Chapter 2
Content Delivery Networks and Its Interplay with ISPs

This chapter surveys measurement studies on Content Delivery Networks (CDNs) in real world. Several representative CDNs with different architecture designs, client mapping policies and redirection mechanisms are examined, and the interplay between CDNs and ISPs are further discussed.

2.1 CDNs in Real World

2.1.1 Akamai

System Architecture. Huang et al. [1] carry out a measurement study on the large-scale CDN of Akamai. The authors take two steps to chart a CDN network. In the first step, they find out all the Canonical NAMEs (CNAMEs) that are used by the target CDN's customers, by resolving 16 million web hostnames. A total number of 282,700 unique open local DNS (LDNS) servers are employed as vantage points in their measurement study. In the second step, the authors query the open LDNS servers all around the world with the CNAMEs, for discovering the Akamai content servers.

With the proposed methodology, the authors have successfully discovered 27,000 Akamai content servers, which the authors believe that complete the Akamai global network. The content servers are distributed in 65 countries in the world, and are in 656 Autonomous Systems (ASes). The authors also find that Akamai employs a two-tier DNS infrastructure with 6,000 name servers for resolving the second level domain names, and they conjecture that at least one name server is run in each distributed cluster of Akamai. The measurement result shows that Akamai adopts "deep-into-ISPs" architecture design in its CDN.

Researchers from Akamai later reveals its system architecture in [2], which is composed of an edge server platform, a mapping system, a communication and control system, and a data collection and analysis system. Akamai's edge servers are

© The Author(s) 2017

Y. Tian et al., *Internet Video Data Streaming*,
SpringerBriefs in Computer Science, DOI 10.1007/978-981-10-6523-1_2

responsible for processing end user requests and serving the requested contents, as well as for acting as intermediaries in the content delivery overlay network.

The mapping system of Akamai has two objectives, scoring and real-time mapping. In scoring, the system creates a topological map capturing the state of connectivity across the entire Internet by dividing the Internet into equivalence classes of IP addresses and represents how well they connect to each other. In real-time mapping, a top-level map first selects a preferred edge server cluster for each equivalence class of end users, according to the scoring result; then a low-level map within a cluster directs content request to a specific server.

Akamai views a content delivery network as a virtual network over the Internet. For video content delivery, Akamai organizes its edge servers as a tiered cache system, which can effectively improve the video delivery performance by reducing the number of content requests back to the origin server. For live video streaming, the transport system transports the stream's packets from the entry point to a subset of edge servers that require the stream using a publish-subscribe model.

Mapping System. Recently, Chen et al. [3] elaborate Akamai's mapping system, which is responsible for making real-time server selection decisions for user clients. They consider the conventional NS-based mapping system previously employed by Akamai as computing the following complex time-varying function:

$$Map_t : \Omega_{Internet} \times \Omega_{Akam} \times Domain \times LDNS \rightarrow IPs \tag{2.1}$$

where $\Omega_{Internet}$ is the current state of the global Internet and Ω_{Akam} is the current state of the Akamai CDN. The two inputs are obtained through network measurement. *Domain* is the domain name of the content accessed by the client and *LDNS* is local recursive DNS server that makes the name resolution request on behalf of the client. A problem for the NS-based mapping is that server selection decisions are made based on the identity of the client's LDNS rather than that of the client itself, and some clients are far away from their LDNS servers, due to the wide usage of the public third-party name service providers such as Google DNS [4] and OpenDNS [5]. As a result, the NS-based mapping may select content servers that are distant from the clients and cause high latencies in downloading the contents.

Chen et al. [3] propose to exploit the recently proposed EDNS0 client-subnet extension, where LDNS forwards a prefix of the client's IP in name resolution requests, to assist the CDN to select servers. Conceptually, with the EDNS0 extension, the mapping system computes the following time varying function:

$$Map_t : \Omega_{Internet} \times \Omega_{Akam} \times Domain \times Client \rightarrow IPs \tag{2.2}$$

where *Client* is the IP prefix of the client.

Akamai deploys the end-user mapping system around the world in the first half of 2014. By comparing the mapping results before and after the deployment, the authors find that end-user mapping provides significant performance benefits. In particular, for clients who use public resolvers, the end-user mapping achieves an eight-fold

decrease in mapping distance, a two-fold decrease in RTT and content download time, and a 30% improvement in the time-to-first-byte.[1]

Performance. Triukose et al. [6] study performance of the Akamai CDN through an active measurement. The authors focus on the performance in terms of download throughput. By carefully constructing requesting URLs, the authors measure and compare the throughputs under the cases of cache hit, cache miss and directly downloading from the origin server. The comparison result shows that Akamai effectively improves the web performance, as in 67 and 41% of the cases, the CDN delivery is at least five times faster than cache-miss and origin delivery respectively.

The authors also examine Akamai's server selection decisions, and find that the CDN rarely selects the best server cluster with the highest download throughput, but it successfully avoids the worst ones. In roughly 75% of the cases, the Akamai-selected server outperformed half of the alternatives, suggesting that there is still space for improvement.

Motivated by the observations, the authors consider consolidating the existing infrastructure of the Akamai CDN into fewer clusters. With simulation experiments driven by the measurement data, it is found that clients would not see noticeable performance difference if the servers were further consolidated into 40 clusters, given that the CDN can always direct clients to the best servers.

2.1.2 Limelight

Huang et al. [1] also employ the same platform for measuring Akamai to chart the Limelight CDN. The authors find that Limelight uses IP anycast to announce its name servers, and reuses its content servers as DNS servers as well. The 4,100 discovered content servers are located in only 19 locations, making Limelight a representative CDN following the "bringing ISPs at home" design philosophy. In addition, it is observed that Limelight's anycast routing does not redirect clients to the closest data centers more than 37% of the time, and the redirection is beyond the top three data centers in about 20% of the time.

The authors then compare Limelight with Akamai regarding the CDN health and the delay performances. Limelight is observed to have better server availability and longer uptime, suggesting that comparing with Akamai, Limelight servers are easier to maintain. For evaluating the delay performance, the authors propose a revised King approach to measure the delay from each LDNS vantage point to the CDN-selected server. The measure result shows that the delay performance has a positive correlation with the coverage of the CDN in a specific target geographic region, and Limelight has considerable longer delays than Akamai in Europe, Asia, Oceania, South America, and Africa, due to its fewer data centers deployed in these regions.

[1] Time-to-first-byte, or TTFB, refers to the duration from when the client makes a HTTP request for the base web page to when the first byte of the requested web page was received by the client.

2.1.3 Amazon CloudFront

Bermudez et al. [7] study Amazon's Web Service (AWS) provided by EC2, S3 and CloudFront. The authors employ one-week traffic collected by Tstat, a traffic monitoring tool, at two ISP PoPs and their campus network. A total number of $6M$ TCP connections between clients and AWS servers were examined in their passive measurement study.

The authors find that for EC2 and S3, workload distributed among different data centers are unbalanced, with 85% European clients being redirected to a data center in US, leading to long latencies, high network cost, and low throughput. The authors find that the CloudFront CDN deploys its servers in 33 locations, and by directing clients to close servers, the CDN shows excellent web performance comparing with EC2 and S3.

2.1.4 Google

Infrastructure. Calder et al. [8] carry out a study of Google's web content delivery infrastructure over a ten-month period. The authors have proposed a suite of methods for mapping Google's infrastructure. More specifically, the EDNS-client-subnet extension is exploited to enumerate addresses of Google's front-end servers, and has collected 28,793 front-end IP addresses. The authors propose a novel technique named client-centric geolocation (CCG) to approximate the location of a server by the geographical mean of the clients that it serves. Finally, the authors measure the RTTs of the front-ends to PlanetLab node landmarks, and cluster them into CDN nodes with the OPTICS algorithm [9].

Armed with the methodologies, the authors present a detailed analysis on Google's web serving infrastructure and its expansion in ten month. It is observed that instead of serving clients from front-end servers on its own backbone network, Google expands its infrastructure to the ASes in lower tiers of the AS hierarchy. As a result, although Google still directs majority of clients to servers in its own network, an additional 8% of clients are served off its network; moreover, the expansion makes clients to be directed to servers that are substantially closer to them. The observation suggests that Google has evolved from a typical "bringing ISP to home" CDN to a hybrid one, as in addition to maintaining its own backbone network and data centers, Google also deploys its server clusters deep into the small eyeball ISPs.

Front-end Server Placement. Chen et al. [10] consider the front-end server placement in dynamic content distribution. The authors consider Google and Bing as two cases, and perform extensive network measurement on their CDNs. The authors propose a simple model-based inference framework to indirectly measure and quantify the "front-end to back-end fetching time" in the two CDNs, and point out that there is a distance threshold within which placing font-end servers further closer to users is no longer helpful; instead, the end-to-end performance is now determined by the

front-end to back-end fetching time, which is determined by the query processing time at back-end data centers and the delivery time between the back-end data centers and the front-end servers. The authors also show that Google outperform Bing in content fetching time, despite that it's front-end servers are slightly farther from the clients.

Path Optimization. Krishnan et al. [11] diagnose the latency inflation of the end-to-end paths between clients and Google's CDN nodes. Google's server selection decisions are made based on latency, more specifically, Google gathers a map of latencies from CDN nodes to IP address prefixes. The latency is collected by redirecting a client to a random node, measuring the RTT, and using it as representative of the client's prefix. With RTT measurements gathered over time, and refreshed periodically, Google determines the closest CDN node for each IP address prefix, and redirects clients in the prefix to the closest node.

For evaluating effectiveness of the latency-based server selection, the authors examine the RTT data measured by 13 nodes in Google's CDN to the 170 K network prefixes they serve in one day. The measurement result shows that although each client is served by the CDN node measured to have the lowest latency to the client's prefix, in 40% cases, the measured RTTs are greater than 400 ms. In other words, the latencies experienced by clients are poor. By further analyze the RTT measurement data and probing prefixes with traceroute, the authors find that there are two major causes for the latency inflation: First, some clients have circuitous routes to or back from the CDN node serving them; Second, TCP connections to most clients are impacted by significant queueing delays.

The authors design and implement *WhyHigh*, a network diagnosis system, to identify the client prefixes that suffer latency inflation and pinpoint the causes for the instances of latency inflation. Inflated prefixes at a node are identified by comparing the minimum RTT measured at the node across all connections to the prefix with the minimum RTT measured at the same node across all connections to clients within the prefix's region. A prefix is declared to be inflated if the RTT difference is greater than 50 ms. For each inflated prefix instance, WhyHigh attempts to pinpoint the set of causes. The plausible causes include lack of peering between the prefix's ISP and Google, limited bandwidth of the peering link, ISP routing misconfiguration, and traffic engineering by ISP.

WhyHigh has identified and diagnosed a total number of 11,862 inflated prefixes globally, and among them, 3,776 instances are due to the lack of peering. The information provided by WhyHigh has helped Google to work with ISPs in the region and significantly improve path latencies.

2.1.5 Bing

Calder et al. [12] study the CDN of Bing, which relies on anycast to direct clients to front-end servers. Anycast is a routing strategy where the same IP address is

announced from many locations throughout the Internet. Then BGP routes clients to one front-end location based on the best BGP path. The authors combine Bing server logs and Javascript-based active measurements to discover front-end servers of the Bing CDN, and measure the latencies between clients and different front-end servers, including the anycasted servers and the top-ten closest servers. The measurement results show that anycast usually performs well despite the lack of precise centralized control, but it directs roughly 20% of clients to a suboptimal front-end server.

The authors also find that the anycast server selections are quite stable, as clients continue going to the same front-end servers over time. The authors propose a simple prediction scheme, and shows with simulation that traditional and recent DNS techniques can improve performance for many of the clients who experience suboptimal anycast routing, thus the DNS-based redirection can be applied for only a small subset of clients that suffer poor server selections, while leaving the others to anycast.

2.1.6 YouTube

Load Balancing. Adhikari et al. [13] conduct an extensive and in-depth study of traffic exchanged between YouTube data centers and its users, as seen from the perspective of a tier-1 ISP in Spring 2008 after YouTube was acquired by Google but before Google did any major restructuring of YouTube. The authors employ the flow-level data related to YouTube that is collected from 44 ISP PoPs in their study, and find that YouTube employs a proportional load balancing strategy to split traffic among the seven data centers, which is very different from latency or location-based strategies that are usually observed, and the proportionality seems to be determined by the "size" of the data centers. The authors also find asymmetric "early-exit" routing paths for client-to-YouTube and YouTube-to-client traffics.

The authors perform a "what if" analysis on the scenario that YouTube adopts a location-based server selection strategy by directing clients to geographical nearby data centers, and show that under such a strategy, the overall video download latency can be reduced, but the workload among the datacenters will become unbalanced.

Server Selection. Torres et al. [14] conduct a detailed study of the YouTube CDN, and seeks to understand the mechanisms and policies for determining which data centers users download video from. The authors employ week-long datasets simultaneously collected from five networks, including ISPs and campus networks in 2010, and apply the delay-based CBG method [15] to geolocate the YouTube video servers. A total number of 33 data centers are identified. By analyzing the traffic datasets, the authors find that the data center that provides most of the traffic is also the data center with the smallest RTT for each dataset.

The authors also investigate the video downloading from non-preferred data centers, and find that a variety of causes, including load balancing across data centers,

variations across DNS servers within a network, alleviation of hotspots due to popular video content, and accesses of sparse video content that may not be replicated across all data centers, are responsible for the video downloading from non-preferred data centers.

2.1.7 *Netflix and Hulu*

Adhikari et al. [16, 17] investigate the systems of Netflix and Hulu, the leading Over-the-Top (OTT) video service providers in the US and Canada. The authors find that both providers employ the same set of three CDNs, namely Akamai, Limelight, and Level3, to distribute video contents. For CDN selection, Netflix assigns a fixed CDN preference order for each user account, and a Netflix client stays with a same CDN as long as possible even if it has to degrade the playback quality level. For each client, Hulu chooses the preferred CDN for each video based on a latent distribution, and when the selected CDN has degraded bandwidth, Hulu adjusts the playback bit rates and continues to use the same CDN server as long as possible.

Motivated by the fact that the CDN preference and selection strategies employed by Netflix and Hulu are agnostic of the varying network conditions of the CDNs, the authors propose a measurement-based adaptive CDN selection strategy and a multiple-CDN-based video delivery strategy. The strategies meet the business constraints between the CDNs and the provider, and can significantly increase users' average available bandwidth in video streaming, by dynamically switching the streaming CDN according to real-time network conditions of the CDNs.

2.1.8 *CDNs in China*

Xue et al. [18] focus on the Chinese CDNs, which are different from global CDNs because of two reasons. First, the Chinese Internet is highly hierarchical with relatively few ASes [19]. Links between ASes are statically configured without using BGP, such a feature makes approaches like anycast fairly ineffective in China. Second, a global CDN usually cannot place content servers in China due to legal reasons, therefore must partner with Chinese CDNs.

The authors carry out a measurement-based study on a number of Chinese CDNs and global CDNs, and have the following observations: (1) all the Chinese CDNs employ the DNS rediection scheme, and tend to select a static set of servers; (2) Chinese CDNs provide customized content delivery service for web sites, leading to different latency performance when accessing these web sites, despite that they are served by a same CDN provider; (3) global CDNs who partner with Chinese CDNs can achieve significantly lower latencies than those who do not, because the contents are provided by servers inside China. The authors also discuss the possible reasons of poor server selections, which should be useful for optimizing the CDNs in China.

2.2 Interplay Between CDN and ISP

2.2.1 ISP Prospective

Poese et al. [20] consider the inherited conflict between ISP and CDN. More specifically, the authors consider the problem that today's CDN optimizes its traffic flows for minimizing its operational cost under the SLA constraint, the traffic scheduling result may be sub-optimal to the ISPs, and impose a heavy burden on them. For example, if the ISP changes its routing, e.g., for the purpose of traffic engineering, the content delivery network may re-optimize this delivery strategy and change the traffic matrix, which may invalidate the traffic engineering choice of the ISP [21].

To overcome this impairment, the authors propose a solution where the ISP offers a *Provider-aided Distance Information System* (*PaDIS*). PaDIS works as a location recommendation service operated by ISP. When a user, a CDN, a DNS resolver, or any other entity queries PaDIS by submitting a list of possible IP addresses and a source, PaDIS ranks the submitted IP addresses according to its metrics such as distance within the Internet topology, path capacity, path congestion, path delay, etc. The authors evaluate PaDIS in real-world measurement experiments, and find that with PaDIS, end-user download time is significantly reduced, while the ISP regains the ability to perform traffic engineering by biasing CDN's server selection decisions.

2.2.2 CDN Prospective

Khare et al. [22] considers the CDN-ISP interplay from the CDN prospective. Concretely, CDNs pay the ISPs for using the underlying networks based on the traffic volume that CDN servers send and receive. If an ISP's operational cost goes up due to the increase of inter-domain payments, it will eventually be reflected in higher traffic charges to the CDNs. Therefore CDN and ISP have the common interest in reducing the ISP's inter-domain payments, which will reduce the operational costs for both of them. The authors propose that to economically incentivize a CDN to consider the underlying ISPs' routing preferences, an ISP should charge differently for content traffic depending upon the type of inter-domain links it traverses, and make this pricing information available to the CDNs.

The authors devise a novel server assignment mechanism, which seeks to minimize a CDN's payment to the ISPs by directing clients to servers in the "right" ISPs under the new traffic pricing model. By comparing with the nearest-available sever selection policy, the authors show that the proposed mechanism provides good content delivery performance for end users, and has significant savings on the CDN's traffic payment.

References

1. Huang, C., Wang, A., Li, J., Ross, K.W.: Measuring and evaluating large-scale CDNs. Technical Report, Microsift Research, MSR-TR-2008-106 (2008)
2. Nygren, E., Sitaraman, R.K., Sun, J.: The Akamai network: a platform for high-performance Internet applications. ACM SIGOPS Oper. Syst. Rev. **44**, 2–19 (2010)
3. Chen, F., Sitaraman, R.K., Torres, M.: End-user mapping: next generation request routing for content delivery. In: Proceedings of ACM SIGCOMM, London, UK, pp. 167–181 (2015)
4. Google Public DNS: http://developers.google.com/speed/public-dns/
5. OpenDNS: http://www.opendns.com/
6. Triukose, S., Wen, Z., Rabinovich, M.: Measuring a commercial content delivery network. In: Proceedings of WWW, Hyderabad, India, pp. 467–476 (2011)
7. Bermudez, I., Traverso, S., Mellia M., Munafo, M.: Exploring the cloud from passive measurements: the Amazon AWS case. In: Proceedings of IEEE INFOCOM, Turin, Italy, pp. 230–234 (2013)
8. Calder, M., Fan, X., Hu, Z., Katz-Bassett, E., Heidemann, J., Govindan R.: Mapping the expansion of Google's serving infrastructure. In: Proceedings of IMC, Barcelona, Spain, pp. 313–326 (2013)
9. Ankerst, M., Breunig, M., Kriegel, H., Sander, J.: OPTICS: ordering points to identify the clustering structure. ACM SIGMOD Record. **28**(2), 49–60 (1999)
10. Chen, Y., Jain, S., Adhikari V.K., Zhang Z.-L.: Characterizing roles of front-end servers in end-to-end performance of dynamic content distribution. In: Proceedings of IMC, Berlin, Germany, pp. 559–568 (2011)
11. Krishnan, R., Madhyastha, H.V., Srinivasan, S., Jain, S., Krishnamurthy, A., Anderson, T., Gao, J.: Moving beyond end-to-end path information to optimize CDN performance. In: Proceedings of IMC, Chicago, IL, USA, pp. 190–201 (2009)
12. Calder, M., Flavel, A., Katz-Bassett, E., Mahajan, R., Padhye, J.: Analyzing the performance of an anycast CDN. In: Proceedings of IMC, Tokyo, Japan, pp. 531–537 (2015)
13. Adhikari, V.K., Jain, S., Zhang, Z.-L.: YouTube traffic dynamics and its interplay with a tier-1 ISP: an ISP perspective. In: Proceedings of IMC, Melbourne, Australia, pp. 431–443 (2010)
14. Torres, R., Finamore, A., Kim, J.R., Mellia, M., Munafo, M.M., Rao, S.: Dissecting video server selection strategies in the YouTube CDN. In: Proceedings of ICDCS, Minneapolis, MN, USA, pp. 248–257 (2011)
15. Gueye, B., Ziviani, A., Crovella, M., Fdida, S.: Constraint-based geolocation of internet hosts. IEEE/ACM Trans. Network. **14**(6), 1219–1232 (2006)
16. Adhikari, V.K., Guo, Y., Hao, F., Varvello, M., Hilt, V., Steiner, M., Zhang, Z.-L.: Unreeling Netflix: understanding and improving multi-CDN movie delivery. In: Proceedings of IEEE INFOCOM, Orlando, FL, USA, pp. 1620–1628 (2012)
17. Adhikari, V.K., Guo, Y., Hao, F., Hilt, V., Zhang, Z.-L., Varvello, M., Steiner, M.: Measurement study of Netflix, Hulu, and a tale of three CDNs. IEEE/ACM Trans. Network. **23**(6), 984–1997 (2015)
18. Xue, J., Choffnes, D., Wang, J.: CDNs meet CN: an empirical study of CDN deployments in China. IEEE Access **5**, 5292–5305 (2017)
19. Tian, Y., Dey, R., Liu, Y., Ross, K.W.: China's Internet: topology mapping and geolocating. In: Proceedings of IEEE INFOCOM, Orlando, FL, USA, pp. 2531–2535 (2012)
20. Poese, I., Frank, B., Ager, B., Smaragdakis, G., Feldmann, A.: Improving content delivery using provider-aided distance information. In: Proceedings of IMC, Melbourne, Australia, pp. 22–34 (2010)
21. Jiang, W., Zhang-Shen, R., Rexford, J., Chiang, M.: Cooperative content distribution and traffic engineering in an ISP network. In: Proceedings of ACM SIGMETRICS, Seattle, WA, USA, pp. 239–250 (2009)
22. Khare V., Zhang, B.: Making CDN and ISP routings symbiotic. In: Proceedings of ICDCS, Minneapolis, MN, USA, pp. 869–878 (2011)



http://www.springer.com/978-981-10-6522-4

Internet Video Data Streaming
Energy-saving and Cost-aware Methods
Tian, Y.; Zhao, M.; Zhang, X.
2017, XI, 60 p. 24 illus., Softcover
ISBN: 978-981-10-6522-4