Characterizing the Deployment and Performance of Multi-CDNs

Rachee Singh
rachee@cs.umass.edu
UMass Amherst

Arun Dunna
adunna@cs.umass.edu
UMass Amherst

Phillipa Gill
phillipa@cs.umass.edu
UMass Amherst

ABSTRACT

Pushing software updates to millions of geographically diverse clients is an important technical challenge for software providers. In this paper, we characterize how content delivery networks (CDNs) are used to deliver software updates of two prominent operating systems (Windows and iOS), over a span of 3 years. We leverage a data set of DNS and ping measurements from 9,000 RIPE Atlas clients, distributed across 206 countries, to understand regional and temporal trends in the use of multiple CDNs for delivering OS updates. We contrast two competing methodologies for distributing OS updates employed by Microsoft and Apple, where the majority of Microsoft clients download Windows updates from their local ISP. But, 90% of Apple clients access iOS updates from Apple's own network. We find an approximate improvement of 70 ms in the latency observed by clients in Asia and Africa when accessing content from edge caches in local ISPs. Additionally, Microsoft provides lower latencies to its clients in developing regions by directing them to Akamai's rich network of edge caches. We also observe that clients in developing regions accessing Windows updates from Level 3 get poor latencies arising from the absence of Level 3's footprint in those regions.

CCS CONCEPTS

• **Networks** ! *Network measurement*; *Location based services*; *Network monitoring*;

ACM Reference Format:
Rachee Singh, Arun Dunna, and Phillipa Gill. 2018. Characterizing the Deployment and Performance of Multi-CDNs. In *2018 Internet Measurement Conference (IMC '18), October 31-November 2, 2018, Boston, MA, USA.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3278532.3278548

1 INTRODUCTION

Content Delivery Networks (CDNs) are employed by various content providers to facilitate low-latency access to clients around the globe. CDN deployments require a significant financial investment by the content providers themselves [1]. With CDNs leveraging their own proprietary store of network measurement data and mapping heuristics, these networks can present different strengths and weaknesses relative to each other in terms of serving certain types of clients (*e.g.,* mobile users) or different geographic regions (*e.g.,* developing regions) better. These strengths combined with a desire to improve reliability in the face of the failure of a single CDN has led some organizations to use multiple CDNs to deliver their content [2].

While the use of multiple CDN providers has been known for nearly a decade, studies of their deployment have been limited. Content providers have an interest in quantifying the performance of multiple CDNs but publishing such data can be challenging if it reflects poorly on their business partners. In this study, we take an alternate approach by studying the deployment of multi-CDNs for content delivery using end-to-end path latency observed by clients. Using data from a recent plenary at RIPE71 [3], we focus on the multi-CDN infrastructure employed by two large software vendors (Apple and Microsoft) for delivering operating system (OS) updates to their customers.

We extend prior analysis of this dataset and develop methods to identify CDN edge caches and assign them to the appropriate CDN provider (§3). With this methodology and dataset we perform one of the first large-scale and longitudinal analyses of multi-CDN deployments. We specifically focus on understanding several facets of multi-CDN deployment: (1) the mix of CDN providers employed by content providers (§4.1), (2) the performance of the different CDN providers (approximated by latency) (§4.2) and in different regions (§4.3), (3) the stability of replica mappings (§5), and (4) the impact of shifting between CDN providers for clients (§6).

Through our analysis, we make the following key observations:

Regional performance trends. There are significant regional variations in the client-side latency across continents. While clients in developed regions (*e.g.,* North America and Europe) observe a median latency of 20 ms, clients in developing regions (*e.g.,* Africa, Asia, South America) observe median latencies as high as 200 ms. There has been a decline in latency of access in developing regions over the last two years but there is significant room for improvement.

Performance impact of edge caches. Clients in developing regions observe significant improvement in latency by accessing content from local edge caches (Section 6.2). In particular, clients in Africa see over 10X reduction in latency by migrating towards edge caches.

2 BACKGROUND

Latency is an important aspect of user experience on the Internet. Due to this, content providers like Google, Microsoft, etc. have either partnered with Content Delivery Networks (CDNs) or developed their own solutions to bring content closer to users. Broadly, CDNs employ one of two approaches for mapping content servers to clients:

IMC '18, October 31-November 2, 2018, Boston, MA, USA

ACM ISBN 978-1-4503-5619-0/18/10. . . $15.00
https://doi.org/10.1145/3278532.3278548

DNS-based redirection of clients: CDNs that provide DNS based client redirection develop a notion of the *best* edge server for clients and use DNS responses to direct clients to the selected replica(s). When the client resolves the URL of an object hosted on the CDN, the authoritative name server of the CDN responds with the IP address of the *best* edge server for the client. One limitation of DNS based redirection is that all clients which have the same local DNS resolver are redirected to the same edge server. While this works well when clients use nearby local DNS resolvers, it fails when a single resolver is responsible for a geographically diverse set of clients (*e.g.,* Google's open DNS resolver [4]).

The Akamai CDN employs DNS based schemes for redirecting clients to edge servers [5]. Building such an infrastructure for content delivery is expensive and requires regular telemetry from clients to determine the client-server mapping. Researchers have studied the performance of DNS based redirection at Akamai and quantified cases where DNS redirection under-performs and have proposed measures to fix the problem [5]. However, their solution relies on ISPs implementing DNS ECS (RFC 7871).

Anycast based redirection of clients. CDNs also use anycast routing for client redirection by announcing an edge server IP via BGP from multiple locations on the Internet. Depending on the client's location, their ISP will choose the best path towards the anycast prefix based on BGP routing policies. This approach is simple to deploy since it does not require infrastructure deployment or telemetry from the CDN. But this ease of deployment of anycast CDNs comes at the cost of fine-grained control. This can lead to overloading of edge servers and inability to migrate specific clients away from the overloaded server. Additionally, if a non-deterministic phenomenon such as congestion in routers affects latency towards an edge server, then anycast, which is BGP-driven, remains unaware.

Microsoft's Bing service uses anycast, with a recent study measuring its performance, and contrasting it with a DNS-based approach [6]. This work analyzes active and passive measurements from Microsoft's edge servers to clients and categorizes client prefixes that observe high latency from edge servers. They find that 20% of client prefixes observe worse latency using anycast CDN as compared DNS based CDNs.

3 METHODOLOGY

In this section, we discuss the measurement methodology used to collect data on multi-CDN deployments and identify specific CDNs and edge caches within the data set. The data set we consider focuses on CDN deployments used to distribute operating system (OS) updates for two large software vendors: Microsoft and Apple.

3.1 Data Collection

We leverage data collected over two years by ~9,000 RIPE Atlas nodes located in ~3,000 autonomous systems (ASes) around the world [3] to perform a detailed characterization of multi-CDN deployment by two large software vendors: Microsoft and Apple. Data is collected by issuing *ping* measurements from all available vantage points in the RIPE Atlas platform to domains used for hosting software updates: download.windowsupdate.com and appdownload.itunes.apple.com for Microsoft and Apple, respectively. Each probe resolves the domain name locally, and then performs 5 pings to the resolved IP address. The average, minimum and maximum round-trip time (RTT) of 5 pings is recorded for each measurement. For Microsoft, both IPv4 and IPv6 pings are performed every hour of the day. IPv4 pings are performed every 15 minutes to the Apple update URL. Table 1 summarizes our dataset.

Table 1: Summary of our data set

	Start Date	End Date	# Measurements
MSFT IPv4	August 1, 2015	August 31, 2018	225,012,410
MSFT IPv6	August 1, 2015	August 31, 2018	62,757,527
Apple IPv4	August 1, 2015	August 31, 2018	992, 988,166

Figure 1(a) shows the number of RIPE Atlas client prefixes (/24 granularity) that issue IPv4 ping measurements towards Microsoft's domain each day. The figure shows that the majority of probes are located in Europe as RIPE Atlas is known to have a bias towards this region. However, our analysis includes measurements from over 250 client prefixes in Africa, over 150 in South America and over 200 in Oceania. The number of clients increases across all continents over the course of the measurement period with an average of 8,060 client prefixes making measurements each day.

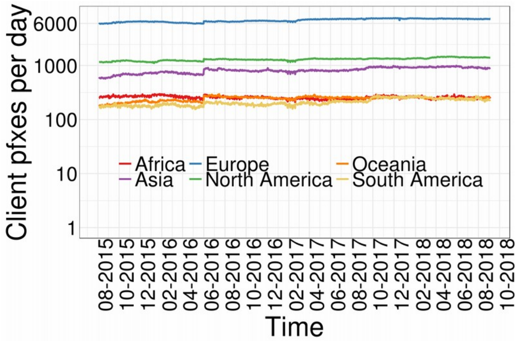
Figure 1(b) shows the number of unique Microsoft server prefixes that respond to our IPv4 *ping* measurements. We see an increase in server IPs observed over time due expansion in CDN infrastructure.

Data Normalization. Since the distribution of RIPE Atlas probes is skewed towards certain geographical regions (*e.g.,* Europe) and networks, we normalized the number of ping measurements analyzed from each AS in a given time window. This normalization can be done in several ways. We experiment with two normalization techniques: (1) sampling a fixed number of latency measurements from each network and (2) sampling pings in proportion to the fraction of all Internet users in that network. We obtain the number of users in a network from the APNIC Labs AS Population dataset [9] which estimates the subscriber could per AS. Both techniques yield similar content provider composition and median latency in our analysis and we present results of the normalization technique where sampling is done in proportion to the number of subscribers in a network. Since the number of users in a network can be a very small fraction of all subscribers on the Internet, we sampled at least 5 pings from each network in a measurement time window.

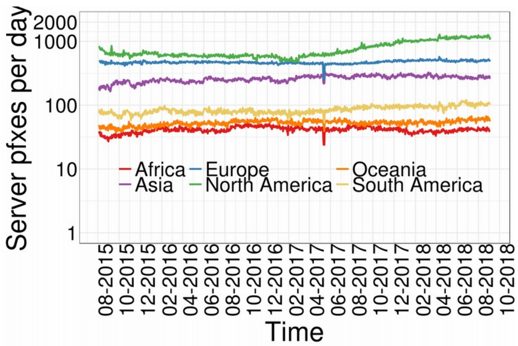
3.2 Identifying CDN instances

In each measurement, the software update URL is resolved using the *resolve on probe* option on RIPE Atlas vantage points. In this section, we describe our method for identifying the organization (content provider or CDN) the client is being referred to.

Identifying Content Providers. Content providers like Microsoft and Apple have multiple ASes in different countries. We refer to these ASes as the content provider's *family* of ASes. We identify content provider families using CAIDA's AS to organization mapping (AS2Org) [7]. We do a regular expression based search on the name field in AS2Org to find members in a content provider's family. Additionally, ASes with same organization IDs in AS2Org are considered to belong to the same organization. Using these techniques, we find 40 ASes corresponding to Microsoft and 11 ASes corresponding to Apple's family of networks.



(a)



(b)

Figure 1: Figure(a) shows the number of unique client prefixes (/24) that *ping* Microsoft's software update domain in our measurements, per day. Figure(b) shows the number of unique server prefixes (/24) that respond to the pings, per day.

Identifying CDN edge caches. Edge caches, deployed in ISP networks, serve an important role in bringing content closer to clients. However, when a client is referred to an edge cache, we will usually observe this as an IP address allocated to an ISP that is unrelated to the CDN. To determine which CDN an edge cache IP corresponds to we perform the following steps:

- We perform a reverse DNS lookup on the server IP address. We develop regular expressions to identify specific CDNs based on the returned hostname (*e.g.,* Akamai hostnames contain deploy.static.akamaitechnologies.com, Microsoft hostnames contain msedge.net).
- Not all server IPs will resolve to a hostname or may resolve to a hostname that does not correspond to a given CDN. In such cases, we use the WhatWeb [8] web scanner to find details about IP addresses. WhatWeb is a web scanner which scans IP addresses and domain names to fingerprint them. We use a set of regular expressions to identify instances of Akamai (fingerprint has the string *AkamaiGHost*) and Amazon (result includes the string *AWS*).

Using AS2Org, we identified 40% of all Microsoft's server addresses that belong to popular CDN families (like Microsoft, Apple, Akamai *etc.*). Regular expression search on hostnames derived from reverse DNS and WhatWeb identified another 12% of all Microsoft IPv4 server addresses observed in our measurements. Finally, not all WhatWeb and DNS responses provide enough information to identify a CDN instance but using a combination of reverse DNS, WhatWeb scanning and AS2Org mappings, we are able identify nearly all CDN edge cache instances, leaving about 0.1% of the ping destination IPs unidentified. For these unidentified IPs, all three of reverse DNS lookup, Whatweb scans and IP-to-AS conversion fails (*Other* category in Figure 2a).

Total CDNs observed. We find 1,059 ASes serving Microsoft's IPv4 clients, of which 930 contain edge caches. Similarly, we find that 449 ASes serve Microsoft's IPv6 clients, and 407 of these contain edge caches. Lastly, we find 879 ASes serve Apple's clients, and 752 of these ASes contain edge caches. Since Akamai's edge caches are located in a large number of countries, we group all Akamai edge caches into one bucket and refer to it as *Edge - Akamai* in our analysis.

3.3 Limitations

In this section we discuss the challenges we face in making inferences from the measurement study and describe ways in which we mitigate them.

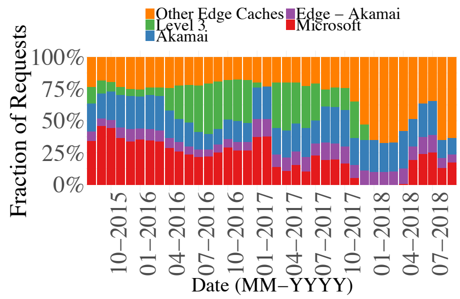
Skewed distribution of RIPE Atlas probes. Overall performance measurement from RIPE Atlas probes will be biased towards the performance observed by European clients, where the majority of RIPE probes are located. We analyze the performance on a per-continent basis to mitigate the geographical bias. A similar bias can occur when a single network hosting disproportionately large number of probes dominates the overall performance statistics. To mitigate this we normalize the ping packet counts from each network such that the contribution of pings from a network is proportional to the number of users in the network [9]. Thus, from the set of RTT measurements from a network in a given time window, we randomly sample pings in proportion to the fraction of users (or eyeballs) in that network.

Failed DNS resolutions and other errors. We observed ping and DNS resolution failures during measurements (2% for Microsoft IPv4, 10% for Microsoft IPv6 and 3% for Apple IPv4 measurements). We excluded these data points from the analyses. Additionally, some RIPE probes tend to be unreliable. Measuring from these probes makes it hard to reason about the longitudinal aspects of their performance. Therefore, we exclude probes with less than 90% availability from our analysis.

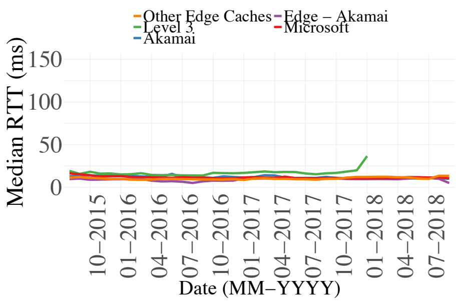
Latency as an approximation for performance. In this study, we measured the latency of accessing OS updates from various content providers. However, content providers often optimize other parameters of client-side performance like throughput and stability of the replica in terms of anycast routing [12].

4 CHARACTERIZING MULTI-CDNS

In this section, we describe the characteristics of multi-CDN deployments that serve content to Microsoft and Apple clients. We discuss the combination of CDN providers that serve OS updates to clients (§4.1) and compare the performance of these providers §4.2). We then analyze regional trends in the latency with a focus on clients in developing regions like Asia, Africa and South America §4.3).

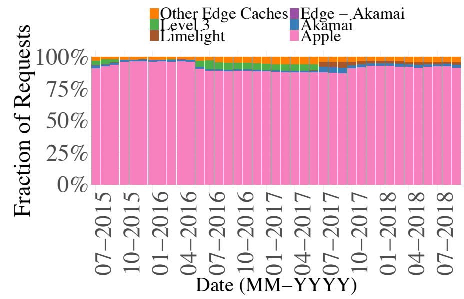


(a) CDNs providing Microsoft's OS updates over IPv4.

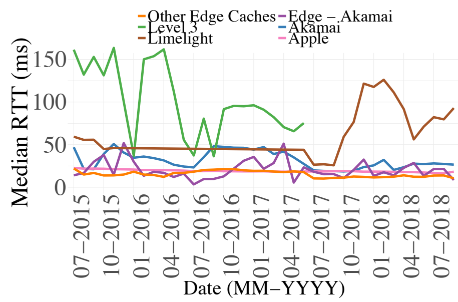


(b) Microsoft IPv4 RTT

Figure 2: CDN usage and Median RTTs by CDNs for Microsoft over IPv4.

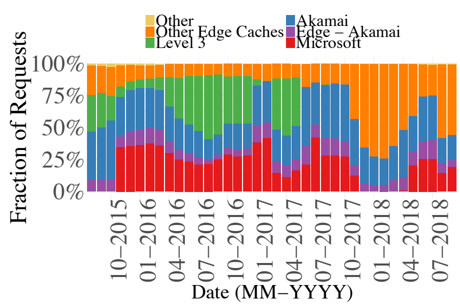


(a) Apple IPv4 Usage

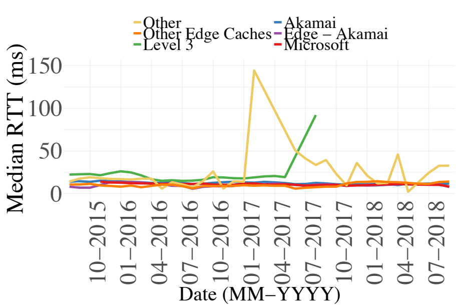


(b) Apple IPv4 RTT

Figure 4: CDN usage and Median RTTs by CDNs for Apple.



(a) CDNs providing Microsoft's OS updates over IPv6.



(b) Microsoft IPv6 RTT

Figure 3: CDN usage and Median RTTs by CDNs for Microsoft over IPv6.

4.1 Multi-CDN Mixture

In Figure 2a, we show the fraction of clients that receive Windows updates over IPv4 from different CDNs over time. We observe that almost all Microsoft IPv4 clients receive software updates from one of 4 providers: Microsoft's own network (registered in the USA), Akamai's network (registered in Europe), Level 3 (a tier-1 network that also provides CDN services), and various Akamai and non-Akamai edge caches. In mid-late 2015, Microsoft's network was directly responsible for serving roughly 45% of the clients, globally. Since then, this percentage has steadily declined to only 11% in April 2017. On the other hand, the percentage of clients served by Level 3 steadily increases until February 2017 when Level 3 served a negligible number of Microsoft IPv4 clients. We discuss the performance impact of the migration away from Level 3 (until the year 2017) in detail (§6). We observe that in August 2017, roughly 40% Microsoft IPv4 clients are served by edge caches (including Akamai's). Since the end of 2017 there has been an increase in non-Akamai edge caches serving Microsoft's OS updates where 70% Microsoft clients in August 2018 receive content from edge caches (Figure 2a). Figure 3a shows the CDNs that serve content to Microsoft's IPv6 clients. Until November 2015, Microsoft's network did not support IPv6. Since then, we see a similar CDN mixture for IPv6 clients as seen for Microsoft's IPv4 clients.

Apple's CDN strategy (see Figure 4) shows a sharp contrast from Microsoft's. Unlike Microsoft, over 85% of Apple's clients get content directly from Apple's own networks. Only 10% to 15% of clients retrieve Apple's software updates from other CDN providers (*e.g.,* Limelight, Akamai, Level 3).

4.2 Performance of CDNs

Figure 2b and 3b show the distribution of median RTT values for Microsoft IPv4 and IPv6 clients respectively. The median RTT observed by IPv4 clients is ↑ 20ms with similar values observed for IPv6 clients (notable exceptions are IPv6 clients served by Level 3).

For both Microsoft (IPv4 and IPv6) and Apple (Figure 4b), edge caches provide the least latency access to their clients with median RTT values in between 10 and 25 milliseconds. As discussed in Section 4.1, Microsoft relies heavily on Akamai's edge caches to distribute content but Apple does not follow this strategy. However, edge caches provide low latency access to Apple's content (Figure 4b), suggesting the usefulness of larger edge cache deployments.

4.3 Regional trends in performance

In Figure 5, we show the median RTT observed by clients in different continents. We see that North American and European clients observe stable low latencies near or below 20 ms for Microsoft's content (Figure 5(a) and (b)). However, other continents go through periods of high variability in latency. For both IPv4 and IPv6, African clients, we observe a general downward trend in median RTT, showing that some progress has been made in improving performance in developing regions; but they still observe worse median RTTs (nearly 50 ms on average) than their North American/European counterparts.

Compared to Microsoft, we observe Apple's clients in Africa and South America receive much worse latency (over 100 ms higher) as seen in Figure 5(c). We attribute this to the lack of Apple edge caches in developing regions (figure omitted for space). Interestingly, in July 2017, we observe a sharp drop in median latency of clients in Africa and South America. We find the bulk of the clients that observe this decrease in latency are shifting from other CDNs to Limelight, another popular CDN provider.

We observe that Level 3 has high latency to both Microsoft and Apple clients. We investigate regions where Level 3's latency is high and find that while Level 3 has short RTTs (⇡ 20ms) to North American clients, clients elsewhere observe high latencies when receiving content from Level 3. For instance, roughly 17% of clients in Africa receive Microsoft's updates from Level 3, and these clients observe large RTT values (⇡ 168ms) in the duration of our study. In case of Apple, we find that roughly 75% of clients in Africa are served Apple updates from Level 3, which leads to overall high latency access in Africa (see Figure 5(c)). We perform a detailed analysis of the performance of Level 3 for content delivery in §6.1.

Since we observe more diversity in the CDNs employed by Microsoft (Figure 4a), we focus on analyzing Microsoft's IPv4 clients in the remaining sections.

5 STABILITY OF CDN ASSIGNMENTS

We now analyze the stability of the mapping between clients and CDN server prefixes. Identifying co-located CDN servers is a hard problem [10], one we do not focus on in this work. Instead, we analyze the mapping of client prefixes to CDN server prefixes with the goal of understanding the relationship between stable mappings and latencies. We quantify stability using two metrics: (1) *prevalence* as defined in previous work [11] and (2) the average number of CDN prefixes observed by clients in a given day.

Prevalence of CDN server prefixes. *Prevalence* of client to server mapping captures how often a client receives content from a given CDN prefix (/24). For each client, we calculate the probability of receiving content from a given prefix in one day (*i.e.,* a prevalence of 1 means that the client is always referred to the same prefix). In Figure 6(a), we show the mean prevalence of the dominant server (the server that responds to the client most often) for Microsoft's IPv4 clients over time. Higher prevalence implies high stability in the client to server mapping. We observe a general decreasing trend in server prevalence for all continents, specifically in case of North America.

CDN prefixes seen in a day. Next, we analyze the number of CDN server prefixes observed by each client over the course of a day. A lower number of prefixes per day would imply more stable mapping. Figure 6(b) shows the mean number of CDN prefixes clients observe each day, over time. We observe a general increasing trend, showing that clients are becoming more likely to receive content from a diverse set of servers. This trend is more pronounced for North American clients.

Stability and latency. We analyze the relationship between the stability of server mappings and client-side latency using linear regression. We note that we are using latency as a proxy for performance as we do not directly measure throughput of the software downloads. Earlier work [12] has found that CDNs trade-off latency by choosing a subset of edge servers for stable anycast routing. Figure 7 shows the linear regression trend between mean RTTs and stability of server mappings for clients in developing regions. We observe that lower RTTs correlate with more stable (high prevalence) server mappings.

6 IMPACT OF CDN MIGRATION

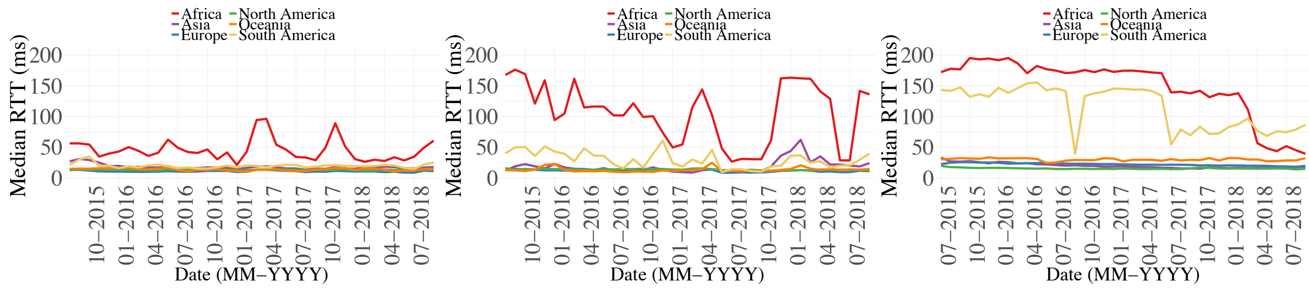
Multi-CDN deployments give content providers options as to which CDN a given client will be referred to. In this section, we investigate how different CDN selection strategies impact client latency. We investigate the impact of migrating to/away from Level 3 (Section 6.1) in developing regions, and how edge caches improve performance for clients (Section 6.2).

6.1 Content Delivery from Level 3

In Section 4, we observe fluctuations in latencies for clients that are being served from Level 3. We now investigate the impact of migrating towards/away from Level 3 on a per-client basis. First, we analyze cases of client migration from Level 3 to any other CDN provider. We find that migration away from Level 3 improves latency for clients in developing regions and Oceania likely due to a dearth of Level 3 servers in these regions. For instance, migration *away* from Level 3 leads to improved RTTs 83%, 75% and 71% of the time for clients in Oceania, Asia and South America, respectively. Figure 8 shows the distribution of the change in RTT ($\frac{old\,RTT}{newRTT}$) when the *same* client migrates towards or away from Level 3. Here, a value of > 1 indicates a decrease in RTT and a value of < 1 indicates that the new RTT was higher than the prior one. Clients located in the developing world can obtain low latency access if they are migrated away from Level 3 but this migration does not significantly impact clients in the developed world.

6.2 Content delivery from edge caches

Second, we analyze cases where Microsoft IPv4 clients migrate from non-edge caches (*i.e.,* servers in Level 3, Microsoft, Akamai's networks) to edge caches of different types (including Akamai's



(a) Microsoft IPv4 (b) Microsoft IPv6 (c) Apple

Figure 5: Median RTTs by continents for Microsoft (IPv4 and IPv6) and Apple.

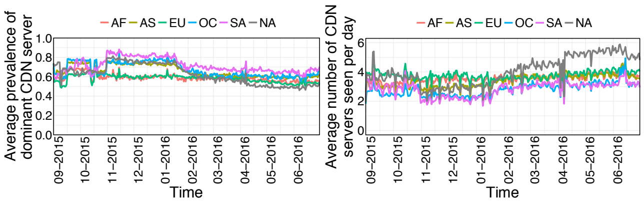


Figure 6: (a) Average probability of seeing the dominant CDN server prefix (/24) by a client in a day (prevalence). (b) Average number of CDN server prefixes seen by each client in a day.

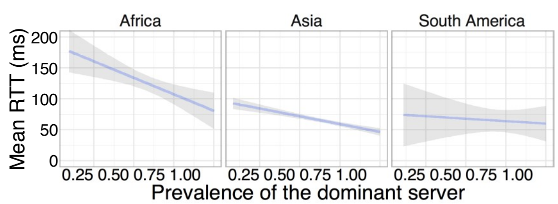


Figure 7: Relationship between CDN server prevalence and RTT for developing regions.

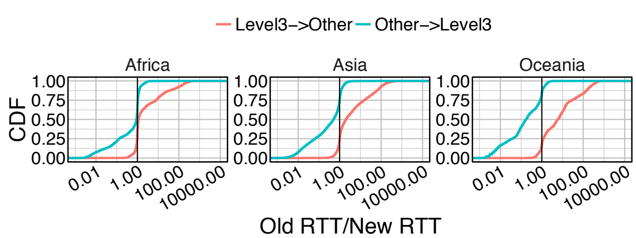


Figure 8: Change in latency observed by clients migrating away and from Level 3.

edge caches and local ISP caches). We find that in 73% of cases, migration towards edge caches leads to improved RTTs for African clients. Migration towards edge caches also leads to improvements in RTT for clients in Oceania and Asia in 76% and 64% of cases, respectively. Additionally, clients suffering with high RTTs benefit the most from migrating to edge caches. Figure 9 shows the change in RTT observed by African clients who had high RTTs (greater than 200 ms) before the migration occurred. Clients that migrated towards edge caches (Other! EC) in 2017 saw an average improvement of 10X to 100X in RTTs. This observation is in agreement with

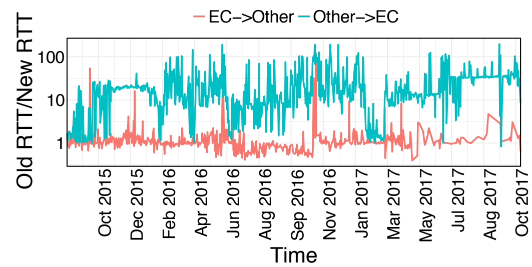


Figure 9: Change in RTT of African clients when they migrate from towards edge caches (Other! EC) and when they migrate away from edge caches (EC! Other). In 2017, the average improvement in RTT when clients migrated towards edge caches was between 10 and 100X.

the intuition that moving content closer to clients, especially in developing regions, improves client performance.

7 CONCLUSION

We perform a longitudinal analysis of distribution and performance of CDNs used by two large content providers, Microsoft and Apple. We find differences in their strategies of pushing software updates to clients. Microsoft makes extensive use of edge caches to put content closer to customers while most Apple clients from across the world receive content from Apple's own networks. As we dig into the performance received by Microsoft's clients in developing regions, we observe that while there is a downward trend in latency, there is room for improvement. Our analysis shows that clients migrating towards edge cache deployments for receiving content observe major improvements in performance. However, non-North American clients experience high latency when served content from the large ISP Level 3. This is especially important for continents such as Asia and Africa, where a significant number of clients receive Microsoft's updates from Level 3.

ACKNOWLEDGEMENTS

We would like to thank Emile Aben, Sharad Agarwal, and David Jensen for their feedback on this work. We also thank our shepherd Philipp Richter and the anonymous IMC reviewers for their feedback. This work was funded by NSF Grants: CNS 1700657, CNS-1651784, CNS-1719386, CNS-1740895, and a Google Faculty Research Award. Rachee Singh is funded by a Google PhD fellowship.

REFERENCES

[1] Erik Nygren, Ramesh K. Sitaraman, and Jennifer Sun. The Akamai Network: A Platform for High-performance Internet Applications. *SIGOPS Oper. Syst. Rev.*, 44(3):2–19, August 2010.

[2] Noe Garcia. A Quick Guide to Multi-CDNs: What to Consider, What Kind to Deploy. https://www.maxcdn.com/blog/multi-cdns/, 2017. Online; accessed June 2017.

[3] Emile Aben. RIPE 71, Measuring Multi-CDNs. https://ripe71.ripe.net/presentations/67-2015-11.multicdn.emileaben.pdf, 2015. Online; accessed May 2017.

[4] Google Public DNS. https://developers.google.com/speed/public-dns/.

[5] Fangfei Chen, Ramesh K. Sitaraman, and Marcelo Torres. End-User Mapping: Next Generation Request Routing for Content Delivery. In *Proceedings of the 2015 ACM Conference on Special Interest Group on Data Communication*, SIGCOMM '15, pages 167–181, New York, NY, USA, 2015. ACM.

[6] Matt Calder, Ashley Flavel, Ethan Katz-Bassett, Ratul Mahajan, and Jitendra Padhye. Analyzing the Performance of an Anycast CDN. In *Proceedings of the 2015 ACM Conference on Internet Measurement Conference*, IMC '15, pages 531–537, New York, NY, USA, 2015. ACM.

[7] CAIDA. Inferred AS to Organization Mapping Dataset. http://www.caida.org/data/as-organizations/, 2017. Online; accessed June 2017.

[8] Andrew Horton. WhatWeb - Next Generation Web Scanner.https://github.com/urbanadventurer/WhatWeb, 2017. Online; accessed May 2017.

[9] Visible ASNs: Customer Populations. http://http://stats.labs.apnic.net/aspop. Online; accessed September 2018.

[10] Xun Fan, Ethan Katz-Bassett, and John Heidemann. Assessing affinity between users and cdn sites. In *International Workshop on Traffic Monitoring and Analysis*, pages 95–110. Springer, 2015.

[11] Vern Paxson. End-to-End Routing Behavior in the Internet. *ACM SIGCOMM Computer Communication Review*, 26(4):25–38, 1996.

[12] Matt Calder, Ryan Gao, Manuel Schröder, Ryan Stewart, Jitendra Padhye, Ratul Mahajan, Ganesh Ananthanarayanan, and Ethan Katz-Bassett. Odin: Microsoft's scalable fault-tolerant CDN measurement system. In *15th USENIX Symposium on Networked Systems Design and Implementation (NSDI 18)*, pages 501–517, Renton, WA, 2018. USENIX Association.

[13] Rupa Krishnan, Harsha V. Madhyastha, Sridhar Srinivasan, Sushant Jain, Arvind Krishnamurthy, Thomas Anderson, and Jie Gao. Moving Beyond End-to-End Path Information to Optimize CDN Performance. In *ACM IMC*, 2009.