A Light-Weight Statistical Latency Measurement Platform at Scale

Xu Zhang, *Member, IEEE*, Geyong Min, *Member, IEEE*, Qilin Fan, *Member, IEEE*, Hao Yin, *Member, IEEE*, Dapeng Oliver Wu, *Fellow, IEEE*, and Zhan Ma, *Senior Member, IEEE*

***Abstract*—The statistical value of latencies between two sets of hosts over a given period, which is referred as to the statistical latency, can benefit many applications in the next-generation networks, for example, Network-in-a-Box-based resource provisioning. However, the existing methods can hardly achieve low measurement cost and high prediction accuracy simultaneously in large-scale scenarios. In this article, we design a light-weight statistical latency measurement platform named DMS (DNS-based statistical latency Measurement platform at Scale). DMS achieves high measurement accuracy by introducing a metric space to select the closest open recursive DNS (Domain Name System) server to a given host, and predicting the end-to-end latency between two hosts via the measured latency between the two corresponding DNS servers. To reduce the overall measurement overhead, DMS clusters the hosts in the metric space with the open recursive DNS infrastructure in the network as the cluster center, thus achieving low measurement cost and good scalability in large scale simultaneously. To evaluate the performance of DMS, we implement a prototype system in the network. Compared to the widely adopted method King, DMS can reduce the relative error by 18.5% for real-time end-to-end latency prediction and 33% for statistical latency prediction.**

Manuscript received March 5, 2021; revised May 18, 2021 and July 6, 2021; accepted July 6, 2021. Date of publication July 26, 2021; date of current version October 27, 2021. This work was supported in part by the National Key Research and Development Program under Grant 2018YFB2100804, in part by the National Natural Science Foundation of China under Grant 61902178 and Grant 62022038, in part by the European Union^^e2^^80^^99s Horizon 2020 research and innovation programme under the Marie Sklodowska-Curie Grant 898588, in part by the Natural Science Foundation of Jiangsu under Grant BK20190295, in part by the Leading Technology of Jiangsu Basic Research Plan under Grant BK20192003, and in part by the National Natural Science Foundation of China under Grant 92067208 and Grant 61972222. Paper no. TII-21-1087. *(Corresponding author: Zhan Ma.)*

Xu Zhang and Geyong Min are with the Department of Computer Science, College of Engineering, Mathematics, and Physical Sciences, University of Exeter, EX4 4QF Exeter, U.K. (e-mail: x.zhang6@exeter.ac.uk; G.Min@exeter.ac.uk).

Qilin Fan is with the School of Big Data and Software Engineering, Chongqing University, Chongqing 401331, China (e-mail: fanqilin@cqu.edu.cn).

Hao Yin is with the Research Institute of Information Technology (RIIT), Tsinghua University, Beijing 100084, China (e-mail: h-yin@mail.tsinghua.edu.cn).

Dapeng Oliver Wu is with the Department of Electrical and Computer Engineering, University of Florida, Gainesville, FL 32611 USA (e-mail: dpwu@ufl.edu).

Zhan Ma is with the School of Electronic Science and Engineering, Nanjing University, Nanjing 210093, China (e-mail: mazhan@nju.edu.cn).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TII.2021.3098796.

Digital Object Identifier 10.1109/TII.2021.3098796

***Index Terms*—IP networks, Domain Name System, Quality of Service.**

I. Introduction

AS THE development of 5G technology is moving to its final phase, the next generation networks, that is, beyond 5G networks (B5G) or 6G, have attracted tremendous interest from industry and academia [1]. Concurrent with the development of cellular network technology has been a steady rise in the latency-sensitive applications, such as cloud gaming [2] and mobile augmented reality and virtual reality (AR/VR) [3], [4]. To deliver these latency-sensitive applications in the network with high users' quality of experience (QoE), service providers usually resort to resources strategically deployed in the network [5], with the expectation that these predeployed resources can enhance the QoE perceived by the targeted set of users over time [6]. An example of the resources is Network-in-a-Box (NIB), which can offers speed in deployment and flexibility to adjust to any circumstances [7]. Therefore, the statistical value of the latencies between the set of predeployed network resources and the targeted users over a given period, which is also referred as to statistical latency, is an important criterion for resource provisioning in the network [8], [9].

Measuring the statistical latencies can also benefit many other scenarios [11], for example, the statistical latency between the set of servers in a content delivery network (CDN) and the set of objective users is an important factor to be considered when Internet content providers (ICPs) select which CDN to use for their content dissemination [12]; for Internet service providers (ISPs), the statistical latency can be used to improve its network performance, such as locating link failures and optimizing routing strategies [10]. Fig. 1 gives an example of the real-world statistical latencies among different countries/regions, which was measured by Verizon in 2020 [10]. However, the statistical latency measurement in large-scale networks faces grand challenges in achieving good scalability, high prediction accuracy, and low measurement cost simultaneously.

Existing methods designed for statistical latency measurements rely on the predeployed dedicated hardwares or softwares, incurring high measurement cost. For example, SSL (Surrogate-based method for large-scale Statistical Latency measurement) [9] takes advantage of the widely distributed end devices in the edge to measure the Internet, but it requires the installation of some softwares and the permission of background measurement

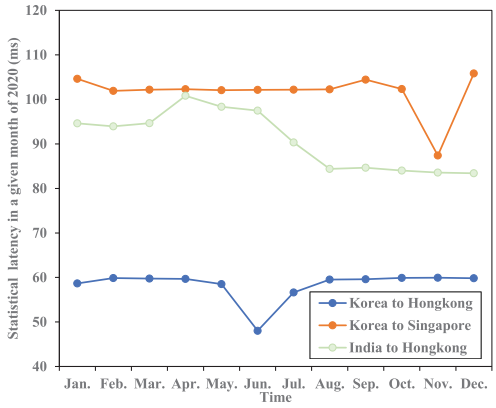

Fig. 1. Statistical latency among countries/regions in a given month of 2020 (Data from Verizon [10]).

from end users. On the other hand, there has been a rich literature on the real-time latency predictions in the Internet. For example, iPlane [13] is proposed to predict the end-to-end path by decomposing it with path segments that can be measured. However, this method is heavy-weighted and cost-inefficient, as it relies on massive amount of active probing from numerous predeployed vantage points. Another representative example is the network-coordinate-based measurement such as Global Network Positioning (GNP) [14], which models the Internet as an Euclidean space and allocates each host a coordinate in the space. This method is light weighted, but the measurement accuracy is low due to the existence of asymmetric routing and triangle inequality violation (TIV). Besides, there also exist proposals to use DNS-based measurement to estimate the latency such as King [15], [16]. However, it suffers from low accuracy nowadays with more and more self-configured domain name servers.

To overcome the challenges, this article introduces a lightweight and scalable framework called DMS, for accurate statistical latency measurement on the Internet. DMS clusters the hosts in the network based on their relative distances in a metric space, where the center of each cluster is an open recursive DNS server in the network. Moreover, DMS measures the latencies between the DNS servers and maintains a latency database. Based on the latency database, DMS can predict the real-time latency between two arbitrary hosts and the statistical latency between two sets of hosts with high accuracy and low overhead simultaneously. In detail, DMS maps the hosts in the network to a metric space and weights the proximity between two hosts in the network based on the distances between them in the metric space. Considering the inaccuracy of the metric space, DMS takes advantage of the high rank accuracy of the metric space and clusters the hosts in the network based on their relative distances in the metric space. To reduce the overall measurement cost, DMS takes the open recursive DNS servers in the network as the cluster centers, avoiding the extra deployment of dedicated infrastructure. Moreover, DMS measures the latencies between two arbitrary DNS servers by DNS tricking. To further reduce the measurement overhead, DMS maintains a database for the latencies between two arbitrary DNS servers. Based on the database, DMS can predict the real-time latency between two arbitrary hosts and the statistical latency between two sets of hosts with high accuracy and low overhead simultaneously.

In this article, we make the following contributions.

1) First, we propose the first lightweight and scalable statistical latency measurement platform called DMS to support both real-time end-to-end latency measurement between two hosts and statistical latency measurements between two sets of hosts over a given period, without resorting to the deployment of numerous extra dedicated measurement servers and the proactive participation from hosts.
2) Second, we map the hosts in the network to a metric space, cluster them in the space with open recursive DNS servers as the cluster centers, and predict the real-time latency between two hosts or the statistical latency measurements between two sets of hosts via the measured latencies among the cluster centers, thus achieving high prediction accuracy, low measurement cost, and good scalability in large scale simultaneously.
3) Third, we analyze the distribution of the statistical latencies and its relationship with the predicted value generated by DMS under a specific scenario for statistical latency measurement. Theoretical analysis results show that the predicted latency is statistically approximate to the actual value.
4) Last but not least, we implement a prototype system and deploy it in the network. The real-world large-scale measurement results show that compared to the existing method King, one of the most widely adopted method in the wild, DMS, can reduce the relative error by 18.5% for real-time end-to-end latency prediction and 33% for statistical latency prediction.

The rest of this article is organized as follows. Section II briefly summarizes the related work. Section III presents the system design of DMS and Section IV analyzes DMS when it is used to predict statistical latencies. Section V shows the implementation of DMS and its evaluation results on real Internet. Finally, Section VI concludes this article.

II. Related Work

Statistical latency measurement can be used in many scenarios, such as CDN selection for ICP. Zhang *et al.* [9] proposed SSL to measure the statistical latency by leveraging the widely distributed end devices in the edge. When predicting the latency between two hosts, it selects multiple end devices closest to one host (A) as its surrogates, utilizes them to send probing packets to the other host (B), and uses the average value of the latencies between the surrogates to B to estimate the latency between A and B. Simulation results show that when about 10% of the users in the network can take participation in the measurement, the accuracy can be higher than 90%. In other words, SSL is designed for large companies that have enormous distributed users with specific softwares installed in their devices. Moreover, permissions from all the participated end users should be warranted. Distinguished from SSL, DMS is more easily to be deployed in the real-world situation.

There are also other related works on real-time network distance estimation [13]–[25]. We classify these works into two categories: the path-fitting methods and the coordinate-based methods. Francis *et al.* [17] propose the Internet Distance Map Service, which predicts the real-time network distance between two hosts by summarizing the distance from each host to its nearest deployed tracer and the distance between the two tracers. To achieve a high prediction accuracy, a large number of dedicated measurement servers or tracers should be deployed in the network, which incurs high deployment costs. Another representative work is iPlane [13], which predicts the end-to-end path latency by composing the latencies of measured segments of Internet paths. To obtain enough measured segments of Internet paths, iPlane maintains a map of the Internet's core by leveraging more than 300 Planetlab servers and nearly a thousand traceroute and Looking Glass servers to send massive packets to Border Gateway Protocol (BGP) atoms. In other words, iPlane requires a large number of measurement servers and incurs high measurement overhead by sending massive probing packets into the network. Gummadi *et al.* [15] present King, which estimates the distance between two hosts via the distance between their domain name servers, assuming that the name servers are located close to their hosts, which does not always hold true in the wild. Apart from these path-fitting methods, Ng *et al.* [14] model the Internet as a geometric space and predict the latency using the distance in the geometric space. Based on the definition of distance in a geometric space, the predicted latency from one host (A) to another host (B) is always equal to that from B to A, which makes the approach suffer from low prediction accuracy due to the existence of asymmetric routing and TIV in the network. Dabek *et al.* [19] propose a fully decentralized system named Vivaldi, which models the network as a spring system, and adjusts the coordinates of hosts progressively in a distributed manner. Fu *et al.* [24] propose a relative coordinate-based distributed sparse-preserving matrix-factorization method to assign a relative coordinate and parameter matrix to each host. However, these methods have limited feasibility by needing the active participation of hosts.

III. System

In this section, we first provide a brief overview of the system DMS, and then, describe how to implement each component in detail.

A. System Overview

DMS maps the hosts and the domain name servers from the network to a metric space, clusters the hosts into groups with the DNS servers as the group centers, and predicts the network latency between two hosts using the measured latency between the two corresponding DNS servers. We describe the main steps as follows.

First, DMS will discover exploitable name servers in the Internet, that is, the open recursive name servers. An open recursive name server can translate domain names to IP addresses for any host in the network. And if it does not have the right information, it will contact other DNS servers as a proxy, and then, pass the information to the host. This property makes it possible to measure the latency between two open recursive DNS servers with DNS tricking, which will be detailed in Section III-B2. Next, DMS maps the hosts and the discovered name servers into a metric space, and allocates each of them with a coordinate. Based on the coordinate, we can calculate the distance between two arbitrary hosts. However, the distance suffers from low accuracy when it is used to predict the latency between the two hosts. To overcome the problem, we only leverage the relative values of the distances, rather than the absolute values. The relative value of the distances means that the comparison of the predicted latencies for two pairs of hosts. Specifically, suppose that there are two pairs of hosts $< A, B >$ and $< C, D >$, where A, B, C, and D are four hosts in the network. The latencies for the two pairs of hosts are $L_{A,B}$ and $L_{C,D}$. With the network coordinates, we can have two predicted latencies for the two pairs of hosts, that is, $d_{A,B}$ and $d_{C,D}$. Instead of using the absolute value of $d_{A,B}$ or $d_{C,D}$, DMS leverages the comparison between $d_{A,B}$ and $d_{C,D}$. This is because that if $d_{A,B} < d_{C,D}$, it is high likely that $L_{A,B} < L_{C,D}$, and vice versa. With the existence of asymmetric routing and TIV in the network, the predicted values $d_{A,B}$ and $d_{C,D}$ suffer from low accuracy. If we use the absolute value directly, the inaccuracy will be incurred, thus deteriorating the performance of DMS.

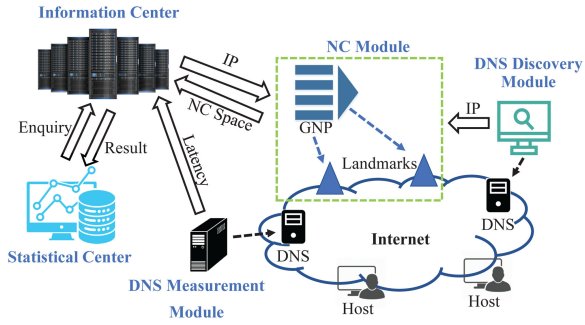

Fig. 2. Framework of DMS.

Then, DMS clusters the hosts into groups, with the name servers as the group centers. Specifically, each host will be allocated to the group whose center is closest to the host based on the calculated distance in the metric space. Finally, the latency between two arbitrary group centers will be measured via DNS tricking. And a database for the latencies between two arbitrary DNS servers will be maintained by DMS. Based on the database, DMS can predict the real-time latency between two arbitrary hosts and the statistical latency between two sets of hosts with high accuracy and low overhead simultaneously.

The system framework is shown in Fig. 2. It consists of one centralized center that computes the statistics, one center for information storage and processing, and three modules for performing the measurement. Among them, *DNS discovery module*, *DNS measurement module,* and *network coordinate module* are always running in the background and exchanging information with the *Information Center*, including the IPs and the network coordinate of the found name servers and the distances among them.

The input to DMS can be a diverse set of measurement objectives. It is provided as a statistical value of two IP lists. When *Statistical Center* accepts operators' measurement objects, it issues a measurement task to the *network coordinate module*, which allocates a coordinate to each host and uploads the coordinate to the *Information Center*. Based on the coordinates of DNS servers and hosts, the *Information Center* takes advantage of the latencies between DNS servers to predict the latencies between the input hosts and provides a lookup interface to *Statistical Center*, which computes and outputs the statistical measurement results to the operators.

Like the most existing work in the field of network latency measurement, we assume that the packets from different applications with different protocols are treated equally in the network due to the principle of net neutrality. In this scenario, the latency experienced by our probing packets are statistically the same as the latency experienced by other packets traversing the same network path at the same time.

With DMS, operators do not need to deploy numerous dedicated measurement servers in the network. What is more, no permission from the hosts is needed, which is critical for many scenarios where the devices from hosts cannot be controlled to actively send probing packets.

B. Components

1) DNS Discovery Module: The DNS infrastructure in the Internet can be divided into multiclasses [26], including authoritative domain name servers maintaining the mapping from hostname to IP address, ingress servers receiving end users' queries, egress servers communicating with authoritative domain name servers, and the hidden servers between ingress servers and egress servers for query intermediating. Note that not all the ingress servers are configured to be connected to a hidden server. In other words, an ingress server can be configured to communicate with an egress server directly. In this article, DMS tries to discover open recursive domain name servers in the network, which play the role of ingress server and egress server simultaneously without hidden servers. Moreover, they can serve users from anywhere without preconfigured rules. The more open recursive name servers are discovered, the more accurate the predicted latency will be.

In detail, *DNS discovery module* first registers a domain and deploys an authoritative domain name server for it. Then, we randomly select an IP address from the IP address space, and send a request to it with the objective to resolve a subdomain within the registered domain. The subdomain will be embedded with the selected IP address. If the selected IP address belongs to an ingress server, the request to resolve the subdomain will finally reach to our deployed authoritative domain name server by the forwarding of an egress server. Through analyzing the logs in the deployed authoritative domain name server, we can extract a set of ingress servers and a set of egress servers, respectively. If an IP address belongs to the two sets simultaneously, it will be regarded as the IP address of an open recursive domain name server. By repeating the aforementioned procedure, DMS can obtain a list of open recursive domain name servers, which will be sent to the *DNS discovery module* for coordinate allocation and to the *DNS measurement module* for measurement.

2) DNS Measurement Module: DNS measurement module takes DNS tricking to measure the latency between two arbitrary two domain name servers. In order to guarantee that the measured latency between two name servers is exactly the latency between them, there should be domain resolution requests sent directly between them. This is why the *DNS discovery module* only takes open recursive domain name servers into consideration.

Suppose that there exist two open recursive name servers with the IP address IP1 and IP2, respectively. To measure the latency between them, we should again register a domain (*e.g.,* mydomain.com) and deploy an authoritative domain name server for it. At the first round, we send to IP1 via our client with the request to resolve abc.IP2.mydomain.com. Our deployed DNS can trick IP1 that IP2 is the authoritative domain name server for resolving *.IP2.mydomain.com. Then, IP1 will cache the information and resend the request to IP2. Finally, IP2 will return an error information to IP1. At the second round, we send to IP1 with another request to resolve xyz.IP2.mydomain.com. As IP1 has the cached information, the request will be directly sent to IP2. Again, an error information will be returned. Through the second round, we can get the overall latency, which is composed of the latency from our client to IP1 and the latency from IP1 to IP2. Subtracting the latency from our client to IP1, the latency between IP1 to IP2 can be obtained. In the aforementioned example, abc and xyz are two random numbers. To reduce the negative effect of cache pollution, we can manually set the TTL of the reply to a small value, *e.g.,* 30 s.

3) Network Coordinate Module: Network coordinate module is responsible for mapping the hosts and the open recursive domain servers in the network to a metric space, and allocating them with coordinates. Based on the coordinates, we can calculate the distance between a host and a domain server, and group the hosts based on the relative values of the distance.

In detail, we will deploy several landmarks in the network, which can be deployed dedicated servers or virtual machines rented from cloud service providers. They should be distributed in the network, with the smaller overall distance from the hosts in the network to its closest landmarks as possible. And they will act as the reference nodes in the network. To construct the metric space, we should first determine the coordinates of the landmarks, and then, allocate a coordinate to each host.

Given that there exist K landmarks $R = \{R_1, R_2, \ldots, R_K\}$. For two arbitrary landmarks R_i and R_j, we can measure the latency L_{R_i,R_j} between them. Furthermore, we can calculate the distance d_{R_i,R_j} between them via their coordinates C_{R_i}, C_{R_j} via a distance function $f(\cdot)$, such as using the Euclidean distance. In other words, $d_{R_i,R_j} = f(C_{R_i}, C_{R_j})$. The process of determining the landmarks' coordinates can be formulated as solving the following optimization problem:

$$\min. \sum_{\forall R_i,R_j \in R | i>j} (L_{R_i,R_j} - d_{R_i,R_j})^2. \tag{1}$$

Assume the coordinate of an ordinary host H is C_H. We can also measure the latency $L_{R_i,H}$ between the landmarks R_i

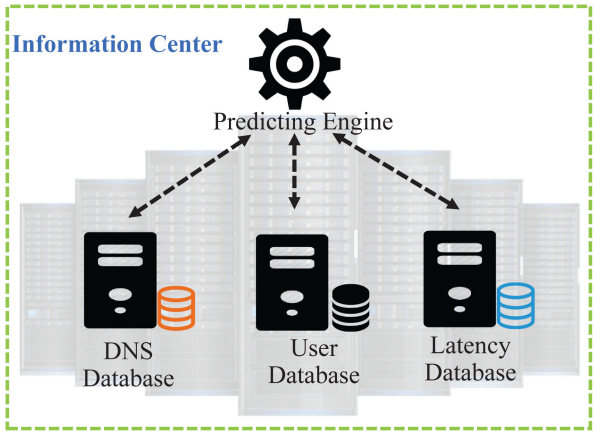

Fig. 3. Structure of the *Information Center*.

and H, and calculate the distance $d_{R_i,H}$ between them via their coordinates C_{R_i} and C_H, that is $d_{R_i,H} = f(C_{R_i}, C_H)$. Similarly, the process of determining the host's coordinate C_H can be formulated as solving the following optimization problem:

$$\text{min.} \sum_{\forall R_i \in R} (L_{R_i,H} - d_{R_i,H})^2. \tag{2}$$

To achieve the objective that two hosts close to each other in the metric space are colocated in the network, thus sharing similar network status at a high probability, the *network coordinate module* determines the coordinates using the latency when the network is idle. Without network congestion, the distance between two hosts in the metric space can reflect their topological distance rather than their real-time network distance. Specifically, the *network coordinate module* measures the latencies between a landmark and an arbitrary host at different time points across one day. With the assumption that the path between the landmark and the host will not be congested all the time, the minimum value of the measured latencies can be regarded as the latency between them when the network is idle. Compared with constructing the metric space based on real-time network latencies, the performance of the metric space based on the network latencies without network congestion will not be deteriorated by the infeasibility to measure all the latencies between the landmarks and end hosts simultaneously at a large scale.

4) Information Center: *Information Center* maintains the overall information about the discovered domain name servers and the allocated coordinates, predicts the latencies between two arbitrary hosts, and provides information support to *statistical center*, as shown in Fig. 3.

In detail, a DNS database will be included in the *Information Center*, which communicates with the *DNS discovery module* and *network coordinate module* periodically and stores the IPs and coordinates of the discovered domain name servers; the latency database stores the real-time latency and the history latency information between DNS pairs. As the name server set varies from time to time, the *Information Center* would exchange DNS information with the *DNS discovery module* at intervals. Apart from the two databases, the *Information Center* also contains a user database, which maintains the information about the input host lists from the operators. Notice that each host or name server is stored in the database by a 2-tuple of $(ip, coord)$, where ip is its IP address and $coord$ is its network coordinate.

To provide information support to the *Statistical Center*, the *Information Center* runs a *Predicting Engine* to predict the latency between arbitrary two hosts. It takes advantage of the information stored in the three databases, maps hosts to their closest domain name servers in the metric space, and predicts the latency between them. In detail, the relative values are leveraged when clustering the end users in the space with open recursive DNS servers as the cluster centers. Specifically, when determining which open recursive DNS server from the open recursive DNS server set $S = \{s_1, s_2, \ldots, s_M\}$ is closest to a given end host h, we first compute the distance $d_{s_i,h}$ between the DNS server s_i and the end host h, where $1 \le i \le M$. Then, it is high likely that the DNS server s_j is the closest one to the end host h in the network if $d_{s_j,h} \le d_{s_i,h} \forall i \in [1, M]$. This will always hold true if the metric space has high consistency between the computed distance and the real-world latency. The overall process of predicting the latency between two hosts is described in Algorithm III-B4.

Algorithm 1: predictLatency($h1$, $h2$, S, M).

```
// < h1, h2 > is the pair of hosts between whom the
   latency should be predicted.
// S is the set of domain name servers.
// M is the matrix of the measured latency between any
   two name servers in S;
dns1 ← NULL
dist1 ← +∞
for s in S do
    tmp ← Distance between h1.coord and s.coord
    if dist1 < tmp then
        dist1 ← tmp
        dns1 ← s.ip
dns2 ← NULL
dist2 ← +∞
for s in S do
    tmp ← Distance between h2.coord and s.coord
    if dist2 < tmp then
        dist2 ← tmp
        dns2 ← s.ip
return M[dns1] [dns2]
```

5) Statistical Center: *Statistical center* is the brain for statistic measurement. On one hand, it receives the measurement tasks from operators, and issues the tasks to relative modules in DMS. On the other hand, it clusters the hosts based on their coordinates and computes the statistical result according to the requirements of operators. For example, if operators want to know the performance of a CDN, the *statistical center* would determine which user should be serviced by which server according to the CDN's

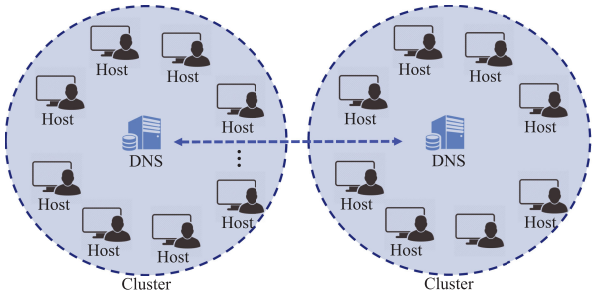

Fig. 4. Hosts are naturally clustered by domain name servers.

request routing strategy. Finally, the *statistical center* would visualize the statistical result and output it to the operators.

IV. Analysis

In this section, we first investigate the measurement overhead of DMS, and then, analyze the relationship between the predicted latency of DMS and the ground truth. The goal of this analysis is to prove that the predicted value is *statistically approximate* to the actual latency value. We mainly concentrate on the statistical latency of the network at a given time point t, as the connectivity status between two groups of hosts over a period T can be divided into a series of snapshots of the network at different time instances within T.

A. Measurement Overhead

Considering there exist N hosts in a network, and we would like to measure the network connectivity via the statistical latency among themselves. For the given time point t, there should be $O(N^2)$ times of measurements to be conducted to get the pair-wise latencies among them.

With DMS, the measurement overhead can be reduced significantly. First, to construct the metric space, K landmarks should be deployed, which would measure the latencies among themselves and the latencies from the landmarks to the hosts. That is, $O(K^2 + KN)$ times of measurements should be conducted, which will be used as input for computing the coordinates of the hosts. Second, to discover the available open recursive DNS servers in the network, $O(N)$ times of measurement should be conducted and M open recursive DNS servers are identified. What is more, $O(M^2)$ times of measurement should be conducted to obtain the latencies between the discovered open recursive DNS servers.

In summary, the measurement overhead is $O(K^2 + KN + N + M^2)$, where K and M are far less than N. That is, the measurement overhead is reduced from $O(N^2)$ to $O(KN)$.

B. Accuracy Analysis

DMS clusters the hosts in the Internet into groups with the domain name servers as the group centers. As Fig. 4 illustrates, the network latency between two hosts is predicted by the measured latency between the corresponding two name servers. Given the metric space has consistency between the computed distance in the metric space and the real-world latency in the Internet, we analyze why the predicted values of DMS can be used to approximate the actual latency in a specific scenario. Inspired by [27], we assume that the latency X_i between the two hosts satisfies a normal distribution $X_i \sim \mathbb{N}(\mu_i, \sigma_i^2)$ if we randomly choose two hosts from two clusters, where the mean value μ_i equal to the latency between the two name servers. As to the standard deviation σ_i, it reflects the accurate extent of DMS in predicting the latency between arbitrary two hosts. Then, the statistical latency between two sets of hosts can be defined as $\overline{X} = \frac{1}{n}\sum_{i=1}^{n} X_i$. According to the property of normal distribution, the mean value of a series of independent normal random variables also satisfies a normal distribution. So, we can assume that

$$\overline{X} \sim \mathbb{N}(\mu, \sigma^2) \tag{3}$$

where μ is the mean value of $\overline{X}$, while σ is the standard deviation of $\overline{X}$.

Based on the definition of $\overline{X}$, we have its mean value as follows:

$$\mu = E\{\overline{X}\} = E\left\{\frac{1}{n}\sum_{i=1}^{n} X_i\right\} = \frac{1}{n}\sum_{i=1}^{n} E\{X_i\}$$
$$= \frac{1}{n}\sum_{i=1}^{n} \mu_i. \tag{4}$$

And the variance of $\overline{X}$ is

$$\sigma^2 = D\{\overline{X}\} = D\left\{\frac{1}{n}\sum_{i=1}^{n} X_i\right\} = \frac{1}{n^2}\sum_{i=1}^{n} D\{X_i\}$$
$$= \frac{1}{n^2}\sum_{i=1}^{n} \sigma_i^2. \tag{5}$$

Moreover, the upper bound of the standard deviation σ is

$$\sigma = \frac{1}{n}\sqrt{\sum_{i=1}^{n}\sigma_i^2} \le \frac{1}{n}\sqrt{\sum_{i=1}^{n}\sigma_{\max}^2} = \frac{1}{\sqrt{n}}\sigma_{\max} \tag{6}$$

while the lower bound of the standard deviation σ is

$$\sigma = \frac{1}{n}\sqrt{\sum_{i=1}^{n}\sigma_i^2} \ge \frac{1}{n}\sqrt{\sum_{i=1}^{n}\sigma_{\min}^2} = \frac{1}{\sqrt{n}}\sigma_{\min} \tag{7}$$

where the equality holds if and only if $\sigma_i = \sigma_{\max} \forall 1 \le i \le n$ for the upper bound, and $\sigma_i = \sigma_{\min} \forall 1 \le i \le n$ for the lower bound.

In conclusion, the standard deviation of the statistical value has also been reduced, just as the following inequation shows:

$$\frac{1}{\sqrt{n}}\sigma_{\min} \le \sigma \le \frac{1}{\sqrt{n}}\sigma_{\max}. \tag{8}$$

From the aforementioned analysis, we can see that the mean of the predicted statistical value is exactly the mean of the latencies between the two sets of hosts, and the standard deviation of the predicted statistical value will decrease as the number of the hosts increases. To this end, it is safe to conclude that we can use the predicted values of DMS to approximate the actual latency.

TABLE I
SUMMARY OF DNS DISCOVERY PROCESS

Sampled IP	14.9M
Ingress DNS servers	192.8k
Egress DNS servers	3.5k
Open Recursive DNS servers	1.0k

TABLE II
CONSTRUCTION OF THE METRIC SPACE

Landmark	15
Open Recursive DNS servers	1.0k
Ordinary Hosts	10k

V. EVALUATION

This section introduces the implementation of DMS in Mainland China, validates the effectiveness of DMS of predicting the real-time latency and the statistical latency, and compares the result with existing methods. After that, we analyze the source of the inaccuracy of DMS.

A. Implementation

In order to help implement and evaluate DMS, we take advantage of 33 vantage points deployed by a company in Mainland China, across the major ISPs in the country. These vantage points have been used by the three measurement modules of DMS. For the DNS measurement module, we utilize 32 vantage servers to measure the latencies between DNS servers, where each vantage server is running with 30 threads simultaneously. In this way, the latencies in the *Information Center* are updated in every ten minutes to reflect the real-time performance. Operators can adjust the frequency according to its own need and balance the tradeoff between the measurement overhead and the real-time performance.

The *DNS discovery module* leverages one of the vantage points to act as the authoritative domain name server for our registered domain, and uses others as probes to send DNS queries within the registered domain. The *DNS measurement module* exploits the vantage points that have not acted as name servers to measure the latencies between the found domain name servers. Besides, the *network coordinate module* selects part of these vantage points as landmarks, and constructs the metric space for the discovered name servers and ordinary hosts in the Internet. Notice that the accuracy of a metric space with 7 to 9 dimensions would be enough [28], and the number of landmarks should be larger than the dimensions. Actually, the minimum number of vantage points required by DMS is limited by the number of landmarks, not necessary to be 33. And operators can determine its own dimension number to strike a balance between the system cost and the accuracy.

To validate the effectiveness of DMS, we sample the IPv4 address space of Mainland China and send requests to the sampled IP address to resolve a subdomain within our controlled domain. The DNS discovery process lasted for about two weeks. Table I summarizes the discovered DNS servers. These name servers are sort of a uniform sample of the whole name server set in Mainland China.

We compare DMS with King, which predicts the latency between two hosts via the latency between their authoritative name servers. King is one of the most widely adopted methods in the wild, and is the state-of-the-art approach that can predict the latency between two arbitrary hosts in the network without the active participation of hosts. It has high feasibility in the wild because it is lightweight with good accuracy. Moreover, it incurs low deployment cost as it does not need to deploy numerous dedicated servers in the network. As King is not always feasible in many cases, we sample the Mainland China's IPv4 address space again and select the hosts that can be measured both by King and Ping simultaneously. After that, the *network coordinate module* constructs a 9-D metric space for the discovered name servers and measurable ordinary hosts. Table II summarize the number of landmarks, discovered DNS servers, and the ordinary hosts in the evaluation.

B. Comparison

In this subsection, we validate DMS's effectiveness when predicting the real-time latency between two hosts and statistically measuring the latency between two groups of hosts. Among the two groups, group A consists of the measurable hosts mentioned in the aforementioned subsection, while group B consists of the vantage points, which we have control with. Group B has the ability to conduct Ping operations to measure their latencies to the group A. We compare DMS's results with the those predicted by King and measured by Ping (real-time measurement).

1) Predicting Real-Time Latency: Real-time latency between two arbitrary hosts is of great significance to many real-time applications, such as the request routing in CDN or the peer selection in P2P systems. In order to verify the accuracy when DMS is used to predict the real-time latency, we compared the real-time latencies from hosts in group B to hosts in group A predicted by DMS and King with the measured latencies by Ping. The evaluation metric, that is, relative error, is defined as follow:

$$\text{relative error} = \frac{|\text{predicted latency} - \text{measured latency}|}{\text{measured latency}}. \tag{9}$$

Thus, the closer to zero the value of the relative error is, the more accurate the predicted end-to-end latency is. As the measurement process of the three methods cannot be conducted absolutely at the same time and the network status stays relatively static within a short period, we think two measurements are conducted synchronously if their conduction time points differs from each other less than one minute.

Fig. 5 shows the cumulative distribution of the relative errors for the methods. DMS can predict the latency between two hosts with relative error less than 0.5 for 75.5% of the cases, while King has relative error less than 0.5 for just 57.0%. In other words, compared with King, DMS can achieve more accurate predicted latency in general. What is more, both DMS and King have the relative error less than 1 with a high probability. Note that the Cumulative Distribution Function (CDF) of DMS is

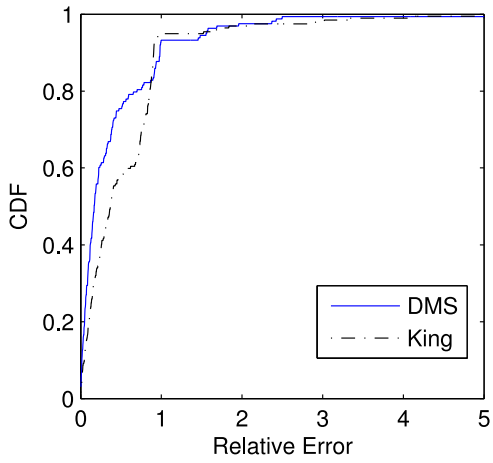

Fig. 5. Cumulative distribution curves of DMS's and King's relative error.

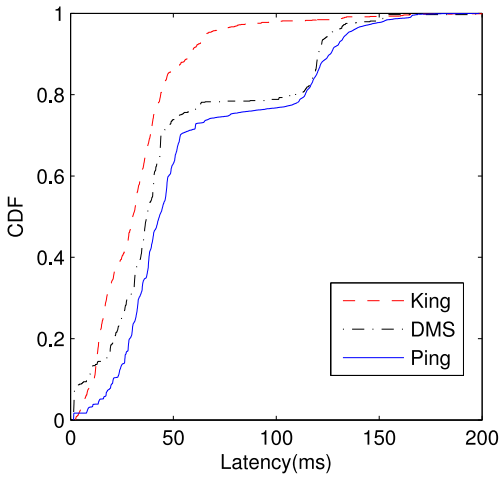

Fig. 6. CDF curves of end-to-end latencies between two groups over a period.

lower than that of King when relative error is about 1. This may be because that DMS will group two hosts to the same group if they are close in the network. In this scenario, the real latency between the two hosts should be small, which can be ignored for many applications. As a result, the error of the real latency between the two hosts is small while the relative error would be 1.

2) Predicting Statistical Latency: As mentioned in Section I, statistical latency can benefit many offline applications, which would take advantage of the long-term statistical result for decision making, such as the CDN selections for ICPs. In order to validate DMS's effectiveness when DMS is used to predict the statistical latency, we compare the statistical latencies from hosts in group B to hosts in group A predicted by DMS and King with the statistical latencies measured by Ping.

Fig. 6 shows the cumulative distribution curves of the latencies between two groups in one day and a half. As shown in the figure, DMS has predicted values closer to the measured value of Ping. The relative error of the predicted statistical latency of DMS is within 2%, while the relative error of King is around 35%.

Fig. 6 is sort of counter-intuitive that the CDF curves measured by Ping and DMS are segmented. We explored the measured data

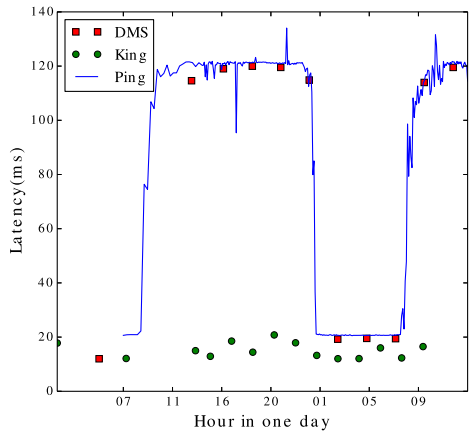

Fig. 7. Latency between a given two hosts with the time.

in depth and found some interesting things, as Fig. 7 shows. There exist links in the network whose latency differs greatly between different time in a day, which may result from their resident network's policy. Fig. 7 also reveals that DMS can effectively predict the dynamic latencies between two hosts with time, while King fails. This can be understood because King predicts the latencies of two hosts using their authoritative name servers, while DMS using their closest name servers, which at most cases have the similar network status with the hosts.

C. Sources of Inaccuracy

So far we have compared the accuracy of DMS with King, but where the inaccuracy of DMS comes from is unclear. In this section, we discuss several sources of the inaccuracy.

1) Subset of DNS Servers: DMS uses the existing DNS infrastructure to help predict the latency between two arbitrary hosts, so the number of discovered usable domain name servers would affect the accuracy of DMS. Actually, the discovered DNS servers can be taken as a uniform sample to the total DNS set, and spread across the Internet. As the *DNS discovery module* continuously samples the IP address space and discovers usable domain name servers, the accuracy of DMS would be enhanced, as shown in Fig. 8. Note that the overall overhead of DMS is $O(K^2 + KN + N + M^2)$, where N is the number of users in the network, K is the number of landmarks, and M is the number of open recursive DNS servers. The measurement overhead of the DNS measurement module will be increased quadratically with M. Based on our experiments, 1K open recursive DNS servers are already sufficient to achieve good enough accuracy for the Internet in Mainland China. In other words, M and K are far less than N, and the overall overhead of DMS is $O(KN)$.

2) Mapping Hosts to DNS Servers: Since DMS takes advantage of the distance in a metric space to map a host to its closest domain server, the inaccuracy of this mapping procedure would incur the inaccuracy of DMS. Then, a question comes: assume that there is a set of discovered domain name servers, what is the probability that a host are mapped to the domain name server closest to it?

Although we can calculate the distances between the name servers and arbitrary hosts in the metric space, we do not have

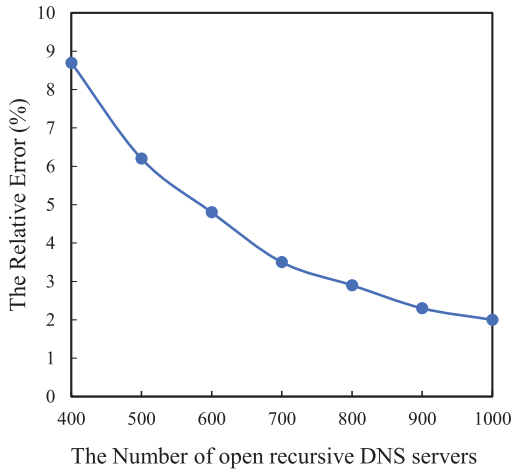

Fig. 8. Relationship between the accuracy of DMS for statistical latency prediction and the number of usable domain name servers.

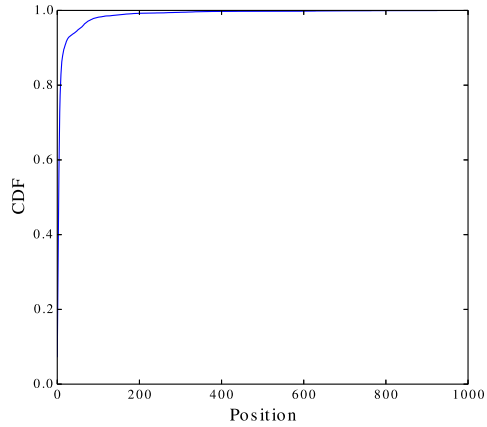

Fig. 9. Cumulative distribution of the probability that the shortest edge in predicted latency is within top x closest edges in measured latency.

enough controlled hosts that can actively measure their distance to the name servers. Due to this limitation, we transform the original question into a new one: there are a set of edges with measured latencies and predicted latencies, what is the probability that the shortest edge based on the predicted latency is just the shortest one based on the measured latency?

To answer this question, we construct a metric space, as described in Section V. Without loss of generality, we sample 1000 edges whose latency can be measured with Internet Control Message Protocol (ICMP) Ping packets, rank them by the measured latency, and number their positions from 1 to 1000. An edge located in position 1 means that it is the shortest edge with measured latency. After that, we select the shortest edge with predicted latency in the metric space, and locate its position in the ranked list. We repeat the process for 10 000 times, and get the probability of each position x, which means that the shortest edge in the metric space have the probability to be the x_{th} closest edge in measured latency.

Fig. 9 shows the cumulative distribution of the probability that the shortest edge in predicted latency is within the x_{th} shortest edge in measured latency. As expected, GNP has relatively high

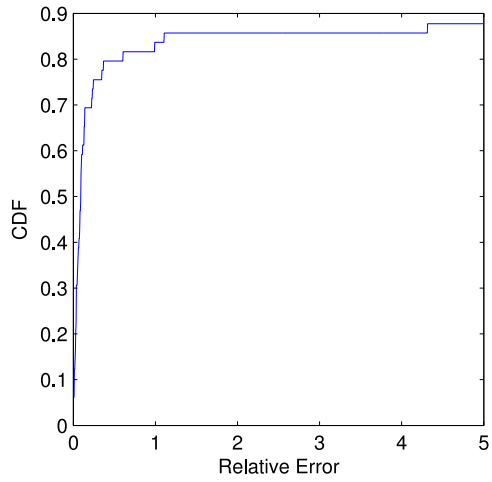

Fig. 10. Cumulative distribution of *DNS measurement module*'s relative error when predicting the latency between two DNS servers.

rank accuracy and a host could be mapped to its top ten closest name servers with probability about 90%.

3) Measuring Latencies Between DNS Servers: As DMS predicts the latency between two hosts by the latency between the corresponding two DNS servers, the inaccuracy of the measured latency between two DNS servers would also incur the inaccuracy.

To quantify the measurement error between DNS servers, we used the deployed name server to measure its latency to the discovered usable name servers in the following two ways: the technique adopted in the *DNS measurement module* and the simple ICMP ping messages. Fig. 10 shows the cumulative distribution of *DNS measurement module*'s relative error. As shown, DMS can measure the latency between domain name servers with relative error less than 0.367 for about 80% of the cases. In the future, DMS can replace the technique in the *DNS measurement module* with other DNS measurement techniques, if there emerge techniques with higher accuracy for measuring the latencies between domain name servers.

VI. Conclusion

In this article, we presented a novel statistical latency measurement framework called DMS. DMS leveraged the existing open recursive domain name servers in the network to reduce the measurement cost and takes advantage of a metric space to guarantee the prediction accuracy. Theoretical analysis showed that the predicted latency is statistically approximate to the actual value. To verify the effectiveness of DMS, we implemented a prototype system in the network. The real-world large-scale measurement results showed that DMS achieves much more accurate predicted values than the existing methods, no matter for real-time end-to-end latencies or statistical latencies over a large number of paths.

Acknowledgment

This article reflects only the authors' view. The European Union Commission is not responsible for any use that may be made of the information it contains.

REFERENCES

[1] M. Katz, P. Pirinen, and H. Posti, "Towards 6G: Getting ready for the next decade," in *Proc. 16th Int. Symp. Wireless Commun. Syst.*, 2019, pp. 714–718.

[2] X. Zhang *et al.*, "Improving cloud gaming experience through mobile edge computing," *IEEE Wireless Commun.*, vol. 26, no. 4, pp. 178–183, Aug. 2019.

[3] S. Livatino, F. Banno, and G. Muscato, "3-D integration of robot vision and laser data with semiautomatic calibration in augmented reality stereoscopic visual interface," *IEEE Trans. Ind. Informat.*, vol. 8, no. 1, pp. 69–77, Feb. 2012.

[4] Z. Lv, X. Li, H. Lv, and W. Xiu, "BIM big data storage in WebVRGIS," *IEEE Trans. Ind. Informat.*, vol. 16, no. 4, pp. 2566–2573, Apr. 2020.

[5] P. P. Ray, N. Kumar, and M. Guizani, "A vision on 6G-enabled NIB: Requirements, technologies, deployments and prospects," in IEEE Wireless Communications, early access, 10 May 2021, doi: 10.1109/MWC.001.2000384.

[6] J. Xue, D. Choffnes, and J. Wang, "CDNs meet CN an empirical study of CDN deployments in China," *IEEE Access*, vol. 5, pp. 5292–5305, Mar. 2017.

[7] M. Pozza, A. Rao, H. Flinck, and S. Tarkoma, "Network-in-a-box: A survey about on-demand flexible networks," *IEEE Commun. Surv. Tut.*, vol. 20, no. 3, pp. 2407–2428, Thirdquarter 2018.

[8] A. A. Atallah, G. B. Hamad, and O. A. Mohamed, "Routing and scheduling of time-triggered traffic in time-sensitive networks," *IEEE Trans. Ind. Informat.*, vol. 16, no. 7, pp. 4525–4534, Jul. 2020.

[9] X. Zhang, H. Yin, D. O. Wu, H. Huang, G. Min, and Y. Zhang, "SSL: A surrogate-based method for large-scale statistical latency measurement," *IEEE Trans. Serv. Comput.*, vol. 13, no. 5, pp. 958–968, Sep./Oct. 2020.

[10] Verizon, IP Latency Statistics, [Online]. Available: http://www.verizonenterprise.com/about/network/latency/ Access date: Feb. 7, 2021.

[11] G. Min and M. Ould-Khaoua, "A performance model for wormhole-switched interconnection networks under self-similar traffic," *IEEE Trans. Comput.*, vol. 53, no. 5, pp. 601–613, May 2004.

[12] *How to Select a CDN*. [Online]. Available: https://www.cdnplanet.com/blog/how-select-cdn/ Access date: Jun. 20, 2020.

[13] H. V. Madhyastha *et al.*, "iPlane: An information plane for distributed services," in *Proc. 7th Symp. Operating Syst. Des. Implementation*, Berkeley, CA, USA, 2006, pp. 367–380.

[14] T. Ng and H. Zhang, "Predicting internet network distance with coordinates-based approaches," in *Proc. 21st Annu. Joint Conf. IEEE Comput. Commun. Soc.*, 2002, pp. 170–179.

[15] K. P. Gummadi, S. Saroiu, and S. D. Gribble, "King: Estimating latency between arbitrary internet end hosts," in *Proc. 2nd ACM SIGCOMM Workshop Internet Meas.*, New York, NY, USA, 2002, pp. 5–18.

[16] D. Leonard and D. Loguinov, "Turbo king: Framework for large-scale internet delay measurements," in *Proc. IEEE 27th Conf. Comput. Commun.*, 2008, pp. 31–35.

[17] P. Francis *et al.*, "IDMaps: A global internet host distance estimation service," *IEEE/ACM Trans. Netw.*, vol. 9, no. 5, pp. 525–540, Oct. 2001.

[18] M. Costa, M. Castro, A. Rowstron, and P. Key, "PIC: Practical internet coordinates for distance estimation," in *Proc. 24th Int. Conf. Distrib. Comput. Syst.*, 2004, pp. 178–187.

[19] F. Dabek, R. Cox, F. Kaashoek, and R. Morris, "Vivaldi: A decentralized network coordinate system," in *Proc. Conf. Appl., Technol., Archit., Protocols Comput. Commun.*, New York, NY, USA, 2004, pp. 15–26.

[20] Y. Mao and L. K. Saul, "Modeling distances in large-scale networks by matrix factorization," in *Proc. 4th ACM SIGCOMM Conf. Internet Meas.*, New York, NY, USA, 2004, pp. 278–287.

[21] Y. Mao, L. Saul, and J. Smith, "IDES: An internet distance estimation service for large networks,"*IEEE J. Sel. Areas Commun.*, vol. 24, no. 12, pp. 2273–2284, Dec. 2006.

[22] J. Ledlie, P. Gardner, and M. Seltzer, "Network coordinates in the wild," in *Proc. 4th USENIX Conf. Networked Syst. Des. Implementation*, Berkeley, CA, USA, 2007, pp. 22–22.

[23] A. Jain and J. Pasquale, "Internet distance prediction using node-pair geography," in *Proc. 11th IEEE Int. Symp. Netw. Comput. Appl.*, 2012, pp. 71–78.

[24] Y. Fu and X. Xu, "Self-stabilized distributed network distance prediction," *IEEE/ACM Trans. Netw.*, vol. 25, no. 1, pp. 451–464, Feb. 2017.

[25] S. A. Mohammed, S. Shirmohammadi, and S. Altamimi, "A multimodal deep learning-based distributed network latency measurement system," *IEEE Trans. Instrum. Meas.*, vol. 69, no. 5, pp. 2487–2494, May 2020.

[26] K. Schomp, T. Callahan, M. Rabinovich, and M. Allman, "On measuring the client-side DNS infrastructure," in *Proc. Conf. Internet Meas. Conf.*, New York, NY, USA, 2013, pp. 77–90.

[27] P. S. Bradley, U. Fayyad, and C. Reina, "Scaling clustering algorithms to large databases," in *Proc. 4th Int. Conf. Knowl. Discov. Data Mining*, 1998, Art. no. 9C15.

[28] L. Tang and M. Crovella, "Virtual landmarks for the internet," in *Proc. 3rd ACM SIGCOMM Conf. Internet Meas.*, New York, NY, USA, 2003, pp. 143–152.

Xu Zhang (Member, IEEE) received the B.Sc. degree in communication engineering from the Beijing University of Posts and Telecommunications, Beijing, China, in 2012, the Ph.D. degree in computer science from the Department of Computer Science and Technology, Tsinghua University, Beijing, in 2017.

He is currently a Marie Sklodowska-Curie Individual Fellowship (Research Fellow) with the College of Engineering, Mathematics and Physical Sciences, University of Exeter, Exeter, U.K. His research interests include artificial intelligence, multimedia communication, and network measurement.

Geyong Min (Member, IEEE) received the B.Sc. degree in computer science from the Huazhong University of Science and Technology, Wuhan, China, in 1995, and the Ph.D. degree in computing science from the University of Glasgow, Glasgow, U. K., in 2003.

He is currently a Professor in high performance computing and networking with the Department of Computer Science, College of Engineering, Mathematics and Physical Sciences, University of Exeter, Exeter, U.K. His research interests include computer networks, wireless communications, parallel and distributed computing, ubiquitous computing, multimedia systems, and modeling and performance engineering.

Qilin Fan (Member, IEEE) received the B.E. degree in software engineering from the College of Software Engineering, Sichuan University, Chengdu, China, in 2011, and the Ph.D. degree in computer science from the Department of Computer Science and Technology, Tsinghua University, Beijing, China, in 2017.

He is currently a Lecturer with the School of Big Data and Software Engineering, Chongqing University, Chongqing, China. Her research interests include network optimization, mobile edge computing and caching, network virtualization, and machine learning.

Hao Yin (Member, IEEE) received the B.S., M.E., and Ph.D. degrees in electrical engineering from the Huazhong University of Science and Technology, Wuhan, China, in 1996, 1999, and 2002, respectively.

He is currently a Professor with the Research Institute of Information Technology (RIIT), Tsinghua University, Beijing, China. His research interests include broad aspects of multimedia communication and computer networks.

Dr. Yin was elected as the New Century Excellent Talent of the Chinese Ministry of Education in 2009, and was the recipient of the Chinese National Science Foundation for Excellent Young Scholars in 2012.

Dapeng Oliver Wu (Fellow, IEEE) received the B.E. degree in electrical engineering from the Huazhong University of Science and Technology, Wuhan, China, in 1990, the M.E. degree in electrical engineering from the Beijing University of Posts and Telecommunications, Beijing, China, in 1997, and the Ph.D. degree in electrical and computer engineering from Carnegie Mellon University, Pittsburgh, PA, USA, in 2003.

He is currently a Professor with the Department of Electrical and Computer Engineering, University of Florida, Gainesville, FL, USA. His research interests include the areas of networking, communications, signal processing, computer vision, machine learning, smart grid, and information and network security.

Zhan Ma (Senior Member, IEEE) received the B.S. and M.S. degrees in electrical engineering from the Huazhong University of Science and Technology, Wuhan, China, in 2004 and 2006, respectively, and the Ph.D. degree in electrical engineering from the Tandon School of Engineering of New York University (formerly Polytechnic University, Brooklyn, NY, USA), New York, NY, in 2011.

He is currently with the Faculty of Electronic Science and Engineering School, Nanjing University, Nanjing, China. From 2011 to 2014, he was with Samsung Research America, Dallas, TX, USA, and Futurewei Technologies, Inc., Santa Clara, CA, USA. His current research interests include the video compression, gigapixel streaming, and multispectral signal processing.