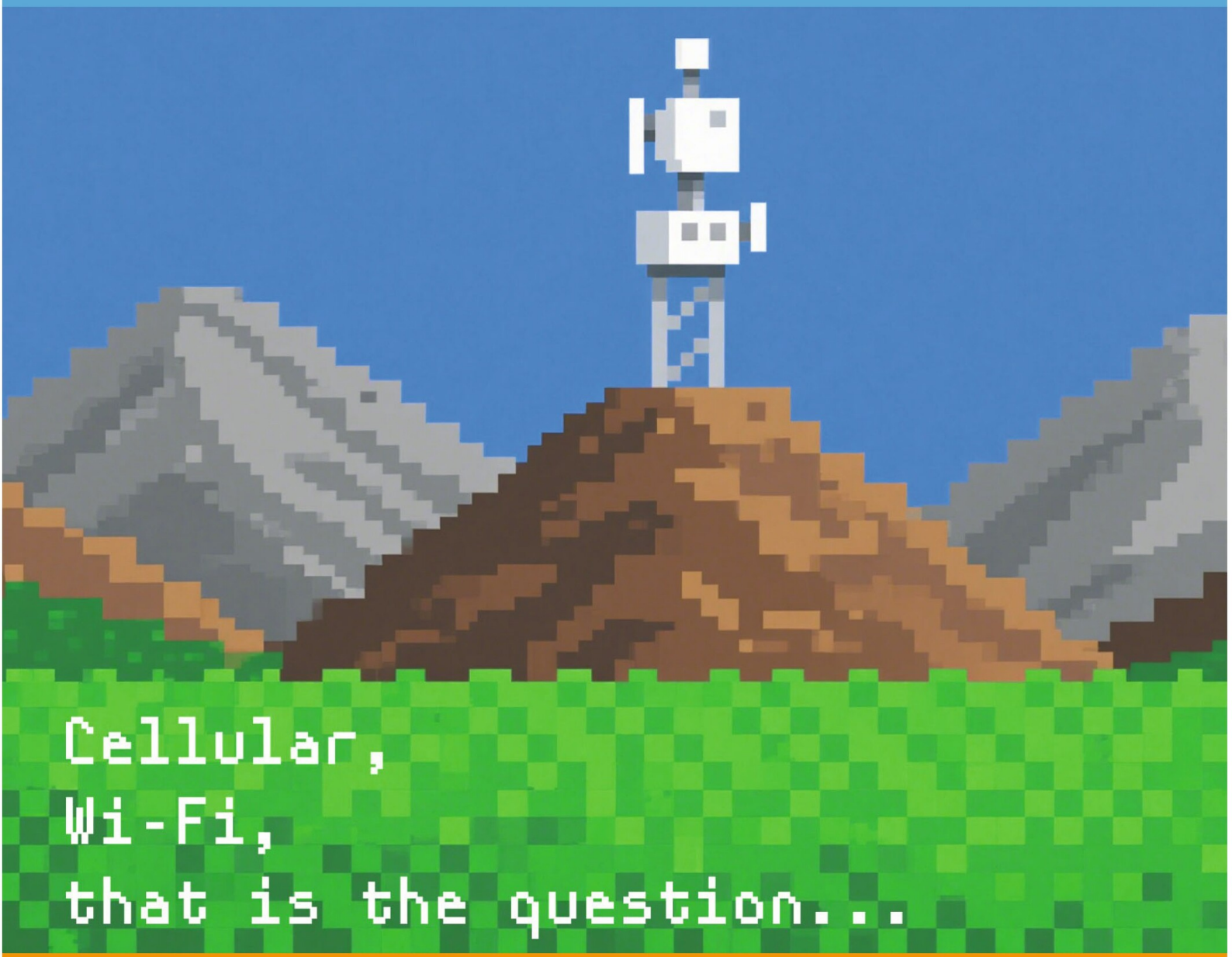
Volume 35 Issue 5 September–October 2025 ISSN 1099-1190

International Journal of

Network Management



WILEY

acm

Received: Added at production | Revised: Added at production | Accepted: Added at production

DOI: xxx/xxxx

Enhancing QUIC Performance in Heterogeneous Networks: A Proactive Connection Migration Approach

Lizhuang Tan[1,2] | Xin Dong[1,2] | Xiaochuan Gao[3] | Peiying Zhang[1,2,4] | Wei Su[5] | James Won-Ki Hong[6]

[1]Key Laboratory of Computing Power Network and Information Security, Ministry of Education, Shandong Computer Science Center (National Supercomputer Center in Jinan), Qilu University of Technology (Shandong Academy of Sciences), Jinan 250014, China

[2]Shandong Provincial Key Laboratory of Computing Power Internet and Service Computing, Shandong Fundamental Research Center for Computer Science, Jinan 250014, China

[3]Bytedance, Beijing 100005, China

[4]Qingdao Institute of Software, College of Computer Science and Technology, China University of Petroleum (East China), Qingdao 266580, China

[5]National Engineering Research Center of Advanced Network Technologies, School of Electronics and Information Engineering, Beijing Jiaotong University, Beijing 100044, China

[6]Department of Computer Science and Engineering, Pohang University of Science and Technology, Pohang 37673, Korea

Correspondence
Lizhuang Tan, Shandong Computer Science Center (National Supercomputer Center in Jinan).
Email: tanlzh@sdas.org

Abstract

The QUIC protocol provides a secure, reliable and low-latency communication foundation for HTTP/3. Connection migration is a key technology of QUIC. When the IP/Port of a connection changes, the connection ID is used to maintain a secure and uninterrupted connection. However, current connection migration is passive, designed to support mobile handover and weak network environments. In this paper, we propose Proactive Connection Migration for QUIC (PCM-QUIC), which combines connection migration and online path selection, enabling QUIC to select the best quality transmission path while maintaining the connection. First, PCM-QUIC integrates the exploration of network quality across different paths into multiple user request actions. Then, considering response completion time and jitter, PCM-QUIC identifies the optimal access path for the current Internet service through online learning. In addition, we propose an Upper Confidence Bound-based path selection algorithm with the goal of minimizing the confidence upper limit of the path reward. The experimental results show that, compared with the original QUIC, PCM-QUIC reduces the average response completion time by up to 59.43%.

KEYWORDS
QUIC, Connection Migration, Heterogeneous Networks, Network Selection, Quality of Service, Performance Management

1 | INTRODUCTION

Different Mobile Service Providers (MSPs) provide varying levels of support for the same Internet services and content. According to our empirical measurement results shown in Table 1, user experience when accessing websites across different MSPs currently varies significantly in mobile networks, primarily due to packet loss and latency caused by commercial arrangements such as inter-MSP settlements. The modern Internet is so complex that it is difficult to pinpoint the exact causes of delay and packet loss. Therefore, it is most effective to improve user experience starting from the client side. By comparing the performance of different access networks with the same service in a heterogeneous network environment[1], providing users with an optimal network becomes a key approach to enhancing the user experience.

TABLE 1 The Average Time to First Contentful Paint of the Same Internet Content by Different MSPs.

FCP / Web \ MSPs	China Mobile	China Unicom	China Telecom
10086.cn	2.46s	**1.95s**	2.20s
10010.com	15.7s	**13.3s**	20.03s
189.cn	1.31s	697ms	**610ms**
google.com	646ms	737ms	**504ms**

QUIC is a UDP-based transport protocol originally designed by Google[2], aiming to provide multiplexed streams over an encrypted transport layer. HTTP/3[3] adopts QUIC instead of TCP as its transport layer protocol[4]. QUIC introduces new features such as low-latency connections, enhanced congestion control, multiplexing without head-of-line blocking, forward error correction, and connection migration, all of which significantly improve Internet transmission efficiency and user experience. In 2021, QUIC was standardized by the IETF[5].

QUIC no longer relies on the five-tuple of IP address and port number to identify a connection. Instead, it uses a 64-bit random number as the connection ID to uniquely identify a transmission. As long as the connection ID remains unchanged, the connection is still considered valid, even if the underlying network changes. In contrast, in HTTP over TCP, any change in the source IP address or port number will result in connection termination. The connection migration feature enables users to seamlessly switch between Wi-Fi and mobile networks while maintaining the upper-layer virtual channel and avoiding losses caused by reconnection[6]. However, the existing connection migration mechanism is passive: QUIC initiates migration only when the user actively changes the network environment. The relevant IETF draft[7] categorizes connection migration into three scenarios: failover, standby, and aggregation mode. Unfortunately, most open-source projects have not yet implemented connection migration.

This paper presents an implementation scheme of Proactive Connection Migration for QUIC (PCM-QUIC) aimed at enhancing user experience. With PCM-QUIC, third-party application services only need to deploy QUIC on the server side, without requiring any additional modifications. PCM-QUIC autonomously performs optimal path exploration and selection on behalf of the user. It is seamlessly integrated into user interactions, such as web browsing or accessing online services, and remains transparent to the user. Experimental results show that PCM-QUIC reduces the average response delay by up to 59.43% compared to the original QUIC. The PCM-QUIC prototype has been open-sourced[8]. An Internet draft on PCM-QUIC has been submitted to the IETF[9], and it is expected to supplement the standard QUIC protocol.

The innovations of this paper include the following.

1. We propose a Proactive Connection Migration (PCM-QUIC) mechanism for QUIC that addresses the gap in optimizing performance in HetNets through active path exploration. PCM-QUIC, connection migration, and MP-QUIC[10] jointly promote QUIC to achieve the most efficient transmission in heterogeneous network environments.
2. We develop a path exploration model based on the multi-armed bandit (MAB)[11] framework that incorporates both response delay and jitter. Depending on specific application requirements, PCM-QUIC can select optimal paths across multiple dimensions.
3. We introduce a Path Selection Algorithm (PSA) based on the Upper Confidence Bound (UCB)[12], which aims to minimize the cumulative response delay and jitter. PSA addresses a limitation of the traditional MAB model, which is optimized solely for cumulative rewards.
4. We have implemented and open-sourced a PCM-QUIC prototype[8]. Its performance has been validated in real-world network environments. The source code is publicly available and can be integrated into commercial applications.

The remainder of this paper is organized as follows. Section 2 introduces the background and related work relevant to PCM-QUIC. Section 3 presents the detailed design of PCM-QUIC, including the protocol and prototype. Section 4 presents the derivation of the path selection model and describes the PSA algorithm in detail. Section 5 provides the implementation and experimental results that demonstrate the feasibility of PCM-QUIC. Section 6 discusses several issues related to PCM-QUIC, including application scenarios, deployment considerations, economic implications, and security. Section 7 concludes the paper and outlines directions for future work. Part of this work[13] was published in ACM/EAI MobiQuitous 2020 (DOI: 10.1145/3448891.3448900). In this journal article, we have extended our earlier work by refining the connection migration model and providing the corresponding background, solution, implementation, and evaluation.

2 | BACKGROUND AND RELATED WORK

In this section, we introduce the background and prior research that motivate and contextualize PCM-QUIC.

2.1 | Background

In the era of mobile Internet, communication devices support multiple network access technologies, such as Wi-Fi and cellular networks. Cellular networks encompass various communication standards, including 4G LTE and 5G.[†]

[†] LTE and 5G are categorized as Wide Area Network (WAN) technologies, whereas Wi-Fi is considered a Local Area Network (LAN) wireless technology. Although Wi-Fi and cellular communication overlap to some extent in their application scenarios, they are largely complementary, enabling seamless network connectivity. In 2019, the Next Generation Mobile Networks Alliance (NGMN) and the Wireless Broadband Alliance (WBA) jointly announced efforts to promote the integration of 5G and Wi-Fi technologies.

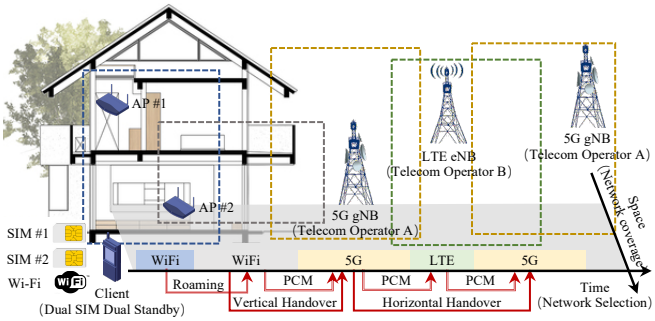


FIGURE 1 Conceptual diagram of the network handover/-connection migration.

As illustrated in Figure 1, the mobile device (client) operates within a heterogeneous converged network environment, which is constantly changing due to factors such as user mobility, signal coverage, and channel scheduling. Wi-Fi and cellular networks have adopted various seamless handover techniques to ensure uninterrupted connectivity. Based on the network types involved before and after the switch, handover scenarios can be categorized into the following three types:

- **Wi-Fi → Wi-Fi**: The handover between different access points (APs) within the same Wi-Fi network is typically achieved through seamless roaming and Mobile IP. The associated switching delay is usually negligible.
- **Wi-Fi → Cellular**: This is a typical vertical handover in HetNets, where the operating system (OS) or the user initiates the network selection and handover. Wi-Fi is generally prioritized over cellular networks. The handover delay is typically very short, usually within 300 ms.
- **Cellular → Cellular**: Seamless handover within cellular networks can be classified as either horizontal or vertical. Horizontal handovers include hard, soft, and relay handovers. In HetNets, a device may connect to cellular networks using multiple standards and MSPs. Cross-MSP handovers are also referred to as cellular-to-cellular handovers. This type of handover often incurs significant delay, as different MSPs must share the same antenna and wireless baseband on the mobile device.

The three types of handover described above can be broadly categorized into vertical and horizontal handovers within HetNets, all aiming to maintain continuous connectivity for mobile devices.

Network handover is a concept that operates at the IP layer and below in the OSI model. In contrast, connection migration operates at the transport layer and above. Connection migration

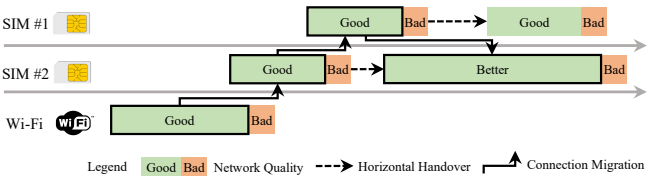


FIGURE 2 Proactive connection migration for QUIC.

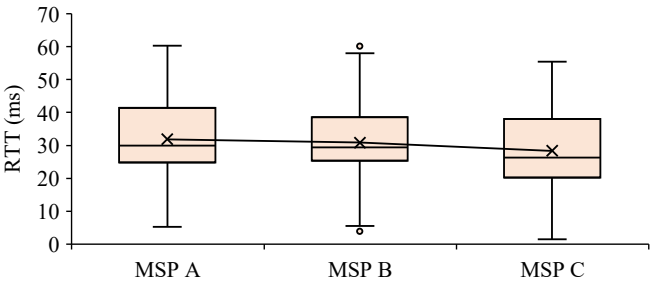


FIGURE 3 RTT statistics result.

is not aware of the current IP address or its associated network. It is solely responsible for maintaining connection continuity between the new IP/port pair after the network handover and the original IP/port pair before the handover.

As illustrated in Figure 2, our approach aims to decouple connection migration from network handover, leverage the redundancy of heterogeneous networks, and enhance transmission performance and reliability. The choice of access network is made at the transport layer rather than at lower layers. PCM-QUIC proactively guides the user connection to the most suitable access network available in the current environment.

We present a comprehensive performance measurement and analysis across different MSPs. We measure the round-trip time (RTT) from 31 provinces in mainland China to a cloud server located in Langfang, Hebei, involving three MSPs: China Mobile, China Unicom, and China Telecom. For clarity, we refer to these providers as A, B, and C. The results are shown in Table 2, and the overall average performance across MSPs is relatively similar, as illustrated in Figure 3. However, in 16 provinces, the RTT performance difference between the three MSPs exceeded 50%, and in 5 provinces, it exceeded 100%. These findings highlight the potential performance gains that proactive connection migration between MSPs can achieve.

2.1.1 | HetNets

A heterogeneous network (HetNet) refers to a multi-protocol communication environment[14]. Unlike the horizontal handoff (HHO) mode used in homogeneous wireless networks, the

TABLE 2 RTT (ms) measurement result.

Province	MSP A	MSP B	MSP C
Shanghai	27.7 (+5.3%)	26.3	26.3
Yunnan	49.2 (+1.2%)	49.7 (+2.3%)	48.6
Inner Mongolia	13.9	14.7 (+5.8%)	14.3 (+2.9%)
Beijing	5.18 (+270.0%)	3.86 (+175.7%)	1.4
Jilin	27.1 (+11.1%)	27.0 (+10.7%)	24.4
Sichuan	35.8 (+6.6%)	35.4 (+5.3%)	33.6
Tianjin	9.11 (+71.2%)	5.45 (+2.5%)	5.32
Ningxia	31.3 (+55.0%)	28.1 (+39.1%)	20.2
Anhui	27.5 (+29.7%)	25.3 (+19.3%)	21.2
Shandong	28.2 (+54.5%)	27.9 (+58.5%)	17.6
Shanxi	24.8 (+73.4%)	32.2 (+125.2%)	14.3
Guangdong	42.3 (+11.3%)	38.6 (+1.6%)	38.0
Guangxi	47.8 (+7.9%)	44.3	45.6 (+2.9%)
Xinjiang	60.3 (+14.6%)	58.0 (+10.2%)	52.6
Jiangsu	28.0 (+6.9%)	28.1 (+7.3%)	26.2
Jiangxi	29.9 (+5.6%)	29.1 (+2.8%)	28.3
Hebei	13.6 (+45.6%)	9.36 (+0.2%)	9.34
Henan	18.4 (+1.6%)	18.1	18.4 (+1.6%)
Zhejiang	33.9 (+1.5%)	33.9 (+1.5%)	33.4
Hainan	50.9 (+10.9%)	47.3 (+3.0%)	45.9
Hubei	24.8 (+2.1%)	26.6 (+9.5%)	24.3
Hunan	33.0 (+11.5%)	29.6	33.3 (+12.5%)
Gansu	32.0 (+18.5%)	29.4 (+8.9%)	27.0
Fujian	44.1 (+2.1%)	43.5 (+0.7%)	43.2
Tibet	60.2 (+8.5%)	60.2 (+8.5%)	55.5
Guizhou	41.4 (+0.9%)	44.0 (+7.3%)	41.0
Liaoning	24.1 (+18.7%)	21.4 (+5.4%)	20.3
Chongqing	33.6 (+2.8%)	32.7	34.1 (+4.3%)
Shaanxi	23.7 (+6.8%)	22.2	23.4 (+5.4%)
Qinghai	36.1 (+33.2%)	31.9 (+17.7%)	27.1
Heilongjiang	29.0 (+18.0%)	33.0 (+34.1%)	24.6

interface switching between different communication systems in HetNets is referred to as vertical handoff (VHO)[15].

In the 5G era, mobile communication networks are envisioned as HetNets incorporating multiple access technologies—such as Wi-Fi, 5G, LTE, and UMTS—supported by both macro and micro base stations to ensure comprehensive signal coverage. MSPs have started exploring the deployment of ultra-dense HetNets to enable efficient operation of low-power access nodes. Ultra-dense HetNets can significantly enhance both power efficiency and spectrum efficiency[16].

HetNets enhance system capacity and user experience by leveraging multi-dimensional diversity techniques at the physical layer[17]. Gai et al.[18] formulated the fundamental problem of multiple secondary users contending for opportunistic spectrum access across multiple channels in cognitive radio networks as a decentralized multi-armed bandit (D-MAB) problem. Their objective was to design distributed online learning policies that minimize regret. Kaori et al.[19] proposed an extended multi-armed bandit algorithm utilizing continuous-valued rewards, applicable to cognitive wireless communication systems with overlapping channels. Cao et al.[20] applied online learning methods to study wireless spectrum management and channel selection. Zhou et al.[21] developed an online learning algorithm for dynamic channel allocation based on contextual multi-armed bandit (CMAB) theory. Huang et al.[22] investigated the coexistence of LTE and Wi-Fi in unlicensed bands using an optimization formulation aimed at minimizing LTE's adverse impact on Wi-Fi users.

However, HetNets can only optimize network performance at the access network level and cannot facilitate end-to-end path selection and performance optimization. In contrast to the performance optimization schemes in HetNets, QUIC operates at a higher level within the network architecture and provides a more comprehensive view of the end-to-end network status. Therefore, as networks evolve, the connection migration mechanism in QUIC will address the limitations of HetNets in end-to-end path selection.

2.1.2 | Multi-path Transmission in Industry

In industry, Apple introduced a multi-path, multi-protocol network operation mechanism known as multipathService[23] in 2017. It is an MPTCP-based service designed to provide seamless handover between Wi-Fi and cellular networks, thereby maintaining connection continuity. It defines four service types: none, handover, interactive, and aggregate.

Huawei Link Turbo[24] is an early client-side solution. It does not require any modifications to application-layer or transport-layer protocols. Based on real-time detection and prediction of current network quality, the system intelligently assigns application requests to the most suitable network. Link Turbo also supports dual-network concurrency to enable rapid switching between networks when quality degrades or when multiple network requests are transmitted simultaneously. This improves service continuity and enhances the overall user experience.

However, both multipathService and Link Turbo require application developers to restructure business logic and implement custom strategy algorithms. The development overhead is substantial, making widespread adoption difficult to achieve in the short term.

2.2 | Related Work

The basic concepts of QUIC have been introduced in Section 1. In this section, we present recent advancements in connection migration, along with related research.

2.2.1 | Connection Migration

The connection migration of QUIC allows a client to change its IP address during an active connection without the need to reestablish it. Connection migration in QUIC involves sending a lightweight token for path validation, offering substantial performance benefits over protocols like TCP, which lack migration support. Govil et al.[25] leveraged connection migration to protect user identity and thwart traffic analysis attacks. Puliafito et al.[26] proposed server-side QUIC connection migration to support microservice deployments.

Tan et al.[13] previously proposed the concept of proactive connection migration, which represents the conference version of this work. Kim et al.[27] used either transmission error events or handover detection timers as migration triggers to support handover in wireless and mobile networks. These two studies represent the only known efforts to proactively improve QUIC performance through connection migration.

The original objective of PCM-QUIC is aligned with that of traditional connection migration—both aim to optimize user experience. However, PCM-QUIC adopts a different triggering mechanism: it relies on proactive path exploration, whereas traditional connection migration is based on passive handover. A combination of PCM-QUIC and conventional connection migration can provide the most efficient solution for maintaining seamless connectivity and improving performance.

2.2.2 | Multi-path QUIC

Multi-path transmission is a common approach to improving throughput, latency, and user experience for mobile terminals in heterogeneous network environments, with Multipath TCP (MPTCP)[28] being a representative example. Researchers have proposed various MPTCP encoding[29] and scheduling[30] strategies to address issues such as packet reordering and receiver buffer overflow caused by path heterogeneity. However, in real-world network deployments, improper traffic scheduling, congestion control, retransmission mechanisms, and sub-path establishment and management can significantly degrade multi-path transmission performance[31]. In some cases, multi-path transmission may even underperform compared to single-path transmission[32].

De Coninck et al.[10] proposed MP-QUIC, which incorporates a path scheduler and a stream framer for packet transmission. The path scheduler employs the shortest round-trip time (SRTT) priority algorithm, while the stream framer adopts either a strict priority or round-robin (RR) scheme to select frames from different application streams.

The existing flow management mechanism in QUIC, characterized by the separation of packet numbers from data offsets, is already sufficient to satisfy most application requirements. This separation facilitates flexible packet scheduling and effectively prevents queue congestion at the receiver. Therefore, incorporating multi-path transmission capabilities into the current QUIC design may not be necessary.

In particular, MP-QUIC performs poorly with small streams, as repeated QUIC handshakes and HTTP/2 requests can significantly increase completion time. Viernickel et al.[33] demonstrated that as the RTT difference between paths increases, the performance of MP-QUIC degrades significantly, approaching that of single-path QUIC. Tong et al.[34] analyzed the impact of the number of sub-streams on MP-QUIC throughput. They found that with two sub-streams, MP-QUIC underperforms compared to TCP, whereas with six sub-streams, it outperforms TCP. Therefore, deploying MP-QUIC on top of existing QUIC to achieve the expected performance gains still presents several challenges.

Mogensen et al.[35] proposed selective redundant MP-QUIC (SRMP-QUIC), an extension of MP-QUIC that enhances the reliability of high-priority services by selectively replicating data packets with strict priority requirements. SRMP-QUIC replicates and transmits data over two LTE networks, achieving a 99.9th percentile delay for critical traffic that is five times lower than that of single-path transmission and three times lower than that of MP-QUIC. While redundant transmission can offer low latency and high reliability, it also comes at the cost of reduced throughput efficiency.

In summary, QUIC transmission optimization has drawn inspiration from MPTCP, aiming to improve performance through multi-path transmission[36]. However, multi-path transmission introduces significant implementation complexity, and numerous factors must be considered in decision-making, which may hinder the expected performance gains[37]. Notably, QUIC already supports multiplexing without head-of-line blocking. Multiple independent streams can be transmitted over a single QUIC connection, and packet loss in one stream does not impact the performance of others.

Unless the service's throughput requirements exceed the available network bandwidth, multi-path QUIC is unlikely to outperform single-path QUIC. Therefore, for most existing network services, single-path QUIC is sufficient to meet latency and bandwidth requirements. Future improvements to QUIC should focus on gaining a comprehensive understanding of the

network environment and selecting the optimal transmission path for the user.

2.2.3 | Online Path Selection

Online learning, game theory, and reinforcement learning have proven to be effective approaches for optimal path exploration and selection in HetNets[38].

Tran et al.[39] proposed a QoE-based server selection algorithm within a CDN architecture. By incorporating realistic characteristics of the server selection process, they formulated the problem as a sequential decision task, which was addressed using the multi-armed bandit (MAB)[40] framework. This approach yielded significant improvements in user-perceived quality compared to traditional strategies such as Fastest, Closest, and Round Robin.

Wu et al.[41] introduced the concept of online learning for traffic-aware network selection. They modeled the problem as a continuous-time multi-armed bandit (CT-MAB) scenario, which matches typical user traffic types to their corresponding optimal networks based on QoE.

Awad et al.[42] formulated a multi-objective optimization problem (MOP) for optimal radio access network (RAN) selection, taking into account user-specific objectives and time-varying network conditions. Their model enables each user equipment (UE) to independently select one or more RANs for simultaneous use and to determine the appropriate data allocation across the selected networks.

Du et al.[43] proposed a second-order reinforcement learning algorithm for end-to-end online routing, which considers both path reward and its variability. Their method adapts to changing network conditions while optimizing long-term performance.

These studies collectively demonstrate that online learning is a viable and effective approach for decision-making in network path selection.

Additionally, Ma et al.[44] proposed a network selection algorithm based on evolutionary game theory, termed NS-EG. This method employs the analytic hierarchy process (AHP) to jointly evaluate user preferences and service requirements in dynamic network environments.

The aforementioned approaches offer valuable insights for path selection in PCM-QUIC. PCM-QUIC incorporates and optimizes these strategies within its actual protocol implementation. Specifically, PCM-QUIC integrates end-to-end response delay and jitter into a unified metric, providing a more comprehensive and fine-grained basis for path selection decisions.

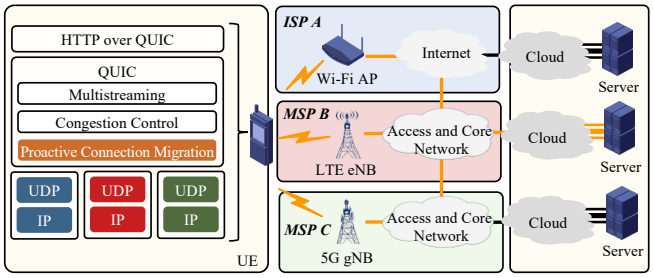


FIGURE 4 QUIC architecture including PCM.

3 | DESIGN OF PCM-QUIC

In this section, we provide a detailed introduction to the design goals, technical requirements, system architecture, network quality evaluation, and interaction workflow of PCM-QUIC.

3.1 | Design Goals

As shown in Figure 4, it is assumed that the UE has three available network access options[45]. The first is direct Internet access via the Wi-Fi network of ISP A, which is generally suitable for most Internet applications. The second and third options involve accessing the Internet through the cellular networks of MSP B and MSP C, respectively. Prior to initiating a service request, the UE has no awareness of the connectivity or network quality between the selected access network and the service provider's server, making it difficult to predict the resulting performance.

As a transport-layer mechanism, PCM-QUIC is designed to provide smoother and more reliable connectivity than existing network handover and selection schemes.

1. **Reliability**: PCM-QUIC should preserve the reliability guarantees of QUIC connection migration to ensure that performance optimization does not come at the expense of stability.
2. **Compatibility**: PCM-QUIC should maintain compatibility with the existing QUIC protocol by minimizing changes to interaction procedures, security negotiation, and connection management.
3. **Efficient utilization of HetNet resources**: PCM-QUIC should leverage the connection migration mechanism together with the path redundancy inherent in HetNets to enhance transmission efficiency while avoiding service interruptions.

3.2 | Network Quality Evaluation

Network quality can be evaluated using a variety of metrics, including received signal strength (RSS) at the physical layer, delay, jitter, and packet loss rate at the network layer, and task completion time (TCT) at the transport layer.

In this work, we select TCT as the primary metric for network quality evaluation. The rationale for this choice is as follows:

1. **User-centric perspective**: As a higher-level metric, TCT more directly reflects the user experience compared to low-level metrics such as RSS, delay, jitter, and packet loss rate.
2. **Ease of measurement**: TCT is simpler and more efficient to measure, as it does not require access to physical-layer or network-layer statistics.
3. **Stability**: TCT tends to exhibit lower variability than metrics like RSS, delay, or jitter, thereby reducing the likelihood of unnecessary connection migrations due to transient fluctuations.

TCT is primarily influenced by two factors: network quality and task size:

$$TCT = F(\text{NetworkQuality}, \text{TaskSize}) \tag{1}$$

In general, better network quality and smaller task size lead to shorter task completion times. Network quality itself can be characterized by bandwidth, delay, and packet loss rate:

$$\text{NetworkQuality} = F(\text{Bandwidth}, \text{Delay}, \text{LossRate}) \tag{2}$$

Higher bandwidth, lower delay, and reduced packet loss generally indicate better network quality. To quantify TCT, we record the establishment time (*EstTime*) and end time (*EndTime*) of each transmission task.

Specifically, *EstTime* denotes the time at which the stream is created, initialized by the *Create Stream* operation. *EndTime* represents the moment when the task is completed, defined as the time at which a frame with the FIN flag set to 1 is received.

For a set of fixed-size tasks, TCT can be used as an effective metric to evaluate the performance of the network responsible for handling these tasks.

$$TCT = EndTime - EstTime. \tag{3}$$

For multiple tasks of varying sizes, we introduce the task completion time per byte (TCTPB) as a metric to evaluate the performance of the network handling these tasks:

$$TCTPB = \frac{(EndTime - EstTime)}{TotalBytes} \tag{4}$$

By combining Eq.3 and Eq.4, the task completion time for each individual request can be accurately calculated and compared across different network paths.

Understanding the variation in TCT is critical for the design and optimization of PCM-QUIC. To this end, we measured the TCT for a mobile device downloading a 1 MB resource from a remote server under four representative network environments: indoor, walking, subway, and high-speed rail. These scenarios cover the majority of real-world situations in which users access Internet services on mobile devices. We conducted separate measurements for three network types: Wi-Fi (CERNET), LTE #1 (China Telecom), and LTE #2 (China Mobile), across the four usage scenarios described above. In total, We collected 9 sets of data, each of which consisted of 360 task completion times lasting 15 minutes.

Figure 5 presents the measurement results described above. Several noteworthy observations can be made:

1. The task completion time of the three networks in indoor environments is relatively stable, but the LTE delay is relatively high. The TCT of LTE #1 and LTE #2 is 1.9 times and 2.1 times that of Wi-Fi, respectively. The average LTE TCT of indoor environment is about 2.3 times that of outdoor environment.
2. Surprisingly, the average task completion time in the subway environment is better than in the outdoor walking scenario. This may be attributed to targeted network optimizations implemented by MSPs specifically for subway systems.
3. In high-speed rail environments, both failure and retransmission rates are considerably higher, indicating that current operator networks are not yet well-optimized for high-speed mobility. The proportion of tasks with completion times exceeding 1000 ms is substantially greater than in subway and walking scenarios.
4. When accessing the same Internet service via different MSPs, users may experience varying task completion times even at the same time and location. Moreover, in the subway, walking, and high-speed rail scenarios, TCT values tend to converge within short time windows, suggesting temporal correlation across network conditions.

We define the relatively consistent task completion times observed over short periods in indoor or stable outdoor environments as stationary task completion times. In contrast, task completion times exhibiting significant long-term fluctuations—typically observed in dynamic outdoor scenarios—are referred to as non-stationary task completion times.

PCM-QUIC should be capable of handling both stationary and non-stationary task completion times within a unified algorithmic framework. To achieve this, we adopt a short-term

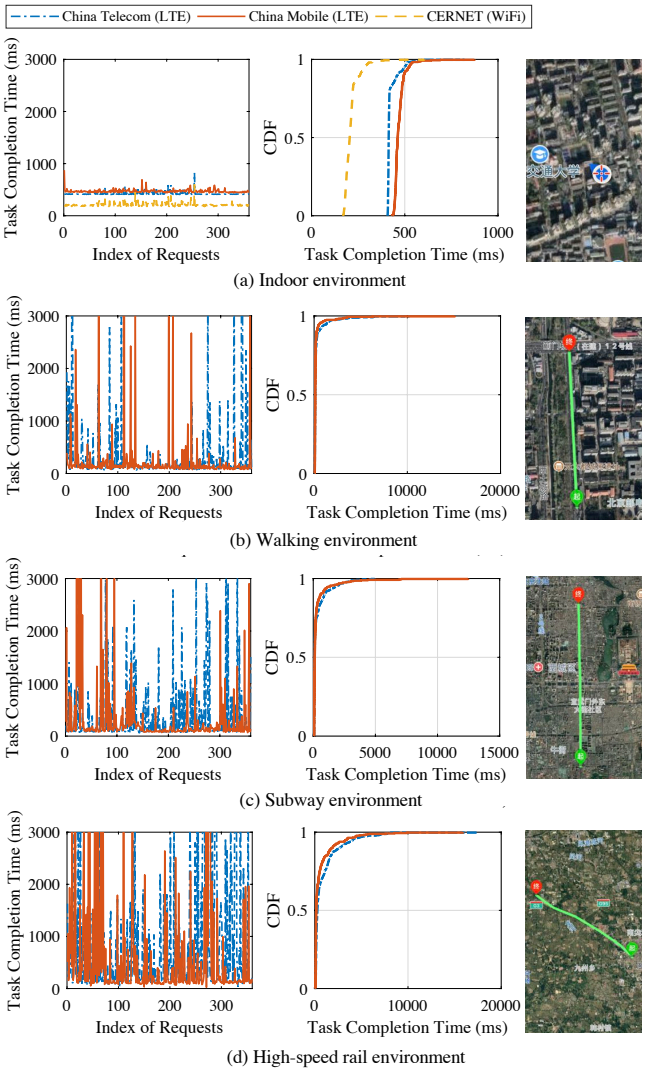


FIGURE 5 The measurement results of task completion time in different scenarios.

memory mechanism that incorporates a limited window of recent task completion time records into the path selection algorithm, while actively discarding outdated measurements. This design enables PCM-QUIC to adapt more effectively to both stable and dynamic network conditions.

3.3 | Technical Requirements of PCM-QUIC

The PCM-QUIC mechanism requires support from the mobile terminal, operating system (OS), and the QUIC protocol itself.

First, PCM-QUIC relies on the physical capabilities of mobile terminals. Most modern smartphones support dual-SIM dual-standby or eSIM functionality, enabling simultaneous access to the Internet via multiple MSPs. Combined with the widespread availability of Wi-Fi, users typically have at least three independent network access options.

However, due to baseband hardware limitations, switching the default SIM card in dual-SIM devices often results in the temporary suspension of signal transmission and data connectivity. Users must wait for the cellular network to recover. Moreover, because of tariff policies, users typically determine the default SIM card, and neither the OS nor applications are authorized to modify this setting dynamically.

To fully support PCM-QUIC, future baseband upgrades and MSP services may need to enable dual-communication modes through user agreements. In contrast, the coexistence and switching between cellular and Wi-Fi networks is more straightforward. Since these two technologies rely on different baseband chips, seamless switching and concurrent access can be achieved through OS-level support without requiring hardware changes.

Second, PCM-QUIC requires OS-level support. The OS must allow applications to maintain simultaneous access to multiple network interfaces. In native Android, network preference is governed by the network score, with Wi-Fi typically prioritized over LTE. As a result, only one active network is usually permitted at a time.

Furthermore, applications and the kernel can typically maintain only one active local socket per connection. Attempting to reuse the same socket often causes the original connection to be released and re-established, leading to service interruption. However, several customized Android distributions have already addressed this limitation by modifying network scoring policies and redesigning socket handling mechanisms. For instance, Huawei Link Turbo[24] supports concurrent LTE and Wi-Fi connections. Similarly, iOS allows simultaneous use of Wi-Fi and 3G/4G/5G by assigning a static IP to Wi-Fi interfaces.

Third, PCM-QUIC requires enhancements to the QUIC protocol. QUIC must continuously probe and evaluate the quality of multiple available network paths, leveraging the interfaces exposed by the OS, in order to assist users in selecting the optimal access path in real time.

3.4 | Protocol Design

PCM-QUIC extends the standard QUIC protocol by introducing a proactive connection migration mechanism. As described in Section 4, when a user initiates a service request, PCM-QUIC actively rebinds the connection to different UDP sockets in order to explore the quality of multiple network paths.

All PCM-QUIC data packets follow the standard QUIC long header format, which includes a connection ID. This connection ID remains bound to a specific QUIC connection, regardless of changes in the client and/or server IP and port.

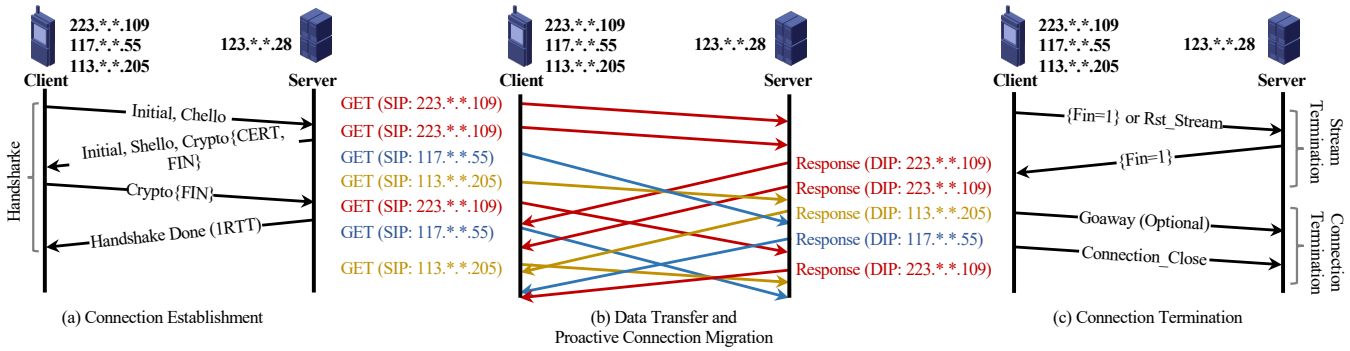


FIGURE 6 The life cycle of QUIC connection.

PCM-QUIC leverages this property by maintaining the same connection ID while modifying the source address (i.e., the UDP socket) used for individual requests. By sending data from different network interfaces, the client can evaluate alternative paths without disrupting the connection. Since all packets retain the same connection ID, the server perceives this as standard connection migration and continues service delivery to the new address.

Importantly, PCM-QUIC requires no modifications to the server-side implementation. Existing QUIC servers are fully compatible with PCM-QUIC, as the mechanism operates entirely at the client side by modifying packet transmission logic and measuring path quality. Furthermore, PCM-QUIC allows reuse of previously negotiated cryptographic keys, eliminating the need for repeated HTTPS handshakes during migration.

As illustrated in Figure 6, the full lifecycle of a QUIC connection consists of three stages: connection establishment, data transmission, and connection termination.

3.4.1 | Connection Establishment

During connection establishment, the QUIC client first retrieves all available IP addresses and excludes loopback addresses. It then randomly selects one IP address to initiate the initial connection.

PCM-QUIC integrates version negotiation and cryptographic setup into the handshake process to minimize connection establishment latency. As part of the handshake, the client and server also negotiate various transport parameters, including the connection ID.

3.4.2 | Data Transfer and Proactive Connection Migration

QUIC provides built-in support for connection reliability, congestion control, and flow control. Its flow control mechanism is tightly aligned with that of HTTP/3. Each QUIC connection maintains a single packet number space, enabling unified congestion control and loss recovery across all streams.

PCM-QUIC builds upon these core mechanisms by introducing proactive connection migration. It dynamically selects source IP addresses using polling, random selection, or the adaptive strategy described in Section 4. For each migration event, PCM-QUIC records the corresponding task completion time, enabling real-time evaluation of path performance.

According to the specifications defined in RFC 9000[5], when a QUIC server receives packets from a new client address, it must verify that the client is able to receive and respond to data on the new path. This is achieved by requiring the client to echo data received from the server, thereby proving its ability to communicate from the new IP address.

Upon successful path validation, the client resets its congestion controller and verifies support for Explicit Congestion Notification (ECN). The server then updates the destination IP address for the next highest-numbered non-probing packet to the new client address. This ensures that subsequent packets are no longer sent to the previous address, thereby avoiding potential packet reordering at the receiver.

In the case of PCM-QUIC, since path changes are initiated by the client immediately after an explicit request to the server, packet reordering is typically not a concern. The server does not receive overlapping packets from both the old and new addresses, which simplifies the migration process.

Skipping path verification can enhance the performance of PCM-QUIC, but it may introduce security risks. To balance performance and security, we adopt a limited address verification scheme. In this approach, the client selects a source IP address from a predefined set of trusted addresses. The server maintains a history of IP addresses previously associated with the client. If the server receives a packet from an IP address that has appeared in the client's history, it can bypasses the standard path validation procedure.

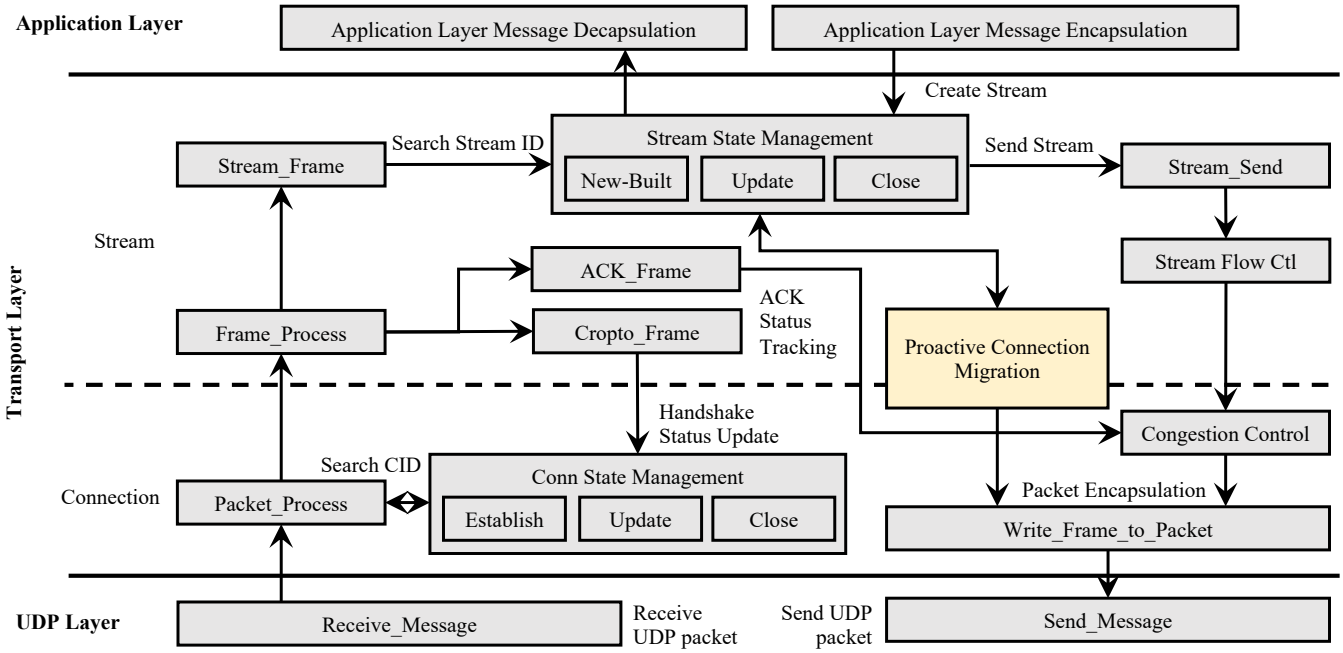


FIGURE 7 Key functional modules including proactive connection migration for QUIC.

3.4.3 | Connection Termination

The stream and connection termination mechanisms in PCM-QUIC are consistent with those defined in the QUIC protocol and can be categorized into three types: normal termination, abrupt termination, and connection termination.

To summarize, the key functional modules of QUIC, including the proactive connection migration mechanism, are illustrated in Figure 7. We introduce the PCM module into the existing QUIC architecture. In order to minimize its intrusion, it helps update the Stream status to record the task completion time corresponding to each Stream and implements network selection in the packet encapsulation stage.

4 | PATH SELECTION MODEL

PCM-QUIC aims to identify the optimal communication path for users as quickly as possible while minimizing exploration overhead. This path selection problem, characterized by a discrete set of candidate paths, can be formulated as a typical multi-armed bandit (MAB) problem[46].

Traditional MAB models aim to maximize cumulative rewards, focusing solely on long-term outcomes while ignoring short-term performance. However, in the context of Internet services, users expect not only low task completion times but also sustained performance stability over time.

As a result, PCM-QUIC prioritizes both the minimization of task completion time and the control of performance fluctuations across different network paths, rather than simply maximizing expected reward. This distinction makes the path selection problem in PCM-QUIC fundamentally different from that of traditional MAB models.

In this section, we formally model the PCM-QUIC path selection problem and propose path selection algorithm (PSA), to address these unique requirements.

We define each complete task lifecycle, from request initiation to completion, as a round. The path selected in each round constitutes an action, and the corresponding task completion time serves as the reward. For detailed parameter definitions, please refer to Table 3.

The path selection strategy of PCM-QUIC can be summarized as the following process. Initially, PCM-QUIC iteratively probes all available network paths in a round-robin manner, collecting performance data for each through task completion time measurements. The cumulative performance of each path is estimated based on the observed task completion times.

After the exploration phase, PCM-QUIC begins exploiting the path with the lowest estimated task completion time, continuing to use it until its performance no longer meets the expected optimality. Each path is assumed to return rewards drawn from a stable probability distribution.

The objective of PCM-QUIC is to minimize the weighted sum of task completion times and their fluctuations over time, thereby achieving both high efficiency and performance stability.

TABLE 3 Notations in the path selection model.

Notations	Description
K	Number of access paths supported by Client
T	Rounds
t	The t-th round
A_t	The actions selected by PCM-QUIC when requested at t
R_t	The task completion time under action A_t
r_k	The average task completion time on k-th path
f_k	The fluctuation (variance) on k-th path
N_k	The cumulative number of times the k-th path has been selected
X_k	task completion time of the k-th path
$\overline{X_k}$	Average task completion time of the k-th path
ρ	The regret after T rounds
μ^*	The optimal task completion time
I_k	The indicator of the k-th path
g	The remaining task completion time, which follow a Gaussian distribution
S	The size of sliding window

4.1 | Probability Distribution of Task Completion Time

It is not feasible to directly evaluate the characteristics of a network path solely based on the observed task completion times. For instance, using the sample average of multiple task completion times on the $k-th$ path to approximate its expected performance is unreliable, as the underlying probability distribution of task completion times is unknown and may not satisfy assumptions such as normality.

As illustrated in Figure 8, we measured eight sets of task completion times corresponding to the same Internet service accessed from different regions and via different MSPs, all within the same time period. Each group consists of 2000 samples collected over a 5-minute window. The empirical distributions of all eight groups exhibit heavy-tailed behavior, with kurtosis values greater than 3.

This phenomenon is primarily attributed to transient network noise, such as queuing delays, which can cause rare but extreme outliers in task completion time.

In order to take care of the stationary task completion time and non-stationary task completion time, we propose a sliding window S. That is to say, PCM-QUIC does not use all the statistical task completion times in history as the basis for decision-making, but uses the latest S task completion time records.

To estimate the distribution of task completion time X_k, PCM-QUIC must account for the fact that it prioritizes minimizing typical latency rather than modeling rare long-tail events. Two common approaches can be considered:

1. **Truncation and Gaussian approximation**: The top $n\%$ of long-tail values in the observed data are removed, and the remaining samples are treated as realizations of a Gaussian random variable.
2. **Lognormal modeling**: The entire distribution of X_k is modeled as a lognormal random variable, which inherently captures the skewed, heavy-tailed nature of the data.

The first approach offers simplicity and computational efficiency, while the second provides a more accurate representation of the underlying distribution.

4.1.1 | Gaussian Model

Let minimum 90% of task completion time $X_k \sim N(\mu_k, \sigma_k^2)$ be a sequence of independent Gaussian random variables with mean μ_k. 90% is an experience value, which means g=10%. The moment generating function of variable X_k is

$$\mathbb{E}[e^{\lambda X_k}] = e^{\mu_k \lambda + \frac{\sigma_k^2 \lambda^2}{2}}, \lambda \in \mathbb{R}. \tag{5}$$

Applying the Cramér Chernoff bound[47], we have

$$\inf_{\lambda>0}\{log\mathbb{E}e^{\lambda(X_k-\mu_k)} - \lambda\epsilon\} = \inf_{\lambda>0}\{\frac{\sigma_k^2\lambda^2}{2} - \lambda\epsilon\} = -\frac{\epsilon^2}{2\sigma_k^2}. \tag{6}$$

We can get the tail probability of the variable X_k:

$$\mathbb{P}[X_k \leq \mu_k - \epsilon] \leq e^{-\frac{\epsilon^2}{2\sigma_k^2}}, \epsilon \geq 0. \tag{7}$$

And the concentration inequality of the variable X_k is

$$\mathbb{P}[|X_k - \mu_k| \geq \epsilon] \leq 2e^{-\frac{\epsilon^2}{2\sigma_k^2}}, \epsilon \geq 0. \tag{8}$$

So, variable X_k is a sub-gaussian random variable with the parameter σ_k. The expectation and variance of a normally distributed random variable X_k are

$$E(X_k) = \frac{1}{N_k}\sum_{i=1}^{N_k} X_{k,i}, \tag{9}$$

$$\begin{aligned} var(X_k) &= \frac{1}{N_k - 1}\sum_{i=1}^{N_k}(X_{k,i} - E(X_k)) \\ &= \frac{1}{N_k - 1}\sum_{i=1}^{N_k}(X_{k,i} - \frac{1}{N_k}\sum_{i=1}^{N_k} X_{k,i}). \end{aligned} \tag{10}$$

In the above equation, $X_{k,i}$ is the i-th value of X_k.

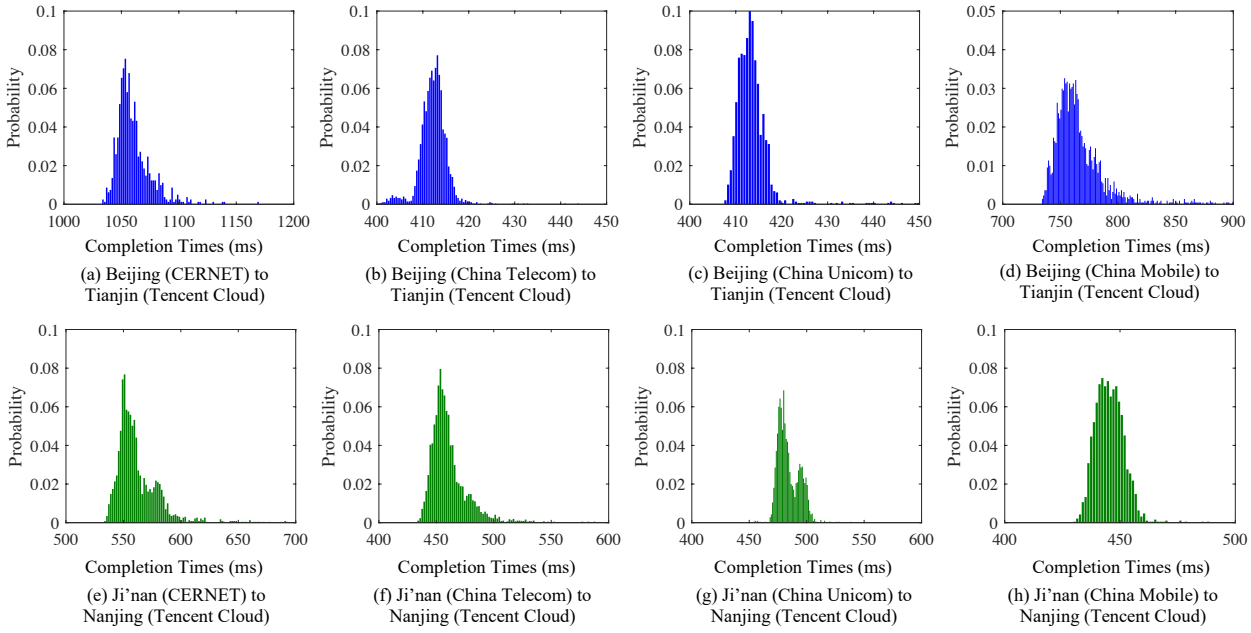


FIGURE 8 Probability distribution of task completion time.

4.1.2 | Lognormal Model

Assuming that X_k is a Lognormal random variable, $\ln(X_k) \sim N(\mu_k, \sigma_k^2)$ is a normal distributed random variable. Given a $x_k > 0$, the probability density function of X_k is

$$f(x_k) = \frac{1}{\sqrt{2\pi}x_k\sigma_k} e^{-\frac{(\ln x_k - \mu)^2}{2\sigma_k^2}}. \tag{11}$$

The Lognormal distribution parameters μ and σ can describe the probability distribution of path k. In order to determine the maximum likelihood estimation of parameters μ_k and σ_k, the same method as the maximum likelihood estimation of normal distribution parameters can be used:

$$f_L(x_k) = \frac{1}{x_k} f_N(\ln(x_k)). \tag{12}$$

Among them, $f_L(\cdot)$ represents the probability density function of Lognormal distribution, and $f_N(\cdot)$ represents the probability density function of normal distribution. Therefore, we can get the maximum likelihood function of x_k:

$$\begin{aligned} l_L(\mu_k, \sigma_k | x_{k1}, x_{k2}, ..., x_{kN_k}) &= -\sum_{i=1}^{N_k} \ln(x_k) \\ &+ l_N(\mu_k, \sigma_k | \ln(x_{k1}), \ln(x_{k2}), ..., \ln(x_{kN_k})). \end{aligned} \tag{13}$$

The maximum likelihood estimations of distribution parameters are

$$\hat{\mu_k} = \frac{\sum_{i=1}^{N_k} \ln(X_{k,i})}{N_k}, \tag{14}$$

$$\hat{\sigma_k^2} = \frac{[\ln(X_{k,i}) - \frac{\sum_{i=1}^{N_k} \ln(X_{k,i})}{N_k}]^2}{N_k - 1}. \tag{15}$$

The expectation and variance of a Lognormal distributed random variable X_k are

$$\begin{aligned} E(X_k) &= e^{\hat{\mu_k} + \frac{\hat{\sigma_k^2}}{2}} \\ &= e^{\frac{\sum_{i=1}^{N_k} \ln(X_{k,i})}{N_k} + \frac{[\ln(X_{k,i}) - \frac{\sum_{i=1}^{N_k} \ln(X_{k,i})}{N_k}]^2}{\frac{N_k - 1}{2}}}, \end{aligned} \tag{16}$$

$$\begin{aligned} var(X_k) &= e^{2\hat{\mu_k} + \hat{\sigma_k^2}}(e^{\hat{\sigma_k^2}} - 1) \\ &= e^{2\frac{\sum_{i=1}^{N_k} \ln(X_{k,i})}{N_k} + \frac{[\ln(X_{k,i}) - \frac{\sum_{i=1}^{N_k} \ln(X_{k,i})}{N_k}]^2}{N_k - 1}} \\ &\cdot (e^{\frac{[\ln(X_{k,i}) - \frac{\sum_{i=1}^{N_k} \ln(X_{k,i})}{N_k}]^2}{N_k - 1}} - 1). \end{aligned} \tag{17}$$

Through the above two models, we can evaluate the expected task completion time and fluctuations of different networks based on statistical values.

4.2 | Evaluation Function

The evaluation on path k consists of two parts. The first part is the expectation of task completion time, which represents the performance of the path. The second part is the fluctuation of the task completion time, which represents the stability of the path. Fluctuation means drastic change in network quality. a and b are the weight parameters of these two parts, and $a + b = 1$. Therefore, the expression of r_k is

TABLE 4 Common application requirements for task completion time, jitter and recommended values of a and b.

Application	TCT	Jitter	Recommended values of a and b
Bulk data transfer	<15s	N/A	a=1, b=0
Command/control	<250ms	N/A	a=1, b=0
Movie clips	<10s	<2s	a=0.5, b=0.5
Online shopping	<2s	<100ms	a=0.8, b=0.2
Realtime games	<75ms	N/A	a=1, b=0
Realtime video	<10s	<2s	a=0.5, b=0.5
Still image	<10s	N/A	a=1, b=0
Surveillance	<10s	<2s	a=0.5, b=0.5
Transaction services	<4s	N/A	a=1, b=0
Videophone	<150ms	<10ms	a=0.2, b=0.8
Voice call	<150ms	<1ms	a=0.2, b=0.8
Voice messaging	<1s	<1ms	a=0.1, b=0.9
Web-browsing	<4s	N/A	a=0.8, b=0.2

$$\begin{aligned} r_k &= aE[X_k] + b \cdot var(X_k) \\ &= aE[X_k] + b\left\{E\left[X_k^2\right] - E^2[X_k]\right\}. \end{aligned} \tag{18}$$

As is shown in Table 4, the values of a and b depend on the upper layer application of the PCM-QUIC.

With reference to the UCB algorithm, we give two path selection strategies, namely optimistic path selection (OPS) and pessimistic path selection (PPS).

In OPS, PCM-QUIC always chooses the path with the most likely smallest r_k. The lower bound confidence for this path is the smallest. The path indicator I_k^{OPS} is

$$\begin{aligned} I_k^{OPS} &= r_k - C\sqrt{\frac{2\ln T}{N_k}} \\ &= aE[X_k] + b \cdot var(X_k) - C\sqrt{\frac{2\ln T}{N_k}}. \end{aligned} \tag{19}$$

In PPS, PCM-QUIC chooses the path with the smallest upper confidence bound. The path indicator I_k^{PPS} is

$$\begin{aligned} I_k^{PPS} &= r_k + C\sqrt{\frac{2\ln T}{N_k}} \\ &= aE[X_k] + b \cdot var(X_k) + C\sqrt{\frac{2\ln T}{N_k}}. \end{aligned} \tag{20}$$

It is worth noting that T in Eq.19 and 20 should be replaced with S, when $T > S$.

When considering its options in round T, the PCM-QUIC has observed N_k samples from path k and received rewards from that path with an empirical of $E[X_k]$ and $var(X_k)$.

$C\sqrt{\frac{2\ln T}{N_k}}$ is path bonus, which means: If the path is selected a few times and the confidence interval is very wide, it will tend to be selected. If the path is selected a lot of times and the confidence interval is very narrow, then the path with a small reward tends to be selected multiple times.

C determines the scope of exploration. The larger the value of C, the more biased toward Breadth First Search (BFS), otherwise the more biased toward Depth First Search (DFS). In the UCB algorithm, the selection of the C value is often an empirical value. The most suitable C value should not interfere too much with the expectation and variance of each path. So, we propose an adaptive C value selection strategy as

$$C = \frac{\sum_{k=1}^{K}\left(aE[X_k] + b \cdot var(X_k)\right)}{K}. \tag{21}$$

4.3 | Accumulated Regret

Since our evaluation of accumulated regret includes TCT and fluctuation, we need to perform a weighted summation of the accumulated regret of task completion time and the accumulated regret of fluctuation to evaluate the effectiveness of the algorithm. The regret ρ after T rounds can be expressed as

$$\begin{aligned} \rho &= \sum_{t=1}^{T} \tilde{r}_t - T\mu^* \\ &= \underbrace{a[\sum_{t=1}^{T} R_t - T \cdot \min(R_t)]}_{\text{The accumulated regret of TCT}} + \underbrace{b\left\{var(R_t) - \min[var(X_k)]\right\}}_{\text{The accumulated regret of fluctuation}}. \end{aligned} \tag{22}$$

4.4 | Path Selection Algorithm

Algorithm 1 describes the pseudo code of the path selection algorithm (PSA). The PSA algorithm consists of two stages. In the first stage (L2-L11), PSA uses each available path in turn to generate the first round of I_k. In the second stage (L13-L30), PSA selects the path with the smallest I_k value for transmission. Since PCM adopts two PSA strategies, namely optimistic and pessimistic, L8-L9 and L27-L28 are calculated according to different formulas.

5 | EXPERIMENT RESULTS

In this section, we evaluate PCM-QUIC from two perspectives: overall performance and sensitivity to parameter settings[‡].

We deploy the experimental topology illustrated in Figure 9. Due to restrictions in the operating systems of existing mobile

[‡] To intuitively demonstrate the effect of PCM-QUIC, readers may refer to our demonstration video at https://www.bilibili.com/video/BV1eaRhY9E4R.

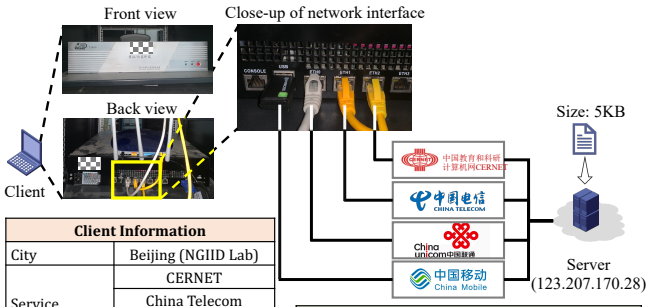
Algorithm 1 Path Selection Algorithm in PCM-QUIC

Require: $a \in [0,1]$, $b = 1-a$, K, T, S
for $t = 1, 2, \ldots, K$ **do**
  $Time_1$ = Time.Now()
  $A_t = t$
  $TaskCompletionTime$ = Time.Now() $- Time_1$
  for $k = 1, 2, \ldots, K$ **do**
    Update N_k
    Update $E[X_k]$ according to Eq.9 or Eq.16
    Update var(X_k) according to Eq.10 or Eq.17
  end for
end for
Calculate C according to Eq.21
for $t = K+1, K+2, \ldots, T$ **do**
  for $k = 1, 2, \ldots, K$ **do**
    $I_{\min} = +\infty$
    Calculate I_k according to Eq.19 or Eq.20
    if $I_k \leq I_{\min}$ **then**
      $PathNumber = k$
    end if
  end for
  $Time_1$ = Time.Now()
  $A_t = PathNumber$
  $TaskCompletionTime$ = Time.Now() $- Time_1$
  for $k = 1, 2, \ldots, K$ **do**
    Update X_k
    Update N_k
    Update $E[X_k]$ according to Eq.9 or Eq.16
    Update var(X_k) according to Eq.10 or Eq.17
  end for
end for

devices, developers do not have privileged access to programmatically control multiple network interfaces. Therefore, we use a Linux server equipped with multiple network interface cards (NICs) to emulate the behavior of a multi-homed QUIC client. This setup does not affect the validity of our experimental results.

The QUIC server is deployed on a cloud instance located in Tianjin, China, hosted by Tencent Cloud. It is responsible for processing incoming requests from QUIC clients. Each client request targets a file of size 5 KB. In our open-source implementation[8], we adopt the Cubic congestion control algorithm. However, due to the small size of the requested file, the impact of the congestion control mechanism is negligible, allowing us to isolate the effect of path selection strategies on performance.

The client accesses the Internet through four distinct network paths, each corresponding to one of the MSPs in the Chinese market. The first path utilizes CERNET, which provides a wired access method. The remaining three are mobile access paths established via personal hotspots enabled on smartphones with SIM cards from different MSPs.



Client Information		
City	Beijing (NGIID Lab)	
Service Provider	CERNET	
	China Telecom	
	China Unicom	
	China Mobile	
System Configuration	CPU	Intel Xeon E3-1270 @ 3.40GHz*8
	RAM	8G, 2.5GT/s
	NIC	Intel 82574L Gigabit*6
	OS	Windows7 (64bit)

Server Information		
City	Tianjin	
Service Provider	Tencent Cloud	
System Configuration	CPU	Intel Xeon E5-26xx v4
	RAM	2GB
	NIC	Virtio network device
	Bandwidth	1Mbps (Public)
	OS	Ubuntu 16.04 (64bit)

FIGURE 9 Experimental environment and topology.

TABLE 5 Statistics of path performance.

Value / Statistic / Path	Ave. RTT	Std. Dev	TTL	Loss
Path-1 (CERNET)	7.120ms	0.446ms	51	0%
Path-2 (China Telecom)	6.707ms	0.636ms	53	0%
Path-3 (China Unicom)	41.004ms	6.458ms	50	0%
Path-4 (China Mobile)	37.568ms	16.756ms	50	0%

To simulate access to different telecom operator networks, the client connects to each mobile phone hotspot through a dedicated wireless router. This setup allows seamless switching between operator networks on the client side without altering hardware configurations.

As shown in Table 5, we use ICMP-based PING measurements to benchmark the baseline performance of these four networks, particularly focusing on RTT. The results clearly indicate performance discrepancies across paths, with significant differences observed in average RTT.

5.1 | Performance of PCM-QUIC

We independently evaluate the impact of four path selection strategies on task completion time: Polling, ε-Greedy, Optimistic PSA, and Pessimistic PSA.

The polling strategy sequentially selects available network paths in a round-robin manner, without adapting to observed task completion times. It serves as a baseline that reflects the average performance across all available paths.

The ε-Greedy strategy primarily selects the path with the best historical performance, while exploring other paths with

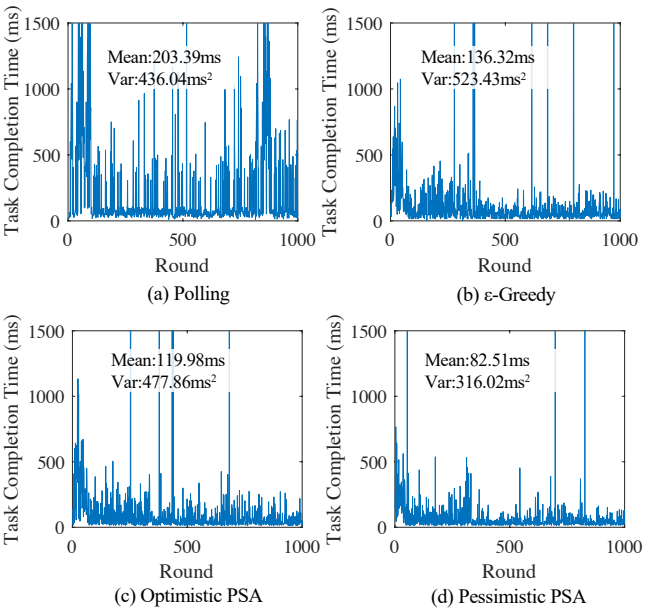


FIGURE 10 The task completion time per round of the four strategies.

a fixed probability ε. This approach offers a stable balance between exploitation and exploration. In our experiment, we set $\varepsilon = 0.05$, a commonly used empirical value recommended in prior studies.

The optimistic PSA and pessimistic PSA choose the path according to Eq.19 or Eq.20 respectively. $S = 100$ and $a = 0.8$. The path expectation and variance are estimated using the Lognormal distribution model.

We evaluate the performance of the three strategies using the following metrics: task completion time per round, cumulative regret, and optimal path selection probability. The reward and regret are computed according to Equation 18 and Equation 22, respectively.

5.1.1 | Task Completion Time Per Round

Figure 10 illustrates the TCT per round for the four path selection strategies.

Compared with the polling strategy, the pessimistic PSA reduces the mean and variance of TCT over 1000 rounds by 41.01% and 9.59%, respectively. When compared to the ε-Greedy strategy, the reductions are 11.99% in mean and 8.71% in variance.

The optimistic PSA outperforms all other strategies. Relative to polling, it achieves a 59.43% reduction in mean TCT and a 27.52% reduction in variance. Compared to ε-Greedy, the improvements are 39.47% and 39.62%, respectively.

The polling strategy, which cyclically utilizes all four paths regardless of their performance variability, exhibits near-periodic behavior in task completion time. In contrast, the

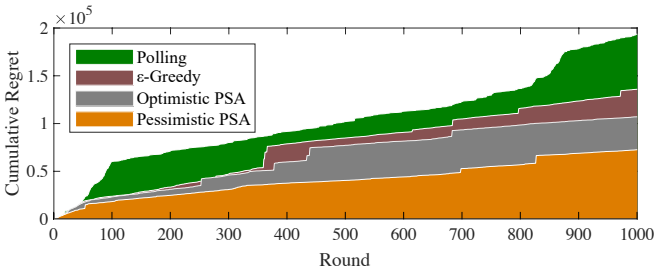


FIGURE 11 Overlapping chart of cumulative regret of four strategies.

ε-Greedy strategy mitigates this periodic fluctuation and shows a general downward trend in TCT due to its adaptive nature.

Both PSA variants further reduce TCT variability by consistently selecting the optimal-performing path. Notably, the pessimistic PSA achieves the lowest overall TCT among the four strategies. This result is attributed to the fact that Path-1 and Path-2 have similar performance levels, and the optimistic PSA is more likely to switch between them, while the pessimistic PSA tends to remain on the more stable path, thereby reducing fluctuations.

5.1.2 | Accumulated Regret

Cumulative regret is a key metric for evaluating the efficiency of path selection in PCM-QUIC. As shown in Figure 11, the cumulative regret of the polling strategy is orders of magnitude higher than that of the other three strategies, indicating its lack of adaptability to dynamic network conditions.

Both optimistic PSA and pessimistic PSA achieve significantly lower cumulative regret compared to polling and ε-Greedy strategies. This demonstrates that the PSA algorithms are more effective in approaching the optimal path selection policy by balancing exploration and exploitation.

5.1.3 | Probability of Selecting the Best Path

Figure 12 illustrates the probability that each strategy selects the optimal path during each round. Under the current experimental conditions, Path-2 consistently yields the best performance, as evidenced by the fact that the latter three strategies predominantly converge on selecting this path.

The polling strategy, by design, selects each of the four paths with equal probability, resulting in a constant 25% hit rate for the optimal path. The ε-Greedy strategy eventually achieves a 91.3% hit rate, limited by its fixed exploration probability ε, which occasionally prevents it from choosing the optimal path.

In contrast, both optimistic PSA and pessimistic PSA rapidly converge to near 100% hit probability, demonstrating their

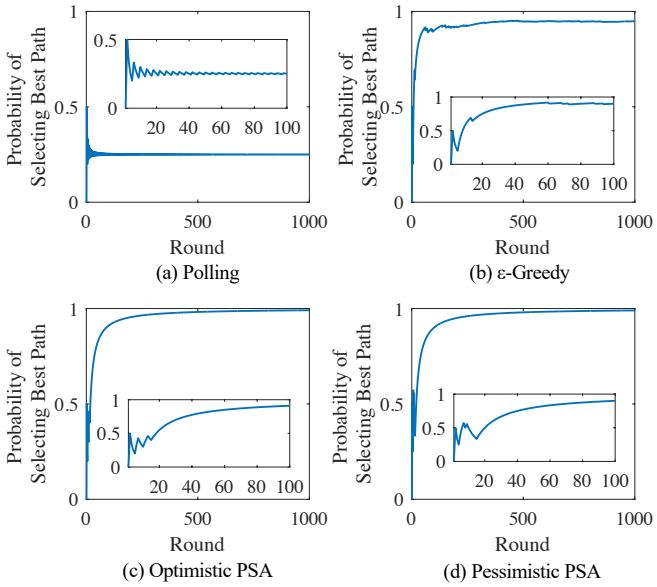


FIGURE 12 Probability of selecting the best path.

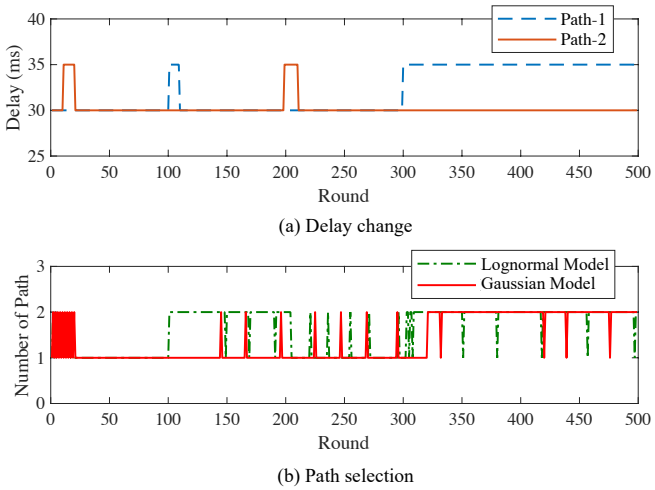


FIGURE 13 Fluctuation in network delay and path selection.

superior efficiency and accuracy in optimal path identification through adaptive exploration.

5.2 | The Influence of Parameters on PCM-QUIC Performance

In this subsection, we discuss the impact of several parameters involved in PCM-QUIC on performance.

5.2.1 | Probability Distribution Model of Task Completion Time

As discussed in Section 4, we proposed two candidate models for the probability distribution of task completion time: the Gaussian model and the Lognormal model. In this subsection, we evaluate the impact of these two modeling approaches on path selection accuracy and overall performance.

We use two virtual NICs on a VMware Linux server (client) to simulate two different access path for user. Network delay, packet loss and bandwidth are set through the network adapter simulation function of VMware Fusion. The delay between the client and the server and the calculation delay of the server in response to the request are almost negligible. The delay of the two NICs (paths) is 30ms. When the VMware Linux server requests the QUIC service deployed on the local server, we artificially change the delays of the two NICs to simulate the performance changes of the two networks in the real space and time dimensions. The network delay change is divided into two situations, the first is the occasional delay change, which often lasts for a short time, and the second is the continuous delay change.

Figure 13 shows the changes in network delay and path selection. In the 20th, 100th and 200th round, we simulate three occasional delay changes by modifying the VMware vNIC. In the 300th round, we simulate a continuous delay change on path-1. The Gaussian model parameter g=10. The sliding window S=100.

The Lognormal model is more sensitive to network fluctuation and the Gaussian model can filter out some sudden fluctuation. At the same time, the Gaussian model is slower to respond to continuous delay change, because this model will filter large delay values when the delay deteriorates. Therefore, the Lognormal model is more suitable for scenarios where the network is stable. The Gaussian model is suitable for scenarios where the network changes drastically.

5.2.2 | Weight Parameter a and b

Building on the analysis presented in Section 5.1, we evaluate the routing performance of the pessimistic PSA under various combinations of parameters a and b.

Figure 14 compares the influence of different combinations of a and b on path selection, they are $a = 0.8$, $a = 0.5$ and $a = 0.2$. In the 15 rounds of path selection, the first two finally selected path-2 as the best transmission path, and the third selected path-1. This result shows that PCM-QUIC can meet the path requirements of different values of a and b. When applying PCM-QUIC, the parameters a and b allow us to control the relative emphasis placed on the expectation and variance of TCT.

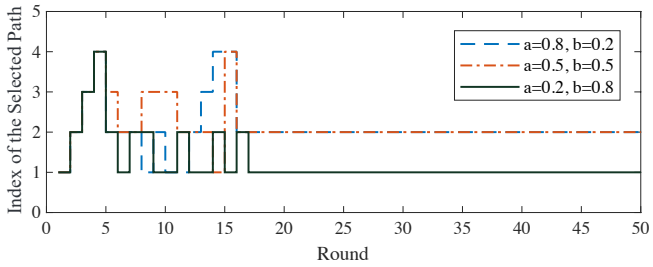


FIGURE 14 Path selection for the first 50 rounds.

TABLE 6 Performance parameters of different paths.

Value / Parameter / Path	Bandwidth	Delay	Packet loss
Path-1 (LTE)	100M	50ms	3.66%
Path-2 (5G)	1000M	30ms	2.79%
Path-3 (Wi-Fi 6)	1000M	30ms	4.06%

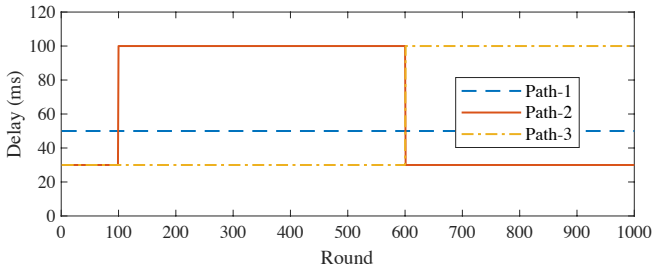


FIGURE 15 Delay variation of three paths.

5.2.3 | Sliding Window

The sliding window parameter S determines how many of the most recent task completion times PCM-QUIC considers when selecting a path. A smaller S places more emphasis on the most recent task completion time, while a larger S extends the influence of historical data. An appropriately chosen S value helps prevent excessive path switching by balancing the need for up-to-date performance information with a thorough understanding of all available paths.

As described in Section 5.2.1, we use three virtual NICs on a VMware-based Linux server to simulate three distinct access methods for the user. The performance of these methods is summarized in Table 6.

Figure 15 illustrates the changes in path delay. The network delays of Path-2 and Path-3 are modified during the 100th and 600th rounds. This experiment allows us to assess PCM-QUIC's ability to adapt to significant network fluctuations under different values of S.

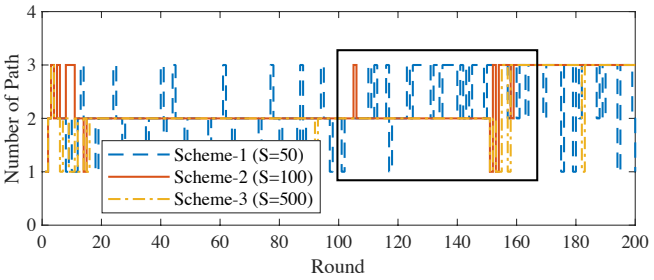


FIGURE 16 Path selection for the first 200 rounds.

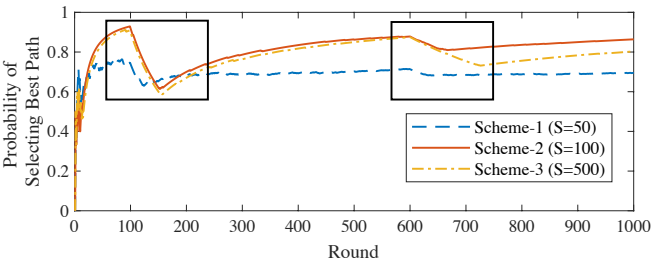


FIGURE 17 Probability of selecting the best path.

As shown in Figure 16, all three schemes identified the optimal transmission path within the first 200 rounds. Scheme-1 (S=50) was the first to switch to the new optimal path, followed by Scheme-2 and Scheme-3. This behavior can be attributed to the smaller sliding window in Scheme-1, which forces the algorithm to use both Path-1 and Path-3 at least once within 50 rounds.

As illustrated in Figure 17, this phenomenon also appears in the 600th round, indicating that a smaller sliding window makes PCM-QUIC more sensitive to changes in task completion time. Ultimately, Scheme-2 demonstrated the highest probability of selecting the optimal path.

Figure 18 presents the cumulative regret changes of the three schemes. Unexpectedly, when S=100, the accumulated regret of PCM-QUIC is the smallest. The reason for Scheme-3 is that PCM-QUIC is slow to respond to task completion time change. The reason for Scheme-1 is more complicated. Because PCM-QUIC not only considers the path reward when selecting the path, but also considers the path bonus. Therefore, Scheme-1 has to frequently switch to a path with poor performance.

6 | DISCUSSION

In this section, we review the research questions and results, and discuss limitations and possible countermeasures.

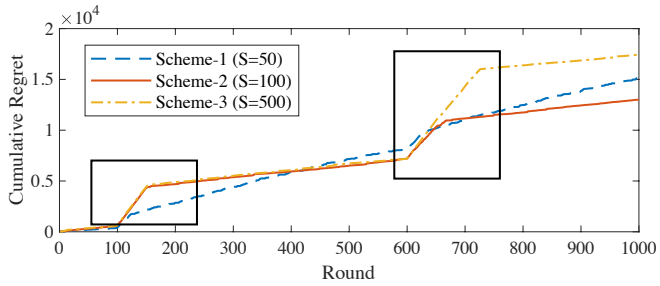


FIGURE 18 Cumulative regret of three schemes.

6.1 | In What Scenarios is PCM-QUIC Applied?

PCM-QUIC uses the characteristics of QUIC connection migration and multiplexing to make up for the defect that QUIC cannot perceive end-to-end path differences at the stream level. The most typical application scenarios of PCM-QUIC are mobile Internet, weak network environment, and audio/video transmission.

6.1.1 | Mobile Internet

It is a fact that mobile devices support multiple NICs. The future mobile access network must be a HetNets. In a mobile HetNets, it will be a valuable technical issue to make full use of the multi NICs characteristics of mobile devices to provide users with more efficient Internet services. At present, PCM-QUIC is a feasible solution for mobile Internet to complete path exploration and utilization.

6.1.2 | Weak Network Environment

Base station congestion, building blockage, and high-speed movement will cause the quality of mobile communication to deteriorate. We refer to scenarios with poor user experience as weak network environments. There are many technologies to improve users' experience in a weak network environment, such as more efficient antennas, more advanced retransmissions, and more concise application functions. PCM-QUIC explores the true end-to-end performance of different network paths in a weak network environment, instead of access network performance, to detect the real network effect closer to the user experience. By switching available networks in a weak network environment, PCM-QUIC can obviously improve user experience.

TABLE 7 Comparison of PCM-QUIC and Traditional Connection Migration.

Value \ Scheme / Comparison	Proactive Connection Migration	Traditional Connection Migration
Type	Active	Passive
Triggering Condition	User experience (Response time)	Physical switching (User switching)
Application Scenes	Live video, Browser, etc.	Weak network, High-speed movement, etc.
Target	QoS/QoE	Keep connected

6.1.3 | Audio/Video Transmission

Streaming media refers to a technology that compresses a series of data and sends them in segments through the network for instant transmission for watching audio and video. The mainstream streaming media real-time transmission protocols include Real Time Messaging Protocol (RTMP)[48], HTTP Live Streaming (HLS)[49] and Web Real-Time Communication (WebRTC)[50]. The main factor that affects the audience experience is the network quality from the CDN node to the audience, including delay and packet loss. Often media providers will make special optimizations for different MSPs.

Although a lot of congestion control schemes[51] and terminal adaptive rate adjustment schemes[52] have been proposed by the academic. But none of these solutions can break through the path bottleneck. If there are two available paths from the media source to the terminal, no matter how the poor path adjusts the congestion window, the effect of the good path is still good. How to choose the best path is the mission of PCM-QUIC. PCM-QUIC combines advanced congestion control and application adaptive technology to enable users to obtain the best viewing and video call experience.

6.2 | Comparison of PCM-QUIC and Traditional Connection Migration

Although both IETF and Google regard connection migration as the main feature and function of QUIC. However, most of the current engineering projects have not shown great interest in connection migration technology. One reason is that connection migration is a top-down cross-Internet technology. It is a passively triggered technology, which leads to its low actual use value. As shown in Table 7, the starting point of PCM-QUIC is to improve user experience, which can better solve user pain points.

6.3 | Comparison of PCM-QUIC and Multi-path QUIC

As shown in Table 8, both PCM-QUIC and multi-path QUIC are designed to improve user experience. PCM-QUIC uses one out of K available networks, and multi-path QUIC uses k out of K available networks. Their underlying technologies and application scenarios are different. PCM-QUIC is suitable for scenarios with sufficient bandwidth, and multi-path QUIC is suitable for weak network scenarios with insufficient bandwidth. They complement each other and can satisfy more application scenarios for users. Especially in Web services, a request may contain multiple servers or CDN resources from different sources. The process of getting each resource using multi-path QUIC is very complicated, and single-path QUIC may be sufficient.

Here, we give the transmission performance model of single-path (PCM-QUIC) and multi-path (MPQUIC).

Assume that the mobile phone accesses the Internet through SIM1, SIM2 and Wi-Fi. The network performance of the three network access methods is bandwidth (B_1, B_2, B_3), latency (D_1, D_2, D_3), packet loss rate (L_1, L_2, L_3). And SIM1 has the best performance, while SIM2 and Wi-Fi are significantly worse. We use effective throughput to measure the performance of PCM-QUIC and MPQUIC.

MPQUIC introduces reordering overhead, congestion control coupling, and scheduling inefficiencies across heterogeneous paths. A penalty term δ represents the processing cost of reordering, retransmission, and inefficient path usage. An efficiency factor $\eta \in [0, 1]$ accounts for how multipath scheduling degrades with increasing path asymmetry.

We can get that single-path throughput is

$$T_{\text{single}} = B_1(1 - L_1). \tag{23}$$

And the multi-path throughput is

$$T_{\text{multi}} = (B_1 + B_2 + B_3)(1 - \bar{L}) \cdot \eta - \delta, \tag{24}$$

where $\bar{L} = (L_1 + L_2 + L_3)/3$ is the average loss rate. So, we can get the condition that T_{single} is greater than T_{multi} is

$$\frac{B_1(1 - L_1) + \delta}{(B_1 + B_2 + B_3)(1 - \bar{L})} > \eta. \tag{25}$$

That is to say, when the average packet loss rate $\bar{L}$ is high, the scheduling efficiency η is low (e.g., the path delay difference is large), or the bandwidth of the poor-quality path is small and the penalty term δ is large, the single-path throughput will be better than the multi-path.

TABLE 8 Comparison of PCM-QUIC and Multi-path QUIC.

Value \ Scheme / Comparison	PCM-QUIC	MP-QUIC
Path Selection	C_K^1	$C_K^k (k \leq K)$
Scenes	Service bandwidth < single path bandwidth	Service bandwidth > single path bandwidth
Target	Improve QoS/QoE	
Disadvantage	Frequent switching may reduce QoS/QoE in the early stages	Out-of-sequence, Retransmission, Path management and congestion control are complex
Relationship	Expand the multi-path selection algorithm for PCM-QUIC, which is equivalent to MP-QUIC	

6.4 | Comparison of PCM-QUIC and VHO

Vertical handoff (VHO) is a mature practice in HetNets. However, how to evaluate the network quality of different heterogeneous networks is always an inevitable and difficult problem for VHO. The parameters of different networks have completely different meanings, and they are not comparable. The information of the physical layer or the data link layer is not enough for VHO to design efficient and reasonable handover decision algorithms. PCM-QUIC does not have this problem at all. Since decision information comes from the transport layer, PCM-QUIC can fairly and efficiently compare the true performance of different heterogeneous networks.

6.5 | The Impact of CDN and Edge Computing on PCM-QUIC

CDN and edge computing technology are widely used, which can greatly improve user access experience. Frankly speaking, PCM-QUIC cannot solve the problem of connection interruption caused by server changes caused by CDN and edge computing. But this problem can be solved in many ways, such as application layer synchronization and server cluster synchronization. It needs to be clarified that this does not mean that PCM-QUIC is not suitable for CDN and edge computing scenarios. PCM-QUIC explores the performance of different service nodes based on CDN and edge computing load balancers, and selects the best.

In addition, there are many scenarios in real life that cannot be covered by CDN and edge computing. For example, exploring the optimal VPN access path in enterprise ERP is also an important application scenario for PCM-QUIC.

TABLE 9 Network Performance Measurement under Different Granularities.

Value \ Scheme / Comparison	Connection level	Stream level	Packet level
Evaluation	Life cycle	Completion time	RTT
Granularity	Coarse	Medium	Fine
Application	L4 or L7 load balancing	PCM-QUIC	Congestion control
Cost	Small	Middle	Big

6.6 | Advantages of Stream-level Measurement

There are three levels of QUIC end-to-end measurement: Packet, Stream, and Connection. As shown in Table 9, stream level has the most appropriate granularity, it is the smallest direct unit of user experience, and it also solves the problem of excessively large measurement granularity at the Connection Level. Since the characteristic of QUIC connection migration is to keep the connection ID unchanged, it is a better granularity to allocate different paths according to different Streams. Stream-level end-to-end measurement only needs to be calculated by the QUIC client. The amount of calculation is minimal. In fact, the operation object of connection migration in QUIC is Connection. The operation object of PCM-QUIC is several Streams belonging to the same Connection.

6.7 | Economic Considerations of PCM-QUIC

As previously mentioned, deploying the PCM-QUIC is not only a technical issue but also an economic one involving equipment manufacturers, MSPs, and users. Although HetNets have become a consensus, the widespread adoption of PCM-QUIC still needs to address three issues.

The first issue is whether the operating systems of equipment manufacturers can support QUIC multi-IP access. We have seen the actions of many equipment vendors. Their mobile phones currently support dual-channel access of 5G and Wi-Fi. However, the cellular communication technology shares a communication baseband, the handover delay is relatively large, and it is still subject to cost constraints.

The second issue is how mobile operators will price their services, which will become a path decision variable for PCM-QUIC. PCM-QUIC that considers tariffs can help users save costs while improving the quality of user experience. PCM-QUIC can also help MSPs reduce costs, balance traffic, and avoid increased network congestion.

The third issue is whether users are willing to confidently hand over the authority to switch their networks to PCM-QUIC. Users should safely allow PCM-QUIC to choose the best network for them. Users can write their expectations or special agreements in the contract between PCM-QUIC and users.

6.8 | Security Considerations of PCM-QUIC

Frequent replacement of the source address by PCM-QUIC may cause traffic amplification attacks. This problem can be solved by address verification. The server receiving the non-detection frame from the client means that the peer has migrated to the new address, and the server needs to send the next data packet to the new address and initiate path verification to prove that the client is on the new address Of ownership. Since the client has used this new address, the above verification can be skipped. After verifying the new client address, the server needs to send a new address verification token to the client.

PCM-QUIC is more sensitive to path attacks. Since the client has multiple IP addresses to communicate with the server, the attacker copies and modifies the address of a data packet. The fake data packet arrives before the original packet arrives at the server, which will make the server think that the client has undergone PCM-QUIC. When the original packet arrives, the original packet is discarded. This problem can also be solved by address verification. The attacker cannot pass the verification because he does not have the necessary encryption key to read or respond to the PATH_CHALLENGE frame. Assuming that there is no channel monitoring on the network, the server will always receive data packets with larger packet numbers from the legal client address. Since the server has verified all possible addresses of the client, the server only needs to send response messages following the latest address of the client.

7 | CONCLUSION AND FUTURE WORK

This paper presents an implementation scheme for the Proactive Connection Migration for QUIC (PCM-QUIC) protocol, which uses the multi-armed bandit model to model and select the optimal path. This is a supplement to the existing connection migration in QUIC, which has improved the transmission performance of QUIC in heterogeneous networks. Future work will focus on improving the PCM-QUIC mechanism in three key areas.

Firstly, the integration of pricing and energy consumption considerations in the PCM-QUIC strategy is a promising area

of research. Different mobile service providers have varying resource pricing models, and the energy consumption per bit of data transmission varies depending on the signal strength and network standards of different access methods. Investigating how to reduce network costs while ensuring optimal user experience represents an important research challenge.

Secondly, multi-path PCM-QUIC is an exciting avenue for future development. Compared to single-path transmission, multi-path QUIC can offer higher throughput and enhanced reliability. In multi-path PCM-QUIC, path selection is influenced not only by the response time of individual paths but also by the overall performance of combined multi-path transmissions.

Finally, given the frequent changes in the client's source address, PCM-QUIC needs to explore mechanisms to mitigate the risk of path attack misjudgments, ensuring secure and reliable path selection.

ACKNOWLEDGMENTS

This work was supported by Institute of Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (RS-2024-00392332, Development of 6G Network Integrated Intelligence Plane Technologies), the Shandong Provincial Natural Science Foundation under Grant No.ZR2022LZH015, the Korea-China Young Scientists Exchange Program grant funded by the National Research Foundation of Korea (NRF-24109).

CONFLICT OF INTEREST

The authors declare no potential conflict of interests.

REFERENCES

1. Zhang T, Mao S. Joint Video Caching and Processing for Multi-Bitrate Videos in Ultra-dense HetNets. *IEEE Open Journal of the Communications Society.* 2020;1:1230-1243.
2. Cui Y, Li T, Liu C, Wang X, Kühlewind M. Innovating transport with QUIC: Design approaches and research challenges. *IEEE Internet Computing.* 2017;21(2):72–76.
3. Bishop M. RFC 9114: Hypertext Transfer Protocol Version 3 (HTTP/3). 2022. [Online; visited Feb-20-2025]. https://datatracker.ietf.org/doc/rfc9114/.
4. Rüth J, Wolsing K, Wehrle K, Hohlfeld O. Perceiving QUIC: Do Users Notice or Even Care?. In: CoNEXT'19. ACM 2019; Orlando, Florida:144-150.
5. IETF . QUIC: A UDP-Based Multiplexed and Secure Transport. 2025. [Online; visited Feb-20-2025]. https://datatracker.ietf.org/doc/html/rfc9000.
6. Biswal P, Gnawali O. Does QUIC Make the Web Faster?. In: GLOBECOM'2016. 2016; Washington, DC, USA:1-6.
7. Paulo R. Exploring QUIC Connection Migration. IETF Draft; 2019.
8. Tan L. PCM for QUIC. 2025. [Online; visited Feb-20-2025]. https://github.com/lzhtan/PCM_for_QUIC.
9. Lizhuang T, Xiaochuan T, Wei Z, Wei S. Connection Migration in QUIC. IETF Draft; 2020. [Online; visited Feb-20-2025]. https://datatracker.ietf.org/doc/draft-tan-quic-connection-migration-00.
10. De Coninck Q, Bonaventure O. Multipath QUIC: Design and Evaluation. In: CoNEXT'17. ACM 2017; Incheon, Korea:160-166.
11. Chen W, Wang Y, Yuan Y. Combinatorial multi-armed bandit: General framework and applications. In: ICML'13. PMLR 2013; Atlanta, Georgia, USA:151–159.
12. Han Q, Khamaru K, Zhang CH. UCB algorithms for multi-armed bandits: Precise regret and adaptive inference. *arXiv preprint arXiv:2412.06126.* 2024.
13. Tan L, Su W, Liu Y, Gao X, Li N, Zhang W. Proactive Connection Migration in QUIC. In: MobiQuitous'21. 2020:476–481.
14. Wang L, Kuo GSG. Mathematical Modeling for Network Selection in Heterogeneous Wireless Networks - A Tutorial. *IEEE Communications Surveys & Tutorials.* 2013;15(1):271-292.
15. Keshavarz-Haddad A, Aryafar E, Wang M, Chiang M. HetNets Selection by Clients: Convergence, Efficiency, and Practicality. *IEEE/ACM Transactions on Networking.* 2017;25(1):406-419.
16. Qin Z, Zhou X, Zhang L, Gao Y, Liang YC, Li GY. 20 Years of Evolution From Cognitive to Intelligent Communications. *IEEE Transactions on Cognitive Communications and Networking.* 2020;6(1):6-20.
17. Himayat N, Yeh SP, Panah AY, Talwar S, Koucheryavy Y. Multi-radio heterogeneous networks: Architectures and performance. In: ICNC'14. 2014; Honolulu, HI, USA:252-258.
18. Gai Y, Krishnamachari B. Distributed Stochastic Online Learning Policies for Opportunistic Spectrum Access. *IEEE Transactions on Signal Processing.* 2014;62(23):6184-6193.
19. Kuroda K, Kato H, Kim SJ, Naruse M, Hasegawa M. Improving throughput using multi-armed bandit algorithm for wireless LANs. *Nonlinear Theory and Its Applications, IEICE.* 2018;9(1):74-81.
20. Cao H, Cai J. Distributed Opportunistic Spectrum Access in an Unknown and Dynamic Environment: A Stochastic Learning Approach. *IEEE Transactions on Vehicular Technology.* 2018;67(5):4454-4465.
21. Zhou P, Xu J, Wang W, Jiang C, Wang K, Hu J. Human-Behavior and QoE-Aware Dynamic Channel Allocation for 5G Networks: A Latent Contextual Bandit Learning Approach. *IEEE Transactions on Cognitive Communications and Networking.* 2020;6(2):436-451.
22. Huang Y, Chen Y, Hou YT, Lou W. Achieving Fair LTE/Wi-Fi Coexistence with Real-Time Scheduling. *IEEE Transactions on Cognitive Communications and Networking.* 2020;6(1):366-380.
23. Apple . URLSessionConfiguration. 2021. [Online; visited Feb-20-2025]. https://developer.apple.com/documentation/foundation/nsurlsessionconfiguration/2875967-multipathservicetype.
24. Huawei . Link Turbo. 2020. [Online; visited Feb-20-2025]. https://consumer.huawei.com/ph/community/details/topicId-138083/.
25. Govil Y, Wang L, Rexford J. MIMIQ: Masking IPs with Migration in QUIC. In: FOCI'20. USENIX 2020; Vitural:1–8.
26. Puliafito C, Conforti L, Virdis A, Mingozzi E. Server-side QUIC connection migration to support microservice deployment at the edge. *Pervasive and Mobile Computing.* 2022;83:101580.
27. Kim SY, Koh SJ. mQUIC: Use of QUIC for Handover Support with Connection Migration in Wireless/Mobile Networks. *IEEE Communications Magazine.* 2023;62(4):128–134.
28. Hurtig P, Grinnemo KJ, Brunstrom A, Ferlin S, Alay Ö, Kuhn N. Low-Latency Scheduling in MPTCP. *IEEE/ACM Transactions on Networking.* 2019;27(1):302-315.
29. Cloud JM, Calmon FDP, Zeng W, Pau G, Zeger LM, Medard M. Multi-path TCP with network coding for mobile devices in heterogeneous networks. In: VTC2013-Fall. IEEE. 2013; Las Vegas, NV, USA:1-5.
30. Choi KW, Cho YS, Aneta , Lee JW, Cho SM, Choi J. Optimal load balancing scheduler for MPTCP-based bandwidth aggregation in heterogeneous wireless environments. *Computer Communications.* 2017;112:116-130.
31. Nikravesh A, Guo Y, Qian F, Mao ZM, Sen S. An in-depth understanding of multipath TCP on mobile devices: measurement and system design. In: MobiCom'16. ACM/IEEE. 2016; New York City, New York, USA:189-201.
32. De Coninck Q, Bonaventure O. Observing Network Handovers with Multipath TCP. In: SIGCOMM'18. ACM 2018; Budapest, Hungary:54-56.
33. Viernickel T, Froemmgen A, Rizk A, Koldehofe B, Steinmetz R. Multipath QUIC: A deployable multipath transport protocol. In: ICC'2018. IEEE. 2018; Kansas City, MO, USA:1-7.

34. Van T, Tran HA, Souihi S, Mellouk A. Empirical study for dynamic adaptive video streaming service based on Google transport QUIC protocol. In: LCN'18. IEEE. 2018; Chicago, IL, USA, USA:343-350.
35. Mogensen RS, Markmoller C, Madsen TK, Kolding T, Pocovi G, Lauridsen M. Selective Redundant MP-QUIC for 5G Mission Critical Wireless Applications. In: VTC'19. IEEE. 2019; Kuala Lumpur, Malaysia, Malaysia:1-5.
36. Coninck QD, Bonaventure O. MultipathTester: Comparing MPTCP and MPQUIC in Mobile Environments. In: TMA'19. IEEE. 2019; Paris, France:221-226.
37. Wang J, Gao Y, Xu C. A Multipath QUIC Scheduler for Mobile HTTP/2. In: APNet'19. ACM 2019; Beijing, China:43-49.
38. Trestian R, Ormond O, Muntean GM. Game Theory-Based Network Selection: Solutions and Challenges. *IEEE Communications surveys & tutorials.* 2012;14(4):1212-1231.
39. Tran HA, Hoceini S, Mellouk A, Perez J, Zeadally S. QoE-Based Server Selection for Content Distribution Networks. *IEEE Transactions on Computers.* 2014;63(11):2803-2815.
40. Liu K, Zhao Q. Distributed Learning in Multi-Armed Bandit with Multiple Players. *IEEE Transactions on Signal Processing.* 2010;58(11):5667-5681.
41. Wu Q, Du Z, Yang P, Yao YD, Wang J. Traffic-Aware Online Network Selection in Heterogeneous Wireless Networks. *IEEE Transactions on Vehicular Technology.* 2016;65(1):381-397.
42. Awad A, Mohamed A, Chiasserini CF. Dynamic Network Selection in Heterogeneous Wireless Networks: A user-centric scheme for improved delivery. *IEEE Consumer Electronics Magazine.* 2017;6(1):53-60.
43. Du Z, Jiang B, Xu K, Wei S, Wang S, Zhu H. Second-order multi-armed bandit learning for online optimization in communication and networks. In: TURC'19. 2019; Chengdu, China:1-6.
44. Ma M, Zhu A, Guo S, Wang X, Liu B, Su X. Heterogeneous network selection algorithm for novel 5G services based on evolutionary game. *IET Communications.* 2020;14(2):320-330.
45. Deng S, Netravali R, Sivaraman A, Balakrishnan H. WiFi, LTE, or both? Measuring multi-homed wireless internet performance. In: IMC'14. 2014; Vancouver, BC, Canada:181-194.
46. Hawkins AS, Berry DA, Fristedt B. Bandit Problems: Sequential Allocation of Experiments. *Journal of the Royal Statal Society.* 1985;83(1):67.
47. Chernoff , Herman . Sequential Design of Experiments. *Annals of Mathematical Statistics.* 1992;30(3):755-770.
48. IETF . Adobe's RTMFP Profile for Flash Communication. 2014. [Online; visited Feb-20-2025]. https://datatracker.ietf.org/doc/html/rfc7425.
49. Bentaleb A, Zhan Z, Tashtarian F, et al. Low latency live streaming implementation in Dash and HLS. In: MM'22. 2022; Lisboa Portugal:7343–7346.
50. Jennings C, Hardie T, Westerlund M. Real-time communications for the web. *IEEE Communications Magazine.* 2013;51(4):20–26.
51. De Cicco L, Carlucci G, Mascolo S. Congestion Control for WebRTC: Standardization Status and Open Issues. *IEEE Communications Standards Magazine.* 2017;1(2):22-27.
52. Mao H, Netravali R, Alizadeh M. Neural Adaptive Video Streaming with Pensieve. In: SIGCOMM'17. ACM 2017; Los Angeles, CA, USA:197-210.

AUTHOR BIOGRAPHY

Lizhuang Tan received his Ph.D. degree from School of Electronic and Information Engineering, Beijing Jiaotong University, P. R. China in 2022. He is currently an Associate Professor with Key Laboratory of Computing Power Network and Information Security, Ministry of Education, Shandong Computer Science Center (National Supercomputer Center in Jinan), Qilu University of Technology (Shandong Academy of Sciences), Jinan, China. And he is also a visiting researcher at DPNM Lab, Department of Computer Science and Engineering, Pohang University of Science and Technology, Pohang, Korea. His research interests include network measurement, management and optimization.

Xin Dong received his B.Sc. degree from School of Computer Science and Technology, Shandong University of Finance and Economics, P. R. China in 2024. He is currently pursuing the M.Sc. degree with the Shandong Computer Science Center (National Supercomputer Center in Ji'nan), Qilu University of Technology (Shandong Academy of Sciences), Jinan, China. His research interests include FPGA, Software Defined Networking and Programmable Networks.

Xiaochuan Gao received his M.S. degree in cyberspace security from School of Cyberspace Security, Beijing University of Posts and Telecommunications in 2019. He is now a senior engineer at Bytedance, Beijing, China. He has been engaged in QUIC research for more than three years. Currently, he is working on the open source of QUIC in the cloud computing environment.

Peiying Zhang received his Ph.D. degree at the University of Beijing University of Posts and Telecommunications, China in 2019. He is currently an Associate Professor with the College of Computer Science and Technology, China University of Petroleum (East China),

Qingdao, China. He has published multiple IEEE/ACM Trans./Journal/Magazine papers, such as IEEE TMC, IEEE TIFS, IEEE TII, IEEE TITS, IEEE TVT, IEEE TNSE, IEEE TNSM, IEEE TETC, IEEE Network and etc. His research interests include Software Defined Networking, Future Internet, and Network Virtualization.

Wei Su received his B.S., M.S., and Ph.D. degrees in communication and information systems from Beijing Jiaotong University in 2001, 2004 and 2008. He is a full professor at the School of Electronic and Information Engineering, Beijing Jiaotong University, Beijing, China. He has studied the Internet for more than 20 years. He was selected for the Beijing Young Talents Program in 2013. He won the second prize of National Technology Invention Award in 2014. He was a visiting scholar with Future University, Hakodate, Japan in 2015. He has published more than 50 research papers and 4 monographs in the areas of communications and computer networks. His research interests are next generation network and Mobile Internet.

James Won-Ki Hong received his HBSc and MSc degrees in Computer Science from the University of Western Ontario, Canada, in 1983 and 1985, respectively, and the PhD degree in Computer Science from the University of Waterloo, Canada, in 1991. He is Professor in the Department of Computer Science and Engineering at Pohang University of Science and Technology (POSTECH), Pohang, Korea. He had worked as the Chief Technology Officer and Senior Executive Vice President for KT (Korea Telecom), the largest telecommunications company in Korea from March 2012 to February 2014, where he was responsible for leading the R&D effort of KT and its subsidiary companies. He was Chairman of National Intelligence Communication Enterprise Association and Chairman of ICT Standardization Committee in Korea. He cofounded and is currently served as Executive Director of SDN/NFV Forum in Korea. He had served as the Head of Department of Computer Science and Engineering, Dean of Graduate School of Information Technology, Director of POSTECH Information Research Labs, and Head of the Division of IT Convergence Engineering at POSTECH. He has served as Chairman of the IEEE Communications Society, Committee on Network Operations and Management. He has also served IEEE ComSoc Director of Online Content (2004–2005, 2010–2011). He is the Editor-in-Chief of International Journal on Network Management (IJNM), IEEE ComSoc Technology News, and KNOM Review Journal. He is the General Chair of NOMS'24, ICBC'19, NetSoft'16 and APNOMS'06. He is an editorial board member of IEEE Transactions on Network and Service Management, Journal of Network and Systems Management and Journal of Communications and Networks. His research interests include network innovation, such as software-defined networking and network function virtualization, cloud computing, mobile services, IPTV, ICT convergence technologies (e.g., Smart Home, Smart Energy, and Health care), and Internet of Things.