Cellulars Go Underground: Practical Obstacles and Effective Mitigation

Hongyi Wang[1,2], Jiaxing Qiu[1], Zhenhua Li[1]✉, Liangyi Gong[3], Feng Qian[4], Yang Li[1], Jiakun Du[1], Zichen Zhang[1], Jiawei Zhen[1], Yunhao Liu[1], Daliang Sun[2], Xiang Li[2], Yumeng Liang[5], Jie Wu[5]✉
[1]Tsinghua University [2]Xiaomi Technology Co. LTD [3]CNIC, CAS
[4]University of Southern California [5]China Telecom Cloud Computing Research Institute

Abstract

Underground cellulars are arguably the most sophisticated public network utilities in modern metropolises due to constrained environments and mobility-driven use cases. To date, their QoS has been largely unsatisfactory to users, especially on subway trains where video apps encounter frequent and lasting video stream stalls (VSSes). With the goal of deeply understanding and practically boosting underground cellular QoS, we conduct the first large-scale study of underground cellular networks, spanning 207 subway lines, 82K BSes, and 1.8M Android phones. Our study reveals the unique multi-stage signal attenuation pattern and the defective infrastructure deployment in today's subway cellular environments, which collectively contribute as the major causes of VSSes. To address this, we enable user devices to actively adapt to the environmental changes, and develop a comprehensive radio/link state collection module to effectively inform video apps. Our optimizations on device-side Android cellular management have reduced 39% VSSes in subway networks; our collaborative optimizations with popular video apps have further reduced 23% VSSes.

CCS Concepts

• **Networks** → **Mobile networks**; **Network measurement**; *Network performance analysis*.

Keywords

Cellular Network; Underground Subway Network; Android Cellular Management; Mobility Management

ACM Reference Format:
Hongyi Wang, Jiaxing Qiu, Zhenhua Li, Liangyi Gong, Feng Qian, Yang Li, Jiakun Du, Zichen Zhang, Jiawei Zhen, Yunhao Liu, Daliang Sun, Xiang Li, Yumeng Liang, Jie Wu. 2026. Cellulars Go Underground: Practical Obstacles and Effective Mitigation. In *The 32nd Annual International Conference on Mobile Computing and Networking (MobiCom '26), October 26–30, 2026, Austin, TX, USA.* ACM, New York, NY, USA, 16 pages. https://doi.org/10.1145/3795866.3796696

MobiCom '26, Austin, TX, USA

ACM ISBN 979-8-4007-2505-0/26/10
https://doi.org/10.1145/3795866.3796696

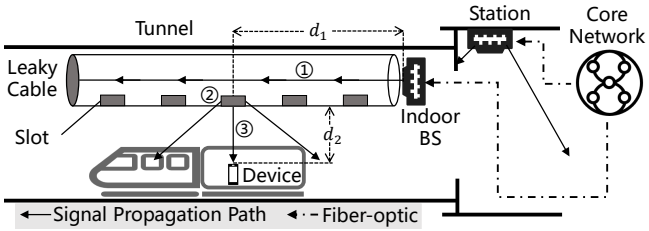

Figure 1: Architectural overview of a typical subway cellular network. Only downlink paths are plotted.

1 Introduction

Cellular networks are indispensable public infrastructures for communication and collaboration in modern society. While their regular deployments are proven to be robust and of high performance [46], they face unique challenges in underground deployments. Unlike overground deployments where base stations (BSes) can be strategically placed based on population density (e.g., in common residences) or along highways/railways, underground deployments are much more constrained and expensive. For example, deploying BSes and fiber-optic interconnect cables in underground stations and tunnels incurs more than 10× higher costs per coverage area, because space constraints force ISPs to adopt small-footprint indoor BSes (which are 5–10× cheaper per unit than regular macro BSes, but require 50–100× denser deployment due to their limited radio coverage [10]).

To balance deployment cost and signal coverage, ISPs usually employ a hybrid solution. In the representative case of subway networks as illustrated in Figure 1, indoor BSes and slotted leaky feeder cables (*leaky cables* for short) are used in combination. ISPs deploy indoor BSes sparsely within tunnels and connect them with leaky cables along tunnels. These cables radiate and aggregate signals through the slots to serve nearby user devices. Due to the substantially lower per-meter cost of leaky cables, this solution delivers cost-efficient cellular infrastructures in today's metropolises.

Motivation. Although underground cellulars have become everyday network utilities, they can hardly meet the Quality

of Service (QoS) requirements of modern mobile applications such as live news, entertainment, and video conferencing. According to our user survey, subway passengers' complaints are mostly (90%) around severe playback stalls of video streaming apps (both on-demand and live streaming).

Unfortunately, prior work on high-mobility networks, especially highway and high-speed railway (HSR) cellular networks [25, 35, 43, 48], can hardly apply to subway networks for at least three main reasons: (1) subway networks make use of indoor BSes rather than regular high-power BSes with broad area coverage; (2) leaky cables have a distinct impact on signal transmission; and (3) underground subway environments incur stronger and irregular ambient interference.

Measurement. To deeply and comprehensively understand the underground cellular QoS, we conduct the first large-scale study of subway cellular networks by collaborating with a major Android phone vendor (i.e., Xiaomi Co. LTD), spanning 207 subway lines across two countries (China and US), 82K BSes, and 1.8M Android phones (whose users took at least one subway ride), over a whole month (Aug. 2024). We focus on Video Stream Stalls (VSSes), the most reported and demanded problem by the surveyed users. We build a continuous monitoring infrastructure on a customized Android system dubbed SubCell, which collects fine-grained device/network state information when users engage with mobile video streaming apps using subway cellular networks.

Our study reveals frequent and lasting VSSes—on average, each user device accesses subway networks for 6 hours and encounters as many as 44 VSSes in the measurement month, where each VSS lasts for as long as 9 seconds. In particular, we find that the majority (70%) of VSSes are accompanied by a BS handover. Counter-intuitively, although 5G on average offers a higher throughput than 4G in overground settings [46], we observe the opposite in underground tunnels (5G: 1.1 Mbps vs. 4G: 1.4 Mbps). In contrast, 5G users experience less frequent VSSes, because 5G devices encounter 42.5% fewer low-downlink-throughput ($\leq$ 0.5 Mbps) events which are more likely to incur VSSes.

Analysis. To uncover the root causes of these observations, we delve into the underground signal attenuation and infrastructure deployment, which critically shape the cellular QoS and fundamentally differ from the overground situations.

Unique in-tunnel signal propagation incurs challenging wireless conditions. Different from the smooth logarithmic attenuation model [38] widely used by major ISPs, we note that underground signal attenuation is governed by a hybrid three-stage model as depicted in Figure 1: ① *Transmission loss* along the leaky cable, increasing linearly with the signal's travel distance inside the cable (d_1); ② *Radiation loss* at the slot where the signal emits into open space, determined by the slot's size, spacing, orientation, etc.; ③ *Propagation loss* from the slot to the device, whose complex pattern is shaped by not only the slot-device distance (d_2), but also the (tunnel + train body) "double-enclosure" structure incurring enormous signal reflections and multi-path effects[1].

MIMO-equipped leaky cables facilitate robustness under weak signals. For both 4G and 5G leaky cables, fast attenuation and strong interference may lead to lower-order modulation [20] (and thus lower throughputs). Nevertheless, we identify one of their major differences to be the adoption of MIMO: in our measurement, 77% of 5G leaky cables support MIMO, compared to only 7% for 4G. We ascribe this gap to the cost-effectiveness of MIMO deployment: a single 4G leaky cable already provides acceptable throughput in most (84%) cases, while a 5G cable frequently falls back to lower-order modulation (due to much faster signal attenuation) and necessitates the enhancement by MIMO. As a result, 5G maintains a relatively stable throughput under weak signal conditions compared to 4G, lowering the likelihood of VSSes.

Heterogeneous subway RAN (Radio Access Network) deployments cause frequent handovers. While 5G BSs are prevalent on subway platforms, inside subway tunnels 5G leaky cables register much less deployment than 4G cables (which are more mature and cost-effective). Such heterogeneity forces a 5G phone to frequently (every 124 seconds on average) switch between 4G and 5G modes (manifested as the 4G/5G *ping-pong handover* phenomenon [25]) while taking the subway, severely harming the cellular connection stability.

Mitigation. In today's cellular ecosystem, mobility management is mostly BS-centered. Based on user-device feedback and pre-configured thresholds, BSes infer the signal attenuation situation and command the handover process, which is however neither accurate nor timely for underground cellulars. To address this limitation, in contrast, we adopt a user-centric mobility management methodology.

First, we enable user devices (by modifying Android's cellular management module) to effectively characterize the underground three-stage signal attenuation with the *Time-Inhomogeneous State Space Model* (TISSM), which leverages a time-varying multi-order state space to finely capture the signal pattern in each stage. Based on TISSM, user devices can precisely identify the optimal BS handover and 4G/5G switch timing, and actively trigger the handover. In addition, since application-layer symptoms usually lag behind underlying signal and link dynamics, we develop a comprehensive radio/link state collection module in SubCell to offer apps useful low-level information, facilitating them to better predict VSSes and adjust video streaming strategies accordingly.

Since the release (on 10/20/2024) of the patched SubCell system with the optimized BS handover mechanism (adopted

[1]Hence, signal attenuation experienced by underground BSes is oftentimes 5–10× higher than that observed in high-speed railway scenarios.

by 5.2M users as of 11/30/2024), we have reduced 39% VSSes and 22% per-VSS duration in subway environments (as of 05/31/2025). We also convinced two major mobile video streaming platforms, Douyin (for on-demand video) and Youku Live (for live video), to practically use the radio/link state collection module (involving 2.4M users), achieving an additional 23% reduction in VSS occurrence and a 25% reduction in per-VSS duration. The two modules incur little overhead on an average user device: a total of <1% CPU utilization, <250 KB memory, and <1.1 MB storage.

We carefully examine the residual VSSes after optimization, finding that they mainly stem from BS signal coverage holes due to terrain limitations, deficient 5G leaky cable deployment, network congestion (esp. during peak hours), and resource bottlenecks (e.g., CPU/GPU contention) on low-end devices. Addressing these issues requires ISPs' providing more reliable signal coverage (e.g., with signal repeaters), accelerating underground 5G rollout, and expanding both terminal and core network capacity, as well as phone vendors' reducing the video decoding and rendering overhead.

Broader Impact. While our work focuses on subway networks, many of our findings and approaches can be extended to other challenging network scenarios (§8). For example, the TISSM-based signal attenuation characterization and the device-driven BS handover strategy can potentially apply to mountainous regions with uneven signal coverage, public transport hubs with intensive noises, underwater cellular networks with immense water resistance, and planet-wide 6G that integrates heterogeneous wireless infrastructures.

Contributions. Our contributions are as follows.

- We conduct the first large-scale and in-depth measurement study on app performance in underground subway cellular networks, identifying frequent and lasting VSSes as the key practical challenge that harms the QoS.
- We reveal the unique signal attenuation pattern and the defective infrastructure deployment in today's underground cellular environments, which collectively contribute as the major causes of VSSes.
- Through both system- and app-level enhancements, we effectively address the VSSes with remarkable real-world impacts on millions of subway passengers.

Code and Data Release. The VSS diagnosis/mitigation code and measurement data involved in the study are publicly available at https://UndergroundCellular.github.io.

Ethical Claim. Our study was reviewed and approved by Xiaomi's IRB. There are no ethical concerns in this work.

2 Study Methodology

Driven by a recent user survey (§2.1) we carried out regarding the QoS of cellular networks in high-mobility scenarios, we develop SubCell, a customized Android system that supports lightweight and in-depth information tracing beyond the capability of the vanilla Android system (§2.2). Based on SubCell, we conduct a large-scale, month-long measurement study on VSSes in subway networks by continuously monitoring 1.8M opt-in user devices (§2.3).

2.1 User Survey

According to our interview with Xiaomi that has hundreds of millions of Android phone users (most of which are located in China), it has been receiving users' reports regarding the poor QoS of cellular networks in high-mobility scenarios, including highways, subways, and high-speed railways (HSRs). In order to figure out which high-mobility scenario presents the most serious challenges to the QoS in a fair setting, we carried out a survey via Xiaomi in Jun. 2024 by recruiting 1,000 participants from its users, each of whom regularly experiences the three scenarios. The participants, aged between 20—50, were of different genders and occupations, and resided in various cities across China. Specifically, they were asked two simple yet critical questions:

- Among the three scenarios, which one brings you the most unsatisfactory cellular networking experience (given the same time duration of staying in each scenario)?
- In the chosen scenario, which app leads to the most pain (given the same usage time) and what is the major pain?

The survey results reveal that 724 participants consider subway networking to bear the worst QoS, followed by HSR networking (228) and highway networking (48). Further, 90% of the 724 participants state that they suffered the most from online video streaming apps (54% on-demand vs. 36% live), with video stream stalls (VSSes) being their primary concern.

2.2 Continuous Monitoring Infrastructure

To practically address the users' pain point on the QoS of subway networking, our measurement study requires 1) precise identification of underground subway scenarios, 2) real-time detection of user-perceived VSSes, and 3) fine-grained, cross-layer data collection for in-depth analysis. However, the vanilla Android system lacks native support for these capabilities. Also, we are unable to capture all the concerned information by simply developing an Android app, given that Android's inherent security restrictions prevent third-party apps from monitoring the status of other apps.

To overcome these limitations, we implement a customized Android system, dubbed SubCell, which enables continuous acquisition of system-level traces whenever users interact

with any of the top-10 popular video streaming apps (see Table 2 in Appendix A) in subway environments. We focus on the 10 apps whose user base accounts for over 90% of all the video streaming apps in China. We modify the Framework-layer programs instead of the hardware abstract layer (HAL) or the kernel layer ones—modifying the latter ones can help collect more underlying and detailed data, but can easily impair the system stability and robustness in practice [17]. SubCell is featured by the following functionalities.

Subway Scenario Identification. We employ a two-stage sensor-based method to detect when a device boards a subway[2]. In the first stage, SubCell detects the device's entering a subway station by identifying a successive increase in air pressure with the barometer data, which indicates the action of descending into an underground environment[3]. Once the pressure increase is detected, it further analyzes the pedometer data to inspect whether there are continuous walking activities both before and after the increase. If the condition is met, it concludes that the device has entered the station. In the second stage, we utilize a binary classifier for detecting train departures, which is based on Multi-Layer Perceptron (MLP) [34] and trained offline on a manually labeled dataset of 150K gyroscope and accelerometer time-series samples.

The above method enables SubCell to recognize subway entry and train departure with a high accuracy in real time. Specifically, the ML model is evaluated using 5-fold cross-validation on the 150K-sample dataset. Defining a successful detection as one falling within ±5s of a ground-truth event, our model achieves >90% precision and >95% recall. We filter out the false positive/negative cases in an offline manner. Similarly, SubCell is also capable of detecting train stops and the device's leaving a subway station, as well as a train's transitioning between underground and overground segments. We also use GPS readings when train stops to approximate locations for counting covered subway lines, a task that tolerates the limited underground positioning accuracy.

Real-Time VSS Detection. Once a device is determined to be in a subway environment, SubCell needs to detect and record VSSes happening to the ten target apps whenever they are in the foreground. Note that we focus on user-perceived visual smoothness (rather than solely on complete playback freezes) and quantify it with the frame rate, a standard metric reflecting the video playback smoothness.

To understand the general frame rate range of *perceptible* VSSes, we conduct a user study involving 50 volunteers of different ages (from 20 to 50) and genders. For each of the 10 apps, we first record a 20-second 30-FPS video when the app performs streaming. Then, we randomly select a one-second segment from the middle ten seconds of the video and progressively reduce its frame rate by 5-30 FPS with an interval of 5 FPS, thus producing 6 videos that contain lower-frame-rate segments. Afterwards, we ask each volunteer to watch the 6 videos in a randomized order and provide feedback on which videos s/he can perceive playback stalls on. The study results show that while not all the VSSes are perceptible to all users, the vast majority (83%) of them (whose frame rates are below 20 FPS) have *obvious* impacts on almost all users. Therefore, we set the threshold for determining an obvious VSS as 20 FPS, and in the remainder of this paper we concentrate on only these obvious VSSes.

To implement real-time frame rate detection, we make lightweight modifications to `SurfaceFlinger`, which is the graphics layer compositor of Android and produces the visual frames for display. We add a timer task to `SurfaceFlinger`, enabling it to count the number of composed graphics layers from the foreground app per second (i.e., the app frame rate). The result is then transmitted to our monitoring service with Andro's efficient Binder IPC (Inter-Process Communication) mechanism [4], realizing real-time frame rate monitoring.

Table 1: Cross-layer data sources collected by SubCell.

Category	Metrics	Data Source / API
Network Performance	RTT, Packet Loss Rate	`TcpInfo` via Netlink
	DNS Success Rate/Latency	`DnsResolver` API
	Uplink/Downlink Throughput	`TrafficStats` API
Radio Condition	RAT, RSRP, SNR,	Modem Chipset
	MCS, MIMO Layers	Customized APIs
BS Connection	MCC, MNC, LAC	`TelephonyManager` APIs
	CID	`ServiceState` APIs

Cross-Layer Data Collection. To investigate the root causes of VSSes, SubCell monitors system resource consumptions including CPU utilization, memory usage, and I/O throughput, as well as network states across three dimensions: network performance, radio condition, and BS connection. Table 1 summarizes the detailed metrics and their corresponding data sources. First, network performance data include the app-level round-trip time (RTT), packet loss rate, DNS query success rate, DNS query latency, and uplink/downlink throughput. Second, radio condition data comprise the Radio Access Technology (RAT), Reference Signal Received Power (RSRP), Signal-to-Noise Ratio (SNR), Modulation and Coding Scheme (MCS), MIMO layers, and Access Point Name (APN). Third, BS connection data include the Mobile Country Code (MCC), Mobile Network Code (MNC), Location Area Code (LAC), and Cell Identity (CID).

SubCell performs such detailed tracing using various Android system APIs. Specifically, we obtain RTT and packet

[2]Although cellular metadata such as Access Point Name and Cell Identity are available in our measurement, they do not reveal the precise physical locations of BSes, and thus cannot be used to identify subway scenarios.

[3]This pressure pattern is universally observable across various entry scenarios (e.g., from underground malls or parking structures), as long as there is a vertical transition from the surface to the subway depth.

loss rate by parsing the `TcpInfo` structure via Netlink sockets, retrieve DNS-related metrics using the `DnsResolver` API, and collect bandwidth metrics via the `TrafficStats` API. For radio-layer metrics, SubCell uses customized APIs provided by the modem chipset manufacturers to access modem logs in real time and parse them from the logs. Other information is accessed with APIs provided by the Android TelephonyManager and ServiceState services.

Overhead. Our modifications to Android introduce trivial overheads through lightweight frame rate monitoring and system-level logging, primarily utilizing existing interfaces. Also, the overheads are incurred only when any of the ten target apps are active in foreground in subway environments. Crucially, our monitoring infrastructure remains dormant on the client side during normal (non-subway) usage, thereby avoiding additional resource consumptions. Even on low-end Android devices, the daily resource consumption when traveling in subways and running a target app is limited to only <2% CPU utilization, <40 KB of memory, <500 KB of storage, and <500 KB of data usage per day.

2.3 Large-Scale Deployment

In July 2024, with the help from Xiaomi, we invited 60M users (through the Android system-level notifications) to participate in our measurement study of VSSes in subway environments by installing SubCell as a system update on their devices. This lightweight update had no impact on their installed apps, existing data, or OS version. Ultimately, 7.1M users consented to participate, contributing data for a whole month in August 2024. Among them, 1,818,629 users (25%) took subway rides during our measurement and used at least one of the ten target apps while on the subway. In our subsequent analysis, we focus solely on these users.

Although the vast majority of participants are located in China, there do exist 21 participants coming from the US, offering us the opportunity to make a comparative analysis when possible. All collected data were uploaded to our backend server via WiFi for centralized analysis.

Ethical Concerns. In the SubCell's update UI, we explicitly informed users about the runtime tracing to be collected. All participants were free to opt out during the measurement. No personally identifiable information (e.g., phone number, IMEI, and IMSI) was collected, and we never (and cannot) link our collected data to users' true identities.

3 Measurement Results

This section first presents the general statistics of our collected data in §3.1, and then describes our major observations from five perspectives in §3.2. We will analyze the root causes of these observations in §4 and §5, in terms of signal attenuation and infrastructure deployment respectively.

3.1 General Statistics

With the help from the 1,818,629 Android devices across 10 distinct phone models (see Table 3 in Appendix A), we record fine-grained, cross-layer traces with regard to 79,401,581 VSSes, involving 207 subway lines in two countries (204 lines in China and 3 lines in the US), four ISPs (three in China and one in the US), and 82,361 BSes, during August 2024. As far as we know, this is the largest dataset regarding VSSes in real-world underground cellular networks to date. In the following sections, we focus on the analysis of the large-scale dataset collected in China. We will discuss the generalization of our findings with the US dataset in §8.

We find that each device accesses subway networks for an average of 6 hours in the month. We then analyze the *prevalence* and *frequency* of VSSes, defined respectively as the proportion of devices experiencing at least one VSS and the average number of VSSes happening to a device. As shown in Figure 2, VSSes occur prevalently and frequently across all the 10 phone models. The majority (74%) of devices experience at least one VSS in the month, and as many as 44 VSSes occur to a device on average. Furthermore, the most affected device encounters as many as 958 VSSes during 62 hours of subway network access. We are also concerned with the duration of VSSes. As shown in Figure 3, the average duration of a VSS is as long as 9 seconds, thus severely degrading the user experience. The duration distribution is notably skewed: while 79.4% VSSes last for less than 10 seconds, the maximum duration reaches 252 seconds (= 4.2 minutes).

3.2 Impact Factor Analysis

To understand the root causes of the observed VSSes, we dissect the measurement results from five key dimensions: temporal patterns, device models, ISP discrepancies, BS connectivity, and radio access technology (RAT) choices.

Temporal Patterns. We observe distinct diurnal patterns in VSS durations. Long-lasting VSSes occur more frequently during commute hours (8:00–10:00, 17:00–19:00), when the high density of passengers results in significant mobile data usage, constantly pushing the cellular network to its capacity limit. This correlation highlights the impact of network congestion on VSS duration. Moreover, prolonged VSSes also occur when subway trains transition between underground and overground segments—although ISPs have deployed leaky cables in both areas, terrain changes substantially impair the effectiveness of leaky cables.

Device Models. Regarding the impact of device configurations, Figure 4 reveals that low-end devices (e.g., Model 1) are notably more VSS-susceptible than high-end devices (e.g., Models 5–10). This discrepancy echoes the fact that VSSes stem from not only network issues, but also client-side processing overheads such as decoding, rendering, and display.

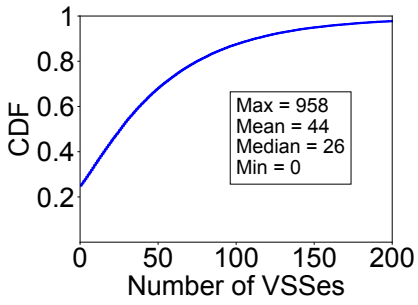

Figure 2: Number of VSSes happening to a single Android phone.

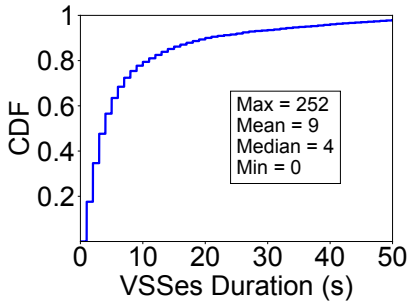

Figure 3: Duration of our recorded VSSes.

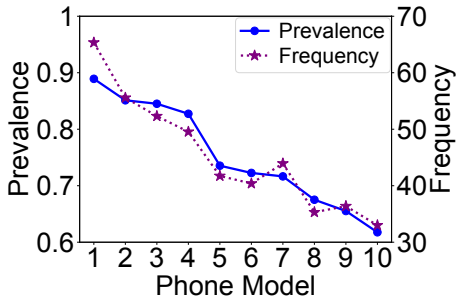

Figure 4: Prevalence and frequency of VSSes on each model of phones.

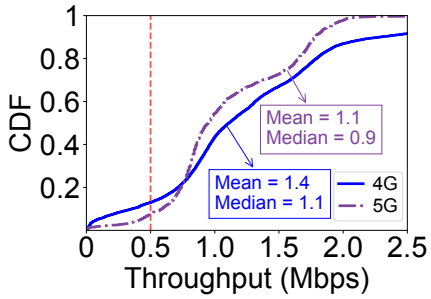

Figure 5: Throughput distribution under 4G and 5G connections.

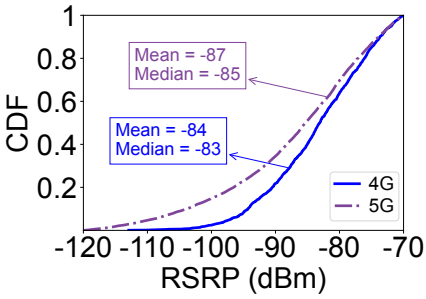

Figure 6: RSRP distribution under 4G and 5G connections.

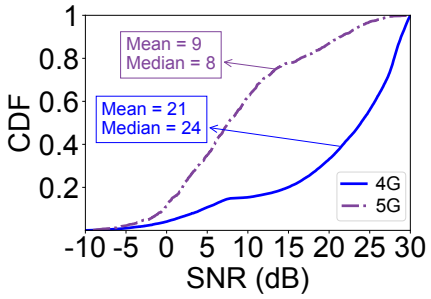

Figure 7: SNR distribution under 4G and 5G connections.

Low-end devices are inherently more prone to VSSes due to limited computational resources (in terms of CPU, memory, GPU, etc.). Besides, high-end devices usually have more sophisticated antenna optimizations, such as multiple-input multiple-output (MIMO) antenna systems and enhanced antenna arrays [30, 42], consequently experiencing fewer VSSes than low-end ones. However, when hardware configurations are comparable, a higher Android version does not translate to significantly fewer VSSes (e.g., Model 2 vs. Model 3), as major updates in Android 14 [5] do not include specific optimizations for underground cellular networking.

ISP Discrepancies. Our study involves three major ISPs in China (China Telecom, China Unicom, and China Mobile) and one in the US (T-Mobile). We focus on the three ISPs in China here and refer them subsequently as ISP-1, ISP-2, and ISP-3. Both the prevalence and frequency of VSSes are higher for the users of ISP-2 (88% and 47) than those of ISP-1 (77% and 40) and ISP-3 (68% and 43). This can be attributed to the fact that the 4G and 5G frequencies allocated to ISP-2 are the highest among the three ISPs, and high-frequency signals are more susceptible to propagation attenuation and interference. Furthermore, ISP-2 shares much of its network infrastructure with ISP-3, potentially leading to inter-modulation interference that further degrades ISP-2's signal quality.

BS Connectivity. We find that the majority (70%) of VSSes are accompanied by a BS handover, and that 94% of these handovers coincide with noticeable signal degradation. Surprisingly, in 86% of such cases, either the source or the target BS still provides accepted signal quality[4]. This indicates that these handovers are triggered at improper timing, either too early—switching from the source BS with sufficient signal quality to a weaker target BS, or too late—lingering on the source BS with degraded signal despite the availability of better target BSes. These findings suggest a close relation between suboptimal mobility management and VSSes.

RAT Choices. When in tunnels, user devices are connected to 4G networks for 74% of the time and to 5G networks for the remaining 26%. This distribution results from the ISPs' infrastructure deployment, which will be explained in §5.2. Counter-intuitively, although 5G generally provides a higher throughput than 4G in overground environments [46], we observe the opposite in underground tunnels. As shown in Figure 5, the average throughput of 5G (1.1 Mbps) is 21.4% lower than that of 4G (1.4 Mbps). More surprisingly, despite the lower average throughput, user devices experience less frequent VSSes under 5G connections. Specifically, they encounter an average of 6.4 VSSes per hour on 5G, compared to 7.6 VSSes per hour on 4G. We reveals that this stems from the different frequencies of low-downlink-throughput (≤ 0.5 Mbps) events experienced by devices when under 4G and

[4]According to the 3GPP standards [1] and our interviews with ISPs, we regard poor signal quality as RSRP < -100 dBm or SNR < 5 dB.

5G connections. Such events, which have a high likelihood (64.7%) of leading to VSSes, occur 42.5% less frequently on 5G. In detail, as illustrated in Figure 5, the probability of encountering these events is 13.1% under 4G but 7.5% under 5G. We will elaborate on the underlying reasons in §5.1.

4 Characterizing Signal Attenu- ation in Subway Environments

To understand the root causes of the aforementioned observations, we first delve into the underground signal attenuation process, which critically shapes the cellular QoS and fundamentally differs from the overground situation.

Unlike the smooth logarithmic attenuation model [38] widely used by today's major ISPs, which generally holds true in overground environments, our detailed analysis of the underground signal propagation process (as illustrated in Figure 1) reveals that the underground attenuation is governed by a hybrid three-stage model. We dissect the three stages in §4.1, including transmission loss, radiation loss, and propagation loss. We then analyze the impact of this model on the final signal quality received by user devices in §4.2.

4.1 Hybrid Three-Stage Model

To characterize the underground signal attenuation, we break down the end-to-end signal propagation path into three components and leverage a hybrid three-stage model that captures the distinct attenuation behavior of each component. We detail these three loss components as follows.

Transmission Loss. The transmission loss ($Loss_{tr}$) refers to the signal attenuation incurred during its propagation through the leaky cable, increasing linearly with the signal's travel distance inside the cable (d_{tr}) and can be modeled as

$$Loss_{tr} = \alpha_{tr} \times d_{tr}, \tag{1}$$

where α_{tr} is the attenuation coefficient. The value of α_{tr} depends on both the signal frequency and the cable's physical design. Specifically, for a given cable, higher-frequency signals exhibit larger α_{tr} and thus increased transmission loss. Moreover, leaky cables that support higher-frequency transmission tend to adopt more confined structures and denser dielectric materials, which amplify both skin effect [18] and dielectric loss [36], ultimately leading to a higher α_{tr}.

Radiation Loss. The radiation loss ($Loss_{ra}$) denotes the loss at the slot where the signal emits from the leaky cable to the open space. $Loss_{ra}$ is primarily determined by the slot design parameters, including the slot size, spacing, orientation, etc.. For a given leaky cable and carrier frequency, $Loss_{ra}$ remains nearly constant at each slot.

Propagation Loss. The propagation loss ($Loss_{pr}$) refers to the signal attenuation that occurs during its propagation from the slot to the device, comprising multiple components. The fundamental component is the free-space loss ($Loss_{pr}^{free}$), which denotes the attenuation experienced by the signal in unobstructed conditions. Its standard expression [38] is:

$$Loss_{pr}^{free} = 20\lg(d_{pr}) + 20\lg(f) + 20\lg\left(\frac{4\pi}{c}\right), \tag{2}$$

where d_{pr} is the distance between the slot and the device, f is the signal frequency, and c is the speed of light in vacuum. This logarithmic attenuation model is widely adopted by ISPs and forms the basis of their planning and strategies.

Beyond this, the unique "double-enclosure" structure (the tunnel and the train body) introduces enormous signal reflections and diffraction, thereby leading to severe multi-path fading (Δmp) [29]. As a result, devices experience significant fluctuations in received signal strength, oftentimes showing 5—10 × higher attenuation than that observed in high-speed railway scenarios. Furthermore, signals also undergo frequent penetration impairment (Δpen) when intervening obstacles such as the train body, seats, and human bodies. Accordingly, we model the total propagation loss as

$$Loss_{pr} = Loss_{pr}^{free} + \Delta mp + \Delta pen. \tag{3}$$

4.2 Device-Side Signal Quality

Based on the above three-stage model, which quantifies the attenuation along a single propagation path through one slot, we now analyze the actual signal strength received by user devices. In practice, a device receives signal from multiple slots along the leaky cable. The received signal strength $RSRP(t)$ can therefore be viewed as the aggregated contribution from all currently serving slots, minus additional attenuation caused by inter-slot interactions (which are not included in the three-stage model capturing the per-slot loss).

As the train moves forward, the set of serving slots gradually shifts along the cable. Importantly, while individual slots are replaced over time, the relative geometric configuration between the device and its serving slots remains approximately stable within the coverage region. As a result, the radiation loss and free-space propagation loss contributions from different slots largely offset each other across adjacent time instances. Consequently, the dominant factor driving $\Delta RSRP(\delta_t)$ is the increase in the signal's transmission distance inside the leaky cable. Assuming the train moves at speed v and there are $N(t)$ serving slots, the variation in RSRP can be approximated as:

$$\Delta RSRP(\delta_t) \approx N(t) \cdot \alpha_{tr} \cdot v \cdot \delta_t + \Delta_{\text{noise}}, \tag{4}$$

where Δ_{noise} captures fluctuations caused by multi-path fading, penetration impairment, and inter-slot interactions.

This formulation reveals that RSRP generally exhibits a linear decline over time while the device remains within the cable's coverage region. Regarding the noise term Δ_{noise}, our

measurement data reveals that it poses fluctuations on the RSRP value as well as the slope of the linear trend.

When the device approaches the edge of the leaky cable coverage, the number of serving slots decreases (i.e., $N(t+\delta_t) < N(t)$). This reduction accelerates the decline in received signal strength, causing the slope of the aforementioned linear trend to increase in magnitude. The noise termalso contributes fluctuations to this acceleration.

In summary, our analysis reveals that underground signal attenuation follows a highly dynamic model (i.e., a linear trend with an accelerating slope and volatile fluctuations), deviating significantly from the smooth logarithmic model assumed by ISPs[5]. This unique complexity also distinguishes subway networks from HSR scenarios (where device-side signal quality is primarily affected by Doppler shifts). Consequently, existing solutions, whether designed for overground or HSR contexts, fail to capture the irregular and rapid signal fluctuations inherent to the "double-enclosure" subway environment, highlighting the necessity for a fine-grained, real-time signal modeling approach.

5 Analyzing the Impact of Infrastructure Deployment

In this section, we analyze how underground infrastructure decisions shape cellular performance. We first examine the physical-layer factors that contribute to the QoS gap between 4G and 5G in tunnels (that 5G delivers lower average throughput yet fewer severe low-throughput events) (§5.1). We then connect these properties to deployment decisions on underground platforms and in tunnels, and show how the 4G/5G heterogeneity drives frequent (and sometimes ill-timed) inter-RAT ping-pong handovers that degrade connection stability (§5.2).

5.1 Physical-Layer Factors Contributing to the 4G/5G Performance Gap

As discussed in §3.2, in underground tunnels, 5G on average offers a lower throughput than 4G, yet devices under 5G connections encounter fewer low-downlink-throughput events. To demystify this counter-intuitive observation, we dive into lower-layer physical parameters, including signal quality, modulation and coding schemes (MCS), and MIMO layers, and analyze their impact on throughput.

Signal Quality. As shown in Figure 6 and Figure 7, 5G exhibits significantly poorer signal quality than 4G. Its average RSRP is 3.6% lower than that of 4G, and more notably, the average SNR is 57.1% lower. This is due to 5G's faster signal attenuation and weaker anti-interference capability. Within the underground "double-enclosure" structure described in Section 4.1, 5G faces more severe multi-path fading yet exhibits lower resilience, substantially reducing the SNR.

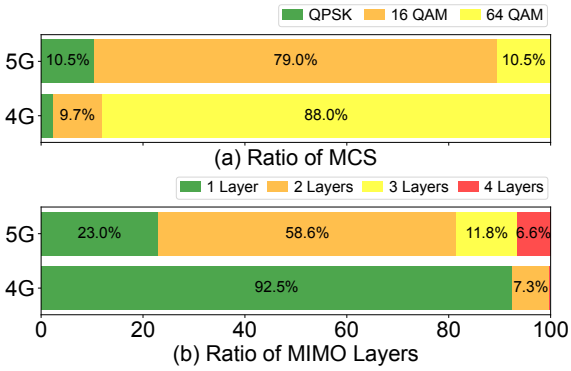

Figure 8: MCS utilization percentages and MIMO layers for 4G and 5G leaky cables.

Modulation and Coding Scheme. The modulation and coding scheme (MCS) is typically selected based on the device-side channel quality indicator (CQI), which is determined by metrics such as RSRP and SNR. For both 4G and 5G leaky cables, fast attenuation and strong interference may lead to lower-order modulation, which sacrifices throughput for transmission robustness. However, due to the significantly poorer signal quality observed on 5G, a 5G leaky cable frequently falls back to lower-order modulation, while a 4G cable is able to sustain higher-order modulation in most cases. In our measurements, both 4G and 5G cables support up to 64 QAM. As shown in Figure 8(a), 5G uses only 16 QAM in 79.0% of the cases, and even falls back to QPSK in 10.5% of the cases. In contrast, 4G maintains 64 QAM in the majority of cases (88.0%). This explains the lower average throughput observed under 5G connections.

MIMO layers. We observe a distinct disparity in MIMO adoption between 4G and 5G leaky cable deployments. As shown in Figure 8(b), 77% of 5G leaky cables support MIMO, compared to only 7% for 4G. This gap stems from the cost-effectiveness of leaky cable–based MIMO deployment: unlike the overground deployment where MIMO can be enabled by adding antennas, equipping leaky cables with MIMO requires deploying additional leaky cables, considerably increasing the deployment cost. For instance, 2×2 MIMO can nearly double the cost. Meanwhile, a 4G single leaky cable generally delivers acceptable throughput in most cases (84%), whereas a 5G cable's frequent utilization of lower-order modulation makes the single-cable throughput insufficient and thus necessitates the enhancement by MIMO. As a result, 5G maintains relatively stable throughput under weak signal conditions, lowering the likelihood of VSSes.

[5]The logarithmic model yields the following RSRP variation:

$$\Delta RSRP'(\delta_t) = \frac{\ln(10)}{20} \cdot \frac{v \cdot \delta_t}{d_p(t)}, \tag{5}$$

where $d_p(t)$ denotes the distance between the device and the BS at time t.

5.2 4G/5G Ping-Pong Handover

Regarding the RAN (Radio Access Network) deployments, we observe that influenced by the aforementioned factors, subway deployments differ significantly from their overground counterparts [46]. Specifically, although 5G BSes are installed on most subway platforms, only ~25% of subway tunnels are equipped with 5G leaky cables. This is largely because 4G cables are more mature and cost-effective. As previously discussed, 5G cables rely heavily on MIMO while equipping leaky cables with MIMO incurs high costs. Moreover, many subway lines were constructed before 5G emerged, making retrofitting tunnels for 5G coverage both financially and technically challenging.

The unique heterogeneous 4G/5G deployment in subway cellular networks forces a 5G device to frequently (every 124 seconds on average) switch between 4G and 5G modes (manifested as the 4G/5G ping-pong handover phenomenon) while taking the subway. Worse yet, ISPs' mobility management based on the smooth logarithmic attenuation model often triggers inter-RAT handovers (i.e., 5G→4G or 4G→5G) either too early or too late in underground settings. Consequently, these frequent and poorly timed ping-pong handovers severely compromise cellular connection stability.

In particular, we observe that 59.5% of 5G→4G handovers, which typically occur when devices transition from platforms into tunnels, suffer from low RSRP at the source BS. This is because platform 5G BSes are optimized for open areas and are poorly suited to signal transmission in enclosed tunnels. This infrastructure limitation results in very high signal attenuation rates as devices enter tunnels. This causes the handover timing, inferred based on ISPs' assumptions of gradual and slow overground attenuation, to be too late.

For 4G→5G handovers, the target BS manifests a low SNR (despite a normal RSRP) in 41.9% cases, primarily due to ISPs' preference for 5G networks. To prioritize 5G connectivity, ISPs often configure user devices—via the currently connected 4G BS—to switch to a nearby 5G BS as long as its RSRP exceeds a relatively low RSRP threshold, even if the 4G signal remains strong. While this approach works well for reliable cellular connectivity in overground settings, it is unfit for underground 4G→5G handovers, which typically occur near tunnel exits with severe ambient interference. Thus, devices frequently undergo premature 4G→5G handovers, where the platform 5G BS's RSRP seems adequate yet its SNR has not reached a usable level.

6 Mitigation

In this section, we introduce two complementary mitigation layers: a system-level enhancement (§6.1) that recalibrates device-side handover decisions, and an app-level enhancement (§6.2) that exposes low-level radio/link state for proactive app streaming strategies. We then evaluate their impact at scale (§6.3), and examine the residual VSSes to identify remaining bottlenecks and opportunities (§6.4).

6.1 System-Level Enhancement

While there exist mobility management optimizations for HSRs, they mainly target Doppler-induced unreliable channel measurements or require precise GPS/trajectory information [25, 35], rendering them inapplicable to subway tunnels where structural interference dominates and GPS is not precise enough. In today's underground cellular ecosystem, mobility management remains mostly BS-centered. ISPs configure BSes with static handover thresholds based on their pre-assumed signal attenuation model (i.e., the smooth logarithmic model). When a device connects to a BS, these handover parameters are passed to the device. The device then monitors signal quality metrics (e.g., RSRP, RSRQ) periodically and reports to the serving BS once certain conditions are met. Based on these reports, the BS infers the signal attenuation trend and commands the handover process.

However, this mechanism is neither accurate nor timely in underground environments. As analyzed in §4, underground handover patterns are highly irregular even around the same BS, making static BS-centered or route-dependent enhancements ineffective. To address this, we propose a user-centric mobility management methodology with real-time signal modeling. By strategically calibrating measurement reports sent from a user device to the serving BS in Android's cellular management module, we can guide the handover timing to align with the underground attenuation pattern.

Specifically, the hybrid three-stage pattern affects device-side signal quality differently across stages. Moreover, to ensure real-time responsiveness, the model must update after each periodic measurement (typically <300 ms). Therefore, we build our method on the lightweight Time-Inhomogeneous State Space Model (TISSM) [22], which employs a time-varying multi-order state space to finely capture the signal pattern in each stage with limited computational overhead.

Concretely, we model the underground signal attenuation state at time t as a vector $\mathbf{x_t} = \left[R_t, b_t, a_t\right]^\top$, where R_t represents the device-side RSRP, b_t denotes the attenuation rate (i.e., the gradient of RSRP), and a_t is the gradient of the attenuation rate which reflects the rate acceleration. Note that we carry both the first-order attenuation rate and its second-order acceleration in $\mathbf{x_t}$ based on our analysis in §4.2, which shows that when the device leaves leaky cable coverage, the RSRP trajectory reflects not only the linear attenuation caused by transmission loss, but also a slope acceleration from the diminishing number of serving slots.

Specifically, the state evolves from t to $t + \Delta t$ as $\mathbf{x_{t+\Delta t}} =$

$$\begin{bmatrix} R_{t+\Delta t} \\ b_{t+\Delta t} \\ a_{t+\Delta t} \end{bmatrix} = \mathbf{M x_t} + \omega_{(\mathbf{t})} = \begin{bmatrix} 1 & \Delta t & \frac{\Delta t^2}{2} \\ 0 & 1 & \Delta t \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} R_t \\ b_t \\ a_t \end{bmatrix} + \begin{bmatrix} \omega_{r(t)} \\ \omega_{b(t)} \\ \omega_{a(t)} \end{bmatrix}, \quad (6)$$

where $\omega_{r(t)}$, $\omega_{a(t)}$, and $\omega_{b(t)}$ represent the noise terms of R_t, b_t, and a_t, caused by multi-path fading, inter-slot interactions, etc.. To embody the transition, we first acquire recent RSRP measurements of the serving BS, which are sampled periodically by the user device. With these samples, we adopt a two-stage Kalman filter framework [13] which iteratively **estimates** the attenuation state using prior system dynamics, as well as **corrects** the state by fusing real-time RSRP observations to minimize the impact of noise. The detailed modeling process is described in Appendix B.

Each RSRP measurement drives an update to the attenuation state $\mathbf{x}$, enabling real-time prediction of signal degradation and data-driven handover guidance. Regardless of the ISPs' pre-set thresholds, upon each RSRP measurement, SubCell determines whether to trigger handover according to what would happen in the *next* measurement period. If a handover is estimated to result in an RSRP below the accepted level (indicating that the optimal handover timing precedes the next measurement), the device proactively sends a measurement report to the serving BS to initiate the handover.

Furthermore, to mitigate the impact of ISPs' blind priority for 5G networks as discussed in §5.2, we augment the BS-provided handover criteria with device-side SNR awareness. Specifically, for 4G→5G handovers, a candidate 5G BS is reported to the serving 4G BS only if (1) its RSRP exceeds the threshold and (2) its SNR reaches a noise-resilient level ($\geq 5dB$). This could effectively prevent premature handovers. We also extend this SNR-aware strategy to other handover types, particularly intra-frequency handovers susceptible to co-channel interference. Provided that the RSRP is above the acceptable level, we trigger handovers earlier when the serving BS's SNR drops critically low, or defer them if the target BS's SNR is insufficient, ensuring handovers are executed with both sufficient signal strength and reliable link quality.

To implement our device-driven handover strategy, Xiaomi engaged in industrial collaborations with modem chipset manufacturers (Qualcomm and MediaTek), who provided customized APIs to SubCell. These APIs enable the system to parse raw modem logs in real-time to retrieve handover configurations and RSRP/SNR measurements, and to send customized measurement reports based on TISSM predictions and SNR conditions. Specifically, SubCell follows the standard event-based handover procedures and constructs the appropriate reports (A3 events for intra-frequency handovers, B1/B2 events for inter-RAT handovers, and measurement reports indicating that the serving cell's RSRP is lower than the threshold for other types [2]) based on current handover configurations and observed candidate cells.

Note that we only optimize the *timing* of sending measurement reports to the serving BS. The actual handover execution remains under the strict control of BSes. Specifically, the target BS retains full authority to perform admission control; if it is congested, it will reject the handover request regardless of when the report is sent. Thus, our approach does not bypass BSes' load balancing or fairness mechanisms.

6.2 App-Level Enhancement

VSSes are oftentimes a result of multiple contributors at different layers, such as the physical-layer radio conditions and the link-layer BS connectivity dynamics. However, since application-layer symptoms usually lag behind these underlying dynamics, app-level streaming strategies are often too late to be effective. To bridge this gap, we develop a comprehensive radio/link state collection module in SubCell to offer apps useful low-level information. Specifically, the module exposes the following metrics to upper-layer apps every second (synchronizing with information collection cycles):

- **Radio state** includes signal quality metrics (i.e., RSRP, SNR, and RSRQ), MIMO layers, and MCS.
- **Link state** includes cellular connectivity events, particularly BS handover events, BS dwell time [3], and RAT.

We collaborate with two major mobile video streaming platforms, Douyin (for on-demand video) and Youku Live (for live video), and integrate information from our radio/link state collection module into their streaming strategies. By combining the low-level information with application-layer information (e.g., client buffer level, content type), they can effectively predict potential upcoming VSSes. Building on this, they further enhance streaming strategies to balance playback smoothness and video quality. Instead of forcing the lowest quality (which hurts user experience), both platforms employ proactive adaptive bitrate (ABR). In detail, they adopt specific mechanisms tailored to their content types:

- **VSS-Informed Pre-Caching for Douyin.** Aside from normal pre-caching[6], when a VSS is predicted to happen in 5 seconds (the threshold is set empirically with internal benchmarks), Douyin will also start pre-caching subsequent video segments and proactively reduce the bitrate according to the signal metrics we provide.
- **Risk-Aware Bitrate Adaptation for Youku Live.** Youku Live augments the origin FESTIVE algorithm [19] by integrating VSS probability into the bandwidth estimation:

[6]We adopt on-demand pre-caching rather than aggressive pre-downloading as user behaviors in short-form video apps are highly unpredictable (e.g., frequent skipping). The latter would consume excessive cellular data without substantially improving QoE.

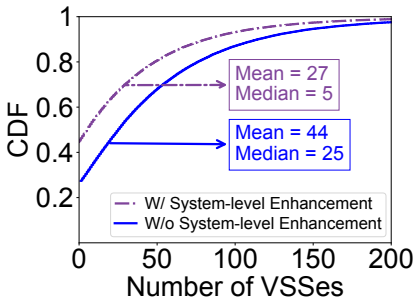

Figure 9: Number of VSSes happening on a single phone with and without our system-level enhancement.

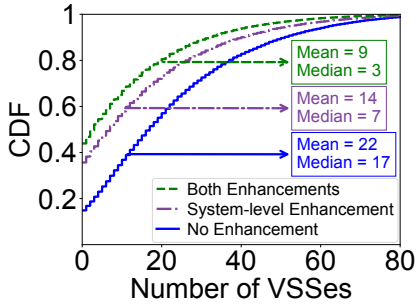

Figure 10: Number of VSSes happening to the two collaborative mobile apps on a single phone.

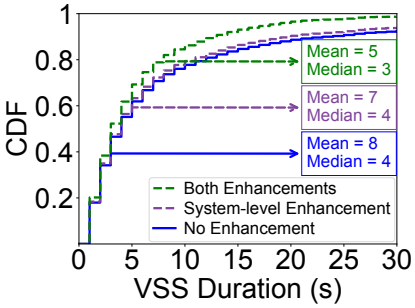

Figure 11: Duration of VSSes happening to the two collaborative apps.

$$B_{adjusted} = \hat{B} \times (1 - \alpha p), \tag{7}$$

where $\hat{B}$ is the historical weighted average and $\alpha = 0.5$ is the empirically optimized sensitivity coefficient. This adjustment proactively reduces bitrate when p rises, achieving 17% reduction of VSSes compared to baseline FESTIVE.

6.3 Deployment and Evaluation

Controlled Benchmark. To quantify the impact of our mitigation efforts, we first implemented the above two-fold enhancements in SubCell and conducted controlled experiments to validate them under identical conditions. We selected 30 devices covering the 10 models listed in Table 3 (3 units per model: one baseline, one with system-level enhancement, and one with both system- and app-level enhancements). We conducted controlled measurements on three representative subway lines in Beijing by carrying all 30 devices simultaneously along the same routes, and evaluating them in both commute and non-commute periods, using Douyin and Youku Live as workload. The benchmark results show that our enhancements effectively reduce VSSes across diverse scenarios. In particular, the system-level enhancement alone reduces VSSes by 32% on average, while the app-level enhancement further reduces VSSes by 26%.

Specifically, the system-level optimization achieves a more significant VSS reduction during non-commute hours compared to commute hours. This is likely because severe network congestion during commute hours forces target BSes to reject more handover requests for load balancing, partially offsetting the benefits of our optimized handover timing. In contrast, the app-level enhancement shows consistent improvements across both periods, as it relies on the VSS prediction derived from cross-layer information, whose accuracy is less affected by congestion fluctuations. Furthermore, the reduction ratio is higher on high-end devices than on low-end ones, as low-end devices suffer more from non-network-induced VSSes (e.g., rendering stalls, as noted in §3.2), which our network-oriented solutions cannot address.

Large-Scale Deployment. We then conducted large-scale evaluation by releasing the patched SubCell to the original 7.1M opt-in users (c.f. §2.3) in 10/20/2024. 5.2M of them upgraded their systems as of 11/30/2024. Among them, 2.4M users also updated Douyin or Youku Live to the latest versions with our app-level enhancement, and agreed to shared data with us for 7 months. We compare users who upgraded with those who did not over the same seven-month period (instead of measurement results in §3). We find no statistically-significant differences in device models or subway usage patterns between upgraded and non-upgraded users.

Regarding the system-level enhancement, as shown in Figure 9, it has reduced 39% VSSes across all the ten concerned apps (as listed in Table 2) on an average phone. Moreover, this enhancement reduces the average duration of a VSS by 22% ($9s \rightarrow 7s$). For Douyin and Youku Live, as illustrated in Figure 10, the system-level enhancement has reduced 36% of their VSSes on an average phone, and the app-level enhancement has further reduced 23% of their VSSes on an average phone. In other words, our two-fold enhancements can prevent the majority (59%) of their VSSes from happening.

In addition, Figure 11 shows that for Douyin and Youku Live, the system-level enhancement has shortened their average VSS duration by 13%, and the app-level enhancement has further shortened their average VSS duration by 25%, adding up to a remarkably 38% reduction in per-VSS duration ($8s \rightarrow 5s$). We attribute this to two factors. First, SubCell mitigates severe signal degradations that would otherwise cause long-duration VSSes. Second, for unavoidable VSSes, apps can sustain fluent playback longer during initial network deterioration and recover faster once connectivity improves.

To evaluate the overhead of SubCell, we perform benchmark experiments using the 10 phone models listed in Table 3 under representative workloads. The results show that our optimizations incur little overhead to a typical user device: a total of <1% CPU utilization, <250 KB memory, <1.1 MB storage, and <5 mAh of battery consumption per hour.

Ethical Concerns. All analysis tasks were adhered to the agreement established with users. No personally identifiable information (e.g., phone number, IMEI) was collected.

6.4 Examination on Residual VSSes

While our two-fold enhancements substantially reduce VSSes, a non-trivial tail (41%) persists. We examine traces that still exhibit VSSes after applying SubCell, and find that they are largely driven by constraints outside the scope of user-centric mobility control. Addressing the residual VSSes requires coordinated efforts from ISPs and phone vendors.

We identify three main classes of causes: radio-side, network-side, and device-side. On the radio side, we observe signal coverage holes in certain tunnel segments, as well as insufficient underground 5G leaky-cable provisioning, which jointly yield deep signal attenuation that client policies can hardly overcome. Beyond radio access, we observe increased RTT and reduced network throughput in peak hours even without degradation of radio access quality, indicating that congestion on indoor BSes and/or the public Internet (e.g., oversubscribed CDN) can lead to VSSes despite stable radio. Finally, device-side resource bottlenecks, especially on low-end or overheated phones, cause video pipeline stalls. For instance, we observe CPU/GPU contention and hardware decoder failures that trigger software fallback, which can manifest as VSSes even with adequate link budget.

Mitigating the long tail therefore calls for complementary actions from both ISPs and device vendors. On one side, to reduce VSSes caused by insufficient radio coverage and network congestion, ISPs should close underground coverage holes (e.g., with signal repeaters), invest more in 5G leaky-cable deployments (preferably with MIMO support), and expand both terminal and core network capacity along the path. On the other side, phone vendors can help combat VSSes by improving the reliability of the video pipeline, and tuning scheduler/thermal behavior for foreground streaming. Together, these efforts complement our system-level and app-level enhancements by addressing the infrastructure and platform limits that our system cannot directly control.

7 Related Work

The increasing penetration of cellular networks in our daily lives has spurred a plethora of researches on their performance, overhead, and reliability across diverse environments. While considerable efforts have been dedicated to optimizing the overground cellular QoS, the unique challenges presented by underground environments, particularly in metropolitan subway systems, have been rarely addressed.

Existing researches have extensively analyzed the cellular networks in general contexts, including urban [6, 44], rural [16, 45], and vehicular environments [11, 40], from various perspectives like signal coverage, network traffic, connection reliability [26, 27], and data rate [46]. Zooming in on a bandwidth- and/or latency-sensitive application scenario, studies on mobile video streaming have highlighted the severe influence of cellular networks' performance fluctuations, thereby proposing a variety of innovations to improve user experience, including adaptive bitrate streaming [14, 23, 47], caching [28, 32, 37], QoS prediction/modeling [31, 33, 41, 50], and so forth. However, they seldom take into account the specific characteristics of underground environments.

There do exist investigations of cellular networks in tunnels and underground spaces [7, 9, 12, 39, 49]. They identify the challenges of constrained radiowave propagation, high user density, and frequent BS handovers. They also highlight the impact of train movement and structural interference on network connectivity. Nonetheless, they typically focus on signal-level propagation modeling and physical-layer solutions (e.g., antenna deployment). They rarely examine how the unique tunnel environment interacts with ISPs' mobility management protocols, or how these interactions ultimately impact the end-user application-level Quality of Experience (QoE), such as the playback smoothness of online video streaming. Moreover, they are conducted at fairly small scales or with limited data, making it difficult to generalize their findings and solutions to large-scale subway systems.

Prior researches have also touched on the mobility management in cellular networks [15, 24, 27]. Most of them are conducted in overground settings with regular macro BSes, and thus can hardly cope with underground subway systems equipped with indoor micro/pico BSes [49]. While measurement studies of cellular networks in highway or HSR environments offer some relevant insights [8, 21, 25, 35, 43, 48], they address fundamentally different challenges. HSR handovers mainly suffer from Doppler-induced unreliable channel measurements, which can be mitigated by delay-Doppler-domain signaling. Conversely, subway handovers suffer from the unique "double-enclosure" structure (tunnel + train body) and unbalanced 4G/5G deployment. These factors cause faster signal attenuation and much more irregular handover patterns than overground scenarios, rendering static BS-centered or route-dependent enhancements ineffective. Therefore, our work necessitates a user-centric, real-time signal modeling approach distinct from HSR solutions.

8 Discussion

Generalization. For regions with 4G-only underground coverage (common in Europe and the US), though they would not suffer from the 4G/5G ping-pong handovers, handovers around tunnel exits and entrances still face similar challenges as those in China. Taking our US dataset for example, we

still observe that serving BSes suffer from abrupt signal attenuation when devices enter tunnels due to the enclosure structure, and BSes near tunnel entrances/exits exhibit normal RSRP but low SNR due to severe ambient interference. Yet, ISPs' handover strategies still assume a gradual signal attenuation rate and rely solely on RSRP thresholds, neglecting SNR conditions [2]. Thus, our real-time SNR-aware signal modeling (detailed in §6) remains effective in 4G-only scenarios by triggering handovers at more appropriate timings.

Limitations. The subway identification method relies on the barometer sensor to detect the pressure increase upon entering an underground station. Consequently, SubCell does not support devices lacking a barometer or scenarios where users board at above-ground stations. However, these limitations do not substantially undermine the practical value of our work: barometers are widely available in modern smartphones (e.g., over 60% of Xiaomi devices), and most subway stations are underground (e.g., >85% in Beijing). For devices without barometers and above-ground scenarios, our system can be extended by fusing location information with motion sensor data to identify train boarding events in future work.

Moreover, despite the effectiveness of our optimizations, they are subject to network capacity and device capabilities. During extreme congestion (e.g., peak commute hours), the target BS may reject handover requests for load balancing purposes, partially offsetting the benefits of our optimized handover timing. Also, for low-end devices, a portion of VSSes may stem from hardware bottlenecks rather than network issues. Thus, the mitigation of our solutions is inherently capped on these devices compared to high-end ones.

Lessons Learned. Beyond specific optimizations for subway networks, our study offers broader insights for future cellular infrastructure and protocol designs. *First, MIMO is critical for robustness, not just throughput.* While MIMO is often deployed to boost data rates in open areas, we find its spatial diversity is even more vital in confined environments (e.g., tunnels) to counteract severe signal degradation. Future deployments in challenging environments can leverage MIMO capabilities to enhance connectivity stability. *Second, mobility management should incorporate device-side intelligence into the BS-centered control loop.* In highly dynamic environments where signal patterns are irregular, the BS-side information may be coarse and lagged. By capturing real-time signal fluctuations, user devices can assist in making more appropriate decisions. *Third, cross-layer information sharing is essential for app resilience.* Existing apps rely on passive throughput estimation, which often fails under rapid fluctuations. Our results suggests a need for operating systems to provide richer network context APIs for apps to enable more proactive adaptation strategies, fostering a more collaborative optimization between the system and apps.

Broader Impact. Most of our experiences gathered in this paper can be applied to other challenging scenarios of cellular networks that deviate from common assumptions. Although our identified multi-fold factors that heavily impact the QoS are based on the measurements in China and the US, they can be reused by subway networks in other areas across the world since these factors essentially stem from international protocol standards (e.g., 3GPP) or de facto infrastructure standards (e.g., indoor BS plus leaky cable deployment).

Apart from subways, many of our findings and optimizations can be extended to underground marketplaces and public transport hubs with fast attenuation and intensive noise, as well as mountainous regions with non-uniform signal coverage, especially when video telephony, online meeting, or multi-player gaming are executed there which require high and stable bandwidth in harsh environments.

In fact, even emerging scenarios like underwater cellular networks (with immense water resistance) and planet-wide 6G (that integrates satellite and terrestrial wireless infrastructures) can benefit from our two-fold enhancements with actionable adaptions. At the system level, the flexibility of our TISSM-based signal model allows for its application to complex environments, by simply adjusting the state transition matrix to align with the gradient variety in the multi-stage signal attenuation. At the app level, our collected low-level radio/link state information can be effectively utilized for timely performance prediction and resource scheduling, so as to ensure seamless QoS across very different links.

9 Conclusion

This paper presents our efforts towards understanding and addressing the QoS problems in underground cellular networks, with a focus on the VSSes in subway environments. Despite considerable existing researches on overground cellular performance, underground cellular networking remains largely unexplored. We fill this knowledge gap through a large-scale, crowdsourcing-based measurement study with the help from millions of Android users. Our results illustrate multi-fold factors that harm the subway cellular QoS, particularly the unique signal attenuation pattern, the defective infrastructure deployment, and the BS-centered mobility management. Driven by the insights, we devise both system- and app-level enhancements to practically mitigate the QoS problems. Our experiences provide valuable guidelines and actionable solutions for mobile ISPs, phone vendors, and app developers to foster a seamless and robust cellular ecosystem.

ACKNOWLEDGMENTS

We appreciate Lei Yang, Dan Xiao, and Yongqing Li for their help in data collection and analysis. We sincerely thank our shepherd and the anonymous reviewers for their insightful

comments. This work is supported in part by the National Natural Science Foundation of China (NSFC) under grants 62332012, 62472245 and 62272440. Zichen Zhang contributed to this work during his visit to Tsinghua University. Jie Wu is also with Temple University.

References

[1] 3GPP.org. 2020. 3GPP TS 36.101. https://www.3gpp.org/ftp/Specs/archive/36_series/36.101/36101-i40.zip.

[2] 3GPP.org. 2020. 3GPP TS 36.331. https://www.3gpp.org/ftp/tsg_ran/WG2_RL2/Specifications/202012_draft_specs_after_RAN_90/Draft_36331-ck0.docx.

[3] Ibtihal Ahmed Alablani and Mohammed Amer Arafah. 2021. Applying a Dwell Time-based 5G V2x Cell Selection Strategy in the City of Los Angeles, California. *IEEE Access* 9 (2021), 153909–153925.

[4] Android.org. 2021. Binder IPC. https://source.android.com/docs/core/architecture/hidl/binder-ipc.

[5] Android.org. 2025. Android 14. https://developer.android.com/about/versions/14.

[6] Tianyang Bai, Rahul Vaze, and Robert W Heath. 2014. Analysis of Blockage Effects on Urban Cellular Networks. *IEEE Transactions on Wireless Communications* 13, 9 (2014), 5070–5083.

[7] Ángel Carro-Lagoa, Tomás Domínguez-Bolaño, José Rodríguez-Piñeiro, Miguel González-López, and José A García-Naya. 2019. Feasibility of LTE for Train Control in Subway Environments Based on Experimental Data. In *Proc. of IEEE EUSIPCO*. IEEE, 1–5.

[8] Haotian Deng, Chunyi Peng, Ans Fida, Jiayi Meng, and Y Charlie Hu. 2018. Mobility support in cellular networks: A measurement study on its configurations and implications. In *Proc. of ACM IMC*. 147–160.

[9] Dirk Didascalou, Jürgen Maurer, and Werner Wiesbeck. 2001. Subway Tunnel Guided Electromagnetic Wave Propagation at Mobile Communications Frequencies. *IEEE Transactions on Antennas and Propagation* 49, 11 (2001), 1590–1596.

[10] EnerSys. 2021. The 5G Power Architecture. https://www.enersys.com/493bb4/globalassets/documents/marketing-literature/esg/communications/white-papers/amer/the-5g-power-architecture.pdf.

[11] Xiaohu Ge, Hui Cheng, Guoqiang Mao, Yang Yang, and Song Tu. 2016. Vehicular Communications for 5G Cooperative Small-cell Networks. *IEEE Transactions on Vehicular Technology* 65, 10 (2016), 7882–7894.

[12] UK Government. 2022. GB Rail Tunnel Signal Propagation and Wireless Connectivity Options. https://assets.publishing.service.gov.uk/media/62fb693ce90e076bb81da6ea/11160-DfT_220303_DfT_Tunnel_Propagation_Study_FINAL_accessible1.pdf.

[13] Mohinder S Grewal and Angus P Andrews. 2014. *Kalman Filtering: Theory and Practice with MATLAB*. John Wiley & Sons.

[14] Yashuang Guo, F Richard Yu, Jianping An, Kai Yang, Chuqiao Yu, and Victor CM Leung. 2020. Adaptive Bitrate Streaming in Wireless Networks with Transcoding at Network Edge Using Deep Reinforcement Learning. *IEEE Transactions on Vehicular Technology* 69, 4 (2020), 3879–3892.

[15] Ahmad Hassan, Arvind Narayanan, Anlan Zhang, Wei Ye, Ruiyang Zhu, Shuowei Jin, Jason Carpenter, Z Morley Mao, Feng Qian, and Zhi-Li Zhang. 2022. Vivisecting Mobility Management in 5G Cellular Networks. In *Proc. of ACM SIGCOMM*. 86–100.

[16] Kurtis Heimerl, Kashif Ali, Joshua Blumenstock, Brian Gawalt, and Eric Brewer. 2013. Expanding Rural Cellular Networks with Virtual Coverage. In *Proc. of USENIX NSDI*. 283–296.

[17] Jann Horn. 2021. Mitigations Are Attack Surface, Too. https://googleprojectzero.blogspot.com/2020/02/mitigations-are-attack-surface-too.html.

[18] Paweł Jabłoński, Dariusz Kusiak, and Tomasz Szczegielniak. 2020. Analytical-Numerical Approach to the Skin and Proximity Effect in Lines with Round Parallel Wires. *Energies* 13 (2020), 6716.

[19] Junchen Jiang, Vyas Sekar, and Hui Zhang. 2012. Improving Fairness, Efficiency, and Stability in Http-based Adaptive Video Streaming with FESTIVE. In *Proc. of ACM CoNEXT*. 97–108.

[20] Rostand A. K. Fezeu, Claudio Fiandrino, Eman Ramadan, Jason Carpenter, Lilian Coelho de Freitas, Faaiq Bilal, Wei Ye, Joerg Widmer, Feng Qian, and Zhi-Li Zhang. 2024. Unveiling the 5G Mid-Band Landscape: From Network Deployment to Performance and Application QoE. In *Proc. of ACM SIGCOMM*. 358–372.

[21] Michail Kalntis, José Suárez-Varela, Jesús Omaña Iglesias, Anup Kiran Bhattacharjee, George Iosifidis, Fernando A Kuipers, and Andra Lutu. 2024. Through the Telco Lens: A Countrywide Empirical Study of Cellular Handovers. In *Proc. of ACM IMC*. 51–67.

[22] Marko Laine. 2020. Introduction to Dynamic Linear Models for Time Series Analysis. *Geodetic Time Series Analysis in Earth Sciences* (2020), 139–156.

[23] Wenzhong Li, Xiang Li, Yeting Xu, Yi Yang, and Sanglu Lu. 2023. Metaabr: A Meta-learning Approach on Adaptative Bitrate Selection for Video Streaming. *IEEE Transactions on Mobile Computing* 23, 3 (2023), 2422–2437.

[24] Yuanjie Li, Haotian Deng, Jiayao Li, Chunyi Peng, and Songwu Lu. 2016. Instability in Distributed Mobility Management: Revisiting Configuration Management in 3G/4G Mobile Networks. In *Proc. of ACM SIGMETRICS*. 261–272.

[25] Yuanjie Li, Qianru Li, Zhehui Zhang, Ghufran Baig, Lili Qiu, and Songwu Lu. 2020. Beyond 5G: Reliable Extreme Mobility Management. In *Proc. of ACM SIGCOMM*. 344–358.

[26] Yang Li, Hao Lin, Zhenhua Li, Yunhao Liu, Feng Qian, Liangyi Gong, Xianlong Xin, and Tianyin Xu. 2021. A nationwide study on cellular reliability: Measurement, analysis, and enhancements. In *Proc. of ACM SIGCOMM*. 597–609.

[27] Yuanjie Li, Jiaqi Xu, Chunyi Peng, and Songwu Lu. 2016. A First Look at Unstable Mobility Management in Cellular Networks. In *Proc. of ACM HotMobile*. 15–20.

[28] Zhuqi Li, Yaxiong Xie, Ravi Netravali, and Kyle Jamieson. 2023. Dashlet: Taming Swipe Uncertainty for Robust Short Video Streaming. In *Proc. of USENIX NSDI*. 1583–1599.

[29] Yu Liu, Ammar Ghazal, Cheng-Xiang Wang, Xiaohu Ge, Yang Yang, and Yapei Zhang. 2017. Channel Measurements and Models for High-Speed Train Wireless Communication Systems in Tunnel Scenarios: A Survey. *Science China Information Sciences* 60 (2017), 101301.

[30] Yuanzhi Liu, Mustapha CE Yagoub, and Mohammed Nassor. 2020. Omni-directional Antenna Array with Improved Gain for 5G Communication Systems. In *Proc. of IEEE USNC/URSI*. IEEE, 33–34.

[31] Gerui Lv, Qinghua Wu, Weiran Wang, Zhenyu Li, and Gaogang Xie. 2022. Lumos: Towards Better Video Streaming QoE Through Accurate Throughput Prediction. In *Proc. of IEEE INFOCOM*. IEEE, 650–659.

[32] Ge Ma, Zhi Wang, Miao Zhang, Jiahui Ye, Minghua Chen, and Wenwu Zhu. 2017. Understanding Performance of Edge Content Caching for Mobile Video Streaming. *IEEE Journal on Selected Areas in Communications* 35, 5 (2017), 1076–1089.

[33] Zili Meng, Xiao Kong, Jing Chen, Bo Wang, Mingwei Xu, Rui Han, Honghao Liu, Venkat Arun, Hongxin Hu, and Xue Wei. 2024. Hairpin: Rethinking Packet Loss Recovery in Edge-based Interactive Video Streaming. In *Proc. of USENIX NSDI*. 907–926.

[34] Fionn Murtagh. 1991. Multilayer Perceptrons for Classification and Regression. *Neurocomputing* 2, 5-6 (1991), 183–197.

[35] Yunzhe Ni, Feng Qian, Taide Liu, Yihua Cheng, Zhiyao Ma, Jing Wang, Zhongfeng Wang, Gang Huang, Xuanzhe Liu, and Chenren Xu. 2023.

POLYCORN: Data-driven Cross-layer Multipath Networking for High-speed Railway through Composable Schedulerlets. In *Proc. of USENIX NSDI*. 1325–1340.

[36] David M Pozar. 2021. *Microwave engineering: theory and techniques*. John wiley & sons.

[37] Jian Qiao, Yejun He, and Xuemin Sherman Shen. 2016. Proactive Caching for Mobile Video Streaming in Millimeter Wave 5G Networks. *IEEE Transactions on Wireless Communications* 15, 10 (2016), 7187–7198.

[38] Theodore S. Rappaport, Yunchou Xing, George R. MacCartney, Andreas F. Molisch, Evangelos Mellios, and Jianhua Zhang. 2017. Overview of Millimeter Wave Communications for Fifth-Generation (5G) Wireless Networks—With a Focus on Propagation Models. *IEEE Transactions on Antennas and Propagation* 65 (2017), 6213–6230.

[39] Md Abdus Samad, Sung-Woong Choi, Chung-Sup Kim, and Kwonhue Choi. 2023. Wave Propagation Modeling Techniques in Tunnel Environments: A Survey. *IEEE Access* 11 (2023), 2199–2225.

[40] Syed Danial Ali Shah, Mark A Gregory, Shuo Li, Ramon dos Reis Fontes, and Ling Hou. 2022. SDN-based Service Mobility Management in MEC-enabled 5G and Beyond Vehicular Networks. *IEEE Internet of Things Journal* 9, 15 (2022), 13425–13442.

[41] Yi Sun, Xiaoqi Yin, Junchen Jiang, Vyas Sekar, Fuyuan Lin, Nanshu Wang, Tao Liu, and Bruno Sinopoli. 2016. CS2P: Improving Video Bitrate Selection and Adaptation with Data-driven Throughput Prediction. In *Proc. of ACM SIGCOMM*. 272–285.

[42] Aliasghar Tarkhan. 2020. Receive Antenna Selection in Uplink Smart Antenna MIMO Systems. In *Proc. of IEEE Comnetsat*. IEEE, 121–124.

[43] Jing Wang, Yufan Zheng, Yunzhe Ni, Chenren Xu, Feng Qian, Wangyang Li, Wantong Jiang, Yihua Cheng, Zhuo Cheng, et al. 2019. An Active-passive Measurement Study of TCP Performance over LTE on High-speed Rails. In *Proc. of ACM MobiCom*. 1–16.

[44] Fengli Xu, Yong Li, Huandong Wang, Pengyu Zhang, and Depeng Jin. 2016. Understanding Mobile Traffic Patterns of Large Scale Cellular Towers in Urban Environment. *IEEE/ACM Transactions on Networking* 25, 2 (2016), 1147–1161.

[45] Elias Yaacoub and Mohamed-Slim Alouini. 2020. A Key 6G Challenge and Opportunity—Connecting the Base of the Pyramid: A Survey on Rural Connectivity. *Proc. of the IEEE* 108, 4 (2020), 533–582.

[46] Xinlei Yang, Hao Lin, Zhenhua Li, Feng Qian, Xingyao Li, Zhiming He, Xudong Wu, Xianlong Wang, Yunhao Liu, Zhi Liao, et al. 2022. Mobile access bandwidth in practice: Measurement, analysis, and implications. In *Proc. of ACM SIGCOMM*. 114–128.

[47] Xu Zhang, Yiyang Ou, Siddhartha Sen, and Junchen Jiang. 2021. SENSEI: Aligning Video Streaming Quality with Dynamic User Sensitivity. In *Proc. of USENIX NSDI*. 303–320.

[48] Zhehui Zhang, Yuanjie Li, Qianru Li, Jinghao Zhao, Ghufran Baig, Lili Qiu, and Songwu Lu. 2022. Movement-based Reliable Mobility Management for Beyond 5G Cellular networks. *IEEE/ACM Transactions on Networking* 31, 1 (2022), 192–207.

[49] Xiaoyin Zhao, Fei Qi, and Weiliang Xie. 2023. From Architecture to Field Trial: A Scheme of mmWave Based IAB and Small Base Station of Commercial Frequency for the Communication of Subway Tunnel Scenario. *IEEE Access* (2023).

[50] Yuhan Zhou, Tingfeng Wang, Liying Wang, Nian Wen, Rui Han, Jing Wang, Chenglei Wu, Jiafeng Chen, Longwei Jiang, et al. 2024. AUGUR: Practical Mobile Multipath Transport Service for Low Tail Latency in Real-Time Streaming. In *Proc. of USENIX NSDI*. 1901–1916.

Table 2: Top-10 popular video streaming apps involved in our study and their main functionalities.

ID	App	Functionality	# Users
1	Douyin	On-demand	755M
2	Douyin Lite	On-demand	521M
3	Kuaishou	On-demand	425M
4	Wechat Video Telephony	Live	354M
5	Tencent Video	On-demand	344M
6	Kuaishou Lite	On-demand	273M
7	Bilibili	On-demand	227M
8	Youku Live	Live	174M
9	Huya	Live	46M
10	Tencent Meeting	Live	35M

Table 3: Hardware configurations of our studied 10 phone models, generally ordered from low-end to high-end. The rightmost two columns correspond to the phone's Android version and user percentage.

Model	CPU	Memory	Storage	Android	Users
1	2.4GHz	8GB	256GB	13.0	8.29%
2	2.6GHz	12GB	256GB	14.0	8.01%
3	2.84GHz	12GB	256GB	13.0	5.81%
4	2.84GHz	16GB	512GB	13.0	6.44%
5	3.0GHz	12GB	256GB	14.0	14.98%
6	3.0GHz	16GB	512GB	14.0	16.17%
7	3.0GHz	16GB	1TB	14.0	6.62%
8	3.3GHz	12GB	256GB	14.0	12.05%
9	3.3GHz	16GB	512GB	14.0	14.11%
10	3.3GHz	16GB	1TB	14.0	7.52%

A Apps and Device Models Involved in Our Study

Table 2 lists the top-10 popular video streaming applications monitored in our study, covering both on-demand streaming and live streaming. Table 3 summarizes the hardware configurations of the 10 phone models involved in our study, detailing their CPU frequencies, memory sizes, storage capacities, Android versions, and user percentages.

B Attenuation State Modeling in TISSM

This appendix details the modeling of device-side signal attenuation in TISSM. Concretely, we model the underground signal attenuation state at time t as a vector $\mathbf{x_t} = \left[R_t, b_t, a_t\right]^\top$, where R_t represents the device-side RSRP, b_t denotes the attenuation rate (i.e., the gradient of RSRP), and a_t is the gradient of the attenuation rate which reflects the rate acceleration. We leverage $\omega_{r(t)}$, $\omega_{a(t)}$, and $\omega_{b(t)}$ to represent the noise terms of R_t, b_t, and a_t. Then, the state evolves from t to $t + \Delta t$ in four steps as follows.

- Step 1: Attenuation rate gradient update

$$a_{t+\Delta t} = a_t + \omega_{a(t)}, \tag{8}$$

where $\omega_{a(t)}$ models the noise influencing rate variability;

- Step 2: Attenuation rate propagation
$$b_{t+\Delta t} = b_t + a_{t+\Delta t} \cdot \Delta t + \omega_{b(t)}, \tag{9}$$
where $\omega_{b(t)}$ captures the impact of noise on attenuation rate;
- Step 3: RSRP update
$$R_{t+\Delta t} = R_t + b_{t+\Delta t} \cdot \Delta t + \omega_{r(t)}, \tag{10}$$
where $\omega_{r(t)}$, analogous to $\omega_{a(t)}$ and $\omega_{b(t)}$, represents the impact of noise on the RSRP value;
- Step 4: The state transition from $\mathbf{x_t}$ to $\mathbf{x_{t+\Delta t}}$ is $\mathbf{x_{t+\Delta t}} =$
$$\begin{bmatrix} R_{t+\Delta t} \\ b_{t+\Delta t} \\ a_{t+\Delta t} \end{bmatrix} = \mathbf{M}\mathbf{x_t} + \omega_{(\mathbf{t})} = \begin{bmatrix} 1 & \Delta t & \frac{\Delta t^2}{2} \\ 0 & 1 & \Delta t \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} R_t \\ b_t \\ a_t \end{bmatrix} + \begin{bmatrix} \omega_{r(t)} \\ \omega_{b(t)} \\ \omega_{a(t)} \end{bmatrix}, \tag{11}$$
which can dynamically infer the current attenuation pattern based on historical attenuation states.

To implement this model, we adopt a two-stage Kalman filtering framework. Following each periodic RSRP measurement, the framework first **estimates** the attenuation state with prior system dynamics, and then **corrects** the state by fusing real-time RSRP observations to suppress noise.

In the **estimation** stage, the predicted state $\hat{\mathbf{x}}_{\mathbf{t+\Delta t}}$ is derived from the prior state $\mathbf{x_t}$ using the aforementioned equations. The noise terms, $\omega_{r(t)}$, $\omega_{b(t)}$, and $\omega_{a(t)}$, are modeled as zero-mean Gaussian processes:

$$\omega_{r(t)} \sim \mathcal{G}(0, \sigma^2_{r(t)}), \omega_{b(t)} \sim \mathcal{G}(0, \sigma^2_{b(t)}), \omega_{a(t)} \sim \mathcal{G}(0, \sigma^2_{a(t)}),$$

where variances $\sigma^2_{r(t)}$, $\sigma^2_{b(t)}$, and $\sigma^2_{a(t)}$ are estimated through Maximum Likelihood Estimation on historical data. These variances populate the diagonal covariance matrix $W_t = diag(\sigma^2_{r(t)}, \sigma^2_{b(t)}, \sigma^2_{a(t)})$, ensuring noise independence.

In the **correction** stage, the predicted state $\hat{\mathbf{x}}_{\mathbf{t+\Delta t}}$ is refined using the observed instantaneous state $\overline{\mathbf{x}}_{\mathbf{t+\Delta t}}$. First, the RSRP residual error is computed as

$$e_{t+\Delta t} = \overline{R}_{t+\Delta t} - \hat{R}_{t+\Delta t}. \tag{12}$$

Afterwards, the error covariance matrix P_t is updated as

$$P_{t+\Delta t} = \mathbf{M} P_t \mathbf{M}^\intercal + W_t, \tag{13}$$

where $\mathbf{M}$ is the state transition matrix in Equation (11).

The Kalman gain $K_{t+\Delta t}$ then weighs the influence of this residual on the state update

$$K_{t+\Delta t} = P_{t+\Delta t}\mathbf{F}^\intercal \left(\mathbf{F} P_{t+\Delta t}\mathbf{F}^\intercal\right)^{-1}, \tag{14}$$

where $\mathbf{F} = \begin{bmatrix} 1, 0, 0 \end{bmatrix}$ isolates the RSRP component for correction. Finally, the corrected state is obtained as

$$\mathbf{x_{t+\Delta t}} = \hat{\mathbf{x}}_{\mathbf{t+\Delta t}} + K_{t+\Delta t} e_{t+\Delta t}, \tag{15}$$

and $P_{t+\Delta t}$ is further updated as

$$P_{t+\Delta t} = (I - K_{t+\Delta t}\mathbf{F}) P_{t+\Delta t}, \tag{16}$$

for estimations in the next measurement period.