the Eleventh ACM International Conference on Future Energy Systems*, ser. e-Energy '20. New York, NY, USA: Association for Computing Machinery, 2020, p. 572–577. [Online]. Available: https://doi.org/10.1145/3396851.3402923

[9] X. Shi, L. Wang, F. Zhang, B. Zhou, and Z. Liu, “Pstream: Priority-based stream scheduling for heterogeneous paths in multipath-quic,” in *2020 29th International Conference on Computer Communications and Networks (ICCCN)*. IEEE, 2020, pp. 1–8.

[10] M. M. Roselló, “Multi-path scheduling with deep reinforcement learning,” in *2019 European Conference on Networks and Communications (EuCNC)*, 2019, pp. 400–405.

[11] H. Wu, Ö. Alay, A. Brunstrom, S. Ferlin, and G. Caso, “Peekaboo: Learning-based multipath scheduling for dynamic heterogeneous environments,” *IEEE Journal on Selected Areas in Communications*, vol. 38, no. 10, pp. 2295–2310, 2020.

[12] H. Wu, G. Caso, S. Ferlin, Alay, and A. Brunstrom, “Multipath scheduling for 5g networks: Evaluation and outlook,” *IEEE Communications Magazine*, vol. 59, no. 4, pp. 44–50, 2021.