How Far Have Edge Clouds Gone? A Spatial-Temporal Analysis of Edge Network Latency In the Wild

Heng Zhang†, Shaoyuan Huang†, Mengwei Xu‡, Deke Guo†, Xiaofei Wang†*, Victor C.M. Leung‖, Wenyu Wang¶
† College of Intelligence and Computing, Tianjin University, Tianjin, China
‡ Beijing University of Posts and Telecommunications, Beijing, China
‖ College of Computer Science and Software Engineering, Shenzhen University, Shenzhen, China and also with the Department of Electrical and Computer Engineering, the University of British Columbia, Vancouver, Canada V6T 1Z4
¶ Paiou Cloud Computing (Shanghai) Co., Ltd, Shanghai, China
hengzhang@tju.edu.cn, hsy_23@tju.edu.cn, mwx@bupt.edu.cn, guodeke@gmail.com,
xiaofeiwang@tju.edu.cn, vleung@ieee.org, wayne@pplabs.org

***Abstract*—The emergence of next-generation latency-critical applications places strict requirements on network latency and stability. Edge cloud, an instantiated paradigm for edge computing, is gaining more and more attention due to its benefits of low latency. In this work, we make an in-depth investigation into the network QoS, especially end-to-end latency, at both spatial and temporal dimensions on a nationwide edge computing platform. Through the measurements, we collect a multi-variable large-scale real-world dataset on latency. We then quantify how the spatial-temporal factors affect the end-to-end latency, and verified the predictability of end-to-end latency. The results reveal the limitation of centralized clouds and illustrate how could edge clouds provide low and stable latency. Our results also point out that existing edge clouds merely increase the density of servers and ignore spatial-temporal factors, so they still suffer from high latency and fluctuations. Based on the observations, we propose a robust prototype edge cloud model based on lessons we learn from the measurement and evaluate its performance in the production environment. The further evaluation result shows that edge clouds achieve 84.1% latency reduction with 0.5ms latency fluctuation and 73.3% QoS improvement compared with the centralized clouds.**

***Index Terms*—Real-world Dataset Collection, Spatial-Temporal Modeling, Edge Clouds**

I. Introduction

Latency is a critical factor that affects user experience in applications. The emergence of next-generation latency-constrained (mission-critical) applications places strict requirements on latency and the stability of network services [1]. For example, autonomous driving requires low and stable latency to sense the road, and high latency will threaten driving safety [2]. VR/AR applications also require low and stable latency to render, and high latency will lead to vertigo [3].

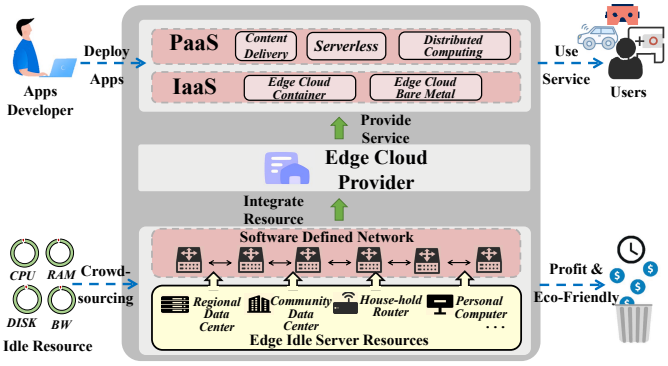


Fig. 1: Architecture of Crowd-Sourcing Edge Clouds.

To mitigate the high and unstable network latency of centralized clouds, edge computing has become a de-facto paradigm and has drawn extensive attention in both academia and industry. Intuitively, edge clouds can effectively guarantee the operation of latency-sensitive applications as edge clouds are deployed in high density. Major cloud providers such as Azure [4], AWS [5], and Alibaba [6] are actively extending their central clouds with small-to-medium-sized edge clouds.

Figure 1 illustrates the architecture of a unique form of edge platform: crowd-souring edge clouds. In the crowd-souring edge clouds, edge cloud provider deploys massive geographically distributed yet lightweight data centers (DCs), as well as rent idle resources in a crowd-sourcing manner. These resources are integrated into an edge cloud platform using software-defined networks, and edge cloud provider

Mengwei Xu was supported by the National Key R&D Program of China (Grant 2021ZD0113001). Victor C.M. Leung was supported by the Guangdong Pearl River Talent Recruitment Program (Grant 2019ZT08X603), the Guangdong Pearl River Talent Plan (Grant 2019JC01X235), Shenzhen Science and Technology Innovation Commission (Grant R2020A045), and the Canadian Natural Sciences and Engineering Research Council (Grant RGPIN-2019-06348). Xiaofei Wang was supported by the National Science Foundation of China (Grant 62072332), the China NSFC (Youth) (Grant 62002260), the China Postdoctoral Science Foundation (Grant 2020M670654), and the Tianjin Xinchuang Haihe Lab (Grant 22HHXCJC00002). Corresponding author: Xiaofei Wang (email: xiaofeiwang@tju.edu.cn).

979-8-3503-9973-8/23/$31.00 ©2023 IEEE

uses this platform to provide service like Infrastructures as a Service(IaaS) and Platforms as a Service(PaaS). All participants can benefit from this edge cloud pattern: users can get better quality of service, and the application developers can easily deploy their applications with the edge cloud providers' platform. The owner of idle resources can get paid by renting their idle resources, and the edge cloud provider can reduce the cost of construction.

However, the current deployment of edge clouds merely increases the density of servers, so most of the current edge clouds look like an extension of centralized clouds [7]. Due to a lack of consideration of spatial-temporal factors on latency, current edge clouds still suffer from high latency and fluctuations. In fact, there lacks of in-depth understanding of how edge clouds have been deployed in the real world and their implications of latency. Specifically, we seek to answer the following research questions about edge clouds: (1) How far from the users should we deploy edge clouds to minimize latency? (2) How about the network topology of edge clouds? Is it still meaningful to continue optimizing network topology for latency reduction in edge clouds? (3) How to mitigate the imbalance of latency and reduce its impact on Server Level Agreement (SLA)? (4) How does edge cloud latency change over time, and can we accurately predict latency variations of edge clouds? (5) How to offload users' requests in edge clouds when facing network congestion in rush hours?

To this end, we collect the dataset from a densely-distributed edge platform and perform a large-scale measurement study on edge-to-edge and in-the-wild network latency. The dataset contains 900 million PING records from 5,174 edge clouds – the largest edge deployment to our best knowledge. Unlike traditional edge cloud platforms, the one we studied is constructed through a *crowd-sourcing* manner as in figure 1: any individuals could install the software provided by the platform maintainer to turn their idle machines (e.g., PCs and workstations) into edge cloud servers.

The crowd-sourcing edge platform we studied mainly depends on renting idle resources by installing the software modified from K8s, so the construction cost is almost negligible. And the crowd-sourcing edge cloud platform only needs to pay for the actual resource used for task execution. One PC can install the software within 10 minutes no matter when and where, so the crowd-sourcing edge cloud has a broader deployment of servers and flexible resource provisioning in the spatial-temporal dimension.

So latency is the most significant factor that impacts the quality of service. Our measurements are focused on latency (end-to-end round-trip latency). We design a script-driven proactive latency probing tool and an information-gathering app to obtain a sufficiently wide range of edge cloud servers' metadata. Besides, unlike most previous investigations, which only analyzed the latency characteristics and did not actually evaluate their solutions in the real world, we also validate our suggestions with a prototype edge cloud based on lessons we learned from the measurements.

Key Observations. (1) The network latency to the edges almost linearly scales with the geographical distance, i.e., around 1.9ms per 100km. Placing edge clouds within 30km of the end users is often enough to saturate such benefits; further reducing the distance below 30km brings negligible improvements as limited by the core network architecture [8]. Instead, the number of intermediate hops does not impose as significant an impact as the distance.

(2) Routing across different Internet Service Providers (ISPs) adds a network latency as high as 20ms. It indicates that two edge clouds accessed through different ISPs, even geographically close, could have high inter-site network latency. Fortunately, we observe that IP prefix matching can effectively mitigate such network overhead.

(3) The edge clouds need dense deployment to mitigate the latency imbalance. The latency of the household network is just 2ms higher than the special line in the edge cloud.

(4) The latency has the same trend every day. The latency becomes 2 to 2.5 times higher during peak periods than during idle periods. So, the time-series algorithm is easy to predict the trend of latency. Our experiments based on DeeAR show that taking hourly granularity can increase the predicting accuracy at most 26.3%.

(5) Since edge clouds are distributed systems, collaboration usually happens between edge clouds. However, if an edge has abnormally high latency to one another edge, it will likely have high latency to 80% of all other edges. This phenomenon indicates that offloading without consideration of fine-grained features (like hourly granularity) may not reduce latency during rush hours.

Implications. (1) For crowd-sourcing edge cloud platforms, our results suggest that the edge cloud platform needs to provide edge resources within 30km to the end users and avoids cross-ISP routing as much as possible to deliver better network performance. Moreover, crowd-sourcing effectively deploys densely distributed edge clouds with almost no performance loss.

(2) For edge computing researchers, our measurements highlight the need for a spatial-temporal scheduling mechanism for a fine-grained (e.g., hourly granularity). We want the mechanism can precisely utilize the 20% relatively free network communication links. One possible design is to predict the latency trend using time-series models like DeepAR and make decisions using Deep Reinforce Learning.

(3) For the edge application developer, our results suggest that edge clouds are good for latency-critical applications. Communication time only consists of 30% of the most strict MTP latency threshold requirement. However, enjoying this superiority of latency also requires that future edge cloud-oriented applications face the problems posed by distributed infrastructure, such as reliability and consistency. More application development frameworks should be proposed to shield applications from the inconvenience of distributed infrastructure.

Contributions. We summarize our key contributions as follows:

(1) Dataset Collection. We cooperated with a commercial crowd-sourcing edge cloud provider to perform this measurement, finally collecting a total of 0.9 billion rows of end-to-end latency data with precise GPS location information and timestamp. [1]

(2) Factor Analysis and Modeling. The accurate multi-variate impact factor analysis and modeling demonstrate the problem of where and when we need edge clouds. We clarify why edge clouds can reduce latency. We also found some factors that may decrease the latency of edge clouds and proposed possible solutions.

(3) Prototype Edge Cloud. We design a prototype edge cloud based on the lessons we learn from factor modeling. The measurements show that the prototype edge cloud can achieve a stable 5ms latency on average, which can achieve at most 84.1% latency reduction with 0.5ms latency fluctuation and 73.3% QoS improvement.

II. Data Collect and Measurement

A. Crowd-sourcing Platform

Our research focused on a unique crowd-sourcing edge platform, an emerging paradigm for edge computing. Such unique crowd-sourcing characteristic allows its resources to sink to very close to end users – perhaps it is the first time the edge servers become "edge enough" as envisioned by edge researchers. Its deployment scale is significant, with a 10x-denser deployment than state-of-the-art edge platforms [9]. This large scale allowed us to observe the unique challenges and opportunities, such that IP prefix matching can mitigate communicating across-ISPs overhead. Even for conclusions that are well known, we for the first time push to its limit: e.g., while the community knows reducing distance can cut down the latency, it's not true when the distance is already below 30km due to the core network architecture. The platform has 5,174 edge servers, with each edge server running edge CDN, video transcoding, and streaming media distribution services.

The commercial edge cloud platform plans the number of crowd-sourcing recruitments according to the population in this area. If the number of participants is less than expected, the platform will adopt some incentives. Our measurements show that the number of devices and the population have a strong Pearson correlation coefficient of 0.667. All our data collection and the prototype edge cloud are based on this platform in the production environment. All 5174 real-world edge servers from this platform were involved in the entire data collection process and experiments. These services were deployed with a Kubernetes-like system. Hence, the topology and collaborations are like typical Kubernetes clusters.

B. Data Collection Framework

We design a framework to probe network latency, as shown in Figure 5. The entire framework is a script-driven proactive measurement tool consisting of three parts. **1) Coordinator:** All measurements are coordinated by the coordinator, including the sampling of edge cloud servers and the assignment of measurement tasks. The coordinator also collects the measured logs at the end of each round. **2) Edge cloud servers:** We utilize edge cloud servers from the commercial edge cloud platform from PPIO as candidate edge measurement nodes. **3) NTP Server:** To ensure the accuracy and consistency of the measurement timestamps, we synchronize the time of the coordinator and all participating edge cloud servers with the NTP server.

The probing process is divided into the following steps:

Step ①: Task Dispatching. The coordinator generates sampling tasks (a list of edge clouds to be probed) and dispatches sampling tasks to all edge cloud servers. The task generation repeats in a certain time interval. To ensure that the massive edge servers complete the data upload, we set the time interval to **5** minutes.

Step ②: Time Synchronizing. Before executing the task, every edge cloud synchronizes its system time with the NTP server. Previous studies, such as Pingmesh [10], often ignore time synchronization. Synchronizing the time ensures the accuracy of the timestamps for every measurement record. Accurate timestamps allow us to understand better how latency changes over time.

Step ③: Task Execution. For every edge cloud server in the measurement task, the probing edge cloud server first sends 32 PING packets to them. After the 32 PING packets are sent, the edge cloud server calculates the average latency of these 32 packets(the unit of latency is millisecond(ms)) for every edge cloud server to be probed. Then the probing edge cloud server uses *traceroute* to obtain the number of hops between the edge cloud server and other edge cloud servers in the measurement task.

Step ④: Result Uploading. After executing the measurement task, the edge cloud server uploads the measured latency, *hops*, machine ids and timestamp to the coordinator.

Besides, we built a proactive information collection app for the edge cloud platform's device providers, allowing device managers to proactively upload additional information such as GPS, ISP, network type, and the online/offline status of edge cloud servers. We collected the latency probing data from 11/27/2021 to 12/17/2021 (21 days total). The full size of the dataset is 181 GB, containing 0.94 billion records. These data cover all the mainstream Internet Service Providers (ISPs).

In this paper, we collect PING-based latency. PING works on OSI-Layer 3. We are using PING to measure the latency of the infrastructure network more accurately. Most previous latency measurement studies are also based on PING. In fact, the higher the layer of the OSI, the more latency is affected by factors and errors. For example, OSI-Layer4 latency depends on congestion control. Although PING is not representative of latency at OSI layers 4 to 7, there existing algorithms for optimizing such latency and they are the same as centralized clouds. E.g., BBR, TCP-Fast-Open, and HTTP3.0. Moreover, we have discussed the applicability.

[1] https://github.com/henrycoding/IWQoS23EdgeMeasurements

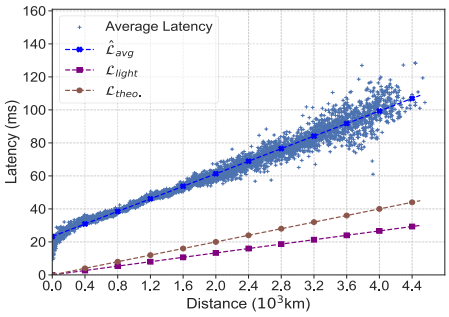


Fig. 2: Scatter plot of distance and average latency between edge clouds.

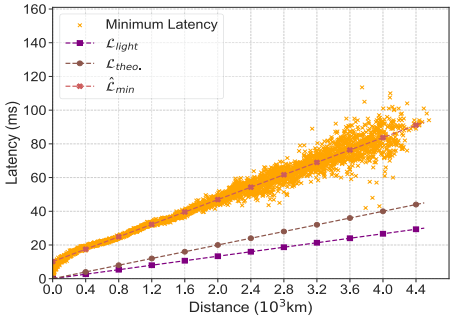


Fig. 3: Scatter plot of distance and minimum latency between edge clouds.

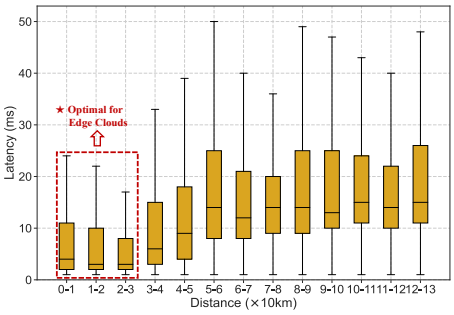


Fig. 4: Boxplot of Latency for Every 10km within 130km.

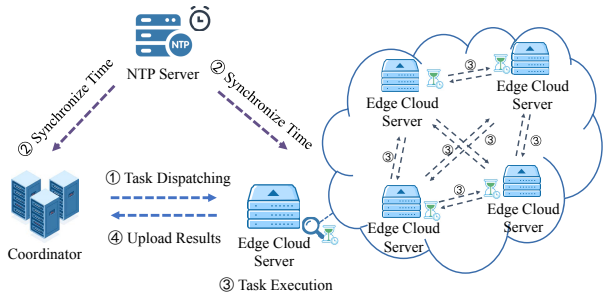


Fig. 5: Latency Measurement Framework.

III. Spatial Analysis and Modeling

In this section, we quantify the impacts of physical distance and network topology distance on latency to answer sub-questions (1) and (2). For (3), we illustrate the impact of geographic imbalance and propose to mitigate the imbalance by crowd-sourcing.

A. The Impact of Physical Distance

To explore the optimal deployment distance for edge clouds, we first need to model the relationship between latency and distance. We analyze the average latency ($\mathcal{L}_{avg}$), the minimum latency($\mathcal{L}_{min}$), the latency between two edge cloud servers propagating at the speed of light ($\mathcal{L}_{light} = 0.00667x_{dist}$), and the theoretical reference latency [11]–[13] ($\mathcal{L}_{theo.} = 0.01x_{dist}$), as shown in figure 2 and figure 3. From figure 2 and figure 3 we can conclude the followings:

End-to-end latency increases 0.0190ms per kilometer on average, and network congestion has negligible influence on this coefficient. $\mathcal{L}$ represents the end-to-end latency, and x_{dist} represents the distance between the two edge cloud servers. The R^2 coefficient of $\hat{\mathcal{L}}_{min}$ and $\hat{\mathcal{L}}_{avg}$ (0.968/0.970) shows that the formula can well fit the relationship between latency and distance. The coefficients between the fitting equation of average and minimum latency (0.01901/0.0184) are similar. The network fluctuations only increase the per unit kilometer latency by about 3.26%. So, network condition has little influence on end-to-end latency per kilometer.

$$\begin{aligned}\hat{\mathcal{L}}_{min} &= 0.0184x_{dist} + 10.1, R^2 = 0.968,\\ \hat{\mathcal{L}}_{avg} &= 0.0190x_{dist} + 23.3, R^2 = 0.970,\end{aligned} \tag{1}$$

This result also indicates the convergence of latency recently. Specifically, the end-to-end latency between two fixed servers is hard to reduce further. [14] measures ping-based latency for the worldwide Bitcoin network in 2019. Their fitted equation is $\mathcal{L} = 0.01912x_{dist} + 70.5457$, where the latency per kilometer (0.01912/0.0190) differs from our measurement by 0.5%. [15] report a speed of 0.014ms/km in 2017. The results do not differ significantly from ours, indicating latency convergence.

Moving services close to users is the most effective way to reduce latency, but excessive proximity to users will lead to extra latency. Figure 4 demonstrates the latency distribution within the serving range of edge cloud (130km). The latency is lower and more stable within 30km. Unexpectedly, the latency within 10km is conversely larger than the latency within 20-30km. [16]–[18] mention that the typical fiber length is 5 km to 20 km, plus the access distance of users, so 30km is a reasonable conclusion. Further reducing the distance below 30km brings negligible improvements as limited by the core network architecture [8].

To conclude, distance is still the most significant factor in latency. Deploying edge clouds close to users is an effective way to reduce latency. So, in edge cloud deployments, it is unnecessary to pursue proximity as long as users can get the edge cloud server within 30km.

B. The Impact of Hops

Since we have already proposed a robust model between latency and physical distance, we next identify the impact of network topology distance. To do so, we counted the number of records under each *hops* to analyze the distribution.

Compared with previous studies, *hops* have decreased recently. But the cost of communicating across ISPs is still very high. Figure 6 clarify the distribution of the number of hops (The *hops* is short for the number of traceroute hops in the following statement). The average *hops* in our measurement is 9.497. Compared to the study in 1998 [19], the *hops* are reduced by about 30% on average over the

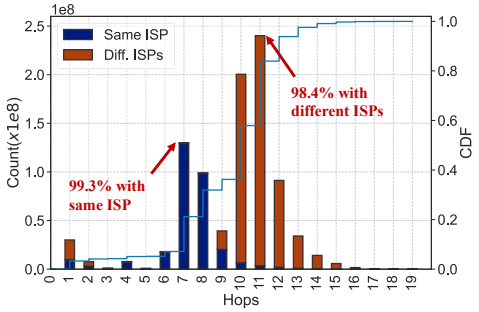


Fig. 6: Overall Hops Distribution. Communication cost across ISPs is still very high.

years. Compared with [20] in 2011, the *hops* is reduced by about 31.4% on average. However, as illustrated in figure 7, communicating across ISPs will increase about 4 *hops* overhead.

Latency optimization based on *hops* may not get significant effects for latency-sensitive applications. We measure the correlation between *hops* and latency for the data within 4-14 *hops* (containing 94.8% of the data). The results of the analysis are shown in figure 7 (overall latency varying with *hops*). We fit the equations for the relationship of average latency and *hops* as equation $\hat{\mathcal{L}} = 3.98x_{hop} + 0.500, R^2 = 0.89$. Although the average latency has a weak linear relationship with hops, the latency in every *hop* deviates greatly from the average due to the network size difference. This leads to the limited effectiveness of hops-based optimization for edge clouds as shown in 7. So, the target of latency reduction and *hops* reduction is not consistent nowadays, but it doesn't mean that *hops* is not important. We then explore the *hops* and ISP.

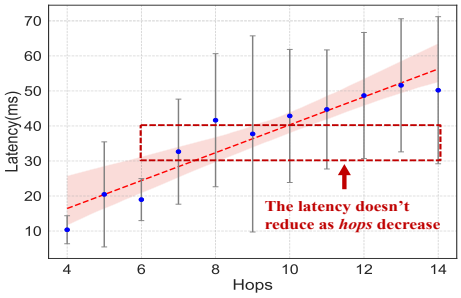


Fig. 7: Overall *Hops* v.s. Latency.

Cross-ISP communications can double edge clouds' latency, and hence it is essential for users to access the edge clouds within the same ISP. From the figure 8, we can see that within 130km, about 90% of the *hops* between two nodes of the same ISP are within 5. However, 90% of the *hops* between two nodes of different ISPs are between 8-10. [21] proposes that "the delay incurred per hop may be negligible when considering small network", but this will never happen in edge clouds. Compared to the same ISP, accessing from different ISPs can double the number of *hops*, resulting in a relative increase in latency of 120.8% (19.46/42.97, at 95% quantile).

However, IP address prefix matching is an easy way to distinguish ISPs. Here we evaluate the performance of IP prefix matching. In Figure 8, 8, 16, 24 represent the number of prefix-matching bits in the source and destination IP addresses. It can be seen that within 130km we can simply tell if they have the same ISP by matching the first 8 bits of the IP addresses. While more IP prefix matching will reduce latency, it is also a requirement for edge cloud service providers to purchase more IP addresses and deploy more edge clouds.

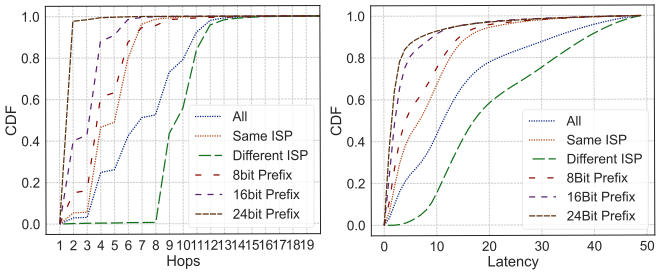


Fig. 8: CDF of Hops and Latency within 130km.

To conclude, the network topology is getting increasingly optimized, but the edge clouds still suffer from high communication costs across ISPs. Keeping users requesting edge clouds with 8-bit IP prefix matching can effectively avoid communication across ISPs.

C. The Geographical Imbalance of Latency

We have studied how latency varies with physical distance and network topology distance, and in this section, we explore how these factors affect edge cloud deployment. Due to ISPs taking different paths for end-to-end traffic between two nodes, such as equal-cost multipath routing (ECMP), the latency from A to B is not always equal to the latency from B to A. So we separately analyze incoming latency and outgoing latency (equation 2), in which $\mathcal{L}$ represents end-to-end latency. We have drawn a contour map of latency based on their relative geographical locations, as shown in figure 9.

$$\begin{aligned} \mathcal{L}^{P}_{incoming} &= \frac{1}{n}\sum \mathcal{L}_{dst\_Province=P}, \\ \mathcal{L}^{P}_{outcoming} &= \frac{1}{n}\sum \mathcal{L}_{src\_Province=P}, \end{aligned} \tag{2}$$

It can be seen from figure 9 that the latency is distributed in a gradient from the center to the surrounding area. So distance is still the dominant factor in this distribution since edge cloud servers in the center have a shorter average distance to edge cloud servers all over the nation. The edge cloud servers in core cities have relatively small latency but are also limited by geographical location. Core cities farther away from the center also have higher latency, so core cities are alternative choices to deploy edge clouds for latency reduction.

The limitation of distance causes the geographical imbalance, but there are still some special cases with great network latency. The incoming and outgoing latency

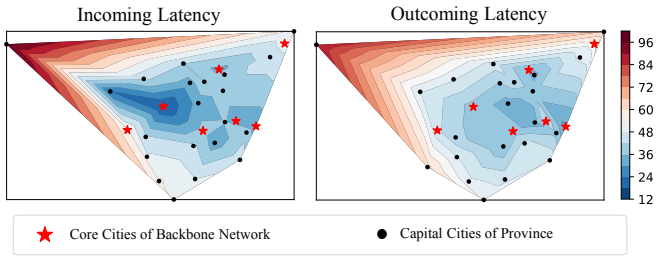


Fig. 9: Latency Geographical Distribution. (The red stars are core cities [22] of the backbone network.)

is asymmetrical by comparing the two figures in figure 9. Actually, incoming latency is 7.18% smaller than outgoing latency on average. This asymmetry results in some edge cloud servers having lower network latency when they are turned into edge clouds. However, to effectively mitigate the imbalance, we should set up abundant edge cloud servers. But deploying density edge clouds needs abundant money and effort; renting idle resources in a crowd-sourcing manner is a good choice to increase the density of edge clouds, but many concerns will arise over the latency of crowd-sourcing edge clouds. We next explore the impact of crowd-sourcing.

The edge clouds platform we studied has two types of networks. One is the household line and the other is the special line. We regard the household line as a crowd-sourcing line and compare it with a special line within 130km and the same ISP. Figure 10 illustrates the difference.

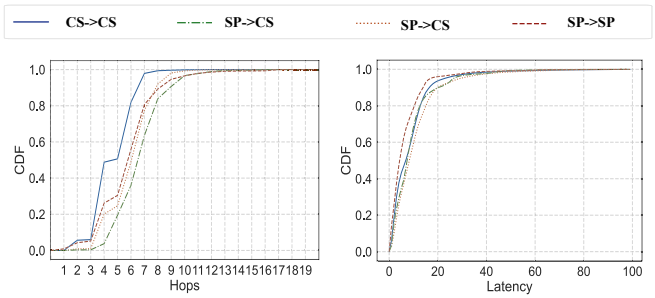


Fig. 10: The latency between lines with different network types. CS means crowd-sourcing edge clouds, and SP means edge clouds with a special line.

From figure 10 we can see that the latency between two edge clouds with both special line networks is the lowest. The latency between crowd-sourcing and the special line follows, and the special line has a significant latency accessing the crowd-sourcing line. The latency between the crowd-sourcing lines only has a 2ms increase compared with the latency between the special lines. Moreover, we can also learn that there is the smallest *hops* between crowd-sourcing lines, effectively reducing latency uncertainty.

To conclude, we can see that latency is highly imbalanced geographically, and the most important reason for this imbalance is the distance limitation. The asymmetry of each province's incoming and outcoming latency also illustrates the asymmetry of resource demand, and satisfying this asymmetry is one of our objectives in building edge clouds. In constructing edge clouds, renting edge cloud servers in a crowd-sourcing manner is a good way to increase the density easily and with little performance loss. Limited by the imbalance, centralized clouds cannot further decrease the latency for the surrounding areas. Bringing clouds close to users like edge clouds is the optimal choice for future network architecture.

IV. Temporal Analysis and Modeling

This section presents the relationship between latency and time. We analyze the latency when network congestion appears and the fluctuations of latency under multiple factors. Specifically, we start focusing on the periodic trend of latency, which tells us when the network congestion happens and the impact of congestion on latency. Then we explore the fluctuations at different granularity. We end up this section with a correlation analysis that clarifies the consistency of the latency fluctuation among different regions.

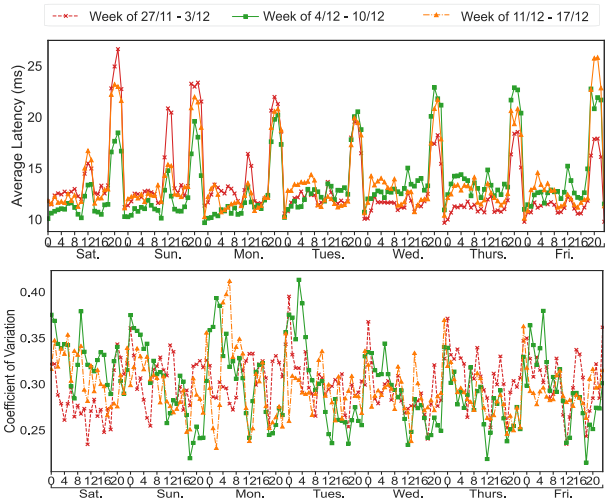


Fig. 11: Diurnal Variation of Latency and Fluctuation within 130km.

A. Diurnal Variation of Latency

We studied the diurnal variation of latency, as shown in figure 11.

Latency varies periodically and becomes 2 to 2.5 times higher during peak periods than during idle periods. We can see that the average latency varies periodically within the day. And the average latency has two peaks, at 12 p.m. and 8 p.m. The peak around 8 p.m. is larger and lasts longer than the one around 12 p.m. We also can see that the average latency level is relatively low during work time (Monday, Tuesday, Wednesday, Thursday, and Friday during the day) and higher during rest time (Friday night, Saturday, and Sunday).

To conclude, network latency has a high temporal homogeneity. The centralized clouds suffer from two latency peaks every day. Since edge clouds are designed to mitigate or

avoid the diurnal variation of latency, we further perform a measurement on the fluctuations to explore how to provide low and stable latency combing with spatial factors.

B. Fluctuations Analysis

Fluctuations of latency are another factor influencing QoS. In this part, we measure the latency fluctuations by measuring the coefficient of variation of latency.

TABLE I: Network Fluctuations within One Day.

Location		Fluctuation (ms)
Same Region (< 2000km)		12.126
Same Province (< 1000km)		13.331
Same ISP		7.618
Same City (< 130km)	Overall	7.282
	Different ISPs	12.779
	Same ISP	**5.539**

The fluctuation of the latency increases along with the increase of end-to-end latency. Fluctuation at the peak is approximately 1.5 - 1.7 times the latency during idle periods. As we can see from figure 11, the dispersion of latency also reaches the peak when the latency is at its maximum. The coefficient of variation at 8 p.m. is about 0.36 to 0.38, showing a high dispersion of the latency. So, as the latency increases, the fluctuation of latency becomes larger, and the uncertainty also increases. We next explored latency fluctuations for different geographical locations.

Table I illustrates latency fluctuations (maximum latency minus the minimum latency) within a single day. From Table I, we can see those edge cloud servers in the same city with the same ISP have the lowest latency fluctuation (5.539ms, about 41.5% of the access latency within the same province), followed by edge cloud servers within the same city (7.282ms, about 54.6% of the access latency within the same province), and between the same ISP edge cloud servers across the network (7.618ms, about 57.1% of the access within the same province. 57.1%). We also note that the latency fluctuations between different ISPs within the same city are huge, even larger than those within the same region, and second only to those within the same province. Therefore, **latency fluctuation between ISPs is an indispensable consideration in the study of proximity latency fluctuation**.

C. The Temporal Homogeneity of Latency

The above analysis shows that latency is highly periodic, and in this section, we continue to analyze the homogeneity of temporal latency variation brought by this periodicity and the impact it has on edge cloud deployment and scheduling.

The variability and fluctuations in latency are strongly correlated with human activities, and it is predictable by time series algorithms. In fact, some studies suggest that the primary traffic within the current network is generated with streaming content [23], [24]. This changing trend is consistent with our daily habits. The increase in human online behavior leads to an increase in packets, making the queue inside the router congested and finally increasing latency. However, human activities are highly homogeneous, making network congestion more severe. So network congestion is hard to avoid. But this homogeneity allows us to predict the network congestion in advance.

TABLE II: Prediction Performance of DeepAR, in which Week represents the day of the week, Day represents the day of the month and Hour represents the hour of the day.

MAPE	Week	Week + Day	Week + Day + Hour
Latency	0.19	0.16	0.14
Fluctuation	0.254	0.236	0.229

We perform a time series prediction based on a widely used state-of-the-art method named DeepAR [25]. DeepAR is an autoregressive recurrent neural network that can consider multi-variable features. We input features into the DeepAR model to test these impacts on prediction. From the table II, we can see that the latency fluctuations are more difficult to predict than the average latency. However, as the granularity of the input features becomes finer, the accuracy of the prediction is improved. Therefore, a fine-grained scheduling policy for edge clouds can better sense the current system state and make the best decisions.

The above analysis is based on end-to-end latency. However, collaboration often exists in edge clouds because they are distributed systems. So, we also analyzed the latency from one area to all other areas when network congestion happens. 6 provinces are involved in this measurement. These 6 provinces contain 15(capital), 14, 8 (The two provinces with the lowest latency to other provinces, 17, 11 (The two provinces with the highest latency to other provinces), and 5 (a province close to the capital).

We perform min-max normalization for the latency to avoid the impact of distance. After normalization, we used the DTW [26] algorithm to calculate the correlation of the latency trend from the same source province to all other destination provinces and plotted its CDF distribution. DTW indicates the correlations of two-time series sequences. The lower the DTW value between two-time series, the more similar they are.

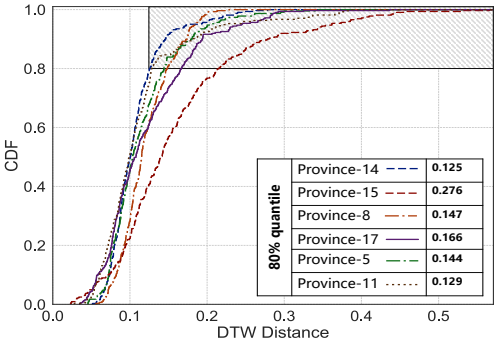


Fig. 12: CDF of DTW Distance from Some Provinces to All Other Provinces in the Nation.

From the figure 12, we can see that the trend of latency follows the power-law (or Pareto distribution) [27]: when we observe a province with higher latency to another province

at rush hour, the probability that it has a higher latency to all other provinces is over 80%, indicating a strong homogeneity in the temporal distribution of latency. This limits the collaboration between edge clouds. **For edge clouds, scheduling during congestion should be noted, edge cloud service providers need to specify fine-grained scheduling strategies. Otherwise, not only does it fail to reduce access latency, but it also increases network congestion.**

V. Feasibility of Edge Cloud

A. Suggestions For Edge Cloud

Based on the above lessons we learned, we have the following suggestions for edge clouds:

- Distance Restriction: Users can access available edge cloud resources within a short distance from them. Here we recommend around 10 to 30 km as the ideal distance for edge cloud deployment. Crowd-sourcing is a good method to increase the density of deployment.
- ISP Restriction: The user's requests should be routed to the edge clouds with the same ISP. No communication across ISPs is allowed. The *hops* between users and edge clouds are less than 10. Scheduling based on IP address prefix matching is easy and effective to avoid communicating across ISPs. Only 8-bit prefix matching can get a good performance.
- Scheduling Restriction: Fine-grained scheduling strategy. We need to achieve hour-level granularity in time, and carefully select the servers to collaborate in space.

B. Performance of Prototype Edge Cloud

In this section, we validate the above suggestions by building a real prototype edge cloud; this prototype edge cloud fully implements the above suggestion. We present the deployment scheme of the edge cloud (Plan d) and 3 centralized cloud plans (Plan a-c). Then we analyze the performance of the edge cloud from the perspective of latency reduction, fluctuation reduction, and QoS improvement.

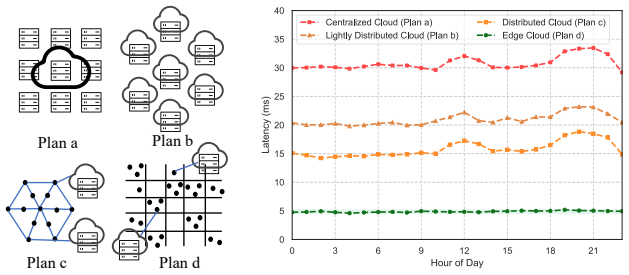
Plan a: Centralized Cloud. We choose the city with the lowest latency to other cities to place a cloud server.

Plan b: Lightly Distributed Cloud. As shown in figure 9, we can see that the core cities of the backbone network have better network resources, so in this deployment plan, we place the edge cloud servers in the seven core cities of the backbone network. 7 edge cloud servers are set up in this plan.

Plan c: Distributed Cloud. As shown in Table I, we can see that the same province, same region, and same city but different ISPs have similar latency fluctuations. Therefore, for simplicity, we place the edge servers here in the capital city of each province. 22 edge cloud servers are set up in this plan.

Plan d: Edge Cloud. As shown in Table I, we can see that it has similar, smaller latency fluctuations within the same city. Figure 4 also shows the superiority of latency within 130km or even within 30km. We set up 5174 edge cloud servers in this plan.

In the measurement stage, we establish an edge cloud with an existing network structure and filter the other edge clouds to meet the suggestion in Sec 5.1 to play the role of users. We make users always choose the server with the lowest latency. Then we measure and analyze the latency between the users and edge clouds. Figure 13 shows the average latency for every hour.



(a) 4 prototypes of edge and cloud deployment

(b) Latency performance

Fig. 13: Latency Measurement of Service Deployment on Centralized Cloud, Distributed Cloud and Edge Cloud. Edge Clouds Can Achieve Low and Stable Latency.

a) Latency Reduction. We calculated the latency averages over 24 hours based on figure 13, and the average latency of Plan a-d are 30.7/20.96/15.82/4.88, respectively, where Edge Cloud (Plan d) shows unparalleled advantages among the four deployment plans. The average latency level is 84.1% lower than that of the Centralized Cloud (Plan a). In addition to the high-density deployment of Edge Cloud (Plan d), we found that deploying service in the capital city of Distributed Cloud (Plan c) also achieved relatively low latency and latency fluctuation, with a 48.4%(15.82/30.7) reduction compared to Centralized Cloud (Plan a). Lightly Distributed Cloud (Plan b) has an average latency reduction of 31.7% (20.96/30.7) compared to the Centralized Cloud (Plan a). However, Lightly Distributed Cloud (Plan b) is still a more practical solution to implement than Distributed Cloud (Plan c), because Lightly Distributed Cloud (Plan b) only requires deployment within 8 core cities, but Distributed Cloud (Plan c) requires deployment in 28 provincial capitals, increasing the number of deployments by a factor of 2.5, but only achieving a limited 32% latency reduction. Therefore, a Lightly Distributed Cloud (Plan b) is more cost-effective than Distributed Cloud (Plan c) in terms of acceptable latency.

b) Fluctuation Reduction. We can see that the diurnal variation of Edge Cloud (Plan d) is very small, so it is not affected by network fluctuation and can provide network service continuously and steadily. The rest of Plans a,b, and c have the same level of latency fluctuation. Thus, a fine-grained edge cloud can significantly reduce the latency fluctuations (up to 87% of the diurnal variation reduction in latency) and continuously provide a stable service with low latency and very small latency fluctuations.

c) QoS Improvement. We counted the CDF of the above four plans as a measure of the QoS. The CDF statistics are shown in figure 14. In Centralized Cloud (Plan a), about 80% of the request latency falls in 11-33ms, and 95% of the requests can be kept within 45ms. Lightly Distributed Cloud (Plan b)

has an overall left shift compared to Centralized Cloud (Plan a), with 95% of requests around 36ms. Edge Cloud (Plan d) has a particularly small overall distribution, with 95% of latency within 12ms, so using Edge Cloud (Plan d) can obtain up to 73.3% QoS improvement.

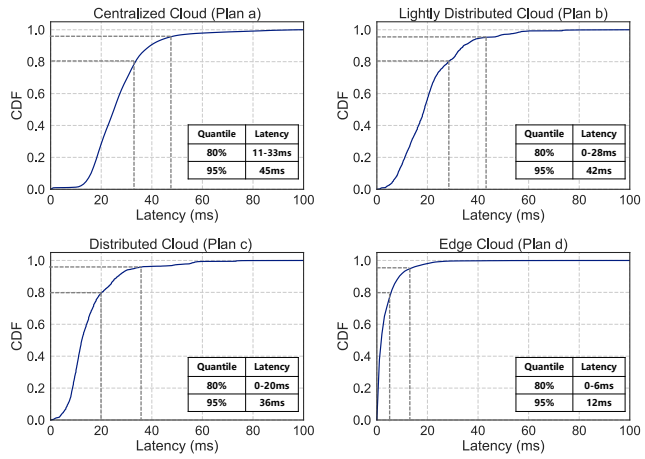


Fig. 14: Service Deployment CDF of Latency on Centralized Cloud, Distributed Cloud, and Edge Cloud.

Although our measurements are PING-based latency, we still want to find how edge clouds impact real-world applications. [28], [29] use three application-level latency thresholds (MTP, HPL, HRT) as latency indicators. MTP is the most strict requirement with interaction latency requirement $\approx$ 20ms. However, combined with technologies such as 5G (5G promising latencies down to 1 ms [28]), the optimal edge cloud architecture can reserve 70% time on average for other procedures (such as computing and rendering) when facing the most strict MTP latency threshold requirement. Such low and stable latency can support the deployment of most latency-sensitive applications.

VI. Related Work

Latency Measurement Methods and Frameworks. [30]–[32] model latency performance based on network parameters. [33] explains and defines latency in various senses, suggests some possible latency influencing factors. Some studies establish latency measurement frameworks to measure latency in different network scenarios. [34] proposes a cloud-based applications speed platform to measure the performance of various networks from virtual machines in cloud regions. [10], [35] propose large-scale systems for the data center network latency measurement framework. Beyond measurements of central clouds and data center networks, few efforts have proposed measurement architectures for large-scale distributed clouds. In our work, we consider latency measurements highly distributed and propose a framework for distributed sampling, including sampling latency, number of hops, and strictly recording the acquisition time and information such as GPS and device location.

Measurement and Analysis of Latency. A lot of studies have identified general patterns of latency in the network and the factors that influence it, which leads us to wonder how latency behaves in edge cloud scenarios. [27] proposes a power-law distribution in network measurements, which provides an essential theoretical basis for our measurement. [36] reveals that some flow requests to controllers in SDN still experience long-tail response latency. [37] proposes a way to achieve last-mile localization using hops count, which shows a correlation between hops count, geographic location, and latency, but no quantitative analysis for the relationship.

Some studies try to prove or question the effect of edge clouds through measurements. [9] measures for edge computing, but its measurements for latency are made at a coarse-grained level and do not provide an analysis of specific factors influencing latency. [28] evaluates the current cloud infrastructure's suitability to meet emerging applications' latency requirements. The authors discuss the necessity of building edge cloud platforms in today's fast-growing cloud computing and put up a question, *"It is not clear what benefits an extensive investment in edge deployment would bring"*. In this paper, we demonstrate that dense deployed edge clouds can effectively address the shortcomings of central clouds in serving next-generation mission-critical applications with a real edge computing platform.

Edge Cloud Platforms Optimization. Measurements of specific network platforms help to suggest optimizations to existing network architecture. [38] proposes incorporating performance information into Facebook's routing decisions to investigate if it can improve its network performance. The study focuses more on optimizing the edge cloud from a routing perspective. It concludes that the current routing strategy is already highly optimized. Hence, optimizing the edge cloud deployment from a routing perspective is not wise.

[39] provides a large-scale latency study using the RIPE Atlas platform and devises several edge deployment strategies to improve cloud access latency. Still, the study doesn't consider many crucial factors like ISPs and temporal factors. Compared with the [39], our study firstly explores the spatial-temporal factors influencing the end-to-end latency, and our four deployment plans are based on the measurement, which is fairer.

VII. Conclusion

In this paper, we have collected a large-scale real-world latency dataset from a commercial crowd-sourcing edge cloud platform. We have analyzed and modeled the factors that influence the spatial-temporal latency distribution and pointed out some potential factors that damage the latency of edge clouds. Based on the lessons we learned from the measurements, we propose a prototype edge cloud and evaluate its latency, volatility, and QoS improvements. Our measurements demonstrate the reliability of the edge clouds and provide foresight for the development and deployment of future latency-sensitive applications.

References

[1] G. Intelligence, "Understanding 5g: Perspectives on future technological advancements in mobile," vol. December, pp. 9–9, 2014.

[2] I. Yaqoob, L. U. Khan, S. M. A. Kazmi, M. Imran, N. Guizani, and C. S. Hong, "Autonomous driving cars in smart cities: Recent advances, requirements, and challenges," *IEEE Network*, vol. 34, no. 1, pp. 174–181, 2020.

[3] M. S. Elbamby, C. Perfecto, M. Bennis, and K. Doppler, "Toward low-latency and ultra-reliable virtual reality," *IEEE Network*, vol. 32, no. 2, pp. 78–84, 2018.

[4] A. edge zone, https://docs.microsoft.com/en-us/azure/networking/edgezones-overview, 03 2022.

[5] A. local zones, https://aws.amazon.com/about-aws/global-infrastructure/localzones/, 03 2022.

[6] E. the boundaries of the cloud with edge computing, https://www.alibabacloud.com/blog/extending-the-boundaries-of-thecloud-with-edge-computing_594214, 03 2022.

[7] N. Mohan, L. Corneo, A. Zavodovski, S. Bayhan, W. Wong, and J. Kangasharju, "Pruning edge research with latency shears," in *Proceedings of the 19th ACM Workshop on Hot Topics in Networks*, ser. HotNets '20. New York, NY, USA: Association for Computing Machinery, 2020, p. 182–189. [Online]. Available: https://doi.org/10.1145/3422604.3425943

[8] X. Jiang, H. Shokri-Ghadikolaei, G. Fodor, E. Modiano, Z. Pang, M. Zorzi, and C. Fischione, "Low-latency networking: Where latency lurks and how to tame it," *Proceedings of the IEEE*, vol. 107, no. 2, pp. 280–306, 2019.

[9] M. Xu, Z. Fu, X. Ma, L. Zhang, Y. Li, F. Qian, S. Wang, K. Li, J. Yang, and X. Liu, "From cloud to edge: a first look at public edge platforms," in *ACM Internet Measurement Conference, Virtual Event, USA*, D. Levin, A. Mislove, J. Amann, and M. Luckie, Eds. ACM, 2021, pp. 37–53.

[10] C. Guo, L. Yuan, D. Xiang, Y. Dang, R. Huang, D. Maltz, Z. Liu, V. Wang, B. Pang, H. Chen, Z.-W. Lin, and V. Kurien, "Pingmesh: A large-scale system for data center network latency measurement and analysis." Association for Computing Machinery, 2015, p. 139–152.

[11] P. K. Wahi, "Optics for microwave applications," in *IEEE MTT-S International Microwave Symposium Digest*. IEEE, 1985, pp. 295–298.

[12] O. Paz, "Infiniband essentials every hpc expert must know," *Retrieved August*, vol. 8, p. 2015, 2014.

[13] B. Lu and Y. Zhang, "A mapping algorithm for low-latency network slices," in *IEEE International Conference on Information and Automation (ICIA)*. IEEE, 2017, pp. 1156–1161.

[14] S. Park, S. Im, Y. Seol, and J. Paek, "Nodes in the bitcoin network: Comparative measurement study and survey," *IEEE Access*, vol. 7, pp. 57 009–57 022, 2019.

[15] G. Hains, Y. Khmelevsky, R. Bartlett, and A. Needham, "Game private networks performance: From geolocation to latency to user experience," in *Annual IEEE International Systems Conference (SysCon)*, 2017, pp. 1–6.

[16] A. X. Zheng, L. Zhang, and V. W. Chan, "Metropolitan area network architecture for optical flow switching," *Journal of Optical Communications and Networking*, vol. 9, no. 6, pp. 511–523, 2017.

[17] A. Acampora, "A high capacity metropolitan area network using lightwave transmission and time multiplexed switching," *IEEE Transactions on Communications*, vol. 38, no. 10, pp. 1761–1770, 1990.

[18] F. Janniello, R. Ramaswami, and D. Steinberg, "A prototype circuit-switched multi-wavelength optical metropolitan-area network," *Journal of Lightwave Technology*, vol. 11, no. 5/6, pp. 777–782, 1993.

[19] A. Fei, G. Pei, R. Liu, and L. Zhang, "Measurements on delay and hop-count of the internet," 09 1998.

[20] X. Chen, L. Xing, and Q. Ma, "A distributed measurement method and analysis on internet hop counts," in *Proceedings of International Conference on Computer Science and Network Technology*, vol. 3, 2011, pp. 1732–1735.

[21] M. Jia, J. Cao, and W. Liang, "Optimal cloudlet placement and user to cloudlet allocation in wireless metropolitan area networks," *IEEE Transactions on Cloud Computing*, vol. 5, no. 4, pp. 725–737, 2017.

[22] B. Baike, "Network node," https://baike.baidu.com/item/%E7%BD%91%E7%BB%9C%E8%8A%82%E7%82%B9/9338583, 03 2022.

[23] T. S. F. of Video Entertainment, https://www.interdigital.com/download/5fa0694a8934bfdf5f00596a, 03 2022.

[24] G. . F. Highlights, https://www.cisco.com/c/dam/m/en_us/solutions/service-provider/vni-forecast-highlights/pdf/Global_2021_Forecast_Highlights.pdf, 03 2022.

[25] D. Salinas, V. Flunkert, J. Gasthaus, and T. Januschowski, "Deepar: Probabilistic forecasting with autoregressive recurrent networks," *International Journal of Forecasting*, vol. 36, no. 3, pp. 1181–1191, 2020.

[26] wikipedia, "Dynamic time warping - wikipedia," https://en.wikipedia.org/wiki/Dynamic_time_warping, 03 2022.

[27] A. Mahanti, N. Carlsson, A. Mahanti, M. Arlitt, and C. Williamson, "A tale of the tails: Power-laws in internet measurements," *IEEE Network*, vol. 27, no. 1, pp. 59–64, 2013.

[28] T. K. Dang, N. Mohan, L. Corneo, A. Zavodovski, J. Ott, and J. Kangasharju, "Cloudy with a chance of short rtts: Analyzing cloud connectivity in the internet," in *Proceedings of the ACM Internet Measurement Conference*. New York, NY, USA: Association for Computing Machinery, 2021, p. 62–79.

[29] N. Mohan, L. Corneo, A. Zavodovski, S. Bayhan, W. Wong, and J. Kangasharju, "Pruning edge research with latency shears," in *Proceedings of the 19th ACM Workshop on Hot Topics in Networks*. New York, NY, USA: Association for Computing Machinery, 2020, p. 182–189.

[30] N. Cardwell, S. Savage, and T. Anderson, "Modeling tcp latency," in *Proceedings IEEE INFOCOM 2000. Conference on Computer Communications.*, vol. 3, 2000, pp. 1742–1751 vol.3.

[31] S.-W. Ko, K. Han, and K. Huang, "Wireless networks for mobile edge computing: Spatial modeling and latency analysis," *IEEE Transactions on Wireless Communications*, vol. 17, no. 8, pp. 5225–5240, 2018.

[32] X. Xu, G. Sun, L. Luo, H. Cao, H. Yu, and A. V. Vasilakos, "Latency performance modeling and analysis for hyperledger fabric blockchain network," *Information Processing & Management*, vol. 58, no. 1, p. 102436, 2021.

[33] P. Svoboda, M. Laner, J. Fabini, M. Rupp, and F. Ricciato, "Packet delay measurements in reactive ip networks," *IEEE Instrumentation Measurement Magazine*, vol. 15, no. 6, pp. 36–44, 2012.

[34] R. K. P. Mok, H. Zou, R. Yang, T. Koch, E. Katz-Bassett, and K. C. Claffy, "Measuring the network performance of google cloud platform," in *Proceedings of the ACM Internet Measurement Conference*. New York, NY, USA: Association for Computing Machinery, 2021, p. 54–61.

[35] Y. Li, Z.-P. Cai, and H. Xu, "Llmp: Exploiting lldp for latency measurement in software-defined data center networks," *Journal of Computer Science and Technology*, vol. 33, pp. 277–285, 03 2018.

[36] J. Xie, D. Guo, X. Li, Y. Shen, and X. Jiang, "Cutting long-tail latency of routing response in software defined networks," *IEEE Journal on Selected Areas in Communications*, vol. 36, no. 3, pp. 384–396, 2018.

[37] C. Xiang, X. Wang, Q. Chen, M. Xue, Z. Gao, H. Zhu, C. Chen, and Q. Fan, "No-jump-into-latency in china's internet! toward last-mile hop count based ip geo-localization," in *Proceedings of the International Symposium on Quality of Service*. New York, NY, USA: Association for Computing Machinery, 2019.

[38] B. Schlinker, I. Cunha, Y.-C. Chiu, S. Sundaresan, and E. Katz-Bassett, "Internet performance from facebook's edge," in *Proceedings of the Internet Measurement Conference*. New York, NY, USA: Association for Computing Machinery, 2019, p. 179–194.

[39] L. Corneo, N. Mohan, A. Zavodovski, W. Wong, C. Rohner, P. Gunningberg, and J. Kangasharju, "(how much) can edge computing change network latency?" in *IFIP Networking Conference (IFIP Networking)*, 2021, pp. 1–9.

Ethics

When performing this study, we take good care of users' privacy and research ethics. The Research Ethical Committee approved the whole data collecting process of the institutes that the authors are currently affiliated with; the edge computing service provider also approved the collection in this study through the service agreement; we collected no sensitive data from individuals except the servers' location, and the location is not directly exposed to us. We only send queries to the database, and it returns the aggregated results without sensitive data. We collected no customer-identifiable information during the study.