Computer Science and Information Systems 21(2):625–643 https://doi.org/10.2298/CSIS230818078L

A Novel Multipath QUIC Protocol with Minimized Flow Complete Time for Internet Content Distribution ⋆

Fang-Yi Lin[1], Wu-Min Sung[1], Lin Hui[2,⋆⋆], Chih-Lin Hu[1], Nien-Tzu Hsieh[1], and
Yung-Hui Chen[3]

[1] Department of Communication Engineering, National Central University
Taoyuan City 320317, Taiwan
{fangyi.lin, wumin.sung, neintzu.hsieh}@g.ncu.edu.tw
clhu@ce.ncu.edu.tw
[2] Department of Computer Science and Information Engineering, Tamkang University
New Taipei City 25137, Taiwan
amar0627@gms.tku.edu.tw
[3] Department of Computer Information and Network Engineering, Lunghwa University of
Science and Technology, Taoyuan City 333326, Taiwan
cyh@mail.lhu.edu.tw

Abstract. The rapid growth of network services and applications has led to an exponential increase in data flows on the internet. Given the dynamic nature of data traffic in the realm of internet content distribution, traditional TCP/IP network systems often struggle to guarantee reliable network resource utilization and management. The recent advancement of the Quick UDP Internet Connect (QUIC) protocol equips media transfer applications with essential features, including structured flow-controlled streams, quick connection establishment, and seamless network path migration. These features are vital for ensuring the efficiency and reliability of network performance and resource utilization, especially when network hosts transmit data flows over end-to-end paths between two endpoints. QUIC greatly improves media transfer performance by reducing both connection setup time and transmission latency. However, it is still constrained by the limitations of single-path bandwidth capacity and its variability. To address this inherent limitation, recent research has delved into the concept of multipath QUIC, which utilizes multiple network paths to transmit data flows concurrently. The benefits of multipath QUIC are twofold: it boosts the overall bandwidth capacity and mitigates flow congestion issues that might plague individual paths. However, many previous studies have depended on basic scheduling policies, like round-robin or shortest-time-first, to distribute data transmission across multiple paths. These policies often overlook the subtle characteristics of network paths, leading to increased link congestion and transmission costs. In this paper, we introduce a novel multipath QUIC strategy aimed at minimizing flow completion time while taking into account both path delay and packet loss rate. Experimental results demonstrate the superiority of our proposed method compared to standard QUIC, Lowest-RTT-First (LRF) QUIC, and Pluginized QUIC schemes. The relative performance underscores the efficacy of our design in achieving efficient and reliable data transfer in real-world scenarios using the Mininet simulator.

⋆ This paper is an extended version of a conference paper, which was published in The 13th International Conference on Frontier Computing (FC 2023), July 10 -14, 2023, after a thorough enhancement.

⋆⋆ Corresponding author

Keywords: Quick UDP internet connect (QUIC), multipath transport, HTTP, content distribution, internet protocol, internet services.

1. Introduction

The HTTP protocol family [1] is the basis for global internet data communications, enabling the rapid development of Web browsers and internet services. HTTP/1.1 and HTTP/2 are two major web protocols. With the proliferation of user demands and mobile services, particularly mobile media streaming and AR/VR flows to an increasing user population, the functions provided by HTTP/1.1 and 2 are no longer sufficient. In 2013, the IETF organization proposed the RFC 9000 [2], i.e., Quick UDP Internet Connect (QUIC) – a UDP-based multiplexed and secure transport protocol. QUIC is often known as the transport layer for HTTP/3. It is recommended to develop HTTP/3 with QUIC and UDP in place of conventional HTTP/1.1 and 2 with TCP or UDP for internet services and applications in wireless and mobile environments.

QUIC provides internet applications with flow-controlled streams for encrypted, multiplexed and reliable communication, low-latency connection establishment, and network path migration. It can sustain high dynamics of traffic loading and resource provision on network hosts, rather than HTTP/2 based on TCP, TLS 1.2, and other HTTP derivatives. Compared with TCP, QUIC need not the 3-way handshake mechanism, so it can greatly reduce the time of network connection establishment and transmission latency. With multiplexing and path migration, it can strengthen the control of congested networks, making it more suitable for emerging mobile services in Wi-Fi and 4G/5G environments.

Prior studies argued that the performance of QUIC can be affected in the case of delivering large-size data between two endpoints [3]. This is because the packet pacing policy is basically used to vary the transmission speed of each stream when numerous packets enter into that stream. The overall completion time of a data flow in a stream, so-called *flow complete time* briefly, will vary as well. Thus, data throughput of each flow going on a link may not reach to the full bandwidth capacity. Moreover, internet operators may operate any self-protection controls by limiting the transmission rate of UDP flows. Regarding the safety of a network system, a self-imposed constraint can be understood to defend against unpredictable threats to the system, although the bandwidth resource along with those links between two endpoints is not used fully.

As the literature review will be mentioned in Section 2, recent studies used Multipath QUIC (MPQUIC) to deal with the above concerns arising from the restrictions of a single path. Similar to Multipath TCP (MPTCP) [4], MPQUIC sends data through different paths and uses the aggregate bandwidth of different paths. It also likely modifies the *path scheduler* policy for increasing the transmission speed and thence decreasing the path delay that definitely corresponds to the end-to-end transmission delay of a QUIC stream between two endpoints in a network. In light of the aforementioned concept of MPQUIC, our study in this paper leverages the functionality of MPQUIC to devise a novel MPQUIC-based path selection strategy for internet content delivery. The contributions of our study are outlined as follows:

- We introduce a novel Multipath QUIC scheme. Its functionality is distinguished by considering both path latency and packet loss rate to identify the most efficient paths.

As a result, network transmission performance is enhanced by minimizing the flow completion time.
- We present the mathematical formulation of the proposed MPQUIC-based path selection strategy and detail the algorithmic procedures. Subsequently, we create a proof-of-concept implementation on the Mininet simulation platform.
- We evaluate the relative performance of our proposed strategy against several standard schemes, including standard QUIC, Lowest-RTT-First (LRF) QUIC [5], and Pluginized QUIC (PQUIC) [6] scheduling strategies, using the Abilene topology on the Mininet emulator. Performance results underscore the superiority of our strategy. Notably, our scheme consistently achieves stable and efficient outcomes in terms of the cumulative distribution function (CDF) of flow completion time. Furthermore, our strategy results in lower path delay and packet loss rates compared to other schemes.
- Performance results demonstrate the superiority of our scheme because this scheme can achieve stable and efficient effects in measure of the cumulative distribution function (CDF) of flow completion time. In addition, this scheme obtains lower path delay and packet loss rate than the other schemes.

The rest of this paper is organized as follows. Section 2 describes background knowledge ad related work. Section 3 details the problem formulation and the path selection algorithm. Section 4 describes the relative performance. Finally, the conclusion is given in Section 5.

2. Background Knowledge & Related Work

Section 2.1 briefly introduces the QUIC protocol to ease the understanding of special functional extensions by contrast to the conventions of TCP and HTTP protocols. Section 2.2 mentions the recent studies on the QUIC-based media transfer techniques in the literature.

2.1. Preliminary Knowledge

With the increasing demand for real-time applications, HTTP/2 shortcomings came to the fore; HTTP/3 aims to provide fast, reliable, and secure web connections. Figure 1 illustrates the architecture of the HTTP/3 protocol. HTTP/3 uses a new transport layer network protocol called QUIC, which runs over the UDP internet protocol instead of the ordered message exchange by TCP. The goal of HTTP/3 is to improve the overall web experience, suitable for IoT, real-time applications, and micro services. In addition, UDP provides greater flexibility, so that it can enables QUIC to exist completely in the user space. When QUIC can be independent of the operating system on the host, users only need to update a Web browser version with QUIC supported to experience the improved network performance by HTTP/3 [7]. Explicitly, QUIC serves as a new message transport layer, featuring Zero Round-Trip Time (0-RTT), flow control, congestion control, multiplexing, built-in security through TLS 1.3, and multiple paths. To aid comprehension, the following describes some of the essential features of QUIC:

Firstly, QUIC's primary attribute is the reduction in connection establishment latency. Unlike traditional TCP connections, QUIC eliminates the need for a three-way handshake, allowing for swift connection establishment. Thus, QUIC can lower initial latency

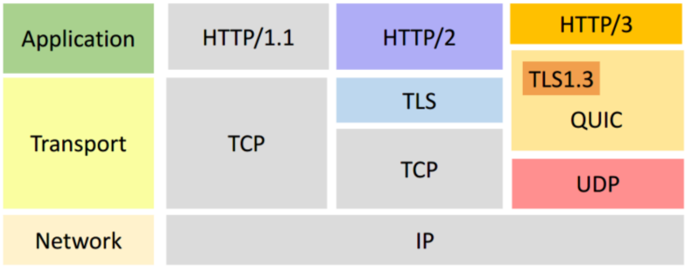


Fig. 1. HTTP architecture

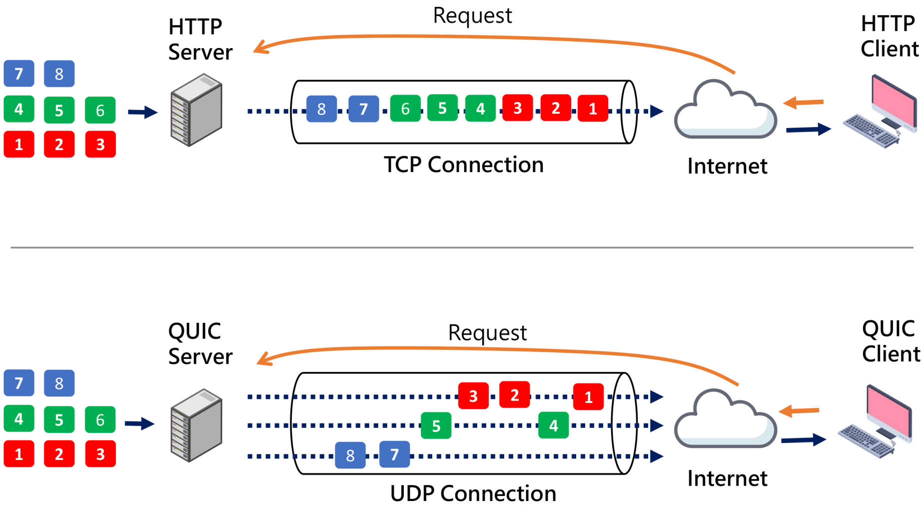


Fig. 2. Difference of the multiplexing features between TCP and QUIC

and quickly respond to end users through 0-RTT by sending data in the very first packet of a connection. Secondly, multiplexing allows for the concurrent transmission of multiple data streams over a single connection, as shown in Figure 2. This improves the efficiency of data transfer and overall performance while addressing the head-of-line blocking problem commonly encountered in HTTP/1.1. Additionally, QUIC possesses built-in error correction mechanisms that swiftly handle corrupted or lost data packets, enhancing the reliability of data transfer in the network. Thirdly, while congestion control in TCP commonly uses the CUBIC algorithm [8], it is not the most optimal for transmitting latency-sensitive network traffic. QUIC offers both the CUBIC and the Bottleneck Bandwidth and Round-trip propagation time (BBR) [9] schemes to address congestion-related issues. BBR actively probes and groups recently sent data, establishing a network model based on the current maximum bandwidth and round-trip time, allowing for the adjustment of transmission rates based on dynamic network conditions, effectively preventing flow con-

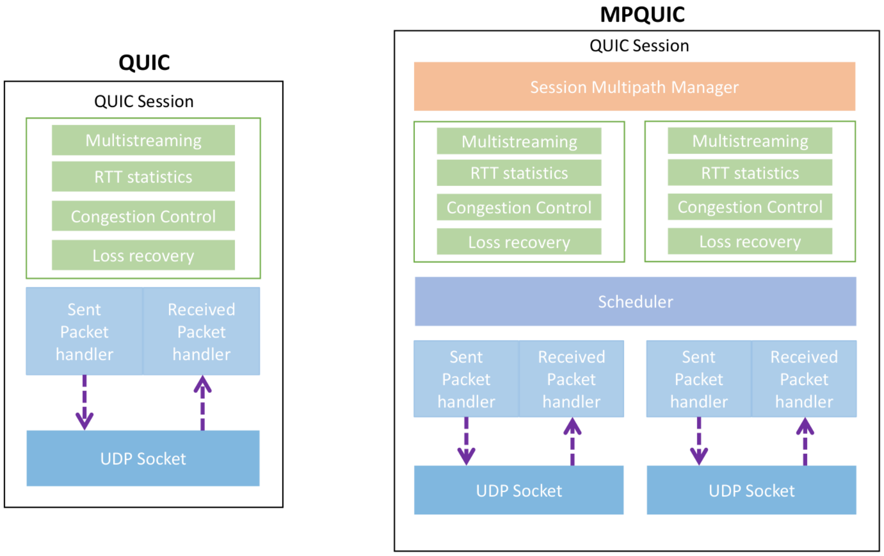


Fig. 3. Comparison between QUIC and MPQUIC network protocols

gestion and optimizing network performance. Fourthly, QUIC integrates Transport Layer Security (TLS) version 1.3 by default, ensuring that data is encrypted and secure during transmission. QUIC's adaptability is notable, allowing for dynamic path and protocol version selection in response to real-time changes in network conditions. Finally, QUIC enables the simultaneous utilization of multiple paths in a network, bolstering network robustness and performance by sending data through various routes, reducing latency, and maximizing bandwidth utilization, as illustrated in the comparison between QUIC and MPQUIC network architectures in Figure 3.

In summary, QUIC offers a comprehensive suite of features that collectively improve internet communication by enhancing speed, reliability, and security, making it well-suited for a wide range of network applications and effectively addressing the demands of modern internet usage, including real-time communication, mobile networks, and high-performance scenarios.

2.2. Related Work

Lots of studies on MPQUIC was inspired by MPTCP. As addressed in [10], the concept of MPQUIC can arrange QUIC connections to go on different paths according to network characteristics. There are two main reasons for the use of the multi-path function. The first is to collect the network resources of different paths to transmit data. Automatically selecting the best path becomes an interesting idea. The second is to maintain user experience against network failures. Given a device with multiple ports, if one of the network interfaces/ports/paths run to failed, the way of immediately switching to another one will not affect the user experience. Using multi-path designs can thus ensure the reliability and stability of network transport services because such designs can distribute and schedule streams to reduce the overall completion time with respect to media transfer in the Internet.

Our literature review summarizes recent studies that proposed various MPQUIC scheduling methods based on a variety of design aspects, such as transmission completion time, path characteristics, data priority, congestion control, and machine learning to enhance the performance of multipath transmission. In course of MPQUIC scheduling, the path selection is crucial for determining the network throughput, reliability, and load balancing with respect to different service requirements. In what follows, we classify prior studies into five categories corresponding to different design aspects.

1. Transmission completion time: [11] investigated MPQUIC's performance on different paths using the proposed Estimated Transfer Completion time (ETC) scheduling method. It considers transmission time and path congestion, reducing the overall file transmission time. However, as a large scheduling unit is used, this method has a drawback of being inefficient for transferring short files. [12] proposed the Delay-based In-Order Decode (DIOD) method which combines Forward Error Correction (FEC) and MPQUIC for reliable and delay-sensitive applications. While it reduces the influence of packet loss, it does not guarantee deadlines and will necessitate a precise loss estimation method for scheduling flexibility.
2. Path characteristics: [13] showed a PStream scheduling method that efficiently matches stream-path characteristics and avoids stream competition for the fastest path. [14] proposed a Nine Tails scheduler that can selectively use redundant paths to reduce latency as sending data in the tail part. By switching between redundant and non-redundant scheduling policies, it can have higher overall throughput and loss recovery. [15] designed an optimal bandwidth allocation strategy which can prioritize streams with a combination of priority and size. However, it underutilizes multi-path aggregation, resulting in suboptimal bandwidth allocation for time-critical stream. When network topology and bandwidth changes, this strategy may fall in performance degradation and flow congestion.
3. Data priority: [16] emphasized prioritized fair bandwidth allocation based on stream priority to prevent delays of individual streams due to varying network paths. [17] developed a Priority Bucket method that categorizes streams into priority-based buckets. Streams with the same priority in a bucket are served in a first-come-first-served order using HTTP/2 expressions. [18] designed Xlink, which is a user-perceived video Quality of Experience (QoE) control mechanism for MPQUIC scheduling. It showed the feasibility and deployability of MPQUIC in the Taobao platform, while substantial bandwidth is potentially required. [19] assumed the server-side has prior knowledge of the web page's dependency tree. It used a flow-aware downlink packet scheduling with stream priorities to optimize the transmission order of streams. This way can reduce flow completion time, page loading time, and expedite loss recovery, but may have efficiency implications for low-priority flows.
4. Congestion control: [20] developed MM-QUIC within the ITSN architecture, utilizing regular satellite orbits for rapid transmission. It also incorporated a basic multipath model for congestion management but noted potential packet loss during handovers in weak signal areas. [21] extended MPQUIC to SR-MPQUIC for 5G networks, improving latency and reliability for prioritizing traffic with redundant path

replication. With the primary focus on delay-sensitive traffic, it may slightly increase bandwidth usage and latency. [22] focused on congestion control and packet scheduling in multipath scenarios. It proposed a Delay-BBR algorithm that complements rate control to reduce packet loss and transmission delay for real-time video.

5. Machine learning: In [23], a reinforcement learning-based scheduling method, Peekaboo, was proposed. It considered temporal certainty and randomness in path characteristics for decision-making. [24] proposed MPQUIC schedulers using the deep reinforcement learning, this design which emphasized fairness to concurrent TCP flows in multipath protocols. [25] introduced a reinforcement learning-based MPQUIC scheduler using Deep Q-Network (DQN) to improve multimedia streaming performance and reduce video download time.

Our study considers the flow completion time in related with two network-oriented factors, i.e., delay and packet loss rate of a path. Accordingly, we formulate a weighting normalization method to calculate the weights of paths, which can be used to facilitate path selection and thus minimize the flow completion time over MPQUIC streams.

3. Design of Path Selection Scheme

This section first describes the problem formulation and then specifies a novel MPQUIC-based path selection scheme for efficient content delivery in the internet.

3.1. Problem Formulation

Give a network topology $G(V, L)$. For every link $l_{i,j} \in L$ from v_i to v_j, the available bandwidth, the delay of the link, and the packet loss rate w.r.t $l_{i,j}$ are denoted as $b_{i,j}$, $t_{i,j}$ and $o_{i,j}$, respectively. Then, $b_{i,j}^{max}$ denotes the maximum amount of bandwidth that $l_{i,j}$ can use.

Let F contain a set of all streams in $G(V, L)$, $\mathcal{P}_f^*$ represent a multipath set in use for a stream $f \in F$, $P_f^*[m]$ be the set of links in the m^{th} path, and likewise $\mathcal{P}_f^*[m][n]$ be the n^{th} link of the m^{th} path. Thus, for the stream and path selection, we take $x_{l_{i,j}}^f$ to be a binary indicator, defined as follows:

$$x_{i,j}^f = \begin{cases} 1, & \text{if a stream } f \text{ passes through a link } l_{i,j} \ , \\ 0, & \text{other conditions.} \end{cases} \tag{1}$$

We further define several expressions regarding the relationship between links and paths, as follows:

$$b_f^P = min\Big(b_{i,j} \times x_{i,j}^f\Big), \quad \forall l \in l_{i,j},\ x_{i,j}^f \neq 0,\ f \in F \tag{2}$$

$$b_{i,j}^{max} \geq \sum_{f \in F} b_{i,j} \times x_{i,j}^f, \quad \forall l \in l_{i,j} \tag{3}$$

$$t_f^P = \sum_{l_{i,j} \in L} t_{i,j} \times x_{i,j}^f, \quad \forall\, f \in F,\ l \in l_{i,j} \tag{4}$$

Table 1. Notations used in MPQUIC domain

Symbol	Description
$G(V, L)$	a graphic representation of a MPQUIC system
V	a set of all nodes in $G(V, L)$
L	a set of all links between two adjacent nodes in $G(V, L)$
$b_{i,j}$	available bandwidth of a link $l_{i,j}$
$b_{i,j}^{max}$	the maximum bandwidth of a link $l_{i,j}$
$t_{i,j}$	transmission delay of a link $l_{i,j}$
$o_{i,j}$	packet loss rate of a link $l_{i,j}$
F	a set of all data flows in the network
b_f	amount of bandwidth required for data stream f
t_f	transmission delay tolerance of a data stream f
o_f	packet loss rate tolerance for a stream f
$\mathcal{P}$	a set of all routing paths between any two nodes in $G(V, L)$
$\mathcal{P}_f$	a set of available paths for a data stream f
$\mathcal{P}_f[m]$	a set of links for the m^{th} path available to the data stream f
$\mathcal{P}_f[m][n]$	n^{th} link of the m^{th} path available to the data stream f
$P[n]$	n^{th} link of path P
$\mathcal{P}_f^*$	a set of multipath that the system ultimately uses for the data stream f
$\mathcal{P}_f^*[m]$	a link set of the m^{th} path in a set of multipath used by the system for stream f
$\mathcal{P}_f^*[m][n]$	the n^{th} link in the m^{th} path in the set of multipath used by the system for the data stream f

$$o_f^P = 1 - \prod_{l_{i,j} \in L} \left(1 - o_{i,j} \times x_{i,j}^f\right), \quad \forall l \in l_{i,j},\ x_{i,j}^f \neq 0,\ f \in F \tag{5}$$

$$y(\mathcal{P}_f^*) = \begin{cases} 1, & \bigcup P_f^* \neq \emptyset, \\ 0, & \bigcup P_f^* = \emptyset. \end{cases} \tag{6}$$

Formula (2) indicates the available bandwidth of a stream f in the set of paths P, and then takes the minimum value. (3) indicates that the bandwidth passed by a link cannot be greater than the maximum bandwidth available of the link. (4) means the sum of transmission delays on a link w.r.t. a stream f. (5) is to multiply the successful rate of each link to get the overall successful rate on a path, so as to obtain the packet loss rate of this path.

To transform a single-path stream into a multipath stream by (6), $y(\mathcal{P}_f^*)$ indicates whether any link and path in the set of paths $\mathcal{P}_f^*$ can be reused or not. Here, we further discuss two cases, as follows.

Case 1 When the links and paths in $\mathcal{P}_f^*$ are not reused.

Since links are not reused, the sum of the available bandwidth of each path can be calculated by (7). Then, for $y(\mathcal{P}_f^*) = 0$ and $\forall v_j \in V$, we can formulate (8) to check the link condition of v_i and v_j: (i) the total number of positive multipaths, (ii) the total number of negative multipaths, and (iii) a balanced state if both v_i and v_j are intermediate relays.

$$b_f^* = \sum_{P \in \mathcal{P}_f^*} b_f^P, \quad \forall f \in F,\ y(\mathcal{P}_f^*) = 0 \tag{7}$$

$$\sum_{l_{i,j} \in L} x_{i,j}^f - \sum_{l_{j,i} \in L} x_{j,i}^f = \begin{cases} |\mathcal{P}_f^*|, & \text{if } v_i \text{ is a start point of } f\ , \\ -|\mathcal{P}_f^*|, & \text{if } v_i \text{ is a target point of } f, \\ 0, & \text{if } v_i \text{ is a relay point of } f. \end{cases} \tag{8}$$

□

Case 2 When the links and paths in $\mathcal{P}_f^*$ can be reused

Let $z_{l_{i,j}}^{\mathcal{P}_f^*}$ indicate whether $l_{i,j}$ is reused in $\mathcal{P}_f^*$:

$$z_{l_{i,j}}^{P_f^*} = \begin{cases} 1, & l_{i,j} \subseteq \bigcup \mathcal{P}_f^*, \\ 0, & l_{i,j} \nsubseteq \bigcup \mathcal{P}_f^*. \end{cases} \tag{9}$$

$n(l_{i,j}, P_f^*)$ indicates the number of times that $l_{i,j}$ is reused by some paths in $\mathcal{P}_f^*$:

$$n(l_{i,j}, P_f^*) = \begin{cases} \sum\limits_{m \in |\mathcal{P}_f^*|} \sum\limits_{n \in |\mathcal{P}_f^*[m]|} l_{i,j} \wedge P_f^*[m][n] - 1, & \forall f \in F,\ l_{i,j} \in L,\ z_{l_{i,j}}^{P_i^*} = 1, \\ 0, & \forall f \in F,\ l_{i,j} \in L,\ z_{l_{i,j}}^{P_i^*} = 0. \end{cases} \tag{10}$$

Then, the bandwidth of a link is divided into two parts: the link bandwidth that has been reused $\bar{b_f^*}$, and the link that has not been reused $\hat{b_f^*}$, as follows.

$$b_f^* = \bar{b_f^*} + \hat{b_f^*}, \quad \forall f \in F, \quad \text{subject to} \tag{11a}$$

$$\bar{b_f^*} = min(b_f^P), \quad \forall f \in F,\ P \in \mathcal{P}_f^*,\ y(\mathcal{P}_f^*) = 1,\ z_{l_{i,j}}^{P_i^*} = 1. \tag{11b}$$

$$\hat{b_f^*} = \sum_{P \in \mathcal{P}_f^*} b_f^P, \quad \forall f \in F,\ y(\mathcal{P}_f^*) = 1,\ z_{l_{i,j}}^{P_i^*} = 0, \tag{11c}$$

Formula (11a) adds the two parts together, which yields the total amount of bandwidth that a path set can provide.

Formula (12) clarifies the link relation in three conditions. (i) If v_i is a start point of a stream f, the total of paths that a steam can still use is given by $|\mathcal{P}_f^*|$ minus the number of times $l_{i,j}$ that is currently used by some paths in $\mathcal{P}_f^*$, i.e., $n(l_{i,j}, P_f^*)$. (ii) If v_i is a target point, the calculation is in opposition to (i). (iii) Finally, if v_i is a relay w.r.t. $\forall\, y(\mathcal{P}_f^*) = 1$ and $v_j \in V$, there are three sub-cases (a)(b)(c). Explicitly, (a) multiple paths converge at this relay point, then $n(l_{j,i}, P_f^*) - n(l_{i,j}, P_f^*)$ is negative. (b) multiple paths to divert from this point, this outcome is positive. (c) in a balanced state, the outcome equals to 0.

$$\sum_{l_{i,j} \in L} x_{l_{i,j}}^f - \sum_{l_{j,i} \in L} x_{l_{j,i}}^f = \begin{cases} |\mathcal{P}_f^*| - n(l_{i,j}, P_f^*), & \text{if } v_i \text{ is a start point of } f\ , \\ -|\mathcal{P}_f^*| + n(l_{j,i}, P_f^*), & \text{if } v_i \text{ is a target point of } f, \\ n(l_{j,i}, P_f^*) - n(l_{i,j}, P_f^*), & \text{if } v_i \text{ is a relay point of } f. \end{cases} \tag{12}$$

□

Note that under the multipath scenario, the delay time and packet loss rate of a path are not affected by whether a path is reused subject to (2). Regardless of the value of (6), the delay time and packet loss rate w.r.t. any $P \in \mathcal{P}_f^*$, denoted as t_f^* and o_f^*, can be given below.

$$t_f^* = \max(t_f^P)\ , \quad \forall f \in F,\ P \in \mathcal{P}_f^*,\ y(\mathcal{P}_f^*) = 0 \tag{13}$$

$$o_f^* = \sum_{P \in \mathcal{P}_f^n} \frac{o_f^P}{|\mathcal{P}_f^*|}, \quad \forall\, f \in F, y(\mathcal{P}_f^*) = 0 \tag{14}$$

According to (13), given a set of final selected multipaths, the delay time is represented by the maximum delay time on the path for $\forall P \in \mathcal{P}_f^*$. The outcome of (14) indicates the average of packet loss rate for those selected paths in $\mathcal{P}_f^*$. After calculating the available bandwidth, delay time, and packet loss rate, now, it is able to figure out the comparison between user requirements and actually available provision, as explained below.:

$$b_f \leq b_f^*, \quad \forall f \in F \tag{15}$$

$$t_f \geq t_f^*, \quad \forall f \in F \tag{16}$$

$$o_f \geq o_f^*, \quad \forall f \in F \tag{17}$$

Particularly, (15) ensures that the multipath bandwidth is available for streaming f, while (16) and (17) enforce that both transmission delay and packet loss rate in the selected path need to be smaller than the tolerable bounds as requested by f.

Hence, in accordance with the above formulae and constraints of the multipath provision, we develop an optimal multipath selection problem of minimizing the flow completion time subject to user requirements, as expressed below:

$$\begin{array}{ll}
\text{arg}\ \min \sum\limits_{f \in F} t_f^*, & \\
\text{s.t.} & \\
\quad x_{i,j}^f = 1, & \forall l_{i,j} \in L, \\
\quad z_{l_{i,j}}^{\mathcal{P}_f^*} \in (0,1), & \forall \mathcal{P}_f^*,\ l_{i,j} \in L, \\
\quad y(\mathcal{P}_f^*) \in (0,1), & \forall \mathcal{P}_f^* \in \mathcal{P}, \\
\quad \text{Eqs.}\ (15), (16), (17). &
\end{array} \tag{18}$$

3.2. MPQUIC-Based Path Selection and Algorithmic Procedures

Our study refers to the research efforts in [26][27], and learns that such a multipath selection problem for QoS-based data streaming is known as NP-Complete [28]. Instead of finding a static optimization in theory, our study in this paper attempts to develop an optimal-approximate solution to figure out a set of appropriate multipaths using heuristic strategies with two design factors, i.e., path delay and packet loss rates. Particularly, we describe a weighting normalization method in 19 with two tuning parameters α and β to change the relative influence of path delay and packet loss rate over MPQUIC streams.

$$p_w = \alpha \times \frac{t_f}{t_f^*} + \beta \times \frac{o_f}{o_f^*}. \tag{19}$$

In what follows, we specify the algorithmic procedures for finding the paths for MPQUIC streams.

Algorithm 1 Path Set Selection with Joint Path Delay and Packet Loss Rate

```
input : $G(V, L)$: network topology,
        $k$: the number of paths in the multipath,
        $\alpha$: a coefficient of path delay,
        $\beta$: a coefficicient of packet loss.
output: $\mathcal{P}_f^*$: the set of multipath.
while Flow $f$ comes into the system do
    $\mathcal{P}_f = \{\emptyset\}$;
    $A[\,][\,] =$ null;
    while $\mathcal{P}_f = \{\emptyset\}$ do
        $\mathcal{P}_f \leftarrow getDefaultPathSet(\mathcal{P}, f)$ ;
        foreach $p \in \mathcal{P}_f$ do
            $A[p][0] \leftarrow getPathBW(\mathcal{P}[p])$ ;                ▷ (2)
            $A[p][1] \leftarrow getPathDelay(\mathcal{P}[p])$ ;             ▷ (4)
            $A[p][2] \leftarrow getPathPL(\mathcal{P}[p])$ ;                ▷ (5)
        end foreach
    end while
    if $(P_f = \{\emptyset\}$ or $|P_f| < k)$ then
        $Reject\ f$ ;
    else
        $\mathcal{P}_f^* \leftarrow getkPath(\mathcal{P}_f, \alpha, \beta, f, k, A)$ ;   ▷ Go to Alg. 2
        if $\mathcal{P}_f^* = \emptyset$ then
            $\mathcal{P}_f^* \leftarrow getShorestkPath \in \mathcal{P}_f$ ;
        end if
    end if
end while
```

Algorithm 1 Path Set Formation with Joint Path Delay and Packet Loss Rate

When the stream enters the MPQUIC, the system initializes the set of available paths $\mathcal{P}_f$ for a data stream f, as well as prepares an empty two-dimensional matrix $A[\,][\,]$. At first, when $\mathcal{P}_f$ is empty, the system refers to (2), (4) and (5) to determine the values of data stream bandwidth, delay, and packet loss rate, which are stored in $A[\,][\,]$. Then, the system checks a condition of whether the set of available paths for f contains equal to or more than k paths. As this condition is valid, the system proceeds to Algorithm 2 with a set of candidate paths for f. Later soon, Algorithm 2 will figure out k shortest paths to form a set of $\mathcal{P}_f^*$.

Algorithm 2 Finding *k* Shortest Paths over MPQUIC Streams

Algorithm 2 is the path selection procedure for finding the k-shortest paths based on QoS requirements. This procedure refers to Yen's k-shortest path algorithm [29] with QoS-specific conditions. To find the k-shortest paths, the procedure runs several routes sequentially: (a) define variables p_w, b_f^* and $\mathcal{P}_f^*[\,][\,]$, (b) calculate the weight value p_w of a stream by (19), (c) sort the weights of streams in descending order, and (d) update the available bandwidth of each link according to (7) and (11a). Then, the procedural

Algorithm 2 Finding *k* Shortest Paths over MPQUIC Streams

```
Function $getkPath(\mathcal{P}_f, \alpha, \beta, f, k, A)$ is
    $p_w[\,] =$ null;
    $b_f^* = 0$;
    $\mathcal{P}_f^*[\,][\,] =$ null;
    foreach $p \in \mathcal{P}_f$ do
        $p_w[p] \leftarrow getPathWeight(\mathcal{P}[p], \alpha, \beta, A)$    ▷ (19)
    end foreach
    $p_w \leftarrow sortByDescendingOrder(p_w)$ ;
    $\mathcal{P}_f^* \leftarrow selectPathTopk(p_w, k)$ ;
    $b_f^* \leftarrow getMultiPathBW(\mathcal{P}_f^*)$    ▷ (7) and (11a)
    while $b_f^* \leq b_f$ do
        if $minBWPath(\mathcal{P}_f^*) \geq maxBWPath(\mathcal{P}_f - \mathcal{P}_f^*)$ then
            $\mathcal{P}_f^* = \emptyset$
            break;
        end if
        $\mathcal{P}_f^* \leftarrow \mathcal{P}_f^* - minBWPath(\mathcal{P}_f^*)$
        $\mathcal{P}_f^* \leftarrow \mathcal{P}_f^* + maxBWPath(\mathcal{P}_f - \mathcal{P}_f^*)$
        $b_f^* \leftarrow getMultiPathBW(\mathcal{P}_f^*)$    ▷ (7) and (11a)
    end while
    return $\mathcal{P}_f^*$
end
```

routine goes into a while-loop with a condition as b_f^* is smaller than the bandwidth b_f asked by a stream f. If the minimum bandwidth of $\mathcal{P}_f^*$ exceeds the currently available path $\mathcal{P}_f$, $\mathcal{P}_f^*$ is still to be null. Then, the routine updates the set of available paths $\mathcal{P}_f^*$ and the bandwidth b_f^*, remove the path of the smaller bandwidth from $\mathcal{P}_f^*$, add a path with the larger bandwidth, update b_f^*, and then push the value of $\mathcal{P}_f^*$ back to Algorithm 1 to allocate available paths. Eventually, the data flow is passed through those suitable and multiple paths in the current network. To better explore the effects of Algorithms 1 and 2, we will present experiments and performance results in Section 4.

4. Performance Results

This section shows the performance of our proposed method in comparison with QUIC, multipath QUIC LRF [5] and the PQUIC schemes [6].

4.1. Experimental Setting

We conducted experiments on the Mininet simulation platform that runs on a computer equipped with an Intel Core i7 processor, 16GB memory, and Ubuntu 18.04.6 LTS. All the algorithmic programs are coded in C language. We used the Wireshark packet analyzer to trace the data flows during the experiments. The following table 2 defines the simulation parameters used in this paper.

Experiments were divided into three sorts with different sizes per data flow: 100, 200, and 400 MB, and produced three measure results of the overall flow completion time, path

Table 2. Simulation Parameter

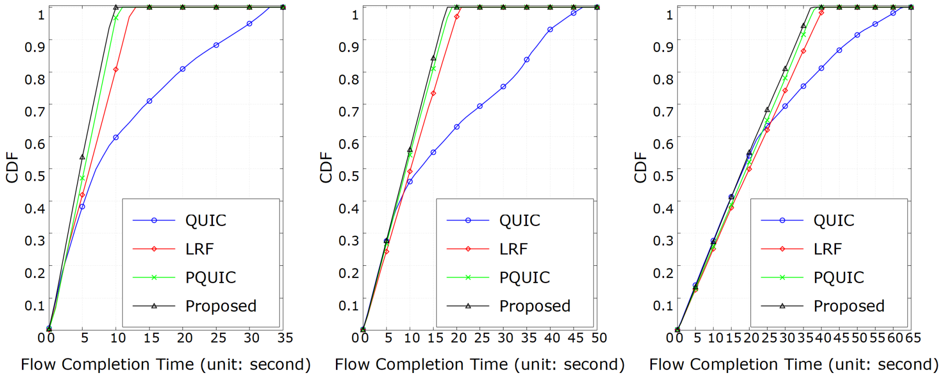
Parameter	Value
Number of nodes in Abilene	11
Number of links in Abilene	14
Data packet size	960-1200 bytes
Transmission bandwidth capacity of a link	100 Mbps
Transmission delay of a link	0-100 ms
Packet loss rate of a link	0.001%
α, coefficient of the measure in 19	0.5
β, coefficient of the measure in 19	0.5
Transmission data size	100MB, 200MB, 400MB

delay, and packet loss rate. We employed the Mininet to adjust simulation parameters. Explicitly, we set $k = 3$, delay coefficient $\alpha = 0.5$ and packet loss coefficient $\beta = 0.5$ as calculating the weighted value p_w. We adopted the Abilene topology [30]: there are 11 nodes and 14 links, the size of each packet is between 960 and 1200 bytes, the path bandwidth is set to 100 Mbps, the delay is from 0 to 100 ms by the binomial distribution, and packet loss rate is set to 0.001%. All experimental cases were run in 20 times to have the results on average.

4.2. Flow Completion Time

Figures 4a, 4b and 4c exhibit the flow completion time in terms of the cumulative distribution function (CDF). As observed, the performance by naive QUIC is the worst, because QUIC only transmits data through a single path, as compared with the other schemes that take multiple paths. Hence, distributing data across multiple paths can obtain better network performance, redundancy, and fault tolerance. It is visible that our scheme outperforms LRF and PQUIC. Explicitly, LRF is based on finding the path with the minimum RTT for transmitting the top-priority data first. Thus, LRF behaves like a greedy way and only focuses on the RTT condition without referring to other network characteristics. The above observations indicate the importance of taking a more comprehensive method for improving network performance.

PQUIC switches between multipaths to ensure that data packets are sent to the receiver fairly. Although PQUIC likely increases the complexity of managing multipath transmissions in dynamic networks, it still suffers from minor performance degradation as path characteristics often change, and as the data size becomes larger. Relatively, our proposed scheme considers both path delay and packet loss rate of path candidates. Such a sophisticated path selection method can lead to better performance to network applications that concern the packet loss. By using a weighting normalization method, it is able to calculate P_w. The higher P_w, the higher priority the data needs to be scheduled for transmission first. This method can dynamically adjust the priorities of data transmission according to network conditions. Our proposed scheme with weighting effects can minimize the flow completion time, resulting in a remarkable comparison with LRF and PQUIC. Thus,



(a) Data size 100 MB **(b)** Data size 200 MB **(c)** Data size 400 MB

Fig. 4. Flow Completion Time

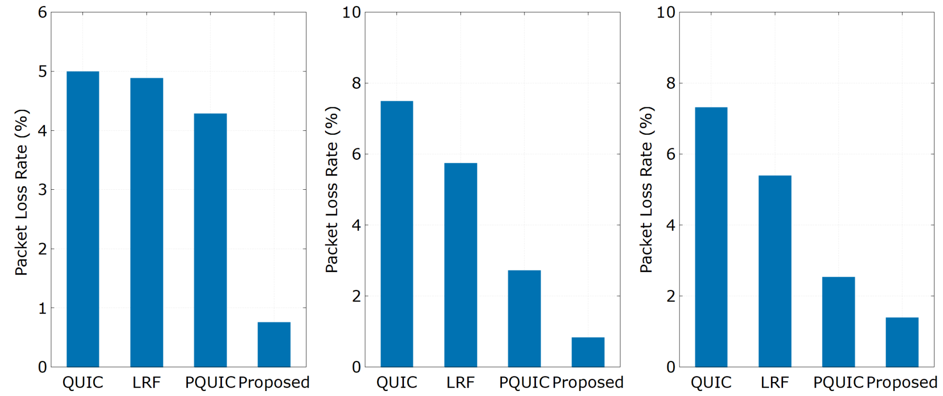
this remarkable result can highlight the importance of intelligent path selection and data prioritization in efficient data transmission and better user experience.

4.3. Packet Loss Rate

Figures 5a, 5b and 5c present the packet loss rate of the overall system performance. As observed, the packet loss rate of QUIC is higher than the other multipath schemes. This is because only the resource allocation of a single path is used, so that the packet loss is apparently affected to a sensitive extent. In the case of data size 100 MB per stream, the packet loss rates of QUIC, LRF, and PQUIC are similar, but become different when the data size per stream increases to 200 MB and even 400 MB. LRF searches for the path of the minimum RTT, which may cause the problem of packet loss in the rear tail of data stream. As examined, this problem is not easy to be resolved when RTT is solely concerned in path selection. PQUIC is fairer as allocating multiple paths to a data stream. Its packet loss rate is lower than the LRF's result. By contrast, our scheme can distribute the data to multiple paths efficiently, thereby being less susceptible to the increase of data size per stream. As seen, our scheme is able to cope with the packet loss rate to be lower than 1 % regardless the increasing data size from 100 MB to 400 MB. Therefore, experimental results provide insights into the relationship between different packet loss rates, data sizes, and transmission schemes. The above findings help in understanding the variance of network performance during the multipath data transmission.

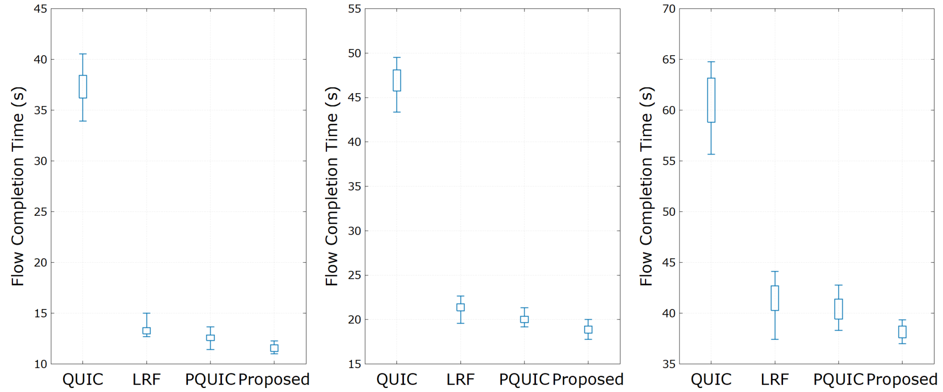
4.4. Overall system stability

Figures 6a, 6b and 6c depict the quartile distribution of flow completion time when the experiment launched 20 data flows one by one repeatedly. Obviously, QUIC needs to take much more time to accomplish the transmission of per data flow. The time gap between QUIC and three multipath QUIC scheme is apparent. This phenomenon shows that employing multiple paths schemes can bring a positive influence on reducing the flow



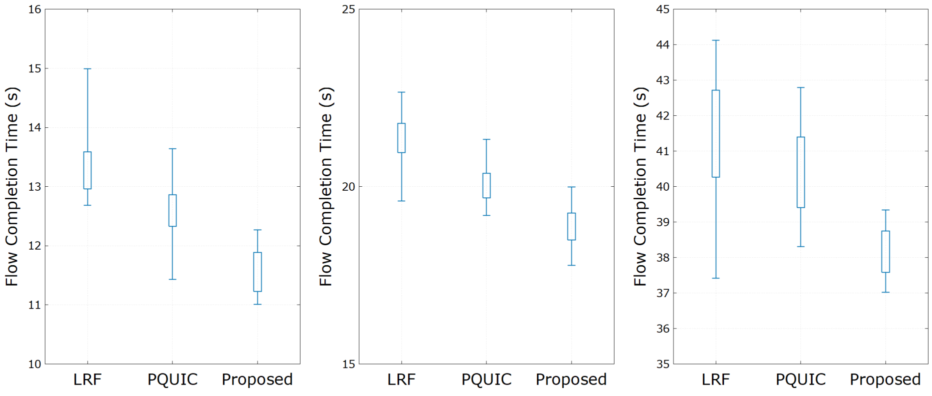
(a) Data size 100 MB **(b)** Data size 200 MB **(c)** Data size 400 MB

Fig. 5. Packet Loss Rate



(a) Data size 100 MB **(b)** Data size 200 MB **(c)** Data size 400 MB

Fig. 6. Overall system stability with QUIC



(a) Data size 100 MB **(b)** Data size 200 MB **(c)** Data size 400 MB

Fig. 7. Overall system stability without QUIC

completion time. Instead, Figures 7a, 7b and 7c exhibit a clear view on the time gap of three multipath QUIC schemes. LRF has not only a larger completion time but also a wider quartile distribution than PQUIC and our scheme. That is, LRF's flow complete time is inconsistent with high variance. We examined that as compared with our scheme, PQUIC cannot perfectly allocate data packets to paths. As the amount of data packets increases rapidly, the probability of head-of-line blocking will increase and then affect the data throughput. Therefore, the results by our scheme are obvious with a minor quartile distribution and the lowest flow completion time. In other words, our scheme can offer stable transport performance since data flows are completed efficiently and with relatively low variability.

5. Conclusion

This paper designs a novel data transport scheme based on MPQUIC. Compared with the traditional network protocol TCP, MPQUIC is based on UDP and keeps the advantages of QUIC from a single-path to multi-path data transport. Our proposed MPQUIC scheme is able to joint sustain transmission delay and packet loss rate with respect to data flows. Performance study is conducted by comparing the proposed scheme with three prior schemes, i.e., QUIC, LRF, and PQUIC. It is remarkable that our proposed scheme performs efficiently and stably in terms of the flow completion time in the system. When the flow completion time is reduced significantly, this scheme also exhibits the effectiveness of reducing path delay and lower packet loss rate under comparative cases with different sizes of data flows.

Our future research will continue to implement MPQUIC and measure the network transport performance in more complicated network scenarios with emerging AR/VR applications, particularly in mobile environments. On the other hand, we notice the adoption of machine learning techniques in internet traffic engineering and management. Our study will further incorporate edge intelligence to network hosts for pro-actively allocating network resources to data flows and streams. Potential effects on network throughput, security, load balancing and user experiences will be investigated.

Acknowledgments. This work was supported in part by the National Science and Technology Council, Taiwan (R.O.C.), under Contracts MOST-109-2221-E-008-051, NSTC-111-2221-E-008-064 and NSTC-111-2410-H-262-001.

References

1. R. Fielding, J. Gettys, J. Mogul, H. Frystyk, L. Masinter, P. Leach, and T. Berners-Lee, "Hypertext transfer protocol–http/1.1," Tech. Rep., 1999.
2. J. Iyengar and M. Thomson, "QUIC: A udp-based multiplexed and secure transport," https://datatracker.ietf.org/doc/html/rfc9000, May 2021, accessed: 2023-7-30.
3. P. Megyesi, Z. Krämer, and S. Molnár, "How quick is quic?" in *2016 IEEE International Conference on Communications (ICC)*. IEEE, 2016, pp. 1–6.
4. C. Raiciu, C. Paasch, S. Barre, A. Ford, M. Honda, F. Duchene, O. Bonaventure, and M. Handley, "How hard can it be? designing and implementing a deployable multipath {TCP}," in *9th USENIX symposium on networked systems design and implementation (NSDI 12)*, 2012, pp. 399–412.
5. T. Viernickel, A. Froemmgen, A. Rizk, B. Koldehofe, and R. Steinmetz, "Multipath quic: A deployable multipath transport protocol," in *2018 IEEE International Conference on Communications (ICC)*. IEEE, 2018, pp. 1–7.
6. Q. De Coninck, F. Michel, M. Piraux, F. Rochet, T. Given-Wilson, A. Legay, O. Pereira, and O. Bonaventure, "Pluginizing quic," in *Proceedings of the ACM Special Interest Group on Data Communication*, 2019, pp. 59–74.
7. R. Marx, J. Herbots, W. Lamotte, and P. Quax, "Same standards, different decisions: A study of quic and http/3 implementation diversity," in *Proceedings of the Workshop on the Evolution, Performance, and Interoperability of QUIC*, 2020, pp. 14–20.
8. S. Ha, I. Rhee, and L. Xu, "Cubic: a new tcp-friendly high-speed tcp variant," *ACM SIGOPS operating systems review*, vol. 42, no. 5, pp. 64–74, 2008.
9. N. Cardwell, Y. Cheng, C. S. Gunn, S. H. Yeganeh, and V. Jacobson, "Bbr: Congestion-based congestion control," *Communications of the ACM*, vol. 60, no. 2, pp. 58–66, 2017.
10. Q. De Coninck and O. Bonaventure, "Multipath quic: Design and evaluation," in *Proceedings of the 13th international conference on emerging networking experiments and technologies*, 2017, pp. 160–166.
11. H. Zeng, L. Cui, F. P. Tso, and Z. Zhang, "Optimizing multipath quic transmission over heterogeneous paths," *Computer Networks*, vol. 215, p. 109198, 2022.
12. V. A. Vu and J. Wolff, "Supporting delay-sensitive applications with multipath quic and forward erasure correction," in *Proceedings of the 17th ACM Symposium on QoS and Security for Wireless and Mobile Networks*, 2021, pp. 95–103.
13. 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.
14. V. A. Vu and B. Walker, "On the latency of multipath-quic in real-time applications," in *2020 16th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*. IEEE, 2020, pp. 1–7.
15. X. Shi, L. Wang, F. Zhang, and Z. Liu, "Fstream: Flexible stream scheduling and prioritizing in multipath-quic," in *2019 IEEE 25th International Conference on Parallel and Distributed Systems (ICPADS)*. IEEE, 2019, pp. 921–924.

16. 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*, 2018, pp. 29–35.
17. 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*, 2020, pp. 572–577.
18. Z. Zheng, Y. Ma, Y. Liu, F. Yang, Z. Li, Y. Zhang, J. Zhang, W. Shi, W. Chen, D. Li *et al.*, “Xlink: Qoe-driven multi-path quic transport in large-scale video services,” in *Proceedings of the 2021 ACM SIGCOMM 2021 Conference*, 2021, pp. 418–432.
19. J. Wang, Y. Gao, and C. Xu, “A multipath quic scheduler for mobile http/2,” in *Proceedings of the 3rd Asia-Pacific Workshop on Networking 2019*, 2019, pp. 43–49.
20. W. Yang, S. Shu, L. Cai, and J. Pan, “Mm-quic: Mobility-aware multipath quic for satellite networks,” in *2021 17th International Conference on Mobility, Sensing and Networking (MSN)*. IEEE, 2021, pp. 608–615.
21. R. S. Mogensen, C. Markmoller, T. K. Madsen, T. Kolding, G. Pocovi, and M. Lauridsen, “Selective redundant mp-quic for 5g mission critical wireless applications,” in *2019 IEEE 89th Vehicular Technology Conference (VTC2019-Spring)*. IEEE, 2019, pp. 1–5.
22. S. Zhang, W. Lei, W. Zhang, Y. Guan, and H. Li, “Congestion control and packet scheduling for multipath real time video streaming,” *IEEE Access*, vol. 7, pp. 59 758–59 770, 2019.
23. 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.
24. E. Quevedo Caballero, M. Donahoo, and T. Cerny, “Fairness analysis of deep reinforcement learning based multi-path quic scheduling,” in *Proceedings of the 38th ACM/SIGAPP Symposium on Applied Computing*, 2023, pp. 1772–1781.
25. S. Lee and J. Yoo, “Reinforcement learning based multipath quic scheduler for multimedia streaming,” *Sensors*, vol. 22, no. 17, p. 6333, 2022.
26. Z. Wang and J. Crowcroft, “Quality-of-service routing for supporting multimedia applications,” *IEEE Journal on selected areas in communications*, vol. 14, no. 7, pp. 1228–1234, 1996.
27. C.-L. Hu, C.-Y. Hsu, and W.-M. Sung, “Fitpath: Qos-based path selection with fittingness measure in integrated edge computing and software-defined networks,” *IEEE Access*, vol. 10, pp. 45 576–45 593, 2022.
28. R. M. Karp, *Reducibility among combinatorial problems*. Springer, 2010.
29. J. Y. Yen, “Finding the k shortest loopless paths in a network,” *management Science*, vol. 17, no. 11, pp. 712–716, 1971.
30. S. Knight, H. X. Nguyen, N. Falkner, R. Bowden, and M. Roughan, “The internet topology zoo,” *IEEE Journal on Selected Areas in Communications*, vol. 29, no. 9, pp. 1765–1775, 2011.

Fang-Yi Lin received the M.S. degree from the Department of Communication Engineering, National Central University, Taiwan.

Wu-Min Sung is currently a Ph.D. student in the Department of Communication Engineering, National Central University, Taiwan. Her research areas are mobile computing and distributed networks.

Lin Hui is currently an associate professor with the department of computer science and information engineering, Tamkang University, Taiwan. Her research interests include machine learning, multimedia applications, and mobile information systems. She has published some journal articles, book chapters, and conference papers related to these research fields. She had served as journal guest editor/reviewer, and program co-chair/chair for many international conferences and workshops.

Chih-Lin Hu received the PhD degree in electrical engineering from the National Taiwan University, in 2003. He was a researcher with BenQ Advanced Technology Center, Taiwan in 2003–2007. In 2008, he joined with the National Central University, Taiwan, and has been a full professor since August 2022. His research interests include mobile and pervasive computing, distributed networks, and Internet of Things.

Nien-Tzu Hsieh is currently a Ph.D. student in the Department of Communication Engineering, National Central University, Taiwan. Her research areas include mobile computing and Internet of Things.

Yung-Hui Chen is currently a full professor and vice-chairman of the Department of Computer Information and Network Engineering, LungHwa University of Science and Technology, Taiwan.

Received: August 18, 2023; Accepted: October 06, 2023.