Understanding the latency to visit websites in China: an infrastructure perspective

Shuying Zhuang[a,b], Jessie Hui Wang[a,b,*], Pei Zhang[c], Jilong Wang[a,b]

[a] *Institute for Network Sciences and Cyberspace, Tsinghua University, China*
[b] *Beijing National Research Center for Information Science and Technology, China*
[c] *Institute of Network Technology, Beijing University of Posts and Telecommunications, China*

Abstract

As the Internet becomes more and more widely used in our daily life, its latency has been regarded as a truly critical issue for various Internet applications. Researchers propose an ambitious goal to pursue "Speed-of-Light Internet". However, there is a large gap between the reality and the goal, and a lot of infrastructure inefficiency caused by simple issues has been observed in some developing or rural regions. In this article, we conduct a measurement study on the latency to visit more than 638K websites in China from five well-connected vantage points, and examine the unnecessary latency caused by the inefficiency of DNS infrastructure and Internet routing infrastructure in China. We find that DNS resolution is the most significant contributor to the overall latency and investigate several important factors that affect DNS latency, *i.e.*, caching, CNAME and delegation. In terms of Internet routing infrastructure in China, we observe that 1) the inter-domain routing in China relies too heavily on the three oldest IXPs while the newly deployed IXPs are under-used significantly, which results in unnecessary routing circuitousness; and 2) most congested links are the links connecting cities with IXPs and these links should be upgraded or utilized with more efficient traffic engineering systems. Furthermore, in order to compute circuitousness ratios and locate congested links, we also propose a method to geolocate IP addresses in China more accurately and we make the geolocation results for IP addresses under the study of this work publicly accessible to facilitate future research efforts.

Keywords: Performance, Website, Network measurement, Routing, Networking infrastructure, DNS

1. Introduction

It is commonly admitted that our daily life is more and more dependent on diverse networking applications. Network users feel an increasing desire to make these applications more responsive, therefore latency has been recognized as one of the most important performance metrics because it is critical for user experience. An Akamai study in 2017 found that a latency deterioration of 100ms in website load time could hurt conversion rates by 7% [1]. Furthermore, low latency is essential for people to realize the full potential of some applications such as telemedicine and telepresence [2]. Researchers have noticed that bandwidth is not the panacea and we should try to remove unnecessary delays at every level of the stack. In 2013, the Internet Society organized a Workshop on Reducing Internet Latency to discuss the action plan to reduce Internet latency [3] and the concept of "Speed-of-Light Internet" was proposed in [2].

On one hand, the networking community is ambitious to pursue an Internet that is close to its theoretical limit by investigating every component of data communication, such as the protocol overheads, CPU, disk, *etc.* On the other hand, the reality is that the Internet infrastructure is deployed and utilized very inefficiently in many developing countries and rural regions and latency can be reduced a lot by exploiting traditional and simple techniques without any complex or novel changes [4]. For example, it was found that traffic going into Cuba was often routed on a high-latency satellite link although the ALBA-1 submarine cable had been available and can be used for the traffic [5]. Researchers also reported that some areas in Africa had consistently high delay because of the use of transit providers that route traffic through Europe and North America [6]. In terms of DNS resolution latency, authors of the work [7] pointed out that the uneven distribution of top level DNS servers had a significant impact on end-user latency in different continents. All these issues are caused by infrastructure inefficiency.

Noticing the ambitious goal and the observed inefficiency, we are curious about how far away the communication latency achieved by current Internet infrastructure is from the optimal in China and the causes of unnecessary latencies if there are. Although China is the country with the largest population of Internet users, the Internet performance in China is known to be unsatisfactory. In recent years, China puts quite a few efforts into Internet infrastructure. For example, ten more Internet Exchange Points (IXPs) were built to foster the local inter-domain

*Corresponding author
Email address: jessiewang@tsinghua.edu.cn (Jessie Hui Wang)

traffic exchange. Are these IXPs used efficiently?

In this article, we conduct a measurement study on the Internet latency in China. We deploy five well-connected vantage points in the three cities, *i.e.*, Beijing, Shanghai and Guangzhou. These cities are selected because the oldest three IXPs are located there and they are thought to be with the best network performance among cities in China. From these vantage points, we *curl*, *ping* and *traceroute* one million websites in China and collect data from about 638K websites successfully. The results of curl include the latencies of four major components, *i.e.*, DNS resolution, TCP connection, server processing and data transfer. Based on them, we present an overview of the latency to visit websites and find that DNS resolution is the most significant contributor to the overall latency.

DNS and IP routing are two most important services provided by Internet infrastructure [8]. We then focus on the two components, *i.e.*, DNS resolution latency and TCP connection latency, to investigate the efficiency of infrastructure. Note that TCP connection latency is equal to the round-trip time of communication, which is determined by ISP's routing policies and traffic load on the selected paths. From the analysis of DNS resolution latency, we report the influence of "CNAME" and delegations caused by delegated DNS providers or CDN providers. We also conduct a controlled experiment to learn the latency to query the authoritative servers at different levels, *e.g.*, top-level domain (TLD), second-level domain (SLD), *etc.*

The efficiency of the routing service of Internet infrastructure can be evaluated by comparing the RTT between two nodes with *f-latency*, which is defined as the geographical distance (measured along the surface of the earth) between the source and the destination divided by 2/3 of the speed of light. Our traceroute results show that the inflation over f-latency is common. In particular, the traces with inflation over f-latency larger than 5 account for 12.6% of all traces we collected. Routing circuitousness and congested links are two major reasons for the inflations. We then investigate routing circuitousness, *e.g.*, the non-optimal IXP selections and the under-utilization of newly deployed IXPs, and the topological and geographical locations of congested links.

In order to calculate f-latency and locate congested links, we have to be able to geolocate IP addresses in China. It has been reported geolocation databases are inconsistent with each other on finer granularities than the country level [9] and many databases are not reliable for China's Internet [10]. Therefore, we develop a two-round voting algorithm to construct a reliable database via a fusion of multiple databases and incorporate multiple techniques to further calibrate the fusion results.

In summary, we make the following contributions.

- We present an overview of the latency to visit about 638K websites in China, including the overall latency and each component latency.
- We analyze the most significant contributor to the overall latency, *i.e.*, DNS resolution latency, and report the influence of "CNAME" and delegations caused by delegated DNS providers or CDN providers.
- We observe that the inter-domain routing in China relies too heavily on the three oldest IXPs and the newly deployed IXPs are under-used significantly.
- By locating congested links, we find that most of congested links are connecting cities of IXPs and they require upgrades of capacity or better traffic engineering systems to reduce traffic loads on them.
- We develop a two-round voting algorithm and multiple calibration techniques to geolocate IP addresses in China more accurately and make the result publicly accessible.

In summary, our study demonstrates that the Internet infrastructure in China is deployed and utilized very inefficiently and even very traditional and simple techniques can reduce communication latency and improve users' experience.

The rest of this article is organized as follows. Section 2 introduces related works. In Section 3, we present an overview of the latency to visit websites and discuss the correlation of websites' performance with their popularities. Section 4 focuses on DNS resolution latency and shows the influence of CNAME and delegations. The TCP connection latency (RTT) is discussed in Section 5. We investigate two major reasons for latency inflation, *i.e.*, routing circuitousness and congested links, and report our observations on the inefficiency of routing infrastructure in China. Section 6 concludes the article.

2. Related Work

Internet latency has been a research focus for a long time because it has a significant influence on user experience of networking applications. As early as about twenty years ago, researchers had conducted measurements to understand the sources of latency in downloading web pages [11] [12] [13] [14]. In [11], the authors present several sources, *i.e.*, DNS, TCP, the Web server, and the network links and routers, and conclude that the bottleneck in accessing pages is due to the Internet latency and TCP mechanism instead of servers. In [13], the authors also notice that DNS, TCP and start-of-session delays of HTTP are three major sources and present some simple techniques to reduce the latency caused by these sources. In [14], the authors find that the DNS lookup contributes more than one second for about 20% of the Web objects.

Besides these early works focusing on web visiting latency at that time, there are also some recent works on Internet latency in general [15]. As the Internet becomes more and more widely used in our daily life, its latency has been regarded as a truly critical issue for various Internet applications. In 2013, the Internet Society organized a Workshop on Reducing Internet Latency to discuss the action plan to reduce Internet latency [3]. The concept of "Speed-of-Light Internet" is proposed in [2]. In PAM 2017, two papers are presented to answer the questions of where my time has gone [16] and why the Internet is so slow [4]. In [6], Formoso *et. al.* focus particularly on inter-country latencies in Africa, whose Internet penetration rate is the lowest in the world. The networking performance of some countries, such as Ghana [17], Cuba [5] and Zambia [18], are also analyzed.

These works focus on different latency contributors. For example, the work [16] studies the end-to-end latency from the application level to the wire and asserts the latency within host should be paid more attention. The authors of [4] aim to achieve the goal of building a speed-of-light Internet. They conduct a study on latency inflation in the Internet across the network stack and reveal the problem that the infrastructure is inefficient in today's Internet. The work [6] also studies the latency issue from the perspective of infrastructure. Similar to [4] and [6], our work also focuses on the issue of infrastructure inefficiency instead of protocol overheads. Particularly, we focus on the DNS and routing infrastructure in China.

The performance of DNS is also a traditional problem and has been investigated since many years ago. Jung *et. al.* study the latency and failures of DNS and evaluate the effectiveness of DNS caching in [19]. In [20], the authors further propose a model to predict hit rate as a function of request arrival times and the choice of TTL. In [21], the relationships among domain names is exploited by authoritative DNS servers to piggyback answers for future queries as part of the response for an initial query. They also report that average DNS latency is about 200-300ms but can be very large (multiple seconds) in worst cases. The work [7] points out that roughly speaking top level DNS provides lower query latency in Europe and North America than other continents. The authors of [22] study how modern web services (including third-party hosting services for websites) deploy authoritative DNS (ADNS) for their domain names, especially the deployment patterns and the characteristics of different deployment styles. The negative impact of "one-time-use" domains on the effectiveness of DNS caching is studied in [23], and the authors also develop a simple classifier to mitigate the negative impact. In this paper, we notice that the recursive resolution of a domain name involves multiple steps and servers (TLD, SLD *etc.*). From controlled experiments, we find caching SLDs can effectively reduce the resolution latency, and the appearance of "CNAME" or delegated name servers (some "NS" records) would incur significant resolution latency in case of no caching.

Routing circuitousness and congestion are two of the major sources of Internet latency and thus draw a lot of attention from researchers. Gao *et. al.* in [24] report AS path inflation in the Internet is more prevalent than expected and they also try to find the routing policies that cause these inflations. In [25], the authors focus on geographical circuitousness (distance inflations) and report that routing circuitousness of our Internet is deteriorating in these years. The work [26] investigates the geographical characteristics of the Internet and reports the ingress-to-egress subpaths are less circuitous than the end-to-end paths, which demonstrates that backbone infrastructures and routing schemes deployed by ASes are efficient. Our measurement results are consistent with [26] on this point.

The challenges of locating congested links also draw attention from some researchers. In [27], [28] and [29], the authors propose a method, named Time Sequence Latency Probes (TSLP), to analyze the intensity and location natures of congested links. They do not find evidence of widespread persistent congestion on monitored inter-domain links, but find some links exhibit recurring congestion patterns. Their works mainly focus on inter-domain links in USA. TSLP is also used in [30] to investigate the causes of congestions on the African IXP Substrate. In this work, we exploit their idea of how to locate congested links and conduct measurement for intra-domain links of ISPs in China. We find some links that are congested for more than 5% of the time each day and suggest these links should be upgraded.

Geolocation of IP addresses [31] is essential for our study. Many researchers have proposed algorithms to locate IP addresses accurately such as [32] [33] [34]. There have been some commercial databases and public databases, but researchers find that geolocation databases are far from being as reliable as they claim [35]. In [9], the authors find databases generally agree on IP-address-to-country mappings but they are doubtful on a finer granularity. The authors of [36] also report that the databases are not reliable for geolocating routers and that there is room to improve their country- and city-level accuracy. More importantly, the work [10] points out that China's Internet is complex and has its unique structural feature, therefore many databases do not adequately cover China's Internet. They propose a new methodology but no database is announced. In [37], the authors try to improve the geolocation accuracy of Chinese IP addresses by the "fusion" of multiple unreliable databases developed by Internet giants in China. In this work, we define a more effective metric to implement the database fusion, and we also develop more steps to further calibrate the results after data fusion. We make the result geolocation mappings public to facilitate other researchers [38].

3. Overview of the Latency

We deploy five well-connected vantage points in three cities, *i.e.*, Beijing, Shanghai and Guangzhou, which are thought to be with the best network performance among cities in China. The five points are located in different ISPs and cloud providers, *i.e.*, Beijing CERNET, Beijing Baidu Cloud, Guangzhou Tencent Cloud, Guangzhou multihoming (Telecom and CERNET), and Shanghai Ali Cloud.

From each vantage point, we run cURL [39] to visit every website in the set of top one million popular websites and more than 638K websites provide answers.

cURL would retrieve the default homepage of the target website and provide us statistics on the time it takes for the whole period and for each step. As we know, when we download a webpage, there are four steps, *i.e.*, 1) a DNS resolution is conducted to find out the IP address of the website; 2) our machine tries to connect to the target by conducting a three-way-handshake; 3) the website receives our request and processes it; 4) the data is transferred from the target website to our machine if the connection is set up successfully. We name these four steps as *DNS resolution*, *TCP connection*, *Processing*, and *Content transfer*. cURL provides the time used for each step. The latency of TCP connection is measured as the time between our machine sending the SYN and our machine receiving the SYN-ACK, which in fact is one round trip time (RTT); the latency of Processing is measured as the time for "request-response", *i.e.* from sending out the request to the receipt of the first byte, which is in fact one RTT plus the processing delay of the target server. In cURL, the latency of transfer is measured as the time from the receipt of the first byte to the receipt of the last byte.

In order to facilitate reading, in this article, we define "processing time" and "transfer time" slightly different from cURL's measurement results. Our "Processing time" is the processing delay of the target server, which is the latency of the third step minus one RTT, therefore the processing time can be used to evaluate the performance of the server. Since the sizes of webpages retrieved vary a lot, we normalize the total transfer latency by the page size, and plot the "Transfer time" as the time it would take to complete the transfer of one KB. We do not perform any processing on the DNS resolution latency, the TCP connection latency and the overall latency collected by cURL. Based on these definitions, we try to conduct measurements to understand the overall latency and each component latency to visit the top one million websites in China.

The Internet is dynamic and its performance varies. Fortunately our focus is on the statistics of the results about all websites instead of individual websites. We first conduct measurements for multiple rounds to see whether the statistics results of the latency to visit one million websites are influenced by temporary or time/day-dependent factors. In a single round, we cURL one million target websites from four vantage points[1]. We run measurements for seven rounds to cover seven days of one week. Each round takes about 15 hours. We intentionally start seven rounds at different time points of their corresponding days, *i.e.*, 19:00, 16:00, 13:00, 10:00, 7:00, 4:00 and 1:00 respectively. The comparison of the results from these seven rounds shows that the CDF curves for seven rounds are very similar and there is no significant difference between the median values of any two rounds. Due to space limitation, we skip the details of seven rounds and the measurement results of all seven rounds can be found at [38]. In Figure 1, we plot the CDF curves for the overall latency and also the latency of each component collected from a round with all five vantage points.

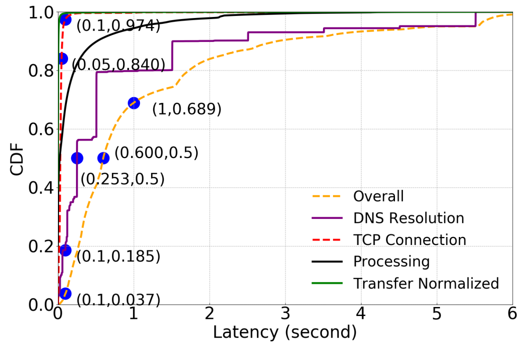

Figure 1: The overall latency and each component latency of visiting websites.

We can see that the overall latency is larger than 100ms for about 96.3% of websites, and about 31.1% of websites have delay of more than 1 second, which suggest the performance of most websites in China needs improvement.

Then we look into each of the four components to see their individual contributions to the overall latency. We find that DNS resolution makes the greatest contribution. The DNS latency is longer than 100ms for about 81.5% of connections. The median DNS latency is 253ms, which is about 42.2% of the median overall latency (600ms). In fact, by a simple calculation we find that there are about 59.4% websites whose DNS latency accounts for more than 50% of overall latency. We would like to make a comparison of the DNS latency with measurement results in other works to help readers better understand the situation. However, the measurement results on DNS latency are often presented for requests, *e.g.*, how many requests can be answered within 10ms. Then we derive an approximate CDF for the latency of requests using the popularities of websites, *i.e.*, the numbers of requests for individual websites. The approximate results show that 65% of requests are answered within 28.7ms and 11.2% of requests take shorter time than 10ms. As a comparison, the authors of [40] reported that "roughly two-thirds of the transactions complete in under 1 msec" and about 75% of the transactions take shorter time than 10ms. Their measurement was conducted in USA in 2011 and 2012, which is obviously better than the DNS latency in our measurement. It motivates our further analysis of DNS resolution latency in Section 4.

The TCP connection latency reflects the performance of the routing path from the measurement point to the target server. About 16.0% of connections are with a latency longer than 50ms, and about 2.6% of connections

[1]One vantage point is not available when the experiments are conducted.

experience a latency even longer than 100ms (which is equivalent to 30,000 Km considering the speed of light). Obviously, they indicate that the Internet in China is far away from the "Speed-of-Light Internet". We will conduct a further measurement study using the command traceroute in Section 5, and study why some paths experience longer connection latency than others.

In the remaining part of this section, we would look into whether websites' performance has correlations with the popularity and the web hosting providers.

3.1. *The Correlation of Websites' Popularity with Performance*

It is a natural conjecture that popular websites are likely to provide better performance. In order to study whether the conjecture is true, we classify all websites into several sets according to their popularities and plot statistics of each set. The popularity of one website is measured by the number of queries of its domain name received by a well-known local DNS server of a major ISP in China. We plot the CDF curve for website popularity in Figure 2. The websites under study are grouped into six sets, *i.e.*, S_i ($i \in [0, 5]$). S_i includes the websites with popularity between 10^{i-1} (exclusive) and 10^i (inclusive). Specially, S_0 includes all websites that are queried only once, and S_5 is the set of all websites with popularity larger than 10^4. The sizes of these sets are different but comparable. The sizes of S_4 and S_5 are relatively smaller than other sets, but we choose to not merge them because their popularities vary significantly and we would like to see more details about the statistics of these popular websites. In order to illustrate it, we mark the dividing points between sets in Figure 2.

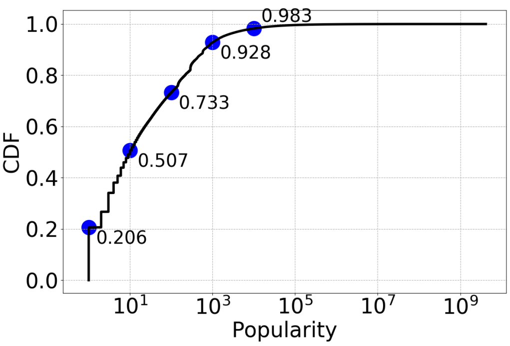

Figure 2: The cumulative distribution function of website popularity.

Then we make box plots for the performance of each set using the percentiles of 10, 25, 50, 75 and 90. The statistics of overall latency of the six sets are plotted in Figure 3. We can see that popular websites statistically tend to have smaller overall latency, although the 90 percentiles of these sets (representing the worst 10% websites) do not have evident difference. We also plot the statistics of each component latency in Figure 4. Statistically, we can see that popular websites tend to have smaller process time

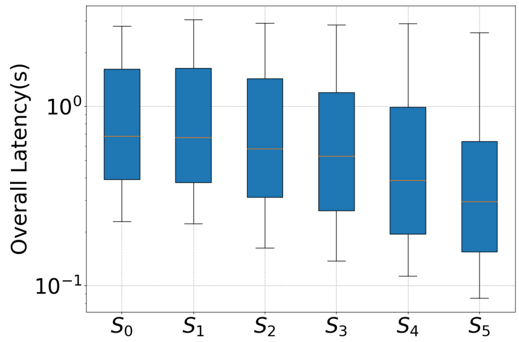

Figure 3: The popularity and overall latency of websites.

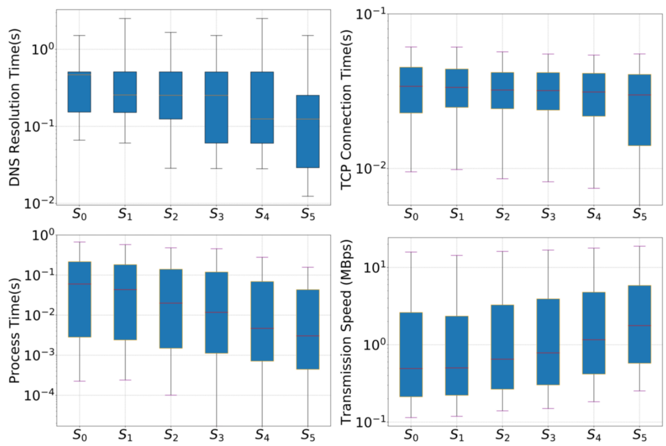

Figure 4: The popularity and performance of websites.

and larger transmission speed, which is consistent with our conjecture. On the DNS resolution time, the trend is not very evident, but we still can find that the websites in the set S_5 perform much better than the less popular websites. On the TCP connection time, the correlation between popularity and performance is not very strong. Roughly speaking, popular websites perform better than less-popular websites, and the median latency keeps decreasing from S_0 (least popular) to S_5 (most popular) although the decrease is relatively small.

3.2. *The Correlation of Websites' Hosting Provider with Performance*

The IP138 database [41] provides information about the corresponding Internet Service Provider (ISP) for each IP address. From it, we can find that IP addresses of some websites belong to "Tencent Cloud", "Ali Cloud" and "Baidu Cloud", which are three major cloud-based hosting providers in China.

It is possible that the performance might be good if we visit one server from the vantage point within the same cloud provider. In order to avoid such influence, we only use two vantage points, *i.e.*, Beijing CERNET and Guangdong multi-homing, in this study on hosting providers. Among all 638188 websites we curled successfully from these two vantage points, there are 158964 websites hosted by Ali Cloud, 14675 websites hosted by Tencent Cloud, and 9095 websites hosted by Baidu Cloud. We plot the overall latency to visit these websites from two vantage points in Figure 5. Statistically, the median overall latency of websites hosted by Ali Cloud and the median overall latency of websites hosted by Tencent Cloud are smaller than the websites not hosted by one of the three major cloud providers, but there is no evident difference between the median overall latency of websites using Baidu Cloud and the websites not hosted by one of the three major cloud providers.

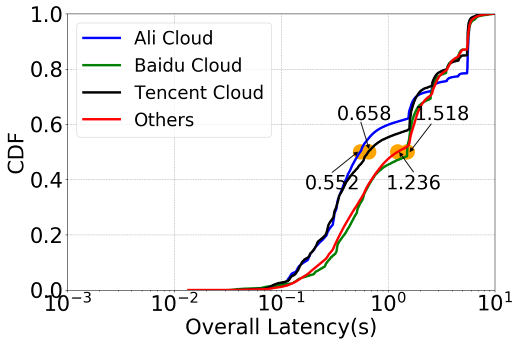

Figure 5: The hosting provider and overall latency of websites ("Others" indicates the websites not hosted by one of the three major cloud providers).

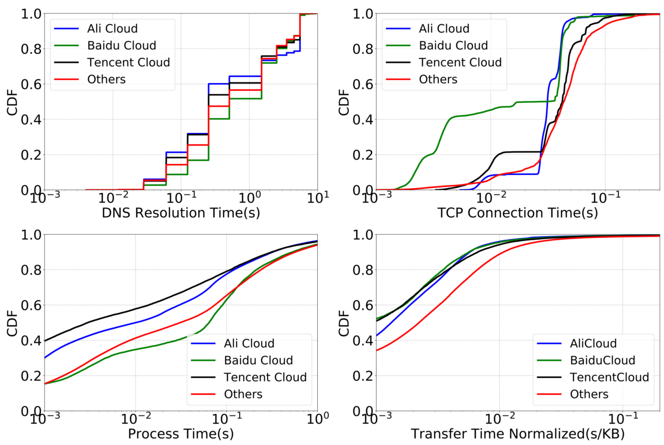

Figure 6: The hosting provider and performance of websites ("Others" indicates the websites not hosted by one of the three major cloud providers).

We further plot each component latency in Figure 6. On the DNS resolution time, statistically the websites hosted by Ali and Tencent are slightly better than the websites hosted by Baidu and the websites not hosted by the three major cloud providers. It is a possible reason for the smaller overall latency of the websites hosted by Ali and Tencent, because DNS resolution latency is the most significant contributor to the overall latency.

In terms of the transfer time (the reverse of transfer speed), it is clear that the websites hosted by clouds perform better than the websites not hosted by the three major cloud providers. Roughly speaking, the TCP connection time shows a similar phenomenon, although the curves of "Ali Cloud" and "Others" cross several times near 10^{-2} seconds. It seems that the websites hosted by cloud providers get better networking services than other websites. In the later sections, we will show that statistically the end-to-end paths associated with cloud providers are less circuitous and less congested, which may explain why the websites hosted by clouds perform better on the TCP connection time and the transfer time.

4. DNS Resolution Time

The measurement result presented in Figure 1 shows that DNS resolution time plays the most significant role in causing latency and the latency is more than 1 second for about 20.3% of websites. We can also see that its CDF curve is step-wise. In this section, we will conduct more measurements and experiments to understand why the DNS resolution time for some websites is so long and if there is any way to improve its performance.

4.1. An Example of DNS Resolution Procedure

Let us look into the procedure of DNS resolution. Figure 7 illustrates an example. The client submits its DNS resolution request about *www.baidu.com* to the pre-configured recursive DNS resolver S_{RDNS}. Let us assume that S_{RDNS} does not have any cached information now. Then it would start at the top of the name hierarchy by asking one of the root name servers, which is the step 2 in the figure. The root name server must know the authoritative name server for the top level domain *com*, and it returns the name and IP address of the top-level domain server, say S_{TLD}, in the step 3. Iteratively, S_{RDSN} continues its quest by sending the query to S_{TLD} and gets the name and IP address of the authoritative name server for the second level domain *baidu.com*, say S_{SLD}, which is step 4 and 5. The query to S_{SLD} is expected to get the information (IP address) of *www.baidu.com*. However, in this case, we only get a "CNAME" resource record instead of an IP address. It means S_{SLD} tells S_{RDNS} *www.baidu.com* has a canonical name *www.a.shifen.com*.

Now S_{RDNS} has to query for the IP address of *www.baidu.com* using this canonical name. Fortunately, it just gets information about the name servers for the domain *com* in step 2-3. So it can skip the step of sending query to root servers, and can go directly to S_{TLD} for the IP address of the name server in charge of *shifen.com*, say S^2_{SLD}. Further query to S^2_{SLD} would return the information about the name server in charge of the third level domain *a.shifen.com* and let us name it as S_{3LD}. Finally, S_{3LD} can return the IP address of *www.a.shifen.com* to S_{RDNS}, which returns the result to the client.

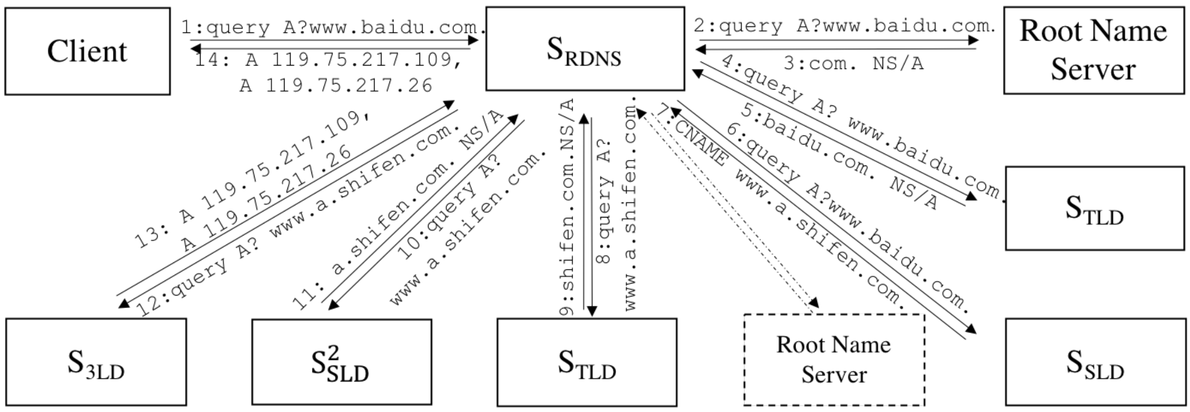

Figure 7: An example of DNS resolution procedure

4.2. The Influence of Popularity and CNAME

From the example, we can see that a query request from one client would result in a sequence of queries of the local recursive DNS server, which are defined as the request's *referrals*. For example, the query request in Figure 7 has six referrals. Obviously, reducing the number of referrals is helpful for improving DNS resolution latency.

In the above example, we assume S_{RDNS} does not have any cached information at the beginning. In real cases, S_{RDNS} would cache the information it gets (with defined TTL) to skip some referrals and reduce latency. For example, if S_{RDNS} has cached information about *baidu.com*, step 2-5 can be skipped and S_{RDNS} can communicate directly with S_{SLD}, then the resolution request can enjoy a shorter latency.

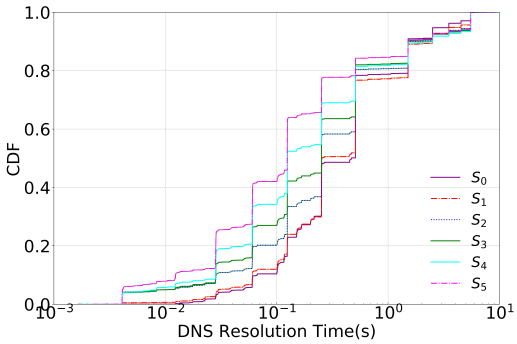

Figure 8: DNS resolution latency for websites with different popularities.

The domain names of popular websites are more likely to be cached by S_{RDNS}, therefore statistically popular websites tend to experience smaller DNS resolution delay. The first subplot of Figure 4 has presented percentiles of DNS resolution times of six sets with different popularities. We further plot CDF curves for DNS resolution times of these sets in Figure 8. It can be seen that statistically the sets of popular websites perform better than less popular websites in DNS resolution.

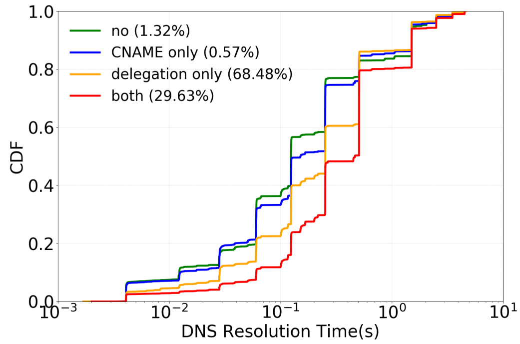

Figure 9: The influence of CNAME and delegation.

From the example we also can have the following observation. The authoritative servers of some websites do not have entries of their IP addresses, instead the servers just return their canonical names, and then more iterative queries are needed, which obviously increases latency. Besides CNAME, the appearance of delegated DNS provider can also increase resolution latency. For instance, one website gets its own domain name zone "*50yoga.cn*" but it does not want to deploy its own authoritative name server to answer queries for names in its zone. It can delegate the resolution of its domain zone to a DNS provider, *e.g.*, *HiChina*. HiChina is a subsidiary of Alibaba and its various name services are widely used by websites in China. In fact, we see many resource records in the format of (**.cn, NS, dns.hichina.com*), which possibly increases resolution latency because more queries are needed or more records should be checked to produce answer. There are also some CDN providers that require delegation of name resolution. An example for the worst case is as follows. A domain name (*www.cgwxq.com*) is delegated to DynDNS (*ns1.bdydns.cn*) that returns a CNAME (**.cnameaddress.top*), and the CNAME is delegated to DNSPod (*f1g1ns2.dnspod.net*). We see that a single query triggers queries to four domain names in this case.

The influence of "CNAME" and delegation on DNS resolution latency is shown in Figure 9. Although the group of "with CNAME only" appears to have better performance than the case "with delegation only" in the figure, our deeper investigation shows that it is not necessarily true. The resolution latency for a name with delegation depends on multiple factors, such as whether the record of the delegated server has been cached locally and the resolution latency of the delegated server.

4.3. Characterizing DNS Latency of Authoritative Name Servers of Different Levels

Caching is the most important way to improve DNS resolution performance. Unfortunately, it is infeasible to cache the information about all domain names. In the case that the whole domain name of the request is not cached, caching the suffix of the domain name, *i.e. top-level domain, second-level domain,* etc., is also helpful. Therefore, we try to conduct experiments to learn the effect of caching information about name servers of different levels.

We cannot control the cache of any recursive resolver that is in operation. Therefore, we deploy a recursive server by ourselves for our experiments, and we then have a full control of its cache. Then we try the following four strategies.

- **Cache NULL**. The server has nothing in its cache, and it has to start from root servers for each query request. In our experiment, we implement this strategy by clearing its cache before we run cURL for each website.

- **Cache TLD**. The server has cached information about all TLD name servers, thus it does not need to communicate with root servers. In our experiments, we implement this strategy by running the command *dig* for the corresponding TLD of the website before we run cURL for each website. For example, we would run *dig com* before we cURL *pop.music.sina.com.*

- **Cache SLD**. The server has cached information about all SLD name servers, thus it does not need to communicate with root servers and TLDs. In our experiments, we implement this strategy by running the command *dig* for the corresponding SLD of the website before we run cURL for each website. For example, we would run *dig sina.com* before we cURL *pop.music.sina.com.*

- **Cache ALL**. The server has cached information about the domain name of the website, thus it does not need to communicate with any authoritative name servers. In our experiments, we implement this strategy by running the command *dig* for the website before we run cURL for each website. For example, we would run *dig pop.music. sina.com* before we cURL *pop.music.sina.com.*

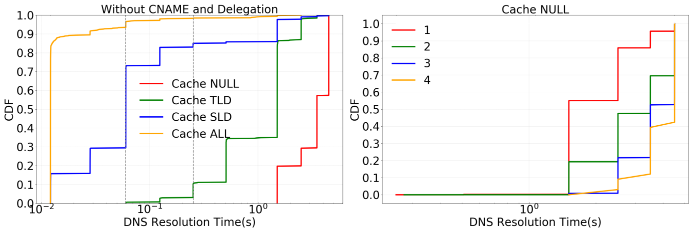

Figure 10: The effect of caching information about name servers of different levels.

We choose 100K websites randomly from all 1M websites and conduct experiments as described above. We plot the DNS resolution time of websites under these strategies in the left part of Figure 10. In order to avoid the influence of CNAME and delegation and concentrate on the effect of caching, in the left part we only focus on websites that have no CNAME and delegation. We also plot two vertical lines, wherein one is for 60ms which we think is acceptable and the other is for 253ms which is the median latency of our measurement results of the current Internet in China. We can see that having its SLD cached is important for a website to provide acceptable resolution latency even in the case without CNAME and delegation.

In the previous subsection, we have presented an example that a query of one website's name involves four different TLDs. Under the strategy of Cache NULL, it would trigger four contacts to DNS root servers. We classify websites according to the number of contacts to root servers to resolve the website's name and plot their latency under the strategy "Cache NULL" in right part of Figure 10. Clearly we see the number of contacts triggered by a resolution request plays a significant role in the DNS resolution latency of the request. Although "Cache NULL" is less likely to occur in the wild, our controlled measurement result indicates the negative effect of delegation and CNAME is exacerbated due to the miss of caching.

5. TCP connection time

TCP connection time is also an important component of the latency to visit websites. It is the RTT from the vantage point to the website, which reflects the performance of the routing path between them. As we know, the end-to-end latency of sending a packet from a source to a destination is the sum of the delay on each hop of the path, and the delay of one hop has four components: transmission delay, propagation delay, queuing delay and processing delay. The transmission delay and processing delay are usually very small, especially when the cut-through switching technology is used. The propagation delay is determined by the geographical distance the packet travels, and the queuing delay is related to the congestion level of the link. Ideally, one packet would like to be transmitted on the geographically shortest path without any queuing delay. The delay in this extremely ideal case is defined as *f-latency*, which is the geographical distance (measured along the surface of the earth) between the source and the destination divided by 2/3 of the speed of light. Please note that the speed of light in fiber is roughly 2/3 of the speed of light in vacuum. *f-latency* is the best latency achievable were a fiber cable laid along the geodesic between two end nodes.

The ratio of the measured latency to the corresponding f-latency is often used to evaluate how far away the performance of the network is from the ideal case [2]. Figure 11 presents RTTs from vantage points to websites in terms of inflation over f-latency. We see that there are about 36% of the traces whose inflation over f-latency is larger than 3. The inflation over f-latency is even larger than 5 for about 12.6% of traces. The same queueing latency would result in a larger inflation for traces with smaller geographical distance than traces with longer distance, and the inflation of a trace with smaller distance is also easy to be affected by the precision of geolocation, therefore we also plot separate curves for traces with different geographical distances, *i.e.*, smaller than 500KM, in the range [500KM, 1000KM), in the range [1000KM, 1500KM), and larger than 1500KM. These four sets take 11.87%, 16.84%, 41.26% and 30.03% of all traces respectively (from sets with shorter distance to sets with longer distance). We can see that the traces with inflation larger than 3 account for about 22% of all traces longer than 1000KM.

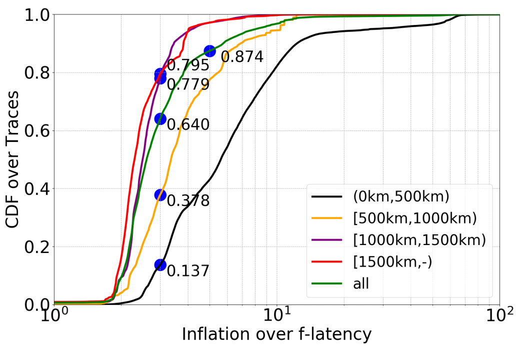

Figure 11: RTT in terms of inflation over f-latency.

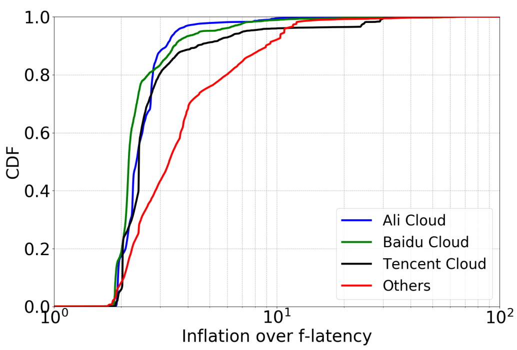

Figure 12: Inflation over f-latency for traces ("Others" indicates the websites not hosted by one of the three major cloud providers).

We also pay special attention to the routing performance of traces to websites hosted by cloud providers in Figure 12. From the figure, we can see that the traces from two non-cloud vantage points, Beijing CERNET and Guangdong multihoming, to websites hosted by three cloud providers statistically have smaller inflation than other inter-domain traces. It shows that routing to the websites hosted by clouds tend to have better performance, *i.e.*, either less circuitous or less congested.

The larger inflation in Figure 11 indicates that many traces are either circuitous or congested. Therefore, we think the following two questions are necessary to be answered to improve TCP connection latency: *1) whether the deployment of networking infrastructure and routing decisions enable that data packets can be transmitted on the geographically shortest path? 2) which links are the most congested and contribute the most to the long latency of some websites?*

In order to answer the above two questions, we have to know the routing paths between sources and destinations, both at the IP layer and the geography layer. *Traceroute* is the most widely used way to measure the IP path and corresponding RTT from a source to a destination. However, it is known that traceroute may lead to incorrect path inference because a series of probe messages during a traceroute to a single destination may traverse different paths [42]. *Paris traceroute* is developed to obtain a more precise view of the actual routing paths, which can ensure that packets in the same flow would traverse the same path even flow-based load balancers appear [43]. In this work, we take the advantage of *Scamper* [44] that supports Paris traceroute to collect IP layer paths. We also develop a method to improve the geolocation dataset for IP addresses in China. Based on the geolocation dataset we achieved, IP layer paths can be mapped to geography layer paths and then we are able to study whether these paths are circuitous geographically.

Paris traceroute also gives the RTT to each hop router along a path. Internet is not necessarily symmetric, which makes two ways to infer latency from traceroute results questionable, *i.e.*, halving the RTT to get one-way latency, and subtracting the RTT to two end-points to get link latency. Although *reverse traceroute* is proposed to measure reverse paths and reverse latencies, its accuracy and coverage cannot be guaranteed and it requires to deploy many vantage points and know the topology in advance [45]. Researchers are forced to use the above ways in their works, *e.g.*, [33] [46]. They try to mitigate its influence in subsequent steps, or validate their results using ground-truth to show their methods based on halving the RTT and subtracting the RTT work well [33] [46]. In this work, we try to reduce the negative influence of asymmetric paths as follows. First, we just geolocate nodes at a granularity of province, which is very coarse. Second, we avoid drawing any conclusion directly from a single inferred latency. For example, in Section 5.1.3, a single inferred latency is just a vote from a single trace, and the final geolocation result is from the votings of all traces.

In the remaining part of this section, we first introduce our method to improve the geolocation dataset for IP addresses in China. Then we analyze the circuitousness of routing paths and characterize congested bottlenecks in these traces.

5.1. Geolocation for IP addresses in China

Geolocation for IP addresses, *i.e.*, estimating the real-world geographical locations of IP addresses, is very useful in many applications. Unfortunately, although there have been a lot of public or commercial geolocation databases, it has been reported that geolocation databases generally agreed on IP-address-to-country mappings but there were a lot of inconsistencies on finer granularities than country [9]. Some works, such as [34], also pointed out that the accuracy of many commercial databases was low for IP addresses outside the USA.

In this work, we use six geolocation databases, *i.e.*, *Maxmind* [47], *IP2Location* [48], *DB-IP* [49], *IP138* [41], *IPIP* [50], and *Chunzhen* [51], wherein the last three databases focus on IP addresses in China. Since the accuracy tends to decrease for finer granularity resolutions, we choose to geolocate IP addresses at the province level using these six databases.

There are $606,103$ IP addresses in total in the traces collected by tracerouting all websites, and the set of these IP addresses is denoted by $\mathbb{P}$. We first investigate the consistency of answers from six databases for each of these IP addresses. Their answers can be classified into four classes: a province in China (which is what we want), China (but without a province name), a location outside China, and NULL (which means the database provides no information). The consistency is evaluated by counting the number of different locations in the answers from six databases for each IP address. Here, we do not count in the answer of "NULL" since there is no useful information.

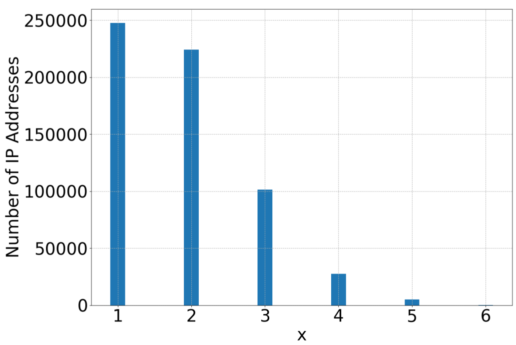

Figure 13: The number of IP addresses that get x provinces in six answers.

The result is plotted in Figure 13. There are about 40.8% of addresses each of that can get a single location from all its answers. 37.0% of addresses get two locations each, and 16.7% get three different locations each. There are even about 5.4% addresses that get more than three different locations each. We see there are a lot of inconsistencies and we have to improve the geolocation accuracy based on these databases.

5.1.1. Two-round voting to improve geolocation

We propose a two-round voting algorithm to determine the location (province) for each IP address. The first round voting is as follows. For each IP address, these databases vote for the location of this IP address, and "NULL" is considered to be an abstention vote. The location who receives the largest number of votes wins. If more than one locations get the same number of votes, we select one from them as the winner randomly. We temporarily assume the winner of one IP address is its location and try to find which database is more credible. Since we only focus on geolocating IP addresses in China at the province level, we do not count in any IP address whose winner is an oversea location or China without a province name. Let us define a set $\bar{\mathbb{P}}$ that includes every IP address whose winner is a province in China.

For each address $\pi_j \in \bar{\mathbb{P}}$, the databases (can be one or more than one) that vote for the winner of the address are considered to be correct on this IP address π_j, and we define a metric c_j to reflect the degree of our confidence in the winner of π_j as the ratio of the number of correct databases on π_j to the total number of usable databases (which is 6 in this paper).

For a database $D_i \in \mathbb{D}$ ($i \in [1, 6]$ in this work) and an IP address $\pi_j \in \bar{\mathbb{P}}$, we define that $S_{i,j} = c_j$ if the database is correct on the IP address and $S_{i,j} = -c_j$ otherwise. $S_{i,j}$ can be viewed as the *score* that D_i gets from π_j. Please note the winner in the first round voting is temporarily assumed to be the "correct" answer, but it is possible that the winner is not the correct answer. Therefore, we should also consider the degree of confidence on the "correct answer" of each IP address. For example, if the winner of the voting on π_1 (say D_3) receives 4 votes (thus $c_1 = 4/6$) and the winner of the voting on π_2 (say D_4) receives only 2 votes (thus $c_2 = 2/6$), obviously the voting result on π_1 is more likely to be the truth. Therefore, the databases who agree with D_3 on π_1 should receive a score of $4/6$ from π_1, larger than $2/6$, which is the score the databases who agree with D_4 on π_2 receive from π_2. On the other hand, the databases who disagree with the more credible "correct answer" should be punished more.

Based on $S_{i,j}$ defined above, we would like to define a metric to evaluate the credibility of a database D_i as follows,

$$
\begin{aligned}
\chi_i &= \frac{\rho_i}{\max\limits_{\forall D_n \in \mathbb{D}} \rho_n} \times \frac{\mu_i}{\max\limits_{\forall D_n \in \mathbb{D}} \mu_n} \times \frac{\min\limits_{\forall D_n \in \mathbb{D}} \sigma_n}{\sigma_i}, \qquad (1)\\
\rho_i &= \frac{|\{\pi_j | S_{i,j} > 0 \quad \forall j\}|}{|\{\pi_j \in \bar{\mathbb{P}}\}|}\\
\mu_i &= \frac{\sum_j S_{i,j}}{|\{\pi_j \in \bar{\mathbb{P}}\}|}\\
\sigma_i &= \sqrt{\frac{1}{|\{\pi_j \in \bar{\mathbb{P}}\}|} \sum_j (S_{i,j} - \mu_i)^2}.
\end{aligned}
$$

Here, χ_i measures the credibility of a database D_i, which will be used as "voting power" in the second round voting. Basically, we have the following considerations in defining χ_i.

- the correct rate ρ_i, *i.e.*, the ratio of IP addresses that D_i is correct on. A credible (good) database should be correct (according to the first round voting) on as many as possible IP addresses.
- the average score μ_i, *i.e.*, the average score D_i gets from addresses in $\bar{\mathbb{P}}$. This metric considers the degree of our confidence on each "correct answer". A credible (good) database should agree with the answers that we are more confident in.
- the standard deviation σ_i, *i.e.*, the standard deviation of the score vector D_i gets from addresses in $\bar{\mathbb{P}}$. A database whose performance (scores) varies a lot across different IP addresses should be punished.

Besides the above the considerations, as shown in Equation 1, we choose to normalize ρ_i, μ_i and σ_i by the best value of the corresponding metric among all databases. It helps us balance the influence of the above three considerations.

In the second round, each database D_i is given a voting power of χ_i and we conduct a weighted voting for each IP address in $\mathbb{P}$ to determine its location. The metrics of six databases are listed in Table 1. We can see that three databases focusing on IP addresses in China, *i.e.*, IPIP, Chunzhen, and IP138, are better than others and IP138 performs best among the six databases.

	ρ_i	μ_i	σ_i	χ_i
MaxMind	50.2%	0.158	0.790	0.068
IP2location	78.5%	0.549	0.589	0.498
DB-IP	70.4%	0.424	0.684	0.297
IPIP	87.5%	0.656	0.467	0.837
Chunzhen	88.8%	0.651	0.474	0.831
IP138	90.3%	0.686	0.422	1

Table 1: The metrics to calculate voting power of six databases.

After weighted votings, we get a new geolocation database $\mathbb{G}_1$ which includes mappings from IP addresses in $\mathbb{P}$ to their corresponding estimated locations. Please note in $\mathbb{G}_1$ there are still some addresses whose estimated locations are "China" or oversea locations.

5.1.2. Necessity of two-round voting and χ_i

If we ping an IP address π_j from a landmark m_i and get a RTT of $t_{i,j}$, the distance between π_j and m_i must be less than $\frac{t_{i,j}}{2} \times \frac{2}{3}c$ (c is the speed of light). It is one of the well-known ways to do geolocation. In this work, we exploit this method to evaluate the accuracy of geolocation databases and demonstrate the necessity of two-round voting and χ_i in improving accuracy.

The key issue here is that it requires a lot of landmarks. Chinaz [52] provides the measurement result of RTTs for a given taget IP address from a lot of landmarks. We then depend on its 124 landmarks in 24 different provinces of China to conduct our measurement. Due to the limitation of this service, the measurement is very time-consuming, therefore we randomly choose $200K$ IP addresses from the set $\bar{\mathbb{P}}$ for our evaluation.

We say one IP address π_j is mistaken in a database if the distance between its location given by the database and a landmark m_i is larger than $\dfrac{t_{i,j}}{3} \times c$. We list the number of mistaken IP addresses in Table 2. The first row is for the result of the first-round voting. The second to fourth row are for the results of two-round votings in which only one of ρ_i, μ_i and σ_i is used to determine the voting power of six original databases. The last row is for $\mathbb{G}_1$, which is derived from our two-round voting algorithm.

	number of mistaken	rate of mistaken
first-round voting	10763	5.38%
two-round using ρ_i	9313	4.66%
two-round using μ_i	8410	4.21%
two-round using σ_i	8349	4.17%
two-round using χ_i	3905	1.95%

Table 2: The necessity of two-round voting and χ_i in improving accuracy.

Although the mistaken rate listed in the table is just a lower-bound, we believe it can demonstrate that the second-round weighted voting using χ_i as voting power improves the accuracy of geolocation.

We also try to correct those mistaken IP addresses in $\mathbb{G}_1$ as follows. If $t_{i,j} \leq 6ms$, we regard it as this landmark-based method is voting for the province of m_i on π_j. Otherwise, we regard it as the method abstains in the voting on π_j's location. Then we go back to the six original databases and find answers that do not violate the measurement result $t_{i,j}$. These answers together with the vote from the landmark-method would vote again and the winner would be regarded as the location of π_j. In this way, we revise the locations of 3292 addresses.

5.1.3. Improving geolocation of backbone IPs

We note that some addresses are labelled as "backbone IPs" of one ISP in Chunzhen database. These IP addresses are used by backbone routers and geolocating routers is regarded to be more difficult than geolocating addresses used by end users, *i.e.*, access addresses [36]. We try to improve the accuracy of geolocating these backbone IPs using traceroute results as follows.

We assume that two nodes with a RTT less than 3ms are likely to be in the same province, wherein a RTT of 3ms corresponds to a geographic distance of 300Km under 2/3 speed of light, but corresponds to a much smaller distance in reality considering inflation. Please note that the routing in China roughly has a hierarchical structure. If two consecutive nodes on a path are in different provinces, they tend to be the provincial nodes of two different provinces, whose distance cannot be too small. When the RTT between two nodes is less than 3ms, if one node has been geolocated successfully and the other node has not, we can try to propagate the location of the known node to the unknown node and see if it can work well for most traces.

Our algorithm works as follows. We mark these backbone addresses as "location unknown", and then traverse traces collected by traceroute. For each trace $\pi = (\pi_1, ..., \pi_i, \pi_{i+1}, ..., \pi_n)$, we conduct a forward propagation and a backward propagation. The forward propagation starts from π_1. If the latency of the link (π_i, π_{i+1}) (for $i \in [1, n-1]$) is less than $3ms$ and π_{i+1} is "location unknown" while π_i's location is known, we then propagate π_i's province-level location to π_{i+1} and remove the "location unknown" mark of π_{i+1} temporarily for the later links during this forward propagation, which means π_{i+1} can further propagate the location to π_{i+2} if the above conditions are satisfied. Please note the mark removal is temporal, and it is still "location unknown" during the backward propagation of this trace and all propagations of other traces. The backward propagation is similar, except the propagation direction is reverse.

If a province-level location, say l, is propagated to π_i during one propagation of one trace, we say this trace is voting for l on π_i's location. Each trace can vote at most two times on the location of one IP address (one for the forward propagation and one for backward), and the two votes can be different. After we collect votes of all traces, if a province receives more than 60% of votes on one IP address's location, we would map the IP address to the province. If no one gets more than 60% of votes, we keep its location unchanged.

There are about 9574 backbone IPs in $\mathbb{G}_1$. Using the above method, we revise the locations for 4372 IP addresses. Let us denote the database we get after revisions based on measurement results of ping from multi-landmarks and traceroute as $\mathbb{G}_2$, which will be used in the analysis in later subsections.

We make $\mathbb{G}_2$ publicly accessible at [38].

5.2. Efficiency of infrastructure and routing

Roughly speaking, ISPs in China deploy their networks in a hierarchical structure. The first level consists of several or about a dozen of *regional cores*. Each regional core is a POP that is primarily responsible for a region, *e.g.*, southwest region or northwest region. The second level includes *provincial nodes*. Each provincial node is a POP that is responsible for the packets from or to the corresponding province. A regional core is always the provincial node of the province in which it is located. One provincial node can be connected directly to some nearby provincial nodes and one or multiple regional cores. The provincial nodes (including regional cores) connect with each other and form the backbone network of the ISP. For one data packet, if there is no direct connection between the source province and destination province, the packet would be sent to one regional core of the source province, and the regional core would forward the packet to its destination provincial node (if connected directly) or its destination regional core (if cross-region is necessary). We illustrate this topology and routing structure in Figure 14. For brevity, we only present partial details about one region and some province nodes as examples.

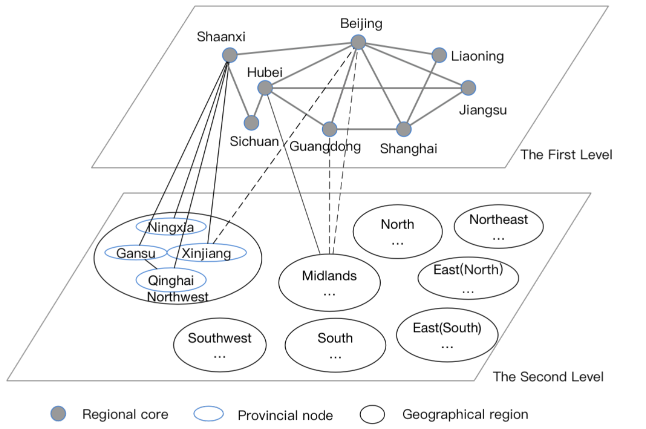

Figure 14: The conceptual topology and routing structure of major ISPs in China.

There are about four major ISPs in China, *i.e.*, China Telecom, China Unicom, China Mobile and CERNET. For one inter-domain packet, whose source and destination are in networks of different ISPs, it must be exchanged across ISPs in national Internet Exchange Points (IXPs). Currently, there are about 13 national IXPs in China, among which ten IXPs are deployed after 2014. Naturally, ISPs tend to deploy their regional cores in these 13 cities because it facilites inter-domain communications. Furthermore, international connections are available only in three supernodes, *i.e.*, Beijing, Shanghai and Guangzhou. They are also the three oldest IXPs in China.

5.2.1. Circuitousness of traces

Limited by the infrastructure and routing policies, it is almost always true that the routing path between two points are longer than the geodesic distance between them. Conventionally, the ratio of routing path length to geodesic distance is defined as *circuitousness ratio* [26]. Mathematically, for a path $\pi = (\pi_1, \pi_2, ..., \pi_n)$, its circuitousness ratio $C(\pi)$ is

$$C(\pi) = \frac{\sum_{\forall i \in [1,n-1]} d(\pi_i, \pi_{i+1})}{d(\pi_1, \pi_n)}, \tag{2}$$

wherein n is the number of IP addresses on π, π_i is the ith address on this path, and $d(\pi_i, \pi_j)$ is the geodesic distance between π_i and π_j.

Due to the hierarchical structure shown in Figure 14, a path of data packets can be viewed as (source endpoint $\rightarrow$ source provincial node $\rightarrow$ a series of provincial nodes $\rightarrow$ destination provincial node $\rightarrow$ destination endpoint). Furthermore, all IXPs are also at these provincial nodes, which means inter-domain paths are in the same format. For a path, the following routers should be in provincial nodes (POPs): the routers in the intermediate provinces, the last hop router of the source province, and the first hop router of the destination province. When we calculate the circuitousness of a path, we remove the segments within the source province and the destination province, and also remove the corresponding latency when necessary. In other words, the path we analyze is (source provincial node $\rightarrow$ a series of provincial nodes $\rightarrow$ destination provincial node). Since we know the geographical location of each provincial node, and we have also known the province of each IP address from the geolocation database, these routers can be mapped to the geographical locations of their individual provincial nodes. Then we can compute $d(\pi_i, \pi_j)$ and then the circuitousness ratio $C(\pi)$. We can see that $C(\pi)$ reflects the circuitousness of backbone networks.

The routing policies of ISPs play an important role in determining the routing paths between sources and destinations. In order to be specific, here we only concentrate on four major ISPs and study the circuitousness of traces that only involve these four ISPs. These traces account for about 70% of all traces in our dataset. We classify these traces into categories according to their source domains and destination domains. The number and proportion of traces in each category are listed in Table 3. For each category, we plot the distribution of circuitousness ratios of traces in Figure 15.

Figure 15(a) shows the circuitousness ratios of intra-domain traces, and Figure 15(b) shows the circuitousness ratios of inter-domain traces. Clearly, *the inter-domain traces are more likely to be circuitous than intra-domain traces.* Then we further split each of the inter-domain traces into multiple intra-domain segments and inter-domain segments, wherein one intra-domain segment is a sequence

Inter-domain			Intra-domain		
Category	Number	Proportion	Category	Number	Proportion
CERNER-Mobile	7431	0.217%	CERNET	85973	2.505%
CERNET-Unicom	56556	1.648%	Telecom	1643293	47.890%
CERNET-Telecom	301780	8.795%	Mobile	23946	0.698%
Telecom-Mobile	7756	0.226%	Unicom	223633	6.517%
Telecom-Unicom	70312	2.049%			
Unicom-Mobile	100	0.003%			

Table 3: The number and proportion of traces in each category.

of continuous links in one ISP and one inter-domain segment is a link connecting two ISPs. Since we focus on the traces that involve only four major ISPs and these ISPs are peering with each other directly, one inter-domain trace is always split into two intra-domain segments and one inter-domain segment. We then plot the circuitousness ratios of these intra-domain segments of inter-domain traces in Figure 15(c). We find that Figure 15(c) is slightly worse than Figure 15(a) but much better than Figure 15(b), which suggests that *the circuitousness of inter-domain traces is mainly caused by the selection of exchange points.*

5.2.2. Policy of selecting IXPs

In this subsection, we conduct an analysis on the policy of selecting IXPs for inter-domain traces. Here, we focus on the inter-domain traces whose source and destination provinces are different, namely *inter-domain cross-province* traces.

Before we analyze how two ISPs select IXPs to exchange traffic, we need to know the IXPs at which two ISPs are connected with each other. We intentionally select more than one hundred of source nodes [53] and about 10K destination websites (200 sites for each ISP in each province with IXP) and conduct traceroutes between them to cover interconnections between ISPs as many as possible. We try our best to spread these source nodes and destination sites across different provinces. Note that we cannot run Paris traceroute on these source nodes, so we discard all traceroute results with different IP addresses on the same hop. The result is shown in Table 4.

From the result, we can find that Telecom-Mobile and Telecom-Unicom have at least tens of traces exchanging traffic at each of 13 IXPs, which indicates that these three major ISPs are connected with each other in all IXPs. But for CERNET, which is relatively smaller, we cannot determine whether it has been connected to some ISPs at Shaanxi (be in operation in 2014), Zhejiang (2017), Fujian (2017), Sichuan (2014) and Chongqing (2014). Please note that the number of traces is zero does not necessarily indicate there is no connection. It just means we have no evidence to safely determine there is a connection.

In summary, any two ISPs have multiple IXPs available for traffic exchange. There can be three cases in terms of selecting IXPs: 1) the IXP at the source province or the IXP at the regional core of the source province (if the

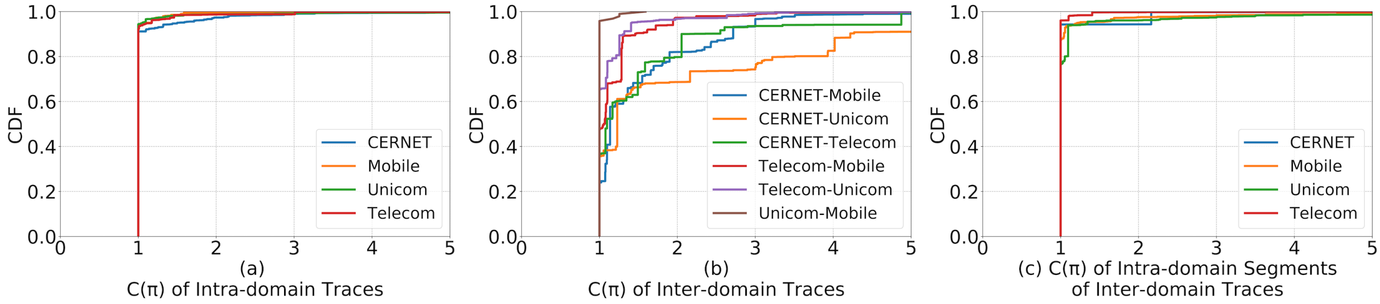

Figure 15: Circuitousness ratios of traces in each category.

	Beijing	Shanghai	Guangdong	Liaoning	Shaanxi	Henan	Jiangsu	Zhejiang	Hubei	Fujian	Guizhou	Sichuan	Chongqing
Telecom-Mobile	326	7969	60	84	1913	54	3501	1499	716	94	28	149	1440
Telecom-CERNET	44167	1052	8232	3192	4232	389	1446	99	955	0	535	0	136
Telecom-Unicom	26057	11432	4084	210	1711	252	4151	663	1745	1124	1021	1534	1689
CERNET-Unicom	42371	21159	102	198	177	16033	892	0	148	436	98	3111	0
CERNET-Mobile	205	18466	211	765	0	38	3349	0	70	0	462	1107	0

Table 4: The traces that exchange packets between two ISPs at each IXP.

source province does not have an IXP) is selected, which indicates a hot-potato routing policy is used. We say the IXP is serving the trace as a *source IXP*; 2) the IXP at the destination province or the IXP at the regional core of the destination province (if the destination province does not have an IXP) is selected, which indicates a cold-potato routing policy is used. We say the IXP is serving the trace as a *destination IXP*; 3) an IXP at other provinces is selected. We say the IXP is serving the trace as a *middle IXP*.

source ISP	# of traces	hot-potato	cold-potato
CERNET	342388	68189(19.9%)	78525(22.9%)
Telecom	72959	14603(20.0%)	46504(63.7%)

Table 5: The routing policy of traces initiated by CERNET and Telecom.

Table 5 presents the number of traces using hot-potato routing and cold-potato routing. Due to the limitation of vantage points, we can only get the result from traces with CERNET or Telecom as their sources. We can see that *Telecom is more likely to use cold-potato routing than CERNET* and *a lot of traces select middle IXPs to complete their cross-domain packet exchanges.*

In terms of traces associated with websites hosted by cloud providers, our data analysis shows that 94% of them exchange packets from ISPs directly to the cloud providers at the closest exit points to destinations (cold potato routing). In other words, at least 94% of these traces are with no inter-domain circuitousness. It is better than the inter-domain circuitousness of traces without cloud providers (see Figure 15). This result is consistent with our intuition that cloud providers generally pay good money to get good networking service from ISPs.

We further check the usage of the 13 IXPs in China. We focus on destination IXPs and middle IXPs to avoid the influence of limited source points. The result is presented in Table 6. There are in total 331458 inter-domain cross-province traces using destination IXPs or middle IXPs. Among them, 72.6% *of traces depend on three oldest IXPs (Beijing, Shanghai and Guangzhou) to complete the data packet exchange.* Except IXPs in Henan, Shannxi and Liaoning, newly deployed IXPs seldom serve as middle IXPs, which indicates *they only accept data packets destined to their own region.*

IXP	M+D	D	M	IXP	M+D	D	M
Shanghai	164802	21984	142818	Fujian	3360	3360	0
Guangdong	64310	18144	46166	Zhejiang	1653	1632	21
Henan	37449	27962	9487	Hubei	1327	1325	2
Shaanxi	29697	27540	2157	Guizhou	493	493	0
Beijing	11594	7886	3708	Sichuan	455	454	1
Liaoning	10099	8924	1175	Chongqing	247	222	25
Jiangsu	5972	5102	870	-	-	-	-

Table 6: The usage of 13 IXPs in China. (The column "D" shows the number of traces in which the IXP is serving as a destination IXP, and "M" shows the number of traces in which the IXP is serving as a middle IXP.)

If an inter-domain trace selects its source or destination IXP to exchange data packets, there tends to be no inter-domain circuitousness at the province level. Therefore we skip these traces. We focus on the 206430 traces using middle IXPs and investigate to what extent the selection of middle IXPs affects circuitousness ratios. We conjecture that a middle IXP is selected for a trace because the source and destination ISP of the trace cannot agree on either hot-potato or cold-potato routing. In order to evaluate the optimality of one middle IXP for a trace, we sort all possible middle IXPs according to the circuitousness ratios if the corresponding middle IXPs is selected. We find that only 9.8% of the traces are using the optimal middle IXP and 48.5% of the traces are using one of its top three middle IXPs. Among the traces that use non-optimal (non-top-three) IXPs, 97.3% (98.5%) of the traces are using the three oldest IXPs.

In summary, we conclude that *the inter-domain routing in China relies too heavily on the three IXPs at Beijing, Shanghai and Guangzhou, and the newly deployed IXPs are underused significantly.* Such an improper selection of IXPs would increase communication latency undesirably. A carefully designed cost-sharing scheme might be helpful for the four major ISPs to reach agreements on more efficient inter-domain routing decisions.

5.3. Identifying and analyzing congested links

Queuing delay caused by congestions is the other potential source of extra latency. In this subsection, we would like to identify links with high level of congestions and find out where these links are located logically (network locations) and physically (geographical locations).

Our idea is as follows. We first find out the links that are measured to be congested in a large number of traces and consider these links as candidates for further analysis. We then conduct long-time time-sequence latency probings for each candidate link, and determine whether the link is congested for a large proportion of time. The links that are congested for at least 5% of time each day (1.2 hours) are regarded as "with high level of congestions" and we then analyze their locations.

We estimate the round-trip queuing delay of a path π as follows,

$$\mathcal{T}^Q(\pi) = \mathcal{T}(\pi) - \frac{2 \sum_{\forall i \in [1,n-1]} d(\pi_i, \pi_{i+1})}{2/3 \times c}. \quad (3)$$

Basically, as we stated at the beginning of this section, we are assuming that transmission delay and processing delay are ignorable, therefore round-trip queuing delay, denoted by $\mathcal{T}^Q(\pi)$, can be calculated as the total RTT, denoted by $\mathcal{T}(\pi)$, minus propagation delay which is the round-trip geographical distance divided by $\frac{2}{3}$ of the speed of light. We plot $\mathcal{T}^Q(\pi)$ in Figure 16. We can see that a lot of traces are suffering from long queuing delay.

As the first step, we split each path into hops and plot the queueing delay of each hop in Figure 17. Here a "hop" refers to an appearance of a physical link in the routing path of a trace during our measurement. We use "link" to refer to a physical link which can be used by multiple traces.

In Figure 17, hops are classified into groups, wherein "intra-cloud" means the two ends of the hop are located in the same cloud, "cloud-ISP" means the hop is connecting one cloud with its ISP, "intra-ISP" means two ends of

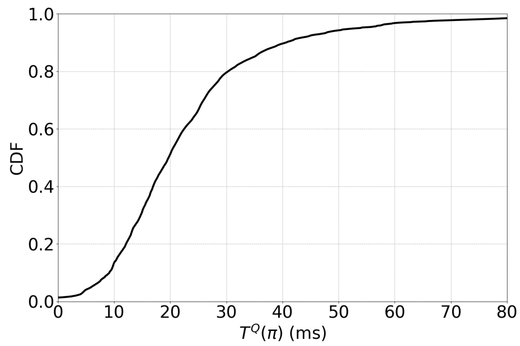

Figure 16: The distribution of queuing delay of all traces.

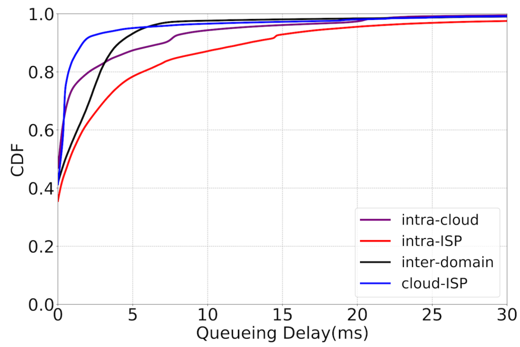

Figure 17: The distribution of queuing delays of all hops.

the hop are located in the same ISP, and "inter-domain" means the hop is connecting two different ISPs. We can see that statistically cloud-ISP hops perform better than inter-domain hops, and intra-cloud hops are better than intra-ISP hops, whose group is worst among four groups.

Let us regard a queuing delay of longer than 10ms as a signal of possible congestion on one hop. We retrieve all congested intra-ISP hops, *i.e.*, the hops whose queueing delays are longer than 10ms, and map these hops to their corresponding links. In this way, we get 95937 links. Each of these link produces a queuing delay of at least 10ms in at least one trace, but the long queuing delay can be an accidental event on the link. In other words, these links can have different levels of congestions. It is infeasible for us to investigate all these links in detail and we would like to focus on the links with high level of congestions. We then rank these links according to the number of appearances in the set of congested hops. The result is plotted in Figure 18.

We select top 3245 links to further examine the nature of their congestions by time-sequence latency probings. These 3245 links are responsible for 80% of the congested hops, and they also account for a majority of paths with long queuing delay, as shown in Table 7. Let us take the second row as an example to explain the ratios we present in Table 7. Among all traces collected by us, 2435693 traces experience a queuing delay of longer than 15ms. The selected links appear in 85.1% of these congested traces, and the selected links contribute a queu-

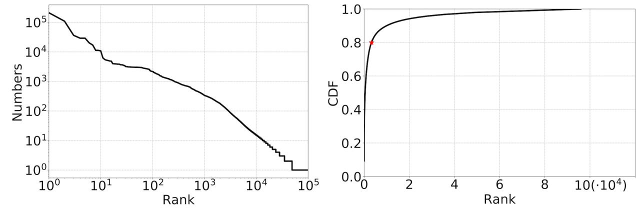

Figure 18: The links and the numbers of their appearances with queuing delay longer than $10ms$.

ing delay of longer than 10ms in 60.6% of these congested traces. To some extent, Table 7 shows that the selected links are representative. Now we conduct time-sequence latency probings for these selected links to study their congestion natures.

	number of traces	selected links appear	selected links appear and congest
$\forall\pi$	3431421	2647985(77.2%)	1624675(47.3%)
$\mathcal{T}^Q(\pi) > 15ms$	2435693	2072511(85.1%)	1476902(60.6%)
$\mathcal{T}^Q(\pi) > 20ms$	1831488	1566456(85.5%)	1163128(63.5%)

Table 7: The ratio of traces affected by the selected links.

TSLP (time-sequence latency probing) is a technical method proposed in [27] to detect the presence of congestion on a link by measuring a sequence of RTT values to the near and far ends (routers) of the link. It is considered as an evidence of congestion that the RTT to the far end increases while the RTT to the near end does not increase.

For each selected link, we try to find five source-destination pairs whose routing paths traverse the link. Here, the source should be one of our five vantage points and the destination is one website. These pairs can be found easily from the set of traces. Then we take the advantage of *scamper* [44] to traceroute from the source point to the destination website of each pair every 10 minutes for one week (from Mar 8, 2019 to Mar 15, 2019).

Figure 19 plots the measurement results of four links as examples. The subplot Figure 19(a) and Figure 19(c) exhibit a strong diurnal pattern, and the RTT to the far end has a flatter peak in Figure 19(c) than Figure 19(a), which indicates that the queue of the link (c) is close to full for a longer time. In the subplot Figure 19(b) and Figure 19(d), the RTT to the near and far ends keep roughly unchanged, but $RTT_{far} - RTT_{near}$ is about $40ms$, which is considerably large. There are two possibilities for the phenomenon in Figure 19(b) and Figure 19(d), *i.e.*, either the load is always more than the capacity that the link can handle and therefore the queue is always full, or the link is with a long propagation delay due to its outmoded technology. Either of these two possibilities indicate that a upgrade of the link is recommended.

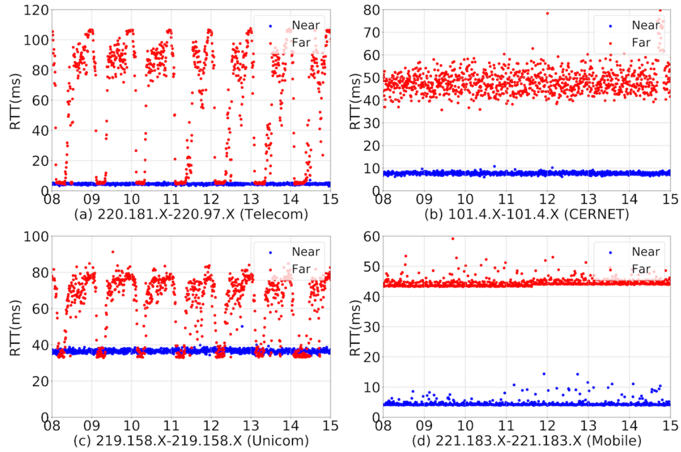

Figure 19: Measurement results of four links as examples.

We characterize the level of congestion of one link l using the $95th$ percentile queuing delay of the link, denoted by $\mathcal{Q}_l^{95}$. We define this metric because the queuing delay of one link is worse than its $95th$ percentile value for 5% of the whole time period, *i.e.*, 1.2 hours per day on the average.

The statistics of $95th$ percentiles of the links are plotted in Figure 20. Experientially, we classify these links into three groups based on their levels of congestion as follows: 1) *slightly congested links* with $\mathcal{Q}_l^{95} < 10ms$, which means the link's queuing delay is less than a threshold smaller than $10ms$ for 95% of the time; 2) *mildly congested links* with $\mathcal{Q}_l^{95} \in [10ms, 20ms]$; 3) *severely congested links* with $\mathcal{Q}_l^{95} > 20ms$. Among the candidate links, we find 362 slightly congested links, 1634 severely congested links and 1240 severely congested links[2]. It also indicates that our method to select candidate links is reasonable.

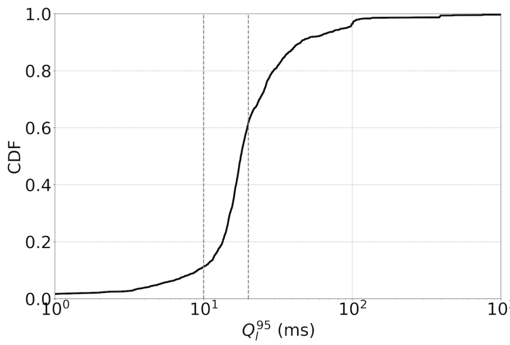

Figure 20: The distribution of the $95th$ percentile queuing delay of each link.

As we introduced at the beginning of Section 5.2, the network of one ISP is hierarchical, wherein each province has its own sub-network and is connected directly or in-

[2]We cannot find usable source-destination pairs for 9 links when TSLP measurements are conducted, thus these links are skipped in later analysis.

directly to the backbone network which includes regional cores and some province nodes. We find most of mildly and severely congested links are connecting provincial subnetworks to the backbone network. We plot our analysis result on it in Figure 21.

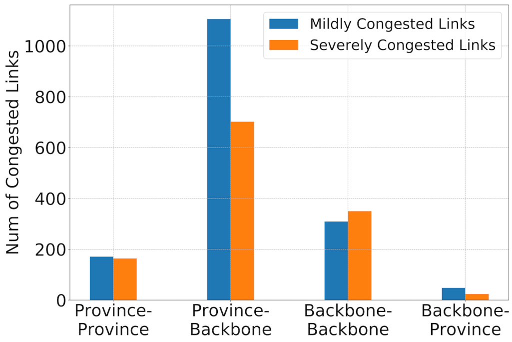

Figure 21: The network locations of congested links.

We further examine their geographical locations of congested links. Here we only focus on 1240 severely congested links. We map the ends of each link to its source province and destination province, and then construct a congestion graph using these links. Each node in the graph is a province and the degree of one node is the number of severely congested links that involve the province. We show the degrees of provinces in the congestion graph in Figure 22. The sizes of the circles for nodes are positively correlated to the degrees of nodes. We see that most of the high-degree nodes are the provinces that have IXPs deployed and three supernodes are with highest degrees. The provinces that have IXPs are usually regional cores of ISPs. Our finding suggests that *most of severely congested links are the links connecting two regional cores and the backbone networks of Chinese ISPs require upgrades or better traffic engineerings.*

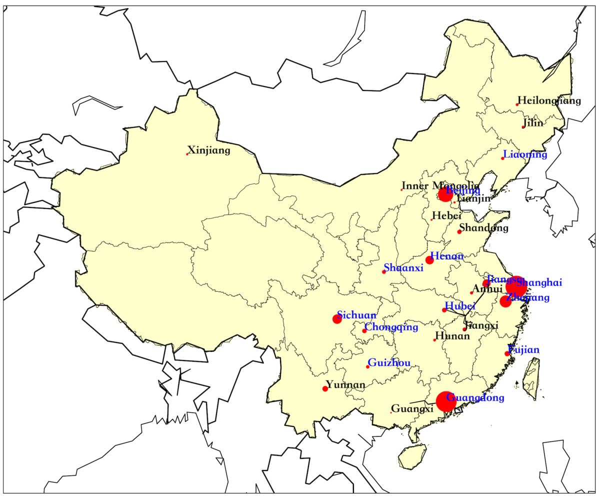

Figure 22: The degrees of nodes in the congestion graph.

6. Conclusion

In this article, we conduct a measurement study on the Internet latency in China. We observe that DNS resolution latency is the most significant contributor to the overlay latency of visiting websites and the DNS latency is more than 1 second for about 20.3% of websites. It indicates that the DNS infrastructure in China requires improvements. The comparison between RTT achieved under current Internet infrastructure and f-latency shows that the selected path between two ends in China is circuitous or congested with a high probability. It indicates that the Internet routing infrastructure also needs improvements.

For the DNS infrastructure, we conduct controlled experiments to characterize DNS Latency of authoritative name servers of different levels. We find that caching SLDs is essential to achieve an acceptable DNS resolution latency. We also report the significant influence of "CNAME" and delegated name servers, especially when they are with different TLDs from the original domain name. For the routing service provided by Internet infrastructure in China, we conduct more measurements to find the reasons for circuitousness and locate congested links. We find that most of circuitous paths are caused by non-optimal selection of IXPs. Although China deploys more IXPs in recent years, they are not used efficiently. Furthermore, most of congested links are those links connecting two regional cores, and they need upgrades or better traffic engineering decisions.

In order to complete our analysis on the routing infrastructure, we geolocate the IP addresses that appear in our dataset. Particularly, we improve the geolocation accuracy by designing a two-round voting algorithm to fuse six unreliable databases and exploiting multiple techniques to further calibrate the fusion results. We make the province-level geolocation mappings publicly accessible.

Acknowledgement

We thank Juncai Liu and Jiali Han for their help in collecting our measurement results. This work was supported in part by the National Key Research and Development Program of China under Grant 2017YFB0503703 and in part by the National Natural Science Foundation of China under 61602055.

[1] Akamai online retail performance report, `https://www.akamai.com/uk/en/about/news/press/2017-press/akamai-releases-spring-2017-state-of-online-retail-performance-report.jsp`, accessed June, 2019 (2018).

[2] A. Singla, B. Chandrasekaran, P. Godfrey, B. Maggs, The Internet at the speed of light, in: Proceedings of the 13th ACM Workshop on Hot Topics in Networks, ACM, 2014, p. 1.

[3] M. Ford, Workshop report: Reducing Internet latency, 2013, ACM SIGCOMM Computer Communication Review 44 (2014) 80–86. doi:10.1145/2602204.2602218.

[4] I. N. Bozkurt, A. Aguirre, B. Chandrasekaran, P. B. Godfrey, G. Laughlin, B. Maggs, A. Singla, Why is the Internet so slow, in: Passive and Active Measurement, Springer International Publishing, Cham, 2017, pp. 173–187.

[5] Z. S. Bischof, J. P. Rula, F. E. Bustamante, In and out of Cuba: Characterizing Cuba's connectivity, in: Proceedings of the 2015 Internet Measurement Conference, ACM, 2015, pp. 487–493.

[6] A. Formoso, J. Chavula, A. Phokeer, A. Sathiaseelan, G. Tyson, Deep diving into Africa's inter-country latencies, in: IEEE INFOCOM 2018-IEEE Conference on Computer Communications, IEEE, 2018, pp. 2231–2239.

[7] J. Liang, J. Jiang, H. Duan, K. Li, J. Wu, Measuring query latency of top level DNS servers, in: Passive and Active Measurement, Springer Berlin Heidelberg, Berlin, Heidelberg, 2013, pp. 145–154.

[8] E. Casalicchio, M. Caselli, A. Coletta, Measuring the global domain name system, IEEE Network 27 (1) (2013) 25–31. doi:10.1109/MNET.2013.6423188.

[9] B. Huffaker, M. Fomenkov, K. Claffy, Geocompare: a comparison of public and commercial geolocation databases - Technical Report , Tech. rep., Cooperative Association for Internet Data Analysis (CAIDA) (May 2011).

[10] Y. Tian, R. Dey, Y. Liu, K. W. Ross, Topology mapping and geolocating for China's Internet, IEEE Transactions on Parallel and Distributed Systems 24 (9) (2013) 1908–1917. doi:10.1109/TPDS.2012.271.

[11] M. A. Habib, M. Abrams, Analysis of sources of latency in downloading web pages., in: WebNet, Vol. 227, Citeseer, 2000, p. 232.

[12] J. Charzinski, Web performance in practice – why we are waiting, International Journal of Electronics and Communications 55 (1) (2001) 37–45. doi:10.1078/1434-8411-00006.

[13] E. Cohen, H. Kaplan, Prefetching the means for document transfer: a new approach for reducing web latency, Computer Networks 39 (2002) 437–455. doi:10.1016/S1389-1286(02)00184-6.

[14] C. E. Wills, H. Shang, The contribution of DNS lookup costs to web object retrieval, Tech. Rep. WPICS.TR.00.12, Worcester Polytechnic Institute (09 2000).

[15] B. Briscoe, A. Brunstrom, A. Petlund, D. Hayes, D. Ros, I. Tsang, S. Gjessing, G. Fairhurst, C. Griwodz, M. Welzl, Reducing Internet latency: A survey of techniques and their merits, IEEE Communications Surveys Tutorials 18 (3) (2016) 2149–2196. doi:10.1109/COMST.2014.2375213.

[16] N. Zilberman, M. Grosvenor, D. A. Popescu, N. Manihatty-Bojan, G. Antichi, M. Wójcik, A. W. Moore, Where has my time gone?, in: Passive and Active Measurement, Springer International Publishing, Cham, 2017, pp. 201–214.

[17] Y. Zaki, J. Chen, T. Potsch, T. Ahmad, L. Subramanian, Dissecting web latency in Ghana, in: Proceedings of the 2014 Conference on Internet Measurement Conference, IMC '14, ACM, New York, NY, USA, 2014, pp. 241–248. doi:10.1145/2663716.2663748.
URL `http://doi.acm.org/10.1145/2663716.2663748`

[18] M. Zheleva, P. Schmitt, M. Vigil, E. Belding, The increased bandwidth fallacy: Performance and usage in rural Zambia, in: Proceedings of the 4th Annual Symposium on Computing for Development, ACM DEV-4 '13, ACM, New York, NY, USA, 2013, pp. 2:1–2:10. doi:10.1145/2537052.2537060.
URL `http://doi.acm.org/10.1145/2537052.2537060`

[19] J. Jung, E. Sit, H. Balakrishnan, R. Morris, DNS performance and the effectiveness of caching, IEEE/ACM Transactions on networking 10 (5) (2002) 589–603.

[20] J. Jung, A. W. Berger, H. Balakrishnan, Modeling TTL-based Internet caches, in: IEEE INFOCOM 2003. Twenty-second Annual Joint Conference of the IEEE Computer and Communications Societies (IEEE Cat. No. 03CH37428), Vol. 1, IEEE, 2003, pp. 417–426.

[21] H. Shang, C. E. Wills, Piggybacking related domain names to improve DNS performance, Computer Networks 50 (11) (2006) 1733 – 1748. doi:https://doi.org/10.1016/j.comnet.2005.06.016.

[22] S. Hao, H. Wang, A. Stavrou, E. Smirni, On the DNS deployment of modern web services, in: 2015 IEEE 23rd International Conference on Network Protocols (ICNP), 2015, pp. 100–110. doi:10.1109/ICNP.2015.37.

[23] S. Hao, H. Wang, Exploring domain name based features on the effectiveness of DNS caching, ACM SIGCOMM Computer Communication Review 47 (1) (2017) 36–42.

[24] Q. Gao, F. Wang, L. Gao, Quantifying AS path inflation by routing policies, International Journal of Future Generation Communication and Networking 9 (1) (2016) 167–186.

[25] J. H. Wang, C. An, A study on geographic properties of Internet routing, Computer Networks 133 (2018) 183 – 194. doi:https://doi.org/10.1016/j.comnet.2018.01.032.

[26] A. Y. Nur, M. E. Tozal, Geography and routing in the Internet, ACM Transactions on Spatial Algorithms and Systems (TSAS) 4 (4) (2018) 11.

[27] D. Clark, S. Bauer, k. claffy, A. Dhamdhere, B. Huffaker, W. Lehr, M. Luckie, Measurement and Analysis of Internet Interconnection and Congestion, in: Telecommunications Policy Research Conference (TPRC), 2014.

[28] M. Luckie, A. Dhamdhere, D. Clark, B. Huffaker, et al., Challenges in inferring Internet interdomain congestion, in: Proceedings of the 2014 Conference on Internet Measurement Conference, ACM, 2014, pp. 15–22.

[29] A. Dhamdhere, D. D. Clark, A. Gamero-Garrido, M. Luckie, R. K. Mok, G. Akiwate, K. Gogia, V. Bajpai, A. C. Snoeren, K. Claffy, Inferring persistent interdomain congestion, in: Proceedings of the 2018 Conference of the ACM Special Interest Group on Data Communication, ACM, 2018, pp. 1–15.

[30] R. Fanou, F. Valera, A. Dhamdhere, Investigating the causes of congestion on the African IXP substrate, in: Proceedings of the 2017 Internet Measurement Conference, IMC '17, ACM, New York, NY, USA, 2017, pp. 57–63. doi:10.1145/3131365.3131394.
URL `http://doi.acm.org/10.1145/3131365.3131394`

[31] R. Barnes, J. Winterbottom, M. Dawson, Internet geolocation and location-based services, IEEE Communications Magazine 49 (4) (2011) 102–108.

[32] S. Laki, P. Mátray, P. Hága, T. Sebők, I. Csabai, G. Vattay, Spotter: A model based active geolocation service, in: Proceedings IEEE INFOCOM, 2011, pp. 3173–3181. doi:10.1109/INFCOM.2011.5935165.

[33] Y. Wang, D. Burgener, M. Flores, A. Kuzmanovic, C. Huang, Towards street-level client-independent IP geolocation, in: NSDI, Vol. 11, 2011, pp. 27–27.

[34] Y. Lee, H. Park, Y. Lee, IP geolocation with a crowd-sourcing broadband performance tool, SIGCOMM Comput. Commun. Rev. 46 (1) (2016) 12–20. doi:10.1145/2875951.2875954.
URL `http://doi.acm.org/10.1145/2875951.2875954`

[35] I. Poese, S. Uhlig, M. A. Kaafar, B. Donnet, B. Gueye, IP geolocation databases: Unreliable?, SIGCOMM Comput. Commun. Rev. 41 (2) (2011) 53–56. doi:10.1145/1971162.1971171.

[36] M. Gharaibeh, A. Shah, B. Huffaker, H. Zhang, R. Ensafi, C. Papadopoulos, A look at router geolocation in public and commercial databases, in: Proceedings of the 2017 Internet Measurement Conference, ACM, 2017, pp. 463–469.

[37] H. Li, P. Zhang, Z. Wang, F. Du, Y. Kuang, Y. An, Changing IP geolocation from arbitrary database query towards multi-databases fusion, in: 2017 IEEE Symposium on Computers and Communications (ISCC), IEEE, 2017, pp. 1150–1157.

[38] A geolocation database for some IP addresses in China, https://github.com/zhuangshuying18/CN2019/, online July 3, 2019 (2019).

[39] curl, `https://curl.haxx.se`, accessed September 10, 2018 (2018).

[40] T. Callahan, M. Allman, M. Rabinovich, On modern dns behavior and properties, SIGCOMM Comput. Commun. Rev. 43 (3) (2013) 7–15.

[41] Ip138, `http://www.ip138.com`, accessed December 1, 2018 (2018).

[42] R. Motamedi, R. Rejaie, W. Willinger, A survey of techniques for Internet topology discovery, IEEE Communications Surveys Tutorials 17 (2) (2015) 1044–1065. doi:10.1109/COMST.2014.2376520.

[43] B. Augustin, X. Cuvellier, B. Orgogozo, F. Viger, T. Friedman, M. Latapy, C. Magnien, R. Teixeira, Avoiding tracer-

oute anomalies with paris traceroute, in: Proceedings of the 6th ACM SIGCOMM Conference on Internet Measurement, IMC '06, ACM, New York, NY, USA, 2006, pp. 153–158. doi:10.1145/1177080.1177100.
URL http://doi.acm.org/10.1145/1177080.1177100

[44] M. Luckie, Scamper: a scalable and extensible packet prober for active measurement of the Internet, in: Proceedings of the 10th ACM SIGCOMM conference on Internet measurement, ACM, 2010, pp. 239–245.

[45] E. Katz-Bassett, H. V. Madhyastha, V. K. Adhikari, C. Scott, J. Sherry, P. Van Wesep, T. Anderson, A. Krishnamurthy, Reverse traceroute, in: Proceedings of the 7th USENIX Conference on Networked Systems Design and Implementation, NSDI'10, USENIX Association, Berkeley, CA, USA, 2010, pp. 15–15.
URL http://dl.acm.org/citation.cfm?id=1855711.1855726

[46] S. Laki, P. Mátray, P. Hága, I. Csabai, G. Vattay, A model based approach for improving router geolocation, Comput. Netw. 54 (9) (2010) 1490–1501. doi:10.1016/j.comnet.2009.12.004.
URL http://dx.doi.org/10.1016/j.comnet.2009.12.004

[47] Maxmind, https://www.maxmind.com/en/home, accessed December 1, 2018 (2018).

[48] Ip2location, http://www.ip2location.com, accessed December 1, 2018 (2018).

[49] Db-ip, https://db-ip.com, accessed December 1, 2018 (2018).

[50] Ipip, https://www.ipip.net, accessed December 1, 2018 (2018).

[51] Chunzhen, https://www.cz88.net, accessed December 1, 2018 (2018).

[52] Chianz, http://ping.chinaz.com, accessed December 10, 2018 (2018).

[53] IPIP Taceroute, https://tools.ipip.net/traceroute.php, accessed Oct 10, 2019 (2019).