Computer Networks 286 (2026) 112478

Contents lists available at ScienceDirect

Computer Networks

journal homepage: www.elsevier.com/locate/comnet

QRoute: A QoE-aware route planning system for enhancing in-vehicle video streaming ☆

Jiahai Hu [a,*], Shutong Chen [b], Lin Wang [c], Qiangyu Pei [a], Fangming Liu [a,d], Bo Li [e]

[a] School of Computer Science and Technology, Huazhong University of Science and Technology, Wuhan, China
[b] Guangxi University, Nanning, China
[c] Paderborn University, Paderborn, Germany
[d] Peng Cheng Laboratory, Shenzhen, China
[e] Department of Computer Science and Engineering, Hong Kong University of Science and Technology, Hong Kong, China

ARTICLE INFO

Keywords:
Route planning
Adaptive video streaming
Quality of experience
5G

ABSTRACT

With the evolution of novel technologies like autonomous driving and in-vehicle entertainment systems, vehicles are evolving into a new form of entertainment space, offering a diverse and enriched in-vehicle experience such as immersive video streaming. However, ensuring consistently high QoE of video streaming while moving is challenging due to variability in network performance, especially in 5G networks. This emphasizes the importance for vehicles to realize QoE-aware route planning. However, QoE-aware route planning for video streaming faces unique challenges. The history-dependent nature of QoE in adaptive video streaming increases the complexity of the route planning problem, rendering traditional algorithms ineffective. Moreover, collecting available bandwidth, which is essential for estimating the QoE of video streaming, presents additional challenges due to the considerable measurement overhead. To address the challenges, we propose a QoE-aware route planning system equipped with a lightweight probe-based bandwidth estimation method and a heuristic QoE-aware routing algorithm to efficiently find the optimal route. Furthermore, our algorithm can flexibly balance QoE and travel distance according to different user preferences through multi-objective route planning. The evaluation results show that our system can achieve an improvement in QoE of up to 97.5% compared to existing route planning systems.

1. Introduction

With advancements in artificial intelligence and autonomous driving technologies, drivers are becoming less involved in driving control. Simultaneously, the variety and richness of entertainment options available within vehicles are increasing, facilitated by cutting-edge in-vehicle entertainment systems. These developments redefine the traditional role of cars as mere transportation tools, transforming them into a new paradigm of dynamic entertainment spaces that offer a diverse and enriched in-vehicle experience for both drivers and passengers. In recent years, governments, auto manufacturers, and streaming service providers have been exploring the potential for entertainment activities within autonomous vehicles. In 2022, the UK government introduced regulations that allowed individuals to legally watch films on motorways [1], signaling a positive position toward embracing in-vehicle entertainment. Furthermore, leading auto manufacturers and technology companies, such as Audi and Intel, are actively developing immersive in-vehicle experiences to keep passengers entertained in a driverless future [2,3]. Recently, Tesla also released RoboTaxi for driverless Taxi services.

Quality of Experience (QoE) for in-vehicle applications, such as video streaming, plays a vital role in shaping the overall in-vehicle experience. Notably, QoE is heavily influenced by the network performance experienced along the route. 5G, as a new generation of mobile communication network, offers the promise of high throughput and low latency. However, despite the growing maturity of commercial 5G networks, providing a consistently smooth video streaming experience

☆ This work was supported in part by National Key Research & Development (R&D) Plan under grant 2022YFB4501703, in part by the Major Key Project of PCL under Grant PCL2024A06 and PCL2022A05, in part by the Shenzhen Science and Technology Program under Grant RCJC20231211085918010, and the High-level Talent Program of Guangxi University under Grant A3070051017, in part by an RGC RIF grant under contract R6021-20, an RGC TRS grant under contract T43-513/23N-2, RGC CRF grants under contracts C7004-22G, C1029-22G and C6015-23G, NSFC/RGC grant CRS_HKUST601/24 and NSFC project under contract 62432008, RGC GRF grants under contracts 16207922, 16207423 and 16203824.

* Corresponding author.
E-mail address: hujiahai@hust.edu.cn (J. Hu).

https://doi.org/10.1016/j.comnet.2026.112478
Received 11 February 2026; Received in revised form 3 June 2026; Accepted 8 June 2026
Available online 13 June 2026

in moving vehicles remains a significant problem. On the one hand, the substantial deployment costs of 5G base stations make achieving comprehensive network coverage difficult, leading to areas with weak signals or subpar network performance. On the other hand, complex environmental factors along roadways can further degrade network performance. For example, obstacles such as trees, buildings, or tunnels can cause signal attenuation or even network disconnection [4], thereby diminishing the in-vehicle experience.

These issues present hurdles for users seeking a consistently high-quality in-vehicle experience. To address this, we argue that vehicles can mitigate QoE degradation by adjusting their routes to avoid roads with poor network connectivity. Specifically, our observations reveal that QoE can be substantially improved by selecting a QoE-aware alternative route, with only a slight increase in travel distance compared to the shortest route (Section 2.2). This highlights the importance of incorporating both road conditions and network performance into vehicle route planning algorithms. Unfortunately, existing navigation applications typically prioritize optimizing travel distance or time, neglecting the importance of in-vehicle application QoE, such as video streaming. While some studies focus on route planning problems with consideration of points of interest [5], safety [6], multi-modal transportation [7], carbon emissions [8], and energy-efficient routes for electric vehicles [9,10], none of these works consider the QoE of in-vehicle applications during route planning.

To bridge this gap, this paper proposes the first QoE-aware route planning system designed to enhance the QoE of in-vehicle applications. The system consists of three key components: *Network Data Collection*, *Application QoE Estimation*, and *QoE-aware Route Planning* (Section 3). Specifically, the system first collects network performance data across roads. Based on the collected network data, it estimates the QoE of in-vehicle applications. Finally, it computes optimal routes considering both the application QoE and user-specific preferences. However, developing such a system is far from trivial. It involves overcoming significant challenges, including the efficient collection of network data across diverse cellular environments, the accurate QoE estimation for complex real-world applications, and the development of an effective QoE-aware routing algorithm. This work tackles these challenges to pave the way for QoE-aware route planning. While various in-vehicle applications have distinct QoE models, this paper focuses on video streaming as a representative example to illustrate the design of our QoE-aware route planning system.

As a popular in-vehicle application, video streaming has become an important part of modern connected car experiences. However, it is worth noting that achieving QoE-aware route planning for video streaming faces unique challenges. One of the critical challenges in QoE-aware routing for adaptive video streaming lies in the *history-dependent* nature of each road segment's weight (i.e., QoE), due to cascading effects of streaming on preceding segments along the route (Section 2.4). This interdependency fundamentally distinguishes it from traditional routing problems, where the weights of path segments are typically static and independent. Moreover, accurately estimating the QoE of adaptive video streaming on a specific road segment based on its network conditions presents a significant challenge in QoE-aware route planning. Beyond network performance, the QoE of adaptive video streaming is also influenced by various factors, such as the behavior of adaptive bitrate (ABR) algorithms and the scheduling patterns of video chunks. Given the intricate interplay of these factors, translating network Quality of Service (QoS) into the QoE of adaptive video streaming becomes a highly complex and challenging task. Additionally, while collecting available bandwidth is essential for estimating the QoE of bandwidth-intensive applications like video streaming, it remains challenging due to the considerable measurement overhead. Note that our system is designed as a general framework that supports multiple application types. We chose video streaming as the representative example because it presents a relatively more challenging scenario for route planning as mentioned above. Our system can naturally accommodate latency-sensitive applications like in-car gaming, as their delay-based QoE metrics are typically not history-dependent. For these applications, the QoE of each route segment can be independently estimated from latency measurements, and directly used as segment weights in route planning.

To address the above challenges, we present a QoE-aware route planning system for in-vehicle adaptive video streaming. The system collects network data through crowdsourcing measurements and employs probe-based bandwidth estimation techniques to efficiently measure available bandwidth. Based on the collected network data, the system estimates the QoE by simulating the streaming behavior of real-world video applications. Finally, to handle the aforementioned QoE dependencies across road segments, we propose a heuristic buffer-based pruning method to efficiently find a QoE-aware route. We evaluate the performance of our system with 5G datasets and further investigate the impact of routing algorithm parameters on its performance. The results show that our algorithm can achieve up to a 97.5% improvement in QoE under real-world 5G networks, while only incurring a slight increase in travel distance. Our contributions are listed as follows:

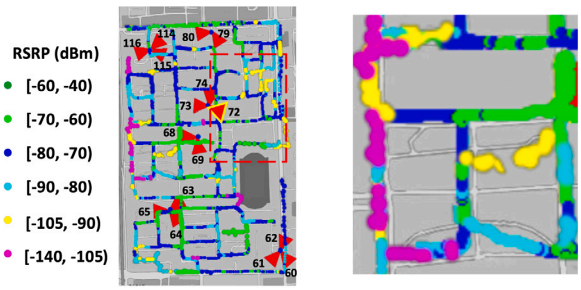
- We propose the first QoE-aware route planning system architecture for enhancing the experience of in-vehicle applications.
- We design a novel probe-based bandwidth estimation method to lightly collect available bandwidth in diverse cellular environments.
- We develop a simulator to swiftly estimate the QoE by carefully and accurately emulating the streaming model of real video streaming applications.
- We define the in-vehicle QoE on a specific road and formulate the QoE-aware route planning problem for adaptive video streaming as a history-dependent shortest path problem.
- We propose a QoE-aware multi-objective route planning algorithm for adaptive video streaming, which achieves considerable QoE improvement while incurring only a slight increase in travel distance.
- We offer a novel perspective on enhancing QoE by QoE-aware route planning, paving the way for research in improving QoE through the lens of routing optimization.

2. Background and motivation

In this section, we first outline the general motivation for QoE-aware routing across diverse mobile scenarios, highlighting the importance and generality of this research problem. We then introduce traditional route planning algorithms and highlight their limitations, followed by a case study whose observations directly reveal the gaps that motivate this work. Next, we introduce adaptive video streaming and its QoE model. Finally, we outline the unique challenges involved in QoE-aware route planning problem for adaptive video streaming.

2.1. QoE-aware route planning

Ensuring satisfactory application QoE in mobile networks is a fundamental and important challenge. For example, modern connected vehicles must support a wide range of bandwidth-hungry or latency-sensitive applications, from passenger entertainment such as video streaming and cloud gaming, to mission-critical services such as teleoperated driving. As vehicles traverse urban road networks, they experience highly heterogeneous network conditions. Similarly, unmanned aerial vehicles (UAVs) are increasingly deployed for tasks such as aerial live streaming and real-time surveillance, where unstable wireless links can lead to frequent video stalls. In these mobile scenarios, a key yet overlooked factor that determines QoE is the movement path of the user equipment (UE). Since network conditions vary significantly across different locations and routes, proactively selecting a path with better



(a) Network overview (b) Network of case study

Fig. 1. 5G coverage of street network (from [16]).

network conditions can promisingly optimize application QoE. This insight motivates us to explore the QoE-aware route planning problem.

Prior studies have investigated QoE-aware routing for UAV-mounted aerial base stations to improve its service coverage and quality [11]. In contrast, our work addresses QoE-aware route planning on the UE side, with a particular focus on driving scenarios. Unlike UAVs, which typically operate in open airspace with dominant line-of-sight (LoS) channels, vehicles traverse dense urban environments where non-line-of-sight (NLoS) propagation caused by buildings and tunnels results in more significant network quality variations across routes. Additionally, among various mobile applications, we focus on adaptive video streaming as a representative case study, since its history-dependent nature introduces unique technical challenges to QoE-aware route planning, which we elaborate in Section 2.4. Other applications without such history dependency are less challenging and thus can be handled more straightforwardly.

2.2. Traditional route planning

Route planning applications are widely used in daily life to assist with navigation and provide route recommendations. Route planning is typically formulated as a shortest path problem, with the primary objective being to identify the shortest or fastest route. Over the years, numerous algorithms have been developed to tackle this problem, including Dijkstra's algorithm [12], the Bellman–Ford algorithm [13,14], and A* algorithm [15], etc.

Current route planning applications usually recommend the shortest or fastest path to users, ignoring the importance of considering application QoE. However, as the following case study will demonstrate, the shortest route suggested by commonly used navigation applications can sometimes result in suboptimal QoE. In contrast, a QoE-aware route can significantly enhance the QoE with only a slight increase in travel distance. This observation highlights the importance of QoE-aware route planning and motivates our work.

Consider a real-world street network, illustrated in Fig. 1, which depicts the distribution of network signal strength (i.e., Reference Signal Received Power, RSRP) across each street. We utilize QoE penalty P to evaluate the QoE of the street:

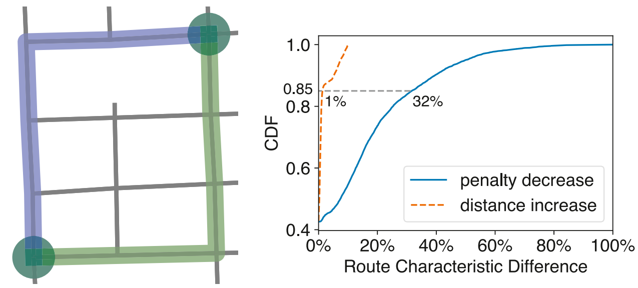
$$P = \frac{\sum_{i=1}^{n} p_i}{n} \cdot d,$$

where p_i is the penalty of a sample point on the street. The penalty of a sample point is defined by its measured RSRP value, as listed in Table 1. And d is the distance of the street.

The shortest route may have poor QoE: We extract two routes with the same source and destination from Fig. 1(b), as shown in Fig. 2(a). Route A (depicted in blue) is the shortest route recommended by the current navigation application, whereas Route B (depicted in green) is an alternative route with a comparable distance. Table 2 illustrates that although the distance and duration of Route A are slightly shorter than those of Route B, Route B provides significantly better QoE compared to Route A.

Table 1
QoE penalty of RSRP values.

Coverage	RSRP (dBm)	Penalty
Good	[−80, −40)	0
Fair	[−90, −80)	1
Bad	[−100, −90)	2
No coverage	[−140, −100)	3



(a) Case study (b) CDF for all pairs

Fig. 2. Characteristic differences of alternative QoE-aware routes compared to the shortest routes.

Table 2
Characteristics of routes.

Characteristics	Route A (blue)	Route B (green)
Distance (m)	384	386
Duration (s)	276.5	277.9
Total penalty	789.8	364.1

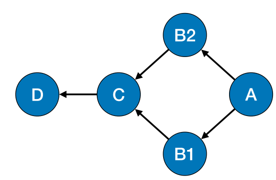


Fig. 3. An example of the invalidation of the optimal substructure property in QoE-aware route planning.

QoE can be substantially enhanced with only a slight distance increase: As shown in Table 2, we observe that Route B achieves approximately a 2× penalty reduction with only a 0.5% increase in travel distance. In fact, it is quite common for two routes with similar distances to deliver noticeably different levels of QoE. Fig. 2(b) compares the shortest route with the QoE-aware route for each pair of nodes in Fig. 1(a). Here, the QoE-aware route refers to the optimal route for better QoE within a 10% distance increase compared to the shortest route. The results reveal that in most cases (∼60%), users can benefit from QoE-aware routing, while in the other cases, the QoE-aware route coincides with the shortest route. Furthermore, in 74% of the beneficial cases, QoE can be improved by up to 32%, while the distance increases by no more than 1%. Thus, we can significantly improve the QoE of mobile applications by planning routes with consideration of application QoE.

Observation: *The QoE of alternative routes can be greatly different, due to distinct network conditions on each route. The shortest path recommended by current navigation applications may have a bad application QoE. Therefore, it is important to take application QoE into consideration while planning routes for users. Moreover, QoE-aware routing is able to significantly improve QoE by trading a slight increase in travel distance.*

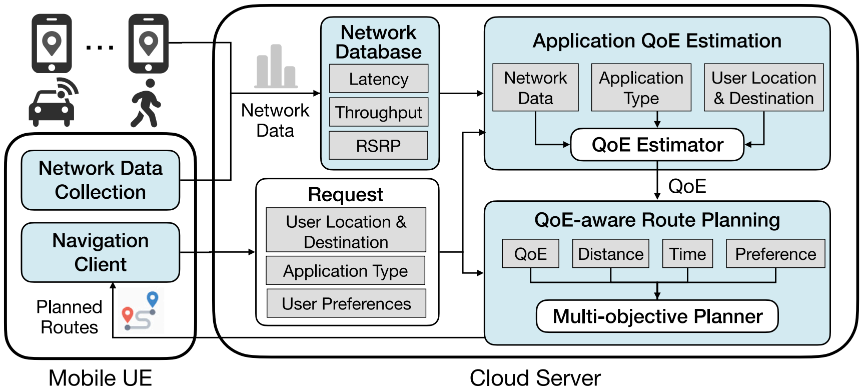


Fig. 4. Overview of QoE-aware route planning system.

2.3. Adaptive video streaming

The adaptive bitrate algorithm is a widely utilized technique in video content delivery, designed to optimize user QoE by dynamically adjusting video bitrate. Specifically, these algorithms dynamically select an appropriate bitrate for each video chunk in real time, based on estimated network throughput and buffer length.

The QoE of adaptive video streaming is affected by various factors, including video quality, quality variation, and stall time caused by rebuffering events [17]. Therefore, the QoE of streaming N video chunks can be generally formulated by

$$QoE = \sum_{n=1}^{N} q(R_n) - \lambda \sum_{n=1}^{N-1} |q(R_{n+1}) - q(R_n)| - \mu \sum_{n=1}^{N} T_n,$$

as proposed in [17], where R_n is the bitrate of the nth video chunk and $q(R_n)$ is the perceived quality of the chunk at bitrate R_n. The second term represents the QoE penalty for quality variation between chunks. T_n is the stall time incurred during the download of the nth video chunk. Coefficients λ and μ are used to weight the penalties for quality variation and stall time, respectively.

2.4. QoE-aware route planning for video streaming

While considering QoE is crucial, it is non-trivial to identify QoE-optimal routes for adaptive video streaming. Traditional shortest path algorithms typically rely on a relaxation process applied to each arc, and solve the problem using the *optimal substructure* property inherent to shortest path problems. Specifically, this property states that the shortest path between two nodes is composed of the shortest subpaths between intermediate nodes.

However, the QoE of adaptive video streaming is affected by is influenced by a complex interplay of factors, including video bitrate, bitrate variation, and stall time caused by rebuffering. Notably, rebuffering events are closely tied to the remaining video content in the buffer, which is determined by the dynamic balance between buffer filling speed and consumption rate along previous segments of the route. These intricate interdependencies break the optimal substructure property, which is a fundamental assumption for traditional shortest path algorithms, making QoE-aware route planning significantly more challenging.

For example, consider the graph shown in Fig. 3, where the objective is to determine the QoE-optimal route from A to D. Assume (A, B_1, C) is the QoE-optimal path from A to C, while an alternative path (A, B_2, C) offers a worse QoE but a longer buffer length at C. Since the cost (i.e., QoE penalty) of the arc (C, D) depends on the buffer length at C, the path (A, B_1, C, D) may not necessarily be the QoE-optimal path from A to D. This is because an insufficient buffer at C increases the likelihood of video stalls along the arc (C, D), thereby degrading overall QoE.

Observation*: In adaptive video streaming, the historical dependency of QoE disrupts the optimal substructure property inherent to shortest path problems. As a result, traditional shortest path algorithms become inapplicable for solving QoE-aware routing problems.*

3. QoE-aware route planning system

To address the above challenges, we propose the first QoE-aware route planning system, designed to enhance the QoE of in-vehicle applications (e.g., video streaming, online gaming, and real-time communications). This system optimizes routes by taking into account network performance, application-specific requirements, and user preferences to achieve better QoE. The architecture of the QoE-aware route planning system is depicted in Fig. 4. This system enables users to plan routes not only based on conventional metrics (e.g., distance or travel time) but also by considering the estimated QoE along different routes. The system is structured into three core modules: *Network Data Collection*, *Application QoE Estimation*, and *QoE-aware Route Planning*. A detailed explanation of each module's design is provided below.

3.1. Network data collection

This module is tasked with gathering network performance data along users' routes to build a crowdsourced network map. The collected data includes essential network metrics such as signal strength (e.g., RSRP), latency, and throughput/bandwidth. Since cellular network operators and crowdsourcing measurement platforms typically only provide network coverage maps, these data are usually not publicly available. Therefore, it is necessary to acquire these data by conducting our own measurements.

Crowdsourcing Network Measurement. Given the expensive cost of constructing a large-scale network map in the real world, our system leverages crowdsourcing measurements, similar to how other navigation platforms gather traffic information. Different applications rely on different network metrics for QoE estimation. For example, video streaming QoE is primarily affected by available bandwidth, while interactive applications (e.g., gaming) are more sensitive to latency. To accommodate this diversity, our system employs the most suitable measurement method for each metric. Cellular network metrics can be accessed via network APIs provided by operating systems (e.g., `CellInfo` on Android), while latency can also be easily measured using tools like `ping`. However, it is non-trivial to accurately estimate available bandwidth, which typically involves injecting probe packets into the network. For example, tools like `Iperf` and `Speedtest` rely on bulk data transfer to precisely measure end-to-end bandwidth, but this method can be costly and extremely intrusive. To address this, our system adopts a more lightweight and less intrusive bandwidth estimation method to balance accuracy and practicality.

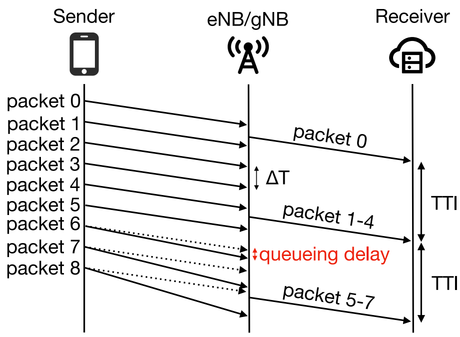


Fig. 5. Impact of packet scheduling intervals in cellular networks on bandwidth estimation accuracy.

Lightweight Probe-based Bandwidth Estimation. To minimize measurement costs and reduce intrusive network traffic, we adopt a packet-train probing method to lightly estimate available bandwidth. Packet-train probing techniques use multiple back-to-back packets sent in a burst (referred to as a "train") to infer the available bandwidth [18]. By observing changes in the intervals between packets as they traverse through the network path, the available bandwidth can be estimated. Specifically, if the available bandwidth is lower than the sending rate, the packets will experience queuing delays, leading to an increase in the intervals between consecutive packets.

However, while packet-train probing is effective in traditional network environments, applying this technique to 5G networks presents unique challenges. One significant challenge lies in the interference caused by packet scheduling intervals in cellular networks. Unlike wired networks, cellular networks typically schedule packets every Transmission Time Interval (TTI). Within each TTI (e.g., 1ms in LTE), the eNB/gNB buffers all incoming packets and transmits them together in a single burst. This behavior shortens the interval between probing packets unexpectedly, leading to inaccurate estimation results (see Fig. 5). This issue becomes even more pronounced in 5G networks due to their variable numerology configurations and TTI durations. In 5G, numerologies, characterized by subcarrier spacing (SCS) and symbol duration, directly influence the packet scheduling granularity (i.e., TTI). The variability in these configurations further complicates achieving precise bandwidth measurements. Additionally, high-bandwidth networks like 5G present challenges due to inherent limitations in packet probing methods. The estimated maximum probing rate is inherently constrained by the Maximum Transmission Unit (MTU) size. To achieve higher estimated rates, probing packets may exceed the MTU size, causing packet fragmentation and distorting bandwidth estimation results [19].

To address the above challenges, we propose a novel packet probing method tailored for bandwidth estimation in diverse cellular network environments. First, to enable high-bandwidth estimation, our method adopts a packet group approach for each probe instead of relying on a single packet, similar to the strategy used by PathRefiner [19]. The total size of each packet group increases exponentially, determined by a configurable spread factor. Each group consists of multiple consecutive packets of identical size, with a consistent number of packets in each group. Second, to mitigate interference caused by packet scheduling in cellular networks, we space each probing packet group with an interval equal to one TTI. This ensures that no consecutive probing groups are transmitted within the same scheduling cycle, reducing inaccuracies induced by queuing at the eNB/gNB. Furthermore, given the variation in SCSs and TTIs across different networks, we adopt an SCS-adaptive probing method that dynamically adjusts the probing interval based on the specific Radio Access Technology (RAT) and network configuration. Specifically, when connected to LTE networks with a fixed SCS of 15 kHz, probing packet groups are sent every 1 ms. For 5G NR networks, where common SCS configurations for FR1 and FR2 bands are 30 kHz and 120 kHz, the probing interval is adjusted to 0.5 ms and 0.125 ms, respectively. Additionally, the probing interval is customizable and can be fine-tuned or learned to align with specific network deployments. Note that an inappropriate interval can be detected by observing identical arrival times of buffered back-to-back packets caused by packet scheduling. This adaptability allows our probing method to achieve precise bandwidth estimation across diverse cellular environments.

The overhead (i.e., bandwidth consumption) of each probe train depends on the probing bandwidth range and granularity, which is determined by the spread factor γ. We denote the minimum and maximum probing bandwidths as R_{min} and R_{max}, respectively. We have $R_{max} = R_{min} \cdot \gamma^{N-1}$, where N is the total number of packet groups in a probe train. Let T be the sending interval between two consecutive packet groups, and let S_i be the total size of the ith packet group. The ith probing rate is defined as $R_i = S_{i+1}/T$, from which the size of the ith packet group can be calculated as $S_i = R_{i-1} \cdot T$. Accordingly, the total size of a single probe train is given by $\sum_{i=1}^{N} S_i$. In practice, the probing range can be dynamically adjusted according to the bandwidth requirements of the target application. For applications with lower bandwidth demands, a narrower probing range with finer granularity is used, which not only reduces probing overhead but also improves estimation efficiency for that specific application. Regarding the probing frequency, our system sends one probe train per second by default. To further reduce overall probing overhead, we perform probing selectively based on the data sparsity and freshness of each road segment. Specifically, frequently traveled roads typically have more recent data and thus requiring less probing, whereas new roads or areas with insufficient historical data require more frequent probing.

Another challenge arises from the reduced estimation resolution as the probing rate increases due to the exponential growth of the probing rate. To address this, we employ iterative estimation during crowdsourcing measurements. As our system progressively gathers more historical bandwidth data, it dynamically refines the granularity of the probing rate, concentrating on bandwidth ranges around the average observed values. This iterative approach not only improves measurement accuracy over time but also enhances the efficiency of bandwidth estimation in high-speed networks. Additionally, inter-packet spacing may be compressed due to network buffering effects, potentially leading to an overestimation of available bandwidth. To mitigate the resulting estimation errors, we leverage the excursion segmentation technique in packet-train probing [20], which identifies contiguous packets experiencing consistent queuing delay increases caused by congestion and uses them to estimate the available bandwidth. This approach is motivated by a fundamental advantage of probe trains over probe pairs. By sending a sequence of multiple packets rather than just two, probe trains can capture trends in queuing delays with a train, which are entirely invisible to packet-pair techniques that directly infer bandwidth from absolute inter-packet spacing. Specifically, the excursion segmentation technique identifies a valid congestion event when the excursion is long enough to be considered statistically meaningful. For each train, we analyze the queuing delay increase signature and detect a potential excursion starting point at packet k where $q_k < q_{k+1}$. The end of an excursion is defined as the first packet j where:

$$q_j - q_i < \frac{max_{i \le k \le j}(q_k - q_i)}{F},$$

where F is a decrease factor. If $j - i > L$ where L is a queue period threshold, the packets between i and j is considered a excursion or a valid congestion event. The decrease factor F and the queue period threshold L are empirically set to 1.5 and 5, respectively. By observing the increasing queuing delay trend within a packet train, this segmentation algorithm makes the probe-based bandwidth estimation robust to sporadic queuing delay drops induced by compressed packet bursts, which could otherwise lead to bandwidth overestimation.

In summary, this module forms the foundation for subsequent QoE estimation by capturing various network performance metrics essential for determining application-specific user experience along users' routes.

3.2. Application QoE estimation

This module estimates application QoE by utilizing the collected network performance data from the network map.

Application-specific QoE Model. Each application type (e.g., video streaming, online gaming) is affected by different network QoS metrics based on its unique requirements, shaping its respective QoE model. For example, the QoE of video streaming is predominantly influenced by throughput, as sufficient throughput ensures smooth and high-quality streaming. In contrast, gaming applications rely heavily on low latency to deliver real-time responsiveness. Therefore, while estimating application QoE, our system considers not only the underlying network conditions but also the specific requirements of each application. In the following, we take adaptive video streaming as an example to illustrate how the system estimates QoE according to its specific requirements and characteristics.

QoE Estimation with Crowdsourced Network Map. Even with the network map and QoE model, there remains a challenge in mapping network QoS metrics (e.g., throughput and latency) to application QoE. For instance, the QoE of video streaming is typically measured by metrics such as video bitrate and stall time. However, it is difficult to capture the potential relationship between these metrics and network throughput, let alone translate it into a precise quantitative formula, due to the complexity of streaming system implementations (e.g., ABR algorithms and chunk scheduling methods).

To achieve efficient QoE estimation for video streaming, we developed a chunk-level simulator that carefully and accurately emulates the streaming logic of real-world video streaming applications. The simulator models key processes, including chunk scheduling, bitrate selection, and buffer consumption (i.e., video playback). A similar idea of the chunk-level simulator is also leveraged in [21]. Our simulator is implemented following the logic of `DASH.js` [22], the official reference implementation for playing videos encoded with MPEG DASH. Given a specific network bandwidth trace and the current buffer length, the simulator can estimate the QoE of the streaming process under specific network conditions and dynamically update the buffer length throughout playback. It is worth noting that network bandwidth is the most critical metric affecting the QoE of video streaming. In contrast, latency-related metrics are often more crucial for live streaming rather than pre-recorded video streaming, which typically uses buffering techniques to preload video content.

3.3. QoE-aware route planning

This module identifies optimal routes by considering application QoE, travel distance or duration, and user preferences, ensuring a balanced and user-centric experience.

Application-specific Heuristic Planning Algorithm. As discussed in Section 2.4, traditional route planning algorithms may fail when considering application QoE. For adaptive video streaming, the QoE dependencies across route segments can break the optimal substructure property, a foundational assumption of many conventional algorithms. Consequently, it is necessary to develop heuristic route planning algorithms tailored to specific application scenarios. We will detail the development of our heuristic QoE-aware route planning algorithm for adaptive video streaming in Section 5.

User Preference and Multi-objective Planning. Routing preferences vary greatly among individuals, shaped by their unique needs and priorities. For instance, some users may prioritize application QoE to ensure a smooth streaming experience throughout their journey, while others may focus on efficiency, favoring the fastest route regardless of application performance. There is also a third group of users who seek a balance between these factors, choosing routes that offer an acceptable trade-off between high application QoE and minimized travel duration or distance. To accommodate this diversity, our route planner is designed to consider multiple objectives, including both application QoE and travel distance or duration. The planner's flexibility allows users to dynamically adjust the relative importance of these factors using configurable weight parameters, as detailed in Section 4.3.

Real-time Route Adaptation. Network performance is inherently dynamic and can fluctuate over time. To address this, our QoE-aware routing system leverages continuously collected crowdsourced network data to detect sudden changes and adapt in real time. By dynamically providing updated route recommendations, the system ensures an optimal experience for the user, similar to how traditional navigation applications reroute drivers in response to unexpected events such as accidents or traffic congestion.

4. Formulation of QoE-aware route planning problem

In this section, we first introduce the formulation of the classic shortest path problem. We then define the QoE model for video streaming along a given path and formulate the QoE-aware route planning problem.

4.1. Classic shortest path problem

Let $\mathbb{G} = (\mathbb{V}, \mathbb{A})$ be a directed network, where $\mathbb{V}$ is the set of vertices and $\mathbb{A}$ is the set of arcs. Let $s, t \in \mathbb{V}$ be the source and target vertices, respectively. Let c_{ij} be the cost for each arc $(i, j) \in \mathbb{A}$. Define x_{ij} as a binary decision variable that indicates whether arc (i, j) is selected as part of a potential path, where $x_{ij} \in \{0, 1\}$, $\forall (i, j) \in \mathbb{A}$. Specifically, $x_{ij} = 1$ if arc (i, j) is included in the path. The objective of the shortest path problem is to find the path from s to t that minimizes the total cost. The problem can be formulated as follows.

$$\min \quad \sum_{(i,j)\in\mathbb{A}} c_{ij} x_{ij} \tag{1}$$

$$s.t. \quad \sum_{j:(i,j)\in\mathbb{A}} x_{ij} - \sum_{k:(k,i)\in\mathbb{A}} x_{ki} = \begin{cases} 1, & \text{for } i = s \\ 0, & \text{for } i \in \mathbb{V}\backslash\{s, t\} \\ -1, & \text{for } i = t, \end{cases} \tag{2}$$

$$x_{ij} \in \{0, 1\}, \quad \forall (i, j) \in \mathbb{A} \tag{3}$$

We define the degree of a vertex to be the number of arcs entering or leaving the vertex. Specifically, the in-degree of a vertex is the number of arcs entering the vertex, while the out-degree is the number of arcs leaving the vertex. Constraint (2) ensures that for all vertices except s and t, the in-degree and out-degree are equal. Meanwhile, the out-degree exceeds the in-degree by exactly 1 for the source s, while the in-degree exceeds the out-degree by exactly 1 for the target t.

4.2. QoE model for video streaming

Next, we define the QoE Q_{ij} for an arc (i, j), which is affected by multiple factors, including video bitrate, bitrate variation, and stall time caused by rebuffering, as discussed in Section 2.3.

On each arc, users may download several video chunks. Let R_{ij}^k, C_{ij}^k, and L be the bitrate, chunk size, and duration of the $k^{th} (k = 1, 2, \ldots, K_{ij})$ video chunk downloaded on the arc (i, j), respectively. Let $q(R)$ be the function that maps bitrate to the video quality perceived by users. In practice, we typically use a linear mapping for this function (i.e., $q(R) = R$) as in [23]. Let $T_{s_{ij}}^k$ be the stall time caused by rebuffering during the download of the kth video chunk. Let Q_{ij} be the total QoE for the arc (i, j). Given these notations, Q_{ij} can be formulated as follows.

$$Q_{ij} = \sum_{k=1}^{K_{ij}} (q(R_{ij}^k) \cdot L - \lambda \max\{q(R_{ij}^{k-1}) - q(R_{ij}^k), 0\} - \mu T_{s_{ij}}^k). \tag{4}$$

In general, the QoE for the arc (i, j) consists of three parts:

- Video Quality: The positive QoE derived from streaming video at a bitrate R_{ij}^k for L seconds of content.
- Quality Variation: The negative QoE incurred due to quality degradation, represented by the change from $q(R_{ij}^{k-1})$ to $q(R_{ij}^k)$.
- Rebuffering Events: The negative QoE resulting from $T_{s\,ij}^{\ k}$ seconds of stalling due to rebuffering

In Eq. (4), λ and μ are penalty coefficients corresponding to quality variation and rebuffering, respectively.

4.3. Problem formulation

In conventional QoE maximization problems for adaptive video streaming, the download throughput trace $\{V_t | t \in [t_1, t_{K+1}]\}$ is typically provided. However, it is not the case in our scenario, as users can select which route to travel, and the network conditions vary significantly across different routes. In this section, we present a formulation of the QoE-aware route planning problem for adaptive video streaming.

To reformulate the QoE maximization problem as a shortest path problem, we use the concept of a *QoE penalty* instead of directly working with QoE as the cost for each road segment. Specifically, the QoE penalty for an arc (i,j) is defined as

$$Q_{ij}^p = \sum_{k=1}^{K_{ij}} ((q(R_{max}) - q(R_{ij}^k)) \cdot L + \mu T_{s\,ij}^{\ k} + \lambda \max\{q(R_{ij}^{k-1}) - q(R_{ij}^k), 0\}), \tag{5}$$

where R_{max} is the highest available video bitrate, and Q_{ij}^p represents the gap in QoE between the ideal case (where all video chunks are streamed at the maximum bitrate and there are no rebuffering events) and the actual QoE Q_{ij} for arc (i,j).

Let d_{ij}^k be the distance from vertex i to the location where the client begins downloading the kth video chunk on arc (i,j). Let t_{ij}^k be the time when the client starts downloading the kth video chunk, with t_{ij}^0 specifically indicating the time when the user enters the arc. Let S_t be the moving speed of the user at time t. Additionally, the function f_v maps a location on arc (i,j) to the corresponding download throughput at that location. We have

$$d_{ij}^k = \int_{t_{ij}^0}^{t_{ij}^k} S_t dt, \tag{6}$$

$$V_{t_{ij}^k} = f_v((i,j), d_{ij}^k). \tag{7}$$

Let D_{ij} be the distance of the arc (i,j), and T_{ij} be the travel duration on the arc (i,j). We have

$$D_{ij} = \int_{t_{ij}^0}^{t_{jk}^0} S_t dt, \tag{8}$$

$$T_{ij} = t_{jk}^0 - t_{ij}^0. \tag{9}$$

Eq. (8) gives the constrain between t_{jk}^0 and t_{ij}^0.

Our objective is to identify a path that simultaneously minimizes the time-averaged QoE penalty for video streaming during travel and the travel distance to the destination. Given that users may have varying preferences, these preferences can typically be modeled as a weighted sum of various objectives [24] (i.e., QoE penalty and travel distance). Let α be a weight parameter that reflects the user's preference ($\alpha \in [0,1]$). The problem can be formulated as:

$$\min_{x_{ij}} \quad \alpha \frac{\sum_{(i,j)\in\mathbb{A}} x_{ij} Q_{ij}^p}{\sum_{(i,j)\in\mathbb{A}} x_{ij} T_{ij}} + (1-\alpha) \sum_{(i,j)\in\mathbb{A}} x_{ij} D_{ij}$$
$$\text{s.t.} \quad (2)(3)$$

5. Heuristic QoE-aware route planning algorithm

In this section, we present solutions to the QoE-aware route planning problems for adaptive video streaming.

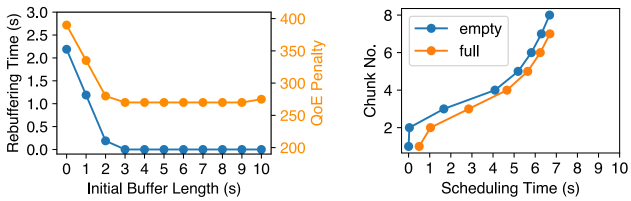
To identify the optimal path with the least QoE penalty, the first step is to determine the cost (i.e., QoE penalty) of each road segment. However, the calculation of QoE penalty depends on the path taken prior to that segment, making these costs path-dependent. Moreover, the QoE dependencies break the optimal substructure property, which prevents the use of traditional shortest path algorithms like Dijkstra's algorithm or the Bellman–Ford algorithm. Therefore, to compute the cost, it becomes necessary to first enumerate all possible complete paths. Once all potential paths are identified, the overall QoE penalty for each path can be computed, and the path with the lowest penalty can be selected as the optimal solution. We refer to this method as the *QoE-optimal* algorithm in the following. However, enumerating all possible paths between the source and the destination can be extremely time-consuming, especially for large-scale graphs. Specifically, the time complexity of identifying all simple paths in a graph is exponential, i.e., $O(2^N)$, where N is the number of vertices in the graph.

One effective way to reduce the search time is to narrow the search space by discarding branches that are unlikely to lead to the optimal result, a technique commonly known as pruning. However, deciding whether to prune a particular path is challenging due to the absence of the optimal substructure property in the problem. To enable efficient pruning of search branches, we begin by analyzing the impact of the buffer on QoE for a given road segment, as the buffer plays a key role in the history-dependent nature of QoE.

5.1. The impact of buffer on QoE

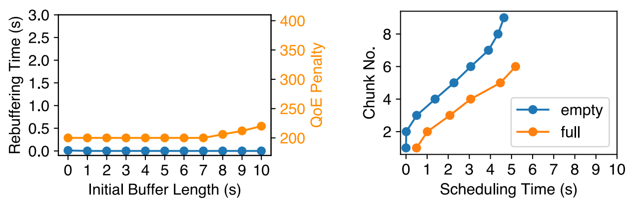
We start with the scenario where the network bandwidth along the road is insufficient to support the requested video bitrate (i.e., bandwidth drops during streaming). Fig. 6(a) shows the rebuffering time and average QoE penalty on the road for varying initial buffer lengths. As shown, rebuffering occurs when the initial buffer length is short. However, the rebuffering time decreases as the buffer length increases. A sufficiently large initial buffer can avoid rebuffering, thereby significantly reducing the QoE penalty or improving the QoE. We also notice a slight increase in average QoE penalty when the buffer is full (i.e., 10s). To figure this out, we analyze the scheduling of video chunk requests. Fig. 6(b) shows the timing of video chunk requests when the initial buffer is either empty or full. Notably, the client requests one fewer video chunk when starting with a full buffer, compared to the scenario with an insufficient initial buffer. This chunk has a lower QoE penalty than the average QoE penalty of previously requested chunks. Consequently, the average QoE penalty slightly increases due to fewer chunks with better QoE being requested. We also observe that the scheduling pattern of video chunk requests is similar across different buffer lengths. In fact, the slope of the curve in Fig. 6(b) is positively correlated with the network bandwidth of the location at that time. Next, we examine the case where the network bandwidth on the road is sufficient to meet the video bitrate requirements. Fig. 7 shows the rebuffering time and average QoE penalty in this scenario. A negligible rebuffering time (i.e., 0.011s) is observed only when there is no initial buffer, resulting in similar QoE penalties across varying buffer lengths.

Observation: *Rebuffering occurs when the network bandwidth is not sufficient, resulting in significant QoE degradation. A sufficient buffer length can avoid rebuffering, thereby improving QoE. Since network bandwidth varies across different roads, the buffer length required to avoid rebuffering also differs accordingly.*



(a) Rebuffering/QoE Penalty (b) Scheduling pattern

Fig. 6. Impact of initial buffer length under insufficient network bandwidth.



(a) Rebuffering/QoE Penalty (b) Scheduling pattern

Fig. 7. Impact of initial buffer length under sufficient network bandwidth.

5.2. Buffer-based pruning for efficient search

Inspired by the observations and insights above, we propose an efficient buffer-based pruning approach to speed up the search for the optimal path.

First, we define a client as *buffer-safe* if rebuffering is unlikely to occur on the next road segment. For example, as shown in Fig. 6(a), a client is considered buffer-safe when entering the road with an initial buffer length of at least 3s. Each road has its specific buffer-safe threshold, which can be easily profiled based on the road's network characteristics. Using this threshold, we can efficiently trade off between buffer length and QoE to make informed decisions.

Next, we demonstrate how paths that are unnecessary to explore can be efficiently pruned. During the graph traversal, each node may be visited multiple times through different incoming arcs depending on the search order. Let us assume that path P_1 is the first traversal of a node and P_2 is a subsequent traversal. To determine whether P_2 should be pruned, we compare the buffer length and QoE penalty of P_1 and P_2. Specifically, P_2 is pruned under one of the following conditions: (1) The QoE penalty of P_2 is greater than that of P_1, and the buffer length of P_2 is shorter than that of P_1. (2) The QoE penalty of P_2 is greater than that of P_1, and while the buffer length of P_2 is longer than that of P_1, P_1 is already buffer-safe. This implies that additional buffer length in P_2 cannot translate into potentially better QoE. By employing this buffer-based pruning method, we can significantly reduce the search space and efficiently identify the QoE-optimal path.

5.3. QoE-aware multi-objective route planning

Building on the buffer-based pruning method, we present our QoE-aware Multi-Objective Route Planning (QMORP) algorithm, as described in Algorithm 1. The objective of the algorithm is to identify the optimal path that minimizes both QoE penalty and travel distance, as detailed in Section 4.3. For clarity, we use pen_{qoe} and pen_{dist} to denote the penalties of QoE and travel distance, respectively. Thus, the total penalty (i.e., the weighted sum of pen_{qoe} and pen_{dist}) can be defined as

$$P = \alpha \cdot pen_{qoe} + (1-\alpha) \cdot pen_{dist}.$$

Generally, we adopt a uniform-cost search [25] strategy to identify the optimal path. This approach prioritizes expanding the node with the lowest total path cost (i.e., penalty) from the source node. We use a min-heap $frontier$ to store paths to be explored, sorted in ascending order by their total penalty. During each iteration, the path with the lowest penalty is selected from $frontier$ (Line 4). If the last node of the selected path u is exactly the target t, the path is added to the candidate list $candidates$ (Lines 5–9). Otherwise, the algorithm examines all neighbors of u to extend the current path (Line 10). For each adjacent node v, the QoE and distance penalties for the edge (u, v) are computed, and the total penalty for the extended path is determined (Lines 15–16). Using the buffer-based pruning method, we then decide whether the newly expanded path should be pruned (Lines 17–23). If the path is not pruned, it is added to the $frontier$ heap, and the current optimal path to node v is updated (Lines 24–29). All potential optimal paths are eventually inserted into the $candidates$ list. Finally, the algorithm selects the path with the lowest total penalty from $candidates$ as the optimal route. It is worth noting that, in large graphs, the search space can become extremely huge, since the algorithm permits detours to explore paths with potentially better networks. To prevent unbounded exploration, a distance threshold d_{limit} is introduced to filter out paths whose travel distances would exceed the threshold before reaching the destination (Lines 11–14). Specifically, $d_{left}[v]$ represents the minimum remaining distance from node v to the target node t. If the sum of $d_{left}[v]$ and the current travel distance $dist$ exceeds d_{limit}, the corresponding path is pruned. This ensures the algorithm maintains computational efficiency and avoids unnecessary expansions.

Algorithm 1 QoE-aware Multi-objective Route Planning

Input: $\mathbb{G}$, source s, target t, preference weight α
Output: path p

1: Initialize a min heap $frontier$, a dictionary $seen$, and a list $candidates$.
2: $push(frontier, (0, 0, [s]))$;
3: **while** $frontier \neq \emptyset$ **do**
4: $(pen, buf, path) \leftarrow pop(frontier)$;
5: $u \leftarrow path[-1]$;
6: **if** $u = t$ **then**
7: $candidates.append((pen, path))$;
8: continue;
9: **end if**
10: **for** $v \in adj[u]$ **do**
11: $dist \leftarrow d + length(u, v)$
12: **if** $dist + d_{left}[v] > d_{limit}$ **then**
13: continue;
14: **end if**
15: $(b, pen_{qoe}, pen_{dist}) \leftarrow estimate(u, v, buf)$;
16: $pen_{new} \leftarrow \alpha * pen_{qoe} + (1-\alpha) * pen_{dist}$;
17: $prune \leftarrow False$;
18: **if** $v \in seen$ **then**
19: $(pen_{old}, b_{old}) \leftarrow seen[v]$;
20: **if** $pen_{new} \geq pen_{old}$ **and** $b \leq b_{old}$ **or** $pen_{new} \geq pen_{old}$ **and** $buffer\_safe(v, b_{old}) = True$ **then**
21: $prune \leftarrow True$;
22: **end if**
23: **end if**
24: **if** $prune = False$ **then**
25: $seen[v] \leftarrow (pen_{new}, b)$;
26: $path_{new} \leftarrow path.append(v)$;
27: $c \leftarrow (pen_{new}, b, path_{new})$;
28: $push(frontier, c)$;
29: **end if**
30: **end for**
31: **end while**
32: Select the optimal path $path_{opt}$ with the least penalty among $candidates$.
33: **return** $path_{opt}$

Fig. 8. Road networks and base station deployments in Melbourne CBD, Australia.

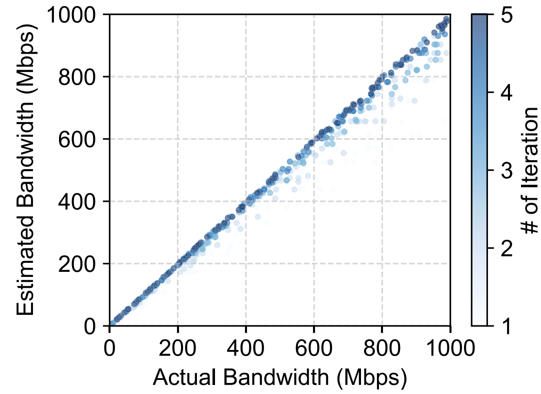


Fig. 9. Estimation results of the bandwidth estimation method.

6. Performance evaluation

In this section, we evaluate the performance of our QoE-aware route planning system, covering both the bandwidth estimation method and the QoE-aware route planning algorithm. Since evaluating bandwidth estimation in cellular networks is challenging due to the lack of ground-truth data, we start with an end-to-end system evaluation in a simulated network, which provides a realistic 5G protocol stack with full control over ground-truth conditions, including interference, network load, and scheduling effects. Furthermore, we extend the evaluation of the QoE-aware route planning algorithm with a real-world 5G dataset to demonstrate its practical effectiveness. Together, these two complementary evaluation approaches provide a comprehensive and robust evaluation of our system. We compare our algorithm against the optimal method (i.e., exhaustive search, as described in Section 5) as an upper bound and a network-aware heuristic method, which directly uses the average bandwidth of each road segment as the routing weight. We further study the impact of various key parameters on its performance to provide a comprehensive analysis.

6.1. End-to-end evaluation in simulated network

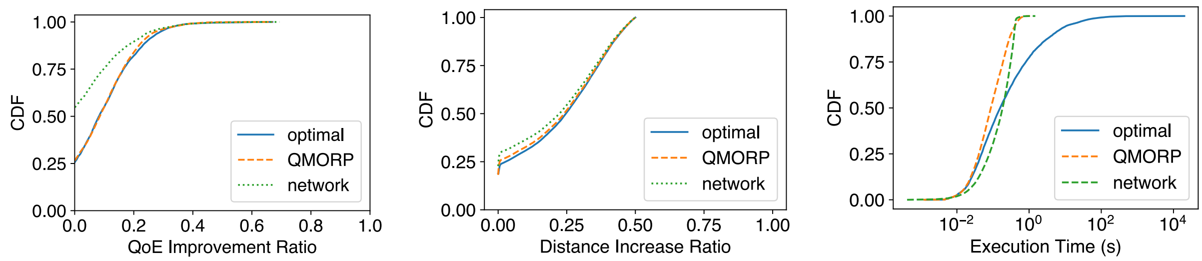
To evaluate our system, we build a simulated cellular network using the widely adopted network simulator, ns-3. We first evaluate the accuracy of our bandwidth estimation method, followed by the performance of the QoE-aware route planning algorithm. The details of the evaluation experiments are introduced below.

Why simulation? Evaluating the accuracy of the bandwidth estimation method in cellular networks is inherently challenging due to the difficulty of the ground truth, i.e., the actual bandwidth of a cellular connection. Without direct access to base stations or detailed knowledge of their configurations, accurately measuring the true available bandwidth becomes highly impractical. While some studies use bulk TCP/UDP throughput as an approximation of the ground truth, evaluating our system requires a large-scale ground-truth bandwidth map of the cellular network. Constructing such a map involves extensive measurements across numerous locations, which incurs significant costs and effort when conducted in the wild. By contrast, simulations can emulate network conditions in a controlled, scalable, and cost-efficient manner, enabling repeatable evaluations and fair comparisons between our system and traditional route planning systems.

Experiment Setup. We build a testbed on the ns-3 simulator integrated with the 5G-LENA module for 5G NR network simulation. Fig. 8 illustrates our simulated scenario with road networks and deployed base stations, a 1.8 km × 0.9 km Central Business District (CBD) in Melbourne. The geographic map of the area is downloaded from OpenStreetMap [26]. The specific locations of these base stations are derived from the radio-comms license dataset [27], released by Australian Communications and Media Authority. The radio access network types (such as 4G LTE or 5G NR) of the base stations are identified using CellMapper [28]. We assume the simulated 5G base stations operate at 28 GHz with a 100MHz bandwidth, while the 4G base stations operate at 1.8 GHz with a 20MHz bandwidth. To simulate the real-world traffic pattern for vehicle UEs, we leverage Simulation of Urban MObility (SUMO) to generate user mobility traces and import them into our simulated scenario in ns-3. As the vehicle UEs move through the simulated scenario, they collect network bandwidth data by our probe-based estimation method, which serves as the network database (i.e., crowdsourced network map) for subsequent QoE-aware route planning. Our QoE-aware route planning algorithm is implemented based on a graph library `networkx` for network analysis. For video streaming application, the 16K video used in our experiments is divided into a sequence of video chunks with 1-second duration, and is encoded into different bitrates (i.e., $\{40, 60, 80, 110, 160, 320\}$ Mbps) as in [23]. The buffer size is set as 10 s.

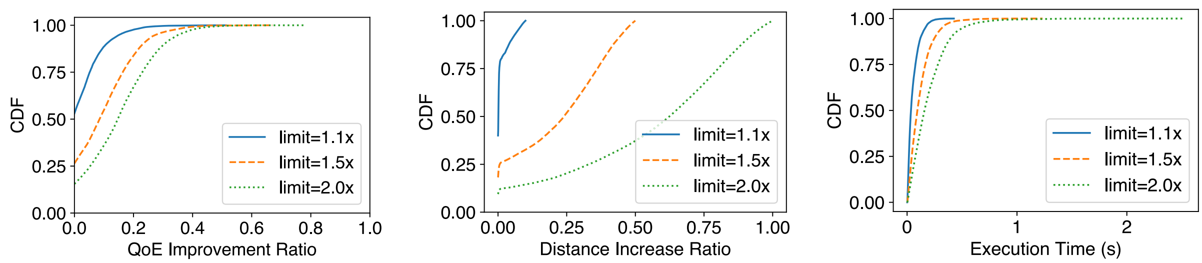
Bandwidth Estimation Accuracy and Overhead. We evaluate the performance of our estimation method by comparing the estimated bandwidth collected by vehicle UEs with the ground truth in the simulated network. To achieve this, we utilize SUMO to randomly generate mobility traces for 50 vehicle UEs, which are then replayed in the ns-3 simulator. Each vehicle UE collects the available bandwidth every second using our estimation method. The actual available bandwidth is measured through bulk data transfers, which is considered as the ground truth (or best-effort ground truth [29]). Additionally, to analyze how the accuracy of our crowdsourcing network map evolves over multiple iterations, we repeat the simulation five times. During each iteration, the bandwidth estimation method dynamically adjusts its probing granularity by leveraging the historical network data accumulated from previous iterations. Fig. 9 presents the estimation results of our bandwidth estimation method. Initially, the estimated bandwidth may exhibit slight deviations from the actual available bandwidth due to coarse-grained probing. However, the accuracy of the estimation improves rapidly after several iterations. It is worth noting that the probing range and granularity can be tailored to meet the specific network requirements of various applications, enabling more efficient and precise probing.

We also evaluate the overhead (i.e., bandwidth consumption) of our packet-train probing method. Generally, the probing is performed once per second with a spread factor of 2. For a 28 GHz 5G network with 120 kHz SCS in our simulation, probing a bandwidth range of [1,1024] Mbps consumes only 255.9kbps. In practice, applications such as video streaming do not require estimating such high bandwidths. For example, a reduced probing range of [1,256] Mbps is sufficient for 16K video streaming in our setup, which further reduces the consumption to just 63.9kbps. By comparison, conventional bandwidth measurement tools such as `Iperf` rely on bulk data transfers, which



(a) QoE (b) Travel distance (c) Execution time

Fig. 10. Performance of QMORP in the simulated scenario.



(a) QoE (b) Travel distance (c) Execution time

Fig. 11. Performance of QMORP under different distance limits in the simulated scenario.

Table 3
Characteristics of the geographic dataset used in the simulated and real-world scenarios.

Scenario	Network	# of Nodes	Distance of pairs (m)		
			Mean	95th	99th
CBD	simulated	528	1412	2307	2589
Campus	real-world	138	491	943	1081

can consume hundreds of Mbps in high-speed 5G networks. Our method reduces measurement overhead by 4–5 orders of magnitude, making it substantially less intrusive. Moreover, the probing traffic is negligible compared to the bitrates of 16K video streaming, ensuring that the probing does not perceptibly degrade the in-vehicle streaming experience.

QoE-aware Route Planning. We then evaluate the QoE-aware Multi-Objective Route Planning (QMORP) algorithm based on the network bandwidth database collected by our system. The network bandwidth used for QoE-aware route planning is obtained by querying our network database. For a specific location, we get all network bandwidth data near the location (i.e., within 10 m) and use the average of these data as the network bandwidth of the location. Table 3 presents characteristics of the geographic map dataset, including the number of nodes and the shortest distances between each pair of nodes. For each pair of nodes (i.e., one as the source, the other as the destination) in the road network, we compare the performance of our algorithm with two baselines. We next present the evaluation results of our algorithms in terms of QoE, travel distance, and execution time.

First, let us take a look at how much our algorithm can improve the QoE. We define the QoE improvement ratio by $\Delta Q = (Q_s - Q_q)/Q_s$, where Q_s is the QoE penalty of the shortest route and Q_q is the QoE penalty of the QoE-aware route. Fig. 10(a) shows the distribution of the QoE improvement ratio by our algorithm, compared with the QoE of the shortest route. As shown, the QoE improvement of QMORP is very close to the optimal method (under the same distance limit). QMORP is able to improve the QoE for 74.4% of the cases. For the other 25.6% of the cases, our algorithm chooses the same route as the shortest route. Overall, our QoE-aware route planning system demonstrates its effectiveness, achieving up to a 64.2% improvement in QoE. Although the network-aware method improves QoE in 45.2% of the cases, it still fails to achieve the optimal QoE in those cases, while leaving potential QoE gains unexploited in the remaining ones. This is because it ignores the gap between network QoS and application QoE, as well as the history-dependent dynamics of QoE.

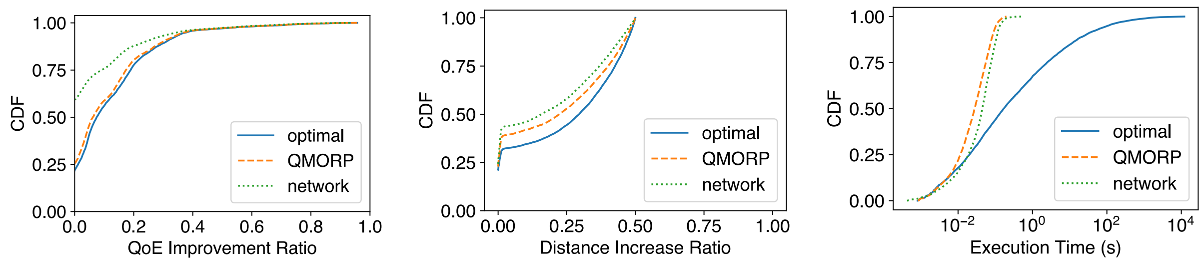
Then we compare the travel distance of the routes recommended by different algorithms with that of the shortest route. Again, we define the distance increase ratio as $\Delta D = (D_q - D_s)/D_s$, where D_s is the travel distance of the shortest route and D_q is the travel distance of the QoE-aware route. Fig. 10(b) shows the distribution of the distance increase ratio. As shown, while achieving near-optimal QoE improvement, our QMORP algorithm ensures the travel distance no longer than that of the optimal method.

We also compare the execution time of different algorithms, including the duration of both weight computation (i.e., QoE estimation) and route planning. Fig. 10(c) shows the CDF of the execution time of different algorithms. As shown, QMORP achieves a tail execution time of under 1.2 s, which falls within an acceptable range for route planning applications. For the optimal method, the exponential time complexity results in extremely high execution time (up to 10^4s), which is totally intolerable for users.

While the above evaluations are performed with a distance limit of $1.5\times$ the shortest distance, we also conduct performance evaluations under varying distance limits. Fig. 11 shows the performance of QMORP under different distance limits. As shown, raising the distance limit can enhance the QoE improvement, although at the cost of longer travel distances. Even with a distance limit of $1.1\times$ the shortest distance, our algorithm is still able to improve QoE in 48.7% of the cases. We will further discuss the impact of the distance limit on QoE improvement under different user preferences later in Section 6.3.

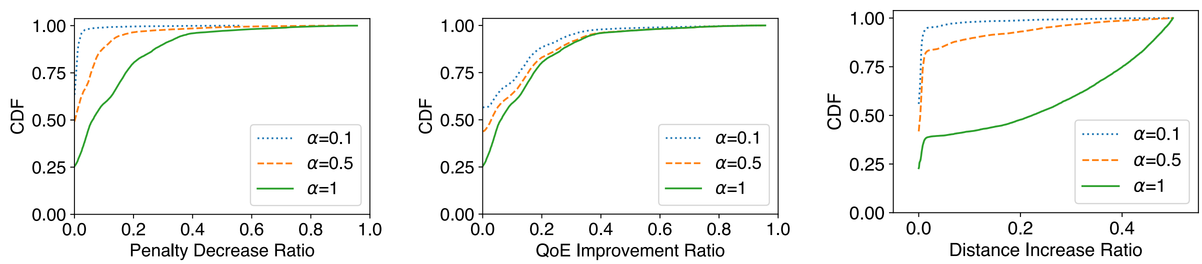
6.2. Evaluation with real-world datasets

In the simulated scenario, the bandwidth data is collected from a simulated cellular network. While simulated networks prove valuable for controlled experimentation, they fall short in accurately capturing the intricate complexities of real-world cellular networks, including



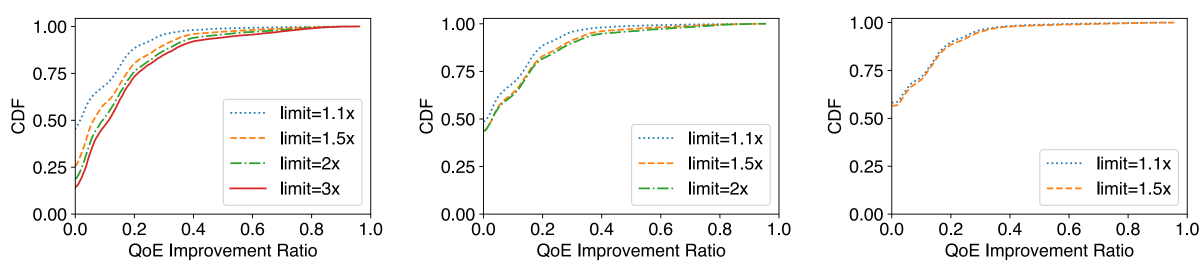
(a) QoE (b) Travel distance (c) Execution time

Fig. 12. Performance of QMORP in the real-world scenario.



(a) Total penalty (b) QoE (c) Travel distance

Fig. 13. Impact of user preference on QMORP performance.



(a) α=1 (b) α=0.5 (c) α=0.1

Fig. 14. Impact of distance limits on QMORP performance under different user preferences.

environmental obstacles and multipath effects, which may lead to fluctuations in signal strength. To ensure a more comprehensive evaluation of QMORP, we further conduct experiments using a real-world network measurement dataset. As previously discussed, the high cost of bandwidth measurements has resulted in a lack of publicly available bandwidth map datasets. Therefore, we synthesize a bandwidth dataset based on a real-world signal strength map. It turns out that our synthetic bandwidth dataset generally matches the actual bitrate observed in the real-world measurement area. While RSRP-based bandwidth synthesis is an approximation of real network conditions, the consistency of results with our NS-3 evaluation demonstrates that the algorithm's performance is robust to real-world network environments. In the following, we elaborate how we generate the synthetic dataset and present our evaluation results in the real-world scenario.

Datasets. The network dataset is measured in a campus by [16], as shown in Fig. 1. Unfortunately, the network dataset only provides RSRP values at each location instead of the bandwidth data required for QoE estimation. To make this dataset practical for evaluation experiments, we generate a synthetic bandwidth dataset based on the original dataset. As shown in Fig. 1, each sampling point represents an RSRP value at that location, which is distributed on and around the road. Since these GPS points may deviate from the road, we first map the network information to the geographic map by finding the nearest road to each point and projecting the point onto the nearest road. Moreover, since the RSRP values cannot be used directly for estimating the QoE of video streaming, we need to map the RSRP values to the network bandwidth. According to the Shannon–Hartley theorem about the logarithmic relationship between channel capacity and signal power [30], we use the following function to synthesize the network bandwidth corresponding to each RSRP value:

$$C = B \cdot \log_2(1 + \frac{10^{\frac{S}{10}}}{P_n}),$$

where C is the estimated capacity or rate in Mbps, B is the physical bandwidth of the radio channel in MHz, P_n is the noise power, S is the RSRP value, $S \in [-140, -60]$dBm. Typically, B is 100MHz for commercial 5G networks and P_n is often approximated as 3.98×10^{-10}W. We assume the RSRP values of those locations with no measurement data as −140 dBm (i.e., no available network). Compared to the bitrate contour in [16], the synthetic bandwidth dataset generally matches the bitrate level on the map. It is worth noting that while such a basic synthesis function may not provide precise bandwidth estimation, it does not affect the fair comparison of different algorithms. The geographic map information (e.g., nodes and paths with their latitude and longitude) of the road network is obtained from an open-source map OpenStreetMap [26].

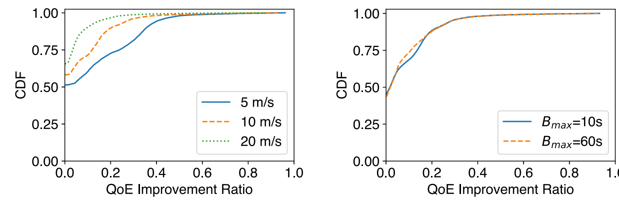
Evaluation results. Fig. 12(a) shows the distribution of the QoE improvement ratio by our algorithm, compared with the QoE of the shortest route. As shown, similar to the simulated scenario, the QoE improvement of QMORP is very close to the optimal method. QMORP is able to improve the QoE for 76.3% of the cases. For the other 23.7% of the cases, our algorithm chooses the same route as the shortest route. Our algorithm can achieve up to 97.5% of QoE improvement in the real-world scenario. Compared to the simulated scenario, our QoE-aware route planning system achieves a 1.2× average QoE improvement in the real-world scenario. In contrast, the network-aware method can only improve QoE in 40.9% of the cases, with an average improvement of 6.3%. The consistency of results across both evaluation approaches demonstrates that our system's performance is robust. Fig. 12(b) shows the distribution of the distance increase ratio. As shown, while achieving near-optimal QoE improvement, our QMORP algorithm keeps the travel distance no longer than 2% of the shortest route for 38.7% of the cases. Compared to the optimal method, QMORP reduces the cost of travel distance while maintaining comparable QoE improvements. It is worth noting that our algorithm can achieve similar QoE improvements at a lower cost of travel distance (Section 6.3). Fig. 12(c) shows the CDF of the execution time of different algorithms. As shown, QMORP achieves a tail execution time of under 0.77 s, while the optimal method consumes up to 10^4s due to its exponential time complexity.

6.3. The impact of parameters

Next, we study how the parameters in our algorithm influence the performance. Given that the proposed algorithm exhibits similar trends in both simulated and real-world scenarios, we focus on the real-world scenario in this section.

User Preference We first study the impact of user preference (i.e., the weight parameter α) on the performance of QMORP. We evaluate the algorithm with different weight parameters (i.e., 0.1, 0.5, 1). We define the penalty decrease ratio by $\Delta P = (P_s - P_q)/P_s$, where P_s is the total penalty of the shortest route and P_q is the total penalty of the QoE-aware route. As shown in Fig. 13(a), the reduction of total penalty becomes increasingly apparent as α increases. This is because a larger α reflects a stronger user preference for higher QoE in video streaming. Specifically, $\alpha = 0$ means only travel distance is considered, whereas $\alpha = 1$ means only QoE is considered (as discussed in Section 5.2). Fig. 13(b) demonstrates that our algorithm consistently achieves high QoE improvement across all values of α. However, the travel distance increases with higher values of α, as a larger α indicates that users place less importance on minimizing travel distance. For example, when α is set to 0.1, 0.5, and 1, the 90th-percentile rates of distance increase are 0.7%, 11.2%, and 46.7%, respectively, while the corresponding 95th-percentile rates are 1.8%, 24.6%, and 48.5%. For users who seek an enhanced QoE with only a slight increase in travel distance, we recommend setting α to below 0.5. Conversely, if users prioritize minimizing travel distance, α should be set to zero.

Distance Limit. We also study the impact of the distance limit on the QoE improvement under different user preferences. As shown in Fig. 14(a), as the distance limit expands, the QoE improvement becomes increasingly significant due to the enlarged search space or optimization space. However, once the distance limit surpasses a certain threshold (~3), raising the limit yields only marginal benefits. It is worth noting that the impact of the distance limit on QMORP's performance also depends on the user's preference. As shown in Fig. 14, as α decreases, the QoE improvement becomes more consistent across varying distance limits, indicating that the marginal effect is observed at shorter distance thresholds. By default, we recommend setting the distance limit to 1.5×the shortest distance, which effectively captures most of the potential QoE improvement while maintaining a reasonable balance with travel distance overhead. Furthermore, since the marginal benefit of increasing the distance limit diminishes at lower values of α (i.e., the weight parameter for QoE penalty), we recommend an adaptive preference-based distance limit. For example, the distance limit can be set to $(1+\alpha)$ times the shortest distance to achieve better search efficiency with comparable QoE benefit.



(a) Impact of Speed (b) Impact of Buffer Size

Fig. 15. QoE improvement of QMORP under different moving speeds and buffer sizes.

Moving Speed. The moving speed of users determines the location where the client downloads video chunks and the duration users spend on the road. To study how the moving speed influences the performance of the proposed algorithm, we compare the performance of the algorithms under different moving speeds. Besides the average driving speed 10 m/s, we also evaluate the performance under 5 and 20 m/s. As shown in Fig. 15(a), when users slow down the moving speed, the QoE improvement achieved by our algorithm becomes more remarkable. When users move at a slow speed, they tend to spend more time on roads with poor/good network conditions. Therefore, users experience a lower QoE in roads with poor network conditions, while enjoying a better QoE in roads with good network conditions. This widens the gap in QoE between the route recommended by our algorithm and the shortest route. In terms of travel distance and execution time, our algorithm performs almost the same under different moving speeds.

Buffer Size. The buffer size B_{max} has an impact on the interval time of chunk scheduling. To study how the buffer size influences the performance of the proposed algorithm, we evaluate its performance with buffer sizes of 10s and 60s. As shown in Fig. 15(b), we observe that the algorithm achieves similar QoE improvements under different buffer sizes. A larger buffer size can reduce the occurrence of rebuffering events, resulting in reduced penalties for both the path recommended by our algorithm and the shortest path. However, the ratio of QoE improvement remains similar.

6.4. Scalability

To further evaluate the runtime scalability on city-scale road networks, we conduct experiments on the road network of Shanghai, which contains 18,376 nodes. Since collecting real-world bandwidth measurements at city scale is prohibitively expensive, we instead use randomly generated bandwidth data, which is sufficient for evaluating execution time alone. As shown in Fig. 16, the median execution time remains below 0.5s even for routes up to 90 km. More importantly, 95% of all route queries are completed within 0.85s, demonstrating that QMORP achieves practical scalability on city-level road networks. The execution time grows approximately linearly with route length, which confirms that our pruning mechanism effectively mitigates the theoretical worst-case exponential complexity in practice.

7. Related work

QoE Optimization in Mobile Scenarios. VSiM proposes to jointly allocate bottleneck bandwidth among multiple users, considering their mobility profiles to achieve max–min QoE fairness [31]. Legato develops an online reinforcement learning framework for improving QoEs of real-time mobile interactive video applications [32]. Floo introduces a QoE-aware congestion control selection mechanism that can adaptively switch the underlying congestion control algorithms to assist the top-level mobile web services to achieve their desired QoEs [33].

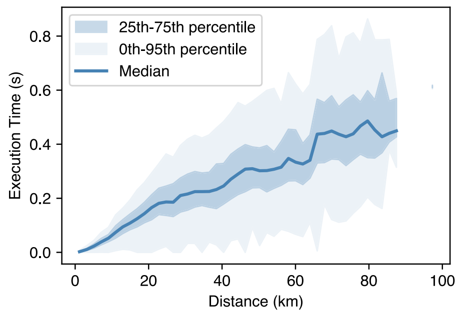


Fig. 16. Execution time versus route distance.

Hassan et al. conduct a data-driven analysis to uncover the impact of mobility on 5G application performance and design an event-based handover prediction system to improve applications' QoE [34]. In addition, current works stand from other perspectives such as caching-based optimizations [35,36], server selection [37,38], multi-path data transport [39], to improve QoEs in mobile scenarios.

Route Planning Considerations. Zheng et al. propose an augmented GPS navigation system GPSView [5] to plan driving routes with scenic landscapes and sights. SocRoutes [6] can find a safer, friendlier, and more enjoyable route based on sentiments inferred from real-time, geo-tagged messages. Lin et al. [8] present an extensive literature review of green vehicle routing problems, which focuses on minimizing system-wide costs related to economic and environmental issues. Wang et al. [9] propose a framework for energy-driven and context-aware route planning for fully electric vehicles. Sarker et al. [10] propose a multi-objective route planner system to guide electric vehicles for multi-objective routing (travel time, energy consumption, charging cost, and range anxiety).

8. Discussion

This section discusses several practical considerations and potential extensions of our system.

Handling Data Sparsity and Freshness in Practice. Real-world deployment of a crowdsourced system such as QRoute introduces several practical challenges, including cold-start problems, data sparsity, and data freshness. When network data is sparse or unavailable for certain roads, QRoute employs the following fallback strategies: (1) For certain roads without sufficient measurement data, we use conservative bandwidth estimates (e.g., 10th percentile of all measurements in the city) to avoid over-optimistic routing; (2) For completely uncovered areas, the system falls back to shortest-path routing. As more users traverse these roads, the crowdsourcing mechanism naturally fills data gaps over time.

Additionally, we apply a confidence-based estimation approach based on data freshness, with older measurements assigned lower confidence weights. Specifically, each bandwidth measurement sample is associated with a timestamp t_i. When constructing the bandwidth map, historical measurements are aggregated using exponential time-decay weighting [40]:

$$w(t_i) = e^{-\lambda \cdot (t_{now} - t_i)},$$

where t_{now} is the current time, t_i is the time when the bandwidth sample B_i was measured, and λ is set empirically based on the historical bandwidth measurements. For example, λ can be adaptively tuned to minimize the root mean squared error (RMSE) of previous estimations for each city's deployed network [41]. By aggregating all the measurement samples, the bandwidth can be estimated as:

$$B = \frac{\sum_i w(t_i) \cdot B_i}{\sum_i w(t_i)}.$$

5G Slice-Aware Routing. 5G's network slicing capability is an important feature that can provide differentiated services for various application types. Incorporating network slicing awareness could further enhance our system. We now discuss how QRoute can be extended to support slice-aware routing as 5G infrastructure evolves. Slice-specific network information can potentially be obtained via 5G `Network Exposure Function` [42], a standardized interface that enables operators to expose network/service information to third-party applications [43]. Leveraging such information, QRoute can maintain separate network maps for different slice types, e.g., an eMBB map for bandwidth-intensive applications such as video streaming, and a URLLC map for latency-sensitive applications such as in-car gaming. Based on the application's QoE requirements, QRoute selects the corresponding map to perform accurate QoE estimation and route optimization. This positions QRoute as a forward-compatible system that can leverage the full potential of 5G slicing as the infrastructure matures. It also highlights the importance of network exposure capability in enabling fine-grained network awareness for mobile applications in 5G and beyond networks.

V2X QoS and Carbon Emission Considerations. Our route optimization framework steers vehicles toward roads with better QoE, which may in turn affect the spatial distribution of V2X-capable vehicles across the road network. If V2X QoS information is available, our system can extend the routing objective to incorporate V2X communication quality:

$$P = \alpha \cdot pen_{qoe} + \beta \cdot pen_{v2x} + (1 - \alpha - \beta) \cdot pen_{dist},$$

where pen_{v2x} captures the V2X communication quality of a road segment, such as RSU coverage density or average V2V link reliability. The weighting coefficients can be dynamically adjusted according to the driving scenario (e.g., collision avoidance vs. infotainment), enabling the system to prioritize safety–critical communication performance when necessary while still optimizing application QoE when V2X constraints become less stringent. Supporting such an extension would require collecting and maintaining V2X-specific link quality maps in addition to the cellular network quality maps. Although this would substantially increase system complexity, we believe it is an important and promising direction for future work.

In addition, growing concerns over carbon emissions make it undesirable to prioritize network quality at the expense of excessive travel distance. We emphasize that our system does not unconditionally optimize QoE regardless of distance cost. Instead, it operates under an explicit distance budget constraint (i.e., the distance limit). This design guarantees that any QoE improvement is achieved within a bounded and controllable increase in travel distance, thus in fuel consumption and carbon emissions. Importantly, our evaluation results demonstrate that significant QoE gains can be achieved with only marginal distance increases. Specifically, our system can achieve up to 97.5% QoE improvement within 10% distance increase. Moreover, in a similar manner to incorporating V2X QoS, our framework can be extended to include energy-related objectives (e.g., fuel consumption [8]), making it possible to jointly optimize application QoE, V2X QoS, and carbon emissions.

Route Convergence Problem. Our system can inherently mitigate the congestion caused by route convergence through the following mechanisms. First, our system measures actual available bandwidth, not merely signal strength. The crowdsourced bandwidth measurements directly reflect the experienced network performance. When multiple users converge on the same route, the consequent bandwidth degradation can be automatically captured by subsequent real-time measurements. Therefore, the "good" route becomes less attractive and the system naturally achieves load balances. Second, our framework supports real-time route adaptation (Section 3.3) during travel. Similar to how traditional navigation systems handle traffic congestion, our framework continuously monitors real-time network performance during a journey. If a route that was initially estimated as high-quality becomes congested, the system detects the throughput degradation and triggers re-routing to alternative paths. This prevents the sustained convergence of all users onto the same routes.

9. Conclusion

In this paper, we propose the first QoE-aware route planning system architecture to enhance the experience of in-vehicle applications. To realize the system, we design a novel probe-based bandwidth estimation method to lightly collect available bandwidth in diverse cellular environments. We also propose a QoE-aware multi-objective route planning algorithm for adaptive video streaming. Our algorithm employs a heuristic buffer-based pruning method to efficiently find the optimal route, while also accommodating varying user preferences. Compared to existing route planning systems, our system can achieve up to a 97.5% improvement in QoE, with only a marginal increase in travel distance. Moreover, our system is capable of finding a QoE-aware route within 0.8s, making it highly adequate for real-world route planning tasks. Overall, our work offers a novel perspective on enhancing QoE by QoE-aware route planning, paving the way for research in improving QoE through the lens of routing optimization.

CRediT authorship contribution statement

Jiahai Hu: Writing – original draft, Software, Methodology, Conceptualization. **Shutong Chen:** Validation, Formal analysis. **Lin Wang:** Writing – review & editing, Methodology. **Qiangyu Pei:** Validation, Conceptualization. **Fangming Liu:** Writing – review & editing, Supervision. **Bo Li:** Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

References

[1] Self-driving car users could watch films on motorway under new DfT proposals, 2022, https://www.theguardian.com/technology/2022/apr/20/self-driving-car-users-could-watch-films-on-motorway-under-new-dft-proposals.
[2] The future of the in-vehicle experience, 2019, https://www.cbinsights.com/research/report/in-vehicle-experience-technology-future.
[3] Holoride: Virtual reality meets the real world, 2022, https://www.audi.com/en/innovation/future-technology/virtual-reality/holoride-virtual-reality-meets-the-real-world.html.
[4] A. Narayanan, E. Ramadan, J. Carpenter, Q. Liu, Y. Liu, F. Qian, Z.-L. Zhang, A first look at commercial 5G performance on smartphones, in: Proceedings of WWW, 2020, pp. 894–905.
[5] Y.-T. Zheng, S. Yan, Z.-J. Zha, Y. Li, X. Zhou, T.-S. Chua, R. Jain, GPSView: A scenic driving route planner, ACM TOMM 9 (1) (2013) 1–18.
[6] J. Kim, M. Cha, T. Sandholm, Socroutes: safe routes based on tweet sentiments, in: Companion Proceedings of WWW, 2014, pp. 179–182.
[7] H. Huang, D. Bucher, J. Kissling, R. Weibel, M. Raubal, Multimodal route planning with public transport and carpooling, IEEE T-its 20 (9) (2018) 3513–3525.
[8] C. Lin, K.L. Choy, G.T. Ho, S.H. Chung, H. Lam, Survey of green vehicle routing problem: past and future trends, Expert Syst. Appl. 41 (4) (2014) 1118–1138.
[9] Y. Wang, J. Jiang, T. Mu, Context-aware and energy-driven route optimization for fully electric vehicles via crowdsourcing, IEEE T-its 14 (3) (2013) 1331–1345.
[10] A. Sarker, H. Shen, J.A. Stankovic, MORP: Data-driven multi-objective route planning and optimization for electric vehicles, Proc. ACM Interact. Mob. Wearable Ubiquitous Technol. 1 (4) (2018) 1–35.
[11] S. Colonnese, F. Cuomo, G. Pagliari, L. Chiaraviglio, Q-SQUARE: A Q-learning approach to provide a QoE aware UAV flight path in cellular networks, Ad Hoc Netw. 91 (2019) 101872.
[12] E.W. Dijkstra, A note on two problems in connexion with graphs, Numer. Math. 1 (1) (1959) 269–271.
[13] R. Bellman, On a routing problem, Quart. Appl. Math. 16 (1) (1958) 87–90.
[14] L.R. Ford, Network flow theory, 1956.
[15] P.E. Hart, N.J. Nilsson, B. Raphael, A formal basis for the heuristic determination of minimum cost paths, IEEE Trans. Syst. Sci. Cybern. 4 (2) (1968) 100–107.
[16] D. Xu, A. Zhou, X. Zhang, G. Wang, X. Liu, C. An, Y. Shi, L. Liu, H. Ma, Understanding operational 5G: A first measurement study on its coverage, performance and energy consumption, in: Proceedings of ACM SIGCOMM, 2020, pp. 479–494.
[17] X. Yin, A. Jindal, V. Sekar, B. Sinopoli, A control-theoretic approach for dynamic adaptive video streaming over HTTP, in: Proceedings of ACM SIGCOMM, 2015, pp. 325–338.
[18] M. Jain, C. Dovrolis, End-to-end available bandwidth: Measurement methodology, dynamics, and relation with TCP throughput, ACM SIGCOMM Comput. Commun. Rev. 32 (4) (2002) 295–308.
[19] J. Huang, N. Ito, T. Oshiba, K. Satoda, T. Murase, PathRefiner: Accurate bandwidth estimation using large-sized virtual packets for high-speed networks, in: Proceedings of IEEE International Conference on Consumer Electronics, 2024, pp. 1–6.
[20] V.J. Ribeiro, R.H. Riedi, R.G. Baraniuk, J. Navratil, L. Cottrell, Pathchirp: Efficient available bandwidth estimation for network paths, in: Proceedings of the Passive and Active Measurement Workshop, 2003.
[21] H. Mao, R. Netravali, M. Alizadeh, Neural adaptive video streaming with pensieve, in: Proceedings of ACM SIGCOMM, 2017, pp. 197–210.
[22] Dash-Industry-Forum, dash.js, 2023, https://github.com/Dash-IndustryForum/dash.js.
[23] A. Narayanan, X. Zhang, R. Zhu, A. Hassan, S. Jin, X. Zhu, X. Zhang, D. Rybkin, Z. Yang, Z.M. Mao, et al., A variegated look at 5G in the wild: performance, power, and QoE implications, in: Proceedings of ACM SIGCOMM, 2021, pp. 610–625.
[24] H.-P. Kriegel, M. Renz, M. Schubert, Route skyline queries: A multi-preference path planning approach, in: Proceedings of IEEE ICDE, 2010, pp. 261–272.
[25] A. Felner, Position paper: Dijkstra's algorithm versus uniform cost search or a case against dijkstra's algorithm, in: Proceedings of the International Symposium on Combinatorial Search, vol. 2, (1) 2011, pp. 47–51.
[26] Openstreetmap contributors, 2017, Planet dump retrieved from https://planet.osm.org.https://www.openstreetmap.org.
[27] Q. He, G. Cui, X. Zhang, F. Chen, S. Deng, H. Jin, Y. Li, Y. Yang, A game-theoretical approach for user allocation in edge computing environment, IEEE Trans. Parallel Distrib. Syst. 31 (3) (2019) 515–529.
[28] CellMapper, Cellular Tower and Signal Map, 2025, https://www.cellmapper.net/.
[29] D. Koutsonikolas, Y.C. Hu, On the feasibility of bandwidth estimation in wireless access networks, Wirel. Netw. 17 (2011) 1561–1580.
[30] C.E. Shannon, Communication in the presence of noise, Proc. IRE 37 (1) (1949) 10–21.
[31] Y. Yuan, W. Wang, Y. Wang, S.S. Adhatarao, B. Ren, K. Zheng, X. Fu, VSiM: Improving QoE fairness for video streaming in mobile environments, in: Proceedings of IEEE INFOCOM, 2022, pp. 1309–1318.
[32] H. Zhang, A. Zhou, H. Ma, Improving mobile interactive video QoE via two-level online cooperative learning, IEEE Trans. Mob. Comput. (2022) 1.
[33] J. Zhang, Y. Zhang, E. Dong, Y. Zhang, S. Ren, Z. Meng, M. Xu, X. Li, Z. Hou, Z. Yang, X. Fu, Bridging the gap between QoE and QoS in congestion control: A large-scale mobile web service perspective, in: Proceedings of USENIX ATC, 2023, pp. 553–569.
[34] A. Hassan, A. Narayanan, A. Zhang, W. Ye, R. Zhu, S. Jin, J. Carpenter, Z.M. Mao, F. Qian, Z.-L. Zhang, Vivisecting mobility management in 5G cellular networks, in: Proceedings of ACM SIGCOMM, 2022, pp. 86–100.
[35] D. Huang, X. Tao, C. Jiang, S. Cui, J. Lu, Trace-driven QoE-aware proactive caching for mobile video streaming in Metropolis, IEEE Trans. Wirel. Commun. 19 (1) (2020) 62–76.
[36] Y. Guo, Q. Yang, F.R. Yu, V.C.M. Leung, Cache-enabled adaptive video streaming over vehicular networks: A dynamic approach, IEEE Trans. Veh. Technol. 67 (6) (2018) 5445–5459.
[37] D.K. Tapang, S. Huang, X. Huang, QoE-based server selection for mobile video streaming, in: 2020 IEEE/ACM Symposium on Edge Computing, SEC, 2020, pp. 435–439.
[38] W. Shi, Q. Li, R. Zhang, G. Shen, Y. Jiang, Z. Yuan, G.-M. Muntean, QoE ready to respond: a QoE-aware mec selection scheme for DASH-based adaptive video streaming to mobile users, in: Proceedings of ACM MM, 2021, pp. 4016–4024.
[39] Z. Zheng, Y. Ma, Y. Liu, F. Yang, Z. Li, Y. Zhang, J. Zhang, W. Shi, W. Chen, D. Li, et al., Xlink: QoE-driven multi-path QUIC transport in large-scale video services, in: Proceedings of ACM SIGCOMM, 2021, pp. 418–432.
[40] M. Zhu, F. Linsalata, S. Mura, L. Cazzella, D. Badini, U. Spagnolini, Exploiting age of information in network digital twins for AI-driven real-time link blockage detection, Comput. Netw. (2025) 111855.
[41] H. Lee, J. Flinn, B. Tonshal, RAVEN: Improving interactive latency for the connected car, in: Proceedings of the 24th Annual International Conference on Mobile Computing and Networking, 2018, pp. 557–572.
[42] 3GPP TS 23.501, System architecture for the 5G system (5GS), 2022, https://www.etsi.org/deliver/etsi_ts/123500_123599/123501/17.05.00_60/ts_123501v170500p.pdf.
[43] 3GPP TS 29.522, Network exposure function northbound APIs, 2022, https://www.etsi.org/deliver/etsi_ts/129500_129599/129522/17.05.00_60/ts_129522v170500p.pdf.