Identifying Influential Factors of CDN Performance with Large-scale Data Analysis

Dawei Wang[1], Shuzhuang Zhang[2], Yibo Xue[3, 4] and Yingfei Dong[5]

[1] *National Computer Network Emergency Response Technical Team / Coordination Center of China, Beijing, China.*

[2] *Institue of Network Technology, Beijing University of Posts and Telecommunications, Beijing, China*

[3]*Research Inst. of Info. & Tech., Tsinghua University, Beijing, China.*

[4]*Tsinghua National Lab for Information Sci. & Tech., Beijing, China.*

[5]*Department of Electrical Engineering, University of Hawaii, Honolulu, HI 96822, USA.*

{stonetools2008@gmail.com, zhangshuzhuang@bupt.edu.cn, yiboxue@tsinghua.edu.cn, yingfei@hawaii.edu

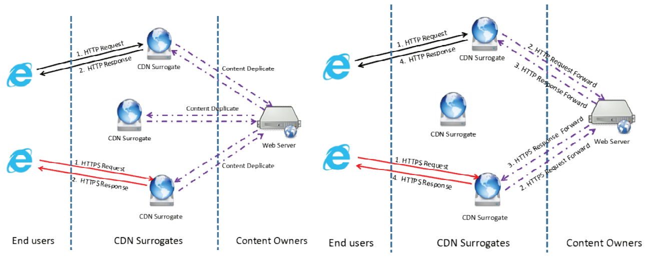
***Abstract*— CDNs manage their own caching or routing overlay networks to provide reliable and efficient content delivery. Currently, CDNs have become one of the most important tools on the Internet, responsible for the majority of today's Internet traffic. The performance of CDNs directly influence the experiences of end users. In this paper, we develop several analyses to figure out the key factors influencing the overall performance of a CDN. The primary results demonstrate that the caching overlay and the routing overlay both have significantly affect CDN performance. Our results also show that the transmission latency between a surrogate and a content owner is a critical factor determining the overall performance of routing overlays. Furthermore, we argue that the surrogate assignment policy of a routing overlay need to seriously take this latency into account. Our analysis results provide a context for the CDN community on preferable surrogate assignment solutions.**

Keywords*— *CDN; Big Data Analysis; Routing Overlays; Caching Overlays; Access Latency.

I. Introduction

A modern content distribution network (CDN) [1] is usually a globally distributed network of proxy servers deployed in multiple data centers. A large CDN service provider, such as Akamai [2], deploys more than 233,000 servers located in 1,600 networks in over 130 countries around the world. According to our observation, there are more than 1.2 million IP addresses registered by Akamai to provide content deliver services for Chinese users. CDNs carry the majority of today's Internet traffic and are expected to carry 71 percent of Internet traffic by 2021 [3].

A CDN manages an overlay network which provides reliable and efficient content delivery [4]. It enables content owner to make their contents widely available. To serve contents to end users, a CDN replicates a content from its owner, i.e., a web site, over the resources at surrogate servers scattered over the global. Then, end user's requests are redirected to the most suitable surrogate based on some criteria such as ISP, geographical position, surrogate server load, and so on. The above process is called a *caching overlay* illustrated as Fig.1 (a). The key component of such an overlay is the cache space on the surrogate servers. Thus, such an overlay is applicable for contents that do not change frequently. The other type of overlay is a *routing overlay*, illustrated in Fig.1(b). Different from the caching overlay, the routing overlay is more like a reverse proxy that forwards an end user's request to a content provider, and then brings the content back to the end user. The routing overlays are often used to deliver dynamic contents that cannot be cached.



(a) Caching overlay (b) Routing overlay

Fig. 1. Two types of CDN overlays.

In this paper, we carry out several data analyses to measure the behaviors of these two types of overlays. The motivation is to figure out the differences between these two types of overlays of a CDN from the respective of end users, based on five large-scale data sets. We further identify the key factors on routing overlays that can affect end user experiences. The analysis results provide a context for the CDN community on preferable surrogate assignment solutions. Our contributions include:

- We developed several analyses to measure behaviors of two types of overlays of a CDN from the end user standpoint. These analyses are carried out based on five large-scale data sets, including three passive measure data sets (*DNS*, *HTTP* and *HTTPS*) and two active measure sets (*whois* information crawled from APNIC and *round-trip-delay* measure data set).
- Based on the comparative analysis, we find that the transmission latency between a surrogate and a content owner is the most critical than other factors, such as the latency between an end user and a surrogate on a routing overlay. We further argue that the cost between surrogates and content owners should be taken into account when assigning surrogates.

The remainder of the paper is structured as follows: Section II presents the related works. We describe our analysis ideas and introduce our large-scale data sets in Section III. In Section IV, we describe the behavior analysis of two types of overlays and present the results in detail. In Section V, we present the key factor analysis of routing overlay performance. We conclude the paper in Section VI with a summary of our contributions.

II. Related Work

The primary goal of a CDN is to deliver contents to end-users with high availability and high performance, which can be affected by several key factors, including surrogate server load, geographical location, performance cost model adopted, and QoS consideration [5]. Most existing solutions revolve around these factors.

A CDN usually deploys its surrogates on the core network close to end users, which may make them far away from content owners' servers [6]. In this case, the replica contents cached in surrogates can be delivered to end users with low latency. The content owners could point their domain name to a CDN's domain name as an alias [7], then an end user request will be redirected to the CDN's edge servers eventually. The key to redirection is how to determine a target surrogate server. The mapping system of a CDN chooses an surrogate based on various policies that considers the server conditions, the locations of end users, and so on, in order to maximize the performance [4]. To meet the scale and quality demands, content providers have employed brokers to spread contents across multiple CDNs[8].

Another key factor that could significantly affect the performance is the cache function. A CDN surrogate employs two levels of caching: in-memory cache called *the hot object cache* and second-level *disk cache*. The performance of in-memory cache is much better than that of the disk cache. The main research issue on this topic focuses on how to manage these two levels of caching so as to improve user experience. In the latest research, Berger proposed and deployed AdaptSize [9], which can improve the object hit ratio by 47%-91%.

Not all the contents on the Internet can be cached for a long time. Dynamic applications (such as dynamic web and banking) generate uncacheable contents based on user interactions in real-time. To deliver those uncacheable contents, researchers have proposed a few solutions. One solution [10] is to replicate the applications instead of contents. This method is vulnerable to the inconsistency of contents, especially real-time contents. To address this issue, CDN vendors proposed a routing overlay architecture, which uses an overlay algorithm to compute a set of overlay paths that each surrogate can use to reach content owners [4] in real-time. An end user request and the corresponding response from its content owner can be forward through the routing overlay network. Besides the surrogate assignment, the selection of overlay paths is another key factor affecting the overall performance.

In summary, a CDN has developed two types of overlays to satisfy various kind of content delivering. As we analyze previously, the caching overlay and the routing overlay both significantly affect the performance of CDNs. Specifically, the critical factors of caching overlays are associated to the path between an end user and a surrogate server, while the routing overlays also include the paths between surrogates and content owners. In this paper, we plan to measure the behaviors of these two types of overlays from the end users' perspective. The purpose is to figure out the difference between the two types of overlays, then further propose a better surrogate assignment solution for routing overlays.

III. Main Ideas and Data Sets

In this section, we first describe the main ideas, and then introduce our five large-scale data sets in this investigation.

A. Main Ideas

As mentioned in the above, a goal of our work is to optimize the surrogate assignment of routing overlays so as to improve use experiences. To achieve this goal, we start from the measurement of surrogate performance from the end user perspective. One critical factor to measure surrogate performance is the access latency. There are several parameters affecting this access latency, include *Round Trip Delay (RTD)*, *surrogate load*, *caching hit ratio* and so on. If we consider a situation that a CDN delivers the same content under the same cache policy over its own surrogates, RTD would be the key to access latency. This is the main reason that a CDN deploys its surrogates on the core network close to end users. For ease of discussion, we assume that the same content distributed by multiple surrogates with the same cache policy; furthermore, the load of these surrogates remain roughly the same due to the balance policy of a CDN. Then we carry out several analyses to measure RTD of the two types of overlays. The main idea is presented in Fig.2.

The first step is to identify the active CDN surrogate IP addresses using *DNS* and *whois* data set. DNS records give us clues to find surrogates, because most end user requests are redirect to the surrogates by the means of DNS redirection. The *whois* data set can help us verify the surrogates IP addresses found via DNS.

In the second step, we carry out several analyses to measure behaviors of two types of overlays using *HTTP* and *HTTPS* data set. A CDN usually provides two types of overlays at the same time, however, the access latency of them are different. For the caching overlay, we measure the latency between an end user and a surrogate since its contents is cached on surrogates. For a routing overlay, there is another important parameter need to be considered: the transmission latency between a surrogate and a content owner. This latency can also

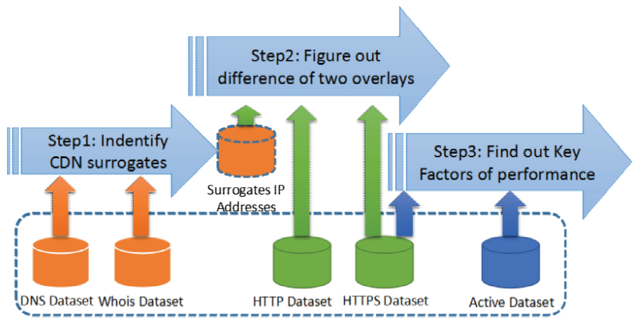


Fig. 2. Main idea.

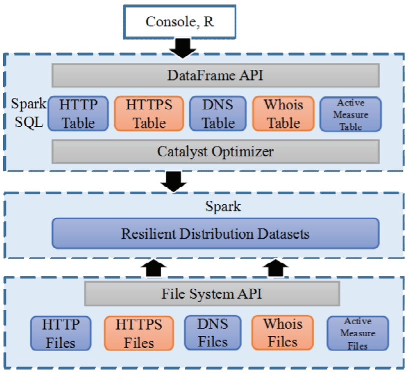


Fig. 3. Processing architecture.

affect the overall content access latency, because a surrogate just plays as a proxy, and the content is actually stored in its owner. Comparing with the caching overlay, we argue that the surrogate assignment of a routing overlay is determined by multiple factors, and need to be examined.

Furthermore, we focus on the routing overlay and further analyze the key factors affecting the surrogate performance. These analyses are carried out based on the *HTTPS* and *round-trip-delay* measure data set. We start from several specific cases of the same dynamic content distributed from different surrogates, and then compare the behaviors of these surrogates to find out the key factors. Based on the results, we further propose a surrogate assignment method to improve the user experience for routing overlays.

We stored our large-scale data sets in a *Hadoop* cluster. We further deployed a *Spark* system over the *Hadoop* cluster. Fig.3 illustrates our data processing architecture. The data set files are collected from our traffic processing system, and loaded into the HDFS. To optimize the analysis performance, we store three copies for each data set. Since the extracted feature data are structured, we employ *Spark-SQL* to deal with it. Then we can conveniently calculate statistics, group and join file large-scale sets using SQL.

B. Our Data Sets

In our work, we develop the analyses based on the following five large-scale data sets shown in Table I. The three passive measurement data sets are captured on a 10 Gbps two-way backbone link of a Chinese ISP.

Because that end user experiences are heavily affected by their access points of Internet, we must collect sufficient large data sets to minimize potential noises and statistical errors. As shown in Table I, we have collected very large data sets to address this concern.

IV. Behaviors Analysis of Two Types of Overlays

In this section, we developed several comparing analyses to figure out the different behaviors between two types of CDN overlays.

A. Observations

Before further discussion, here is our observation which is obtained based on large-scale data analysis.

TABLE I. Description of Data Sets

Data Set	Record Number	Description
DNS	2.6 billion	This data set is collected from the DNS request and response packets. The main attributes include request domain name, domain name aliases (CNAME) and IP addresses (A and AAAA).
HTTP	12.7 billion	This data set is extracted from HTTP traffic, which include multiple attributes such as flow duration, host, user agent, transmission statistics and so on.
HTTPS	20.7 billion	As same as the HTTP data set, this set records the information of HTTPS whose attributes include flow duration, server name indication (SNI), transmission statistics and so on.
Whois	1.04 million	This data set is crawled from APNIC. The main attributes include issuer information and IP address range.
round-trip-delay measure	12.3 million	We measure the RTD from multiple points to the surrogates of CDNs. This data set mainly records the million seconds of RTD.

The caching overlay runs over the HTTP protocol, and the routing overlay employs the HTTPS protocol as its transport layer protocol.

This observation shows the following facts:

- The dynamic contents usually contain user private information. Thus, it requires the routing overlay to employ HTTPS to deliver dynamic sensitive contents.
- The static contents, on the other hand, rarely involve individual information, and are suitable to deliver over HTTP in plain text. Moreover, comparing to HTTPS, HTTP obviously has better performance. Therefore, a a caching overlay mostly use HTTP.

The purpose of this observation is to provide more clear discussion via simplifying the classification of two types of overlays. There are certain exceptions in which a caching overlay uses HTTPS or a routing overlay uses HTTP. These circumstances are rare such that we can ignore them, especially when we analyze a large number of CDNs on large-scale data sets.

B. Identifying surrogate IP addresses

As pointed out in the above, our first task is to find out CDN surrogate IP addresses. We accomplish this task using *DNS* and *whois* data sets.

First, we use the *DNS* data set to screen out the possible surrogate IP addresses. DNS-based request-routing is the most common means for a CDN to redirect user requests to the assigned surrogates. A content owner can point its domain name to a CDN's domain name as an alias to redirect requests. Thus, the domain name alias (canonical name, *CNAME*) gives us sufficient clues to find surrogate IP addresses. We processed the *DNS* data set and found 32 million CNAME records. We further located 3.27 million IP addresses associated with these CNAME records by joining DNS address records.

Then we used the *whois* data set to filter these 3.27 million IP addresses. Due to the fact that the *whois* data set is relatively "dirty", we developed our filter program based on some keywords collected by hand. The keywords are mainly the CDN vendor's names. After filtering, we identify 1.38 million surrogate IP addresses. On our surrogate list, the first three CDN vendors are Akamai, CloudFlare, and Fastly, which account for 87.78%, 5.92% and 4.57% of the total list, respectively.

C. Behavior Analysis

In this section, we select the records from the *HTTP* and *HTTPS* data sets whose server IP addresses are the surrogates, and then compare the behaviors of two types of overlays from the following aspects:

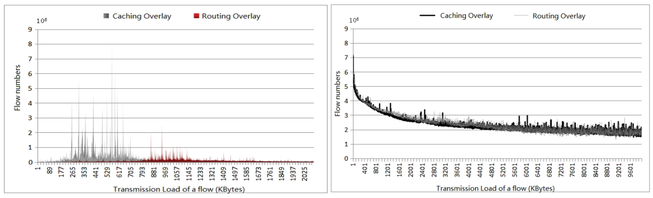
- **Upload and download statistics**. As the goal of a caching overlay is to accelerate user access to static contents, the downloading traffic is usually much more than uploading. In the case of a routing overlay, we believe the uploading and downloading traffic show their interactive characteristics.
- **Access latency**. A caching overlay delivers contents from the local cache of surrogates, while a routing overlay works as a proxy between an end user and a content owner. There must be some differences in the access delays of two types of overlays.

We first examine the transmission load of two types of overlays, illustrated in Fig.4. The x-axis represents the distribution of kilo bytes of transmission of a flow, and the y-axis represents the numbers of flows. From these two figures, we can see that there are obvious differences between the uploading transmission load of the two types of overlays. In specific, the transmission load of routing overlays are noticeably more than caching overlays. This is a clear demonstration of interactive characteristics of routing overlays. On the other direction, illustrated in Fig.4 (b), there are no distinct differences between two types of overlays. It seems that the contents carried by a flow are roughly the same, no matter on a caching overlay or a routing overlay.

Then we calculate the average packet length of two types of overlays. Fig.5 shows the distribution of uploading, and the downloading distribution is illustrated in Fig.6. The uploading distributions of two types of overlays seem the same, since the main payloads from end users to surrogates are the requests, which are usually short with a given range. Nevertheless, the two types of overlays differ greatly in downloading distribution. The caching overlays tend to employ large packets to improve efficiency. The average packet lengths of routing overlays distribute evenly from 300 bytes to 1400 bytes. The routing overlays have a wider range than the caching overlays. This is also because the interactive characteristics of routing overlays, so the dynamic contents can hardly transfer using large packets, compared to the static contents in caching overlays.

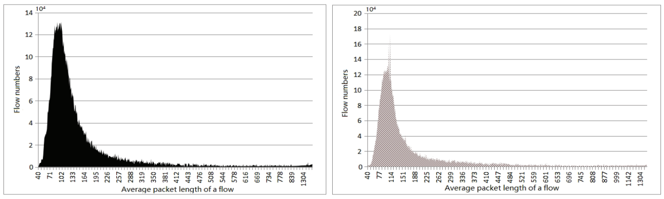
From the above analyses, we can see that the behaviors of routing overlays show obvious interactive characteristics. The interacting is possibly associated with more access latency. Therefore, we further examine the RTD time of two types of overlays, as shown in Fig.7 and Fig.8.

As shown in Fig.7, the average flow duration of routing overlays distributes much wider than that of caching overlays. This is because user interactions usually take more time. Then we further examine the packet arrival intervals of these two types of overlays as shown in Fig.8. It should be noted that the packet arrival intervals of both directions of routing overlays are roughly the same, but there are noticeably differences of the two directions of caching overlays. This is because that the intervals of routing overlays are bound up with interactive behaviors, while the caching overlay is mainly used to deliver



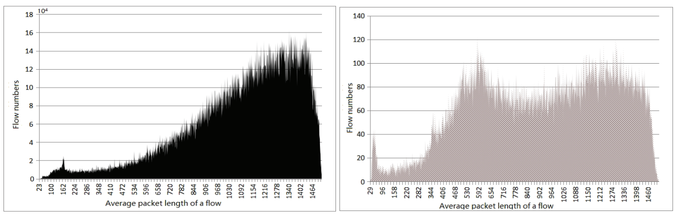
(a) Uploading (b) Downloading

Fig. 4. Distribution of transmission load.



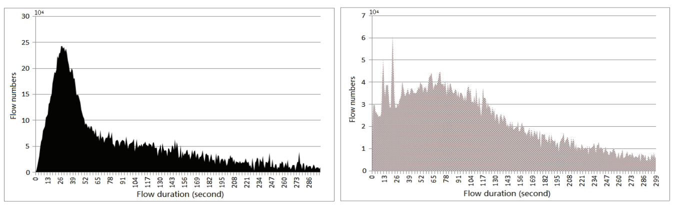
(a) Caching overlays (b) Routing overlays

Fig. 5. Distribution of average packet length of uploading.



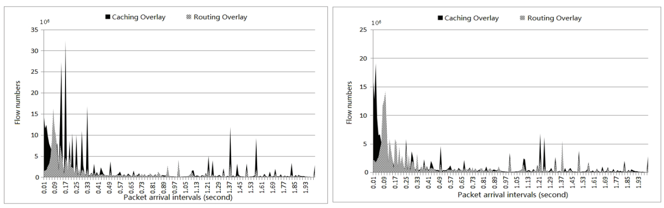
(a) Caching overlays (b) Routing overlays

Fig. 6. Distribution of average packet length of downloading.



(a) Caching overlays (b) Routing overlays

Fig. 7. Flow duration of two types of overlays.



(a) Uploading (b) Downloading

Fig. 8. Packet arrival intervals.

TABLE II. DESCRIPTION OF DATA SETS

IP Address	Country	Access Number	Average duration (s)	Average Packets	Active measure delay (RTD1)(ms)
23.42.189.62	Hong Kong	1,823,714	58.92	12.34	53.75
23.2.138.14	Japan	863,633	34.15	11.08	78.63
104.76.21.248	Japan	666,630	30.25	11.22	62.33
104.95.218.170	USA	493,478	8.91	11.18	130.62
23.58.248.102	Malaysia	328,300	19.09	12.04	175.92
104.96.42.200	Germany	251,969	8.25	12.06	311.01
104.70.143.114	Japan	232,806	45.94	10.94	122.32
23.43.5.44	Korea	170,621	61.76	13.34	91.2
104.115.113.147	USA	154,796	5.73	11.96	165.11
2.17.44.166	Netherlands	153,267	10.75	10.8	222.82

static contents. In the case of caching overlays, the TCP sliding window mechanism could work at its full capacity to improve performance. Besides the initial requests, the remaining of uploading are usually TCP acknowledge packets, which are used to confirm a batch of downloading packets.

In summary, this section demonstrates the key differences between two types of overlays. The fundamental cause lies in the interactive behaviors. As the surrogates of routing overlays just work as proxy servers, the overall access delay of end users should embrace the delay time between end users and surrogates as well as between surrogates and owner providers. This interactive characteristic also reflects to packet arrival intervals. As we carefully examine the above results, we find that the distribution of routing overlay intervals has several pulses, especially in Fig.8 (b). We believe that these pulses demonstrate the effect of delays between surrogates and content owners on the overall access delay, because the delays between end users and surrogates should be relatively stable. From this point, we argue that it is worth to study the access latency of routing overlays in depth.

V. ACCESS DELAY OF ROUTING OVERLAYS

Fig.9 shows some details of routing overlay networks. The whole network works as a reverse proxy, which is used to forward end user requests and bring responses back to them. As we consider the access latency of routing overlays, there are two parts need to be considered, including RTD1 (a delay between an end user and a surrogate) and RTD2 (a delay between a surrogate and a content owner). For the current surrogate assignment, the RTD1 is a major consideration. However, the assigned surrogates may be far away from the content owners. In this case, the overall access delay of the assigned surrogate may be a little higher than some other surrogates. In this section, we try to measure RTD2, and then propose a solution for better surrogate assignments.

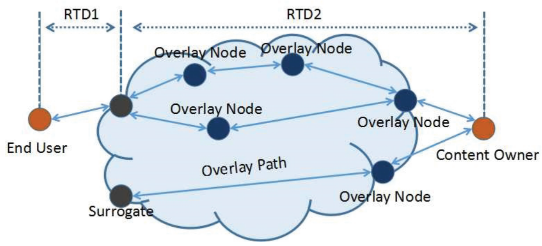


Fig. 9. Routing overlay network.

We plan to start the analysis with a specific application. According to our previously assumption, the routing overlay use HTTPS as its application layer protocol. In our work, we use a server name indication (*SNI*) to identify an application. Our target application is selected by the following conditions:

- The target application should employ enough surrogates to deliver its contents so that we can collect sufficient large scale data to analysis.
- The service provided by the target application should be relatively homogeneous.

We first group the *HTTPS* data set by SNIs and surrogate IP addresses first, and calculate the mean and standard deviation values of flow duration, packet numbers, and transmission bytes of two directions. Then, we filter the grouped data set with the conditions of standard deviations of packet number and transmission bytes less than their mean values. At last, we sort the filtered data set according to the number of employed surrogates. Finally, we determine to use an application named "api.accuweather.com" as a case for the following analyses. This application provides real-time weather forecast services. According to our observation, this application employs 201 Akamai surrogates. Due to the limitation of our observation, we select 100 surrogates whose number of flow records per day is more than 10,000. Fig. 10 shows the statistics of flow duration and packet numbers associating with this application. For most surrogates, the average number of packets per flow ranges from 10 to 12, however the average duration fluctuates greatly. We further analyze the most frequently used surrogates, illustrated in Table II.

Table II shows some an interesting result. Of all the listed surrogates, the first three surrogates (have lower RTD1s) have a relatively long flow duration. This shows that the RTD2 of these three surrogates are longer than other surrogates. As examining the geographical location, we find the surrogates

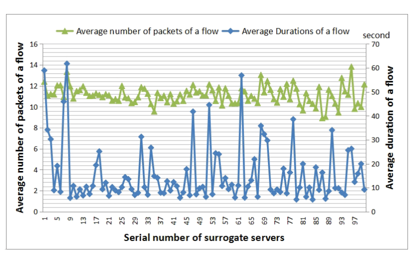


Fig. 10. Statistics of "api.accuweather.com".

located at East Asian have lower RTD1, therefore they are usually assigned to Chinese users. The surrogates at American or Europe can provide better performance, although the their RTD1 are relatively long.

From the case study, we conclude that the surrogates with low RTD1 to end users may be not the best choice to deliver dynamic contents, since the RTD2 from these surrogates to content owners may be larger than the others not assigned to the end users. Therefore, the current assignment policy which only consider the RTD1 is not applicable for the routing overlays. The current assignment policy can be estimated as the sum of two costs: (1) The cost of reserving computational resources at the candidate surrogates, including CPU, memory, and storage resources; (2) The cost of transferring the contents between end users and surrogates. However, this policy cannot satisfy routing overlays. The cost of transferring contents between end users and surrogates must be taken into the account. In this case, the assigned surrogates can provide the best overall performance for end users.

VI. Conclusions and Future Works

In this paper, we developed several analyses based on five large-scale data sets. The analysis results reveal the difference between caching overlays and routing overlays of CDNs. Aiming at the routing overlays, we further examine a specific application to figure out the influence of access delays. The results show that the delay between surrogates of a CDN and content owners is a significant factor of the overall performance of routing overlays. Furthermore, we argue that the surrogate assignment policy of routing overlays need to take this delay into account in order to improve user experiences. In the future work, we plan to divide routing overlays into sub-types, measuring the access latency of each type and analyze their performance.

Acknowledgment

This work is supported by the project of National Natural Science Foundation of China (No. 61571144).

References

[1] J. Dilley, B. M. Maggs J. Parihk and et al. Globally distributed content delivery. IEEE Internet Computing, 2002, Vol. 6, Issue 5, pp: 50–58.

[2] E. Nygren, R. K. Sitaraman, and J. Sun. The Akamai Network: A platform for high performance Internet applications. ACM SIGOPS Operating Systems Review, 2010, Vol. 44, Issue 3, pp: 2–19.

[3] The Zettabyte Era: Trends and Analysis. http://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/vni-hyperconnectivity-wp.html, July, 2017.

[4] R. K. Sitaraman, M. Kasbekar, W. Lichtenstein and et al. Overlay Networks: An Akamai Perspective. Advanced Content Delivery, Streaming, and Cloud Services, 2014, pp:305 - 328.

[5] C. Papagianni, A. Leivadeas and S. Papavassiliou. A Cloud-Oriented Content Delivery Network Paradigm: Modeling and Assessment. IEEE Transactions on Dependable and Secure Computing, 2013, Vol. 10, Issue 5, pp: 287–300.

[6] A. Passarella. A survey on content-centric technologies for the current Internet: CDN and P2P solutions. Computer Communications, 2012, Vol. 35, Issue 1, pp: 1–32.

[7] J.J. Liang, J. Jiang, H.X Duan and et al. When HTTPS Meets CDN: A Case of Authentication in Delegated Service. 2014 IEEE Symposium on Security and Privacy, pp: 67 – 82.

[8] M. K. Mukerjee, I. N. Bozkrut, B. Maggs and et al. The Impact of Brokers on the Future of Content Delivery. In proceedings of the 15th ACM Workshop on Hot Topics in Networks, 2016, pp: 127-133.

[9] D. S. Berger, R. K. Sitaraman, M. Harchol-Balter. AdaptSize: Orchestrating the Hot Object Memory Cache in a Content Delivery Network. In the Proceedings of the 14th USENIX Symposium on Networked Systems Design and Implementation. 2017, pp: 483 - 498.

[10] M. Rabinovich, Z. Xiao, A. Aggarwal, Computing on the edge: a platform for replicating internet applications, Web content caching and distribution: proceedings of the 8th international workshop, 2004, pp. 57–77.