

Dissecting Apple's Meta-CDN during an iOS Update

Jeremias Blendin*

Technische Universität Darmstadt /
DE-CIX

Fabrice Bendfeldt

Technische Universität Darmstadt

Ingmar Poesse*

BENOCS

Boris Koldehofe

Technische Universität Darmstadt

Oliver Hohlfeld*

RWTH Aachen University

ABSTRACT

Content delivery networks (CDN) contribute more than 50% of today's Internet traffic. Meta-CDNs, an evolution of centrally controlled CDNs, promise increased flexibility by multihoming content. So far, efforts to understand the characteristics of Meta-CDNs focus mainly on third-party Meta-CDN services. A common, but unexplored, use case for Meta-CDNs is to use the CDNs mapping infrastructure to form self-operated Meta-CDNs integrating third-party CDNs. These CDNs assist in the build-up phase of a CDN's infrastructure or mitigate capacity shortages by offloading traffic. This paper investigates the Apple CDN as a prominent example of self-operated Meta-CDNs. We describe the involved CDNs, the request-mapping mechanism, and show the cache locations of the Apple CDN using measurements of more than 800 RIPE Atlas probes worldwide. We further measure its load-sharing behavior by observing a major iOS update in Sep. 2017, a significant event potentially reaching up to an estimated 1 billion iOS devices. Furthermore, by analyzing data from a European Eyeball ISP, we quantify third-party traffic offloading effects and find third-party CDNs increase their traffic by 438% while saturating seemingly unrelated links.

ACM Reference Format:

Jeremias Blendin, Fabrice Bendfeldt, Ingmar Poesse, Boris Koldehofe, and Oliver Hohlfeld. 2018. Dissecting Apple's Meta-CDN during an iOS Update. 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/10.1145/3278532.3278567>

1 INTRODUCTION

Content delivery networks (CDNs) have become a key component of the Internet [5, 10]. In order to reduce latencies and increase the availability of content for its consumers they serve content from nearby servers, which has flattened the hierarchical structure of the Internet [19]. CDNs achieve this by providing three major functions: geographically distributed content caches, direct connections to ISPs or IXPs, and a request mapping to select the best cache location.

*Corresponding authors:

Jeremias Blendin: jblendin@kom.tu-darmstadt.de, Ingmar Poesse: ipoese@benocs.com, Oliver Hohlfeld: oliver@comsys.rwth-aachen.de

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from [permissions@acm.org](https://permissions.acm.org).

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

© 2018 Copyright held by the owner/author(s). Publication rights licensed to ACM.

ACM ISBN 978-1-4503-5619-0/18/10...\$15.00

<https://doi.org/10.1145/3278532.3278567>

Unsurprisingly, CDNs cause high traffic shares: e.g., more than half of the traffic of a North American [16] and a European [26] Internet service provider (ISP) can be attributed to only a few CDNs. Despite these characteristics, customers of a single CDN are bound to its cost model, performance, and geographic distribution—a limitation solved by publishing content on *multiple* CDNs, which requires an additional request mapping for CDN selection.

We refer to CDN selecting infrastructures as Meta-CDN services. Known instances of third-party Meta-CDNs [15, 22, 23, 30] provide services to implement custom and dynamic request mapping policies to direct traffic to the different CDNs. The characteristics of third-party Meta-CDNs have been studied in part on the example of Conviva [2, 13, 22] and Cedexis [17] as the prevalent operators.

As an alternative to using third-party Meta-CDN providers, large content providers often prefer to build up their own infrastructure and only depend on third-party CDNs when necessary. This approach leads to a hybrid model where a content provider uses third-party CDNs to supplement its own infrastructure, e.g., to handle overload. In this model, the content provider effectively becomes a *self-operated Meta-CDN* by directing traffic either to its own infrastructure or to third-party CDNs. However, little is known about this type of Meta-CDN and its implications.

In this paper, we shed light on self-operated Meta-CDNs using a detailed analysis of the Apple Meta-CDN as a prominent example. It is used to deliver Apple software updates for mobile devices, i.e., iOS updates for up to 1 billion devices including the iPhone, iPad, and iPod [18]. The rollout of major updates only happens a few times a year but for all devices at the same time, which creates substantial traffic demands. In this study, we show how the Apple Meta-CDN handles these update events by examining a major iOS update. From this, we discover that the Apple Meta-CDN indeed operates a Meta-CDN service by involving third-party CDNs in addition to their own CDN infrastructure. We further demonstrate the *consequences* of this update event and particularly the Meta-CDN service *on ISP traffic* by analyzing detailed traffic traces from a European Eyeball ISP. To reason about the effects of the Meta-CDN service, we correlate the RIPE measurement with the ISP dataset. Our contributions are as follows:

- (1) We provide the first characterization of a self-operated Meta-CDN. We describe the involved CDNs, the request-mapping mechanism, and discover the cache locations of the Apple CDN using more than 800 RIPE Atlas [27] probes worldwide.
- (2) We observe its request mapping and load sharing behavior during a major iOS update in Sept. 2017.
- (3) We provide the first study on the impact of a Meta-CDN service on ISP traffic. In this study, we cross-correlate RIPE Atlas DNS measurements with ISP traffic data to quantify

the effect of offload and overflow traffic for the ISPs with regards to the Apple iOS update. We find third-party CDN traffic spikes to reach 438%, that the distribution of offload traffic is dynamic on a daily basis and that, due to overflow, seemingly unrelated links suddenly saturate.

2 BACKGROUND & RELATED WORK

Building on a geographically distributed infrastructure, CDNs [5, 10] enable high availability and low latency when content is retrieved from close-by servers. Multiple studies contribute towards an understanding of these infrastructures, including their performance [5, 10, 24, 25], and their request mapping mechanisms [11, 28]. Further optimizations of request mapping mechanisms are proposed, e.g., based on anycast [11, 14], by enabling ISP-CDN collaboration [26], or both [29]. Using multiple CDNs can offer cost reductions (e.g., due to different prices to serve content at different times, or due to traffic volume contracts) and higher availability (e.g., in high-load scenarios). This case of *multi-homed* content—i.e., content that is served by multiple CDNs—requires an additional request mapping layer to route requests to CDNs selected for delivery [15, 20, 30].

A Meta-CDN service can be provided by third-party infrastructures, most prominently Conviva for video delivery [2, 13] and Cedexis for general (web) services [1, 17, 30]. Within these Meta-CDNs, the request mapping function of a participating content provider is unknown. Hence, the demand and corresponding traffic are harder to predict for Eyeball ISPs. This lack of information can cause unexpected traffic skews as well as significant variance in demand over short timescales [22]. Understanding the request mapping potential and its consequences on traffic—as addressed by our study—is thereby needed to understand Meta-CDNs and their consequences on traffic better.

3 DISSECTING THE APPLE CDN

We start our analysis by dissecting the operating principle of the Apple Meta-CDN and will show *i)* how iOS devices discover and download updates, *ii)* how download requests are mapped to CDNs, and *iii)* the architecture of Apple’s content cache infrastructure.

3.1 iOS Device Behavior

We examine the iOS update discovery and download behavior by analyzing traffic from an Apple TV and an iPhone 7 Plus device. We found that iOS devices download two manifest files from *mesu.apple.com* once per hour to check for available updates. The first file [7], termed manifest, contains the version and download URL for every device and OS version combination with about 1800 entries as of July 2017, and the second file [8] contains only six entries. We did not observe the second file being used in the collected data, and therefore, assume it to be a last-resort mechanism that enables devices with outdated iOS software to upgrade their software. If the manifest contains information on a new software update, the user is notified on its availability. When the user manually initiates the update process, the update file is downloaded from *appldnld.apple.com* via HTTP.

3.2 CDN Selection Process

The Apple iOS update CDN is a Meta-CDN that relies on a mixture of its own infrastructure and infrastructure from other CDN providers for both the request mapping and the file delivery. The request mapping in Apple’s self-operated Meta-CDN uses location-based dynamic DNS resolution [6, 28].

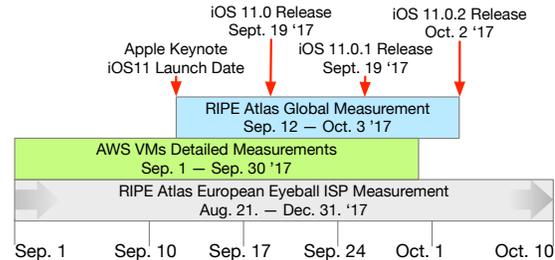


Figure 1: Active measurement timeline.

Measurement Setup. To account for the location-dependence in the request mapping, we queried the Apple Meta-CDN from globally distributed RIPE Atlas [27] probes and Amazon AWS VMs. We show the measurement period in Figure 1. Full recursive DNS resolution measurements and checking the availability of the relevant files on the Apple CDN servers was done on nine AWS VMs distributed over all continents except Africa. To understand the global CDN request mapping behavior in detail, we monitored the Apple Meta-CDN from 800 globally distributed RIPE Atlas probes issuing DNS requests and traceroute probes. In contrast to the AWS VMs, these probes only collect the DNS reply data. These Atlas probes issued DNS requests every 5 minutes during one week before and after the iOS 11 update released on Sep. 19, 2017; the data is publicly available [9]. We perform traceroutes to all server IPs identified via DNS every hour. Finally, to understand the specific behavior of the Apple Meta-CDN from the viewpoint of the European Eyeball ISP studied in Section 5, an additional 400 Atlas probes are used, dedicated to measuring inside the ISP every 12 hours between Aug. 20 and Dec. 31, 2017. The measurement approach is designed to capture the diversity of DNS request mappings using the RIPE Atlas probes and the DNS mapping infrastructure as well as the availability of the content using AWS VMs. The approach is generic, which means it could be applied to any other CDN.

Apple CDN Request Mapping Infrastructure. In Section 3.1, we observed *appldnld.apple.com* to be the entry point for downloading update images on iOS devices. By using our measurements, we dissect the involved request mapping infrastructure, as shown in Figure 2. The parts depicted in an orange checker pattern are modifications observed during the rollout of the iOS 11.0 (see Section 4). The depiction contains the parts of the request mapping process where decisions are made. Each box, except for ⑤, represents a DNS name. Each arrow represents a CNAME redirect with the time-to-live (TTL) in seconds of the following DNS name—which we found to be stable throughout our measurements. The part named edge site ⑤ illustrates the cache server structure of Apple’s own CDN and is discussed in Section 3.3.

The first step ① uses an Akamai DNS service to differentiate whether the request originates from India or China, or from other

scheme. Their IPs are distributed via request mapping system to clients, in the depiction their labels denote $\langle \# \text{ of sites} \rangle / \langle \text{total} \# \text{ of cache servers} \rangle$. One desired property of a CDN is to have a globally distributed set of edge servers to serve nearby clients.

Table 1: Apple server naming scheme

Naming Scheme: ab-c-d-e.aaplimg.com
Example: usnyc3-vip-bx-008.aaplimg.com

Identifier	Meaning
a	UN/LOCODE location (e.g., deber for Berlin)
b	Location site id (e.g., 1)
c	Function: vip, edge, gslb, dns, ntp and tool
d	A secondary function identifier: bx, lx and sx
e	Id for same function server (e.g., 004)

Interestingly, the *internal structure* of *edge sites* can be revealed by analyzing *HTTP* header information used during downloads; an example of the relevant part of the header is shown below:

```
X-Cache: miss, hit-fresh, Hit from cloudfront
Via: 1.1 2db316290386960b489a2a16c0a63643.cloudfront.net (CloudFront),
http/1.1 defra1-edge-lx-011.ts.apple.com (ApacheTrafficServer/7.0.0),
http/1.1 defra1-edge-bx-033.ts.apple.com (ApacheTrafficServer/7.0.0)
```

From that, we infer that client requests are directed to nodes with the *vip-bx* function that forwards requests to one of four associated nodes with the *edge-bx* function as denoted by ⑤ in Figure 2. If the file is not found there, the request is forwarded to a node with the *edge-lx* function. The term *vip* suggests “virtual IP” and that the associated server is a load balancer for four *edge-bx* nodes. The use of load balancers in the delivery sites instead of purely relying on DNS suggests that a single Apple CDN IP represents the download capacity of four servers. This is why the number of servers per location in Figure 3 refers to the number of *edge-bx* nodes.

Takeaway. *Besides using third-party CDNs, Apple has created a substantial cache infrastructure of CDN sites. Their density of sites is the highest in the USA followed by Europe and East Asia, while the South American and African continents lack distribution data centers.*

4 CHARACTERIZING THE APPLE CDN DURING AN IOS UPDATE ROLLOUT

To study the behavior of the Apple Meta-CDN during operation *and* under load, we measured the rollout of iOS 11, a major iOS software update released by Apple on Sep. 19, 2017 at 17h UTC. For this perspective, we use the RIPE Atlas DNS measurements described in Section 3.2, which started on Sep. 12–7 days before the update was made available to iOS users (see Figure 1).

To provide a global perspective of the changes in the CDN infrastructure during the update, we show the involved CDNs and their number of unique server IPs per continent in Figure 4 as seen in responses to our DNS queries. Cache IPs that are used by Akamai or Limelight but not located within their respective autonomous systems (ASs) are denoted as “other AS”. The focus of this discussion is the changes in the Meta-CDN service, specifically the choices of the Apple Meta-CDN operators when it comes to selecting third-party CDNs. As discussed in Section 3.2, we assume commercial interests to be the driving factor for Apple’s Meta-CDN service design. Information on the actual traffic caused by the Apple Meta-CDN during the event is not available globally but for a single European Eyeball ISP. The results of which are discussed in Section 5.

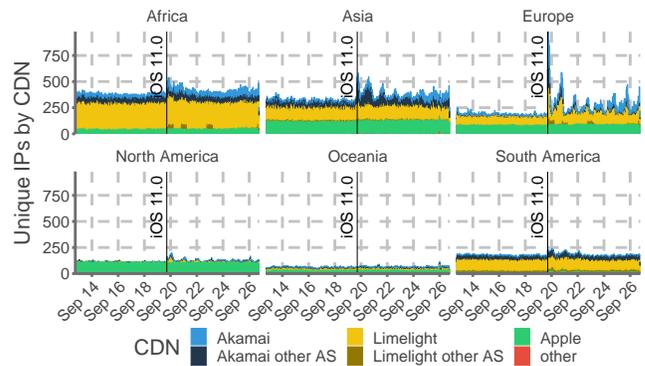


Figure 4: Number of unique CDN cache IPs of the worldwide measurement.

Global perspective. As expected from the distribution of Apple CDN sites, North America features the highest ratio of Apple cache IPs, while South America and Africa show the highest ratio of third-party CDN IPs. Interestingly, despite the high number of Apple CDN sites in Europe, ~50% of the observed IPs belong to third-party CDNs. Furthermore, Europe is the only continent that shows a considerable spike in the number of unique IPs following the update event. We observe a maximum of 977 IPs immediately after the iOS software release on Sep. 19 at 18h UTC. This peak is more than four times the average of 191 of unique cache IPs that were observed in the two days before. The major part of the increase in unique IPs is caused by Limelight and, to a lesser part, Akamai, with the latter increasing the number of cache IPs that are located in third-party networks. Akamai’s increased load correlates with the appearance of a new CNAME: six hours after the update started on Sep. 19 around 23h, *a1015.gi3.akamai.net* was added in the Akamai CDN for requests coming from *ios8-eu-lb.apple.com.akadns.net*. We assume the reason for this behavior, in comparison to North America, is twofold. First, Apple’s infrastructure is not as built out yet in Europe. Second, according to estimates by the APNIC [4], the ISP market in the USA is more consolidated with roughly 60% market share for the ten largest ISPs, while in Europe the ten largest ISPs only have a market share of about 30%. The higher market fragmentation in Europe requires a more complex CDN deployment.

The new delivery path is depicted in an orange checker pattern in Figure 2. This observation shows that the Apple Meta-CDN handles the increased traffic demand of the update by delegating traffic to third-party CDNs using their Meta-CDN service. We did not observe any proactive changes to Apple’s request mapping infrastructure before the release of the update.

European ISP. We show the same measurement performed by 400 RIPE Atlas probes located in the network of the European Eyeball ISP studied in Section 5 in Figure 5. Here, the load increase by the update is not distributed equally among all CDNs, and the spike in the number of IPs is not as pronounced as in the whole of Europe. Still, we observed an increasing number of IPs for other CDNs, most notably the number of Akamai CDN IPs rise by 408% from Sep. 18 to Sep. 20. We find Apple’s CDN to have a somewhat stable number of IPs that are not increased, e.g., suggesting that

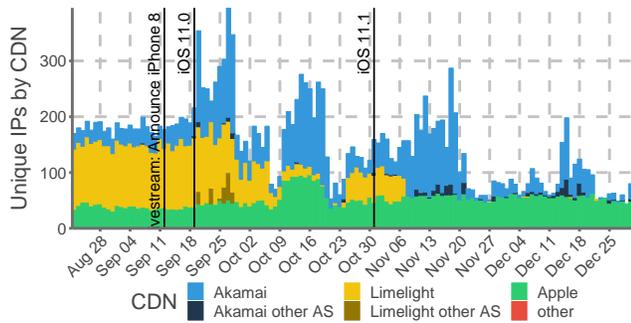


Figure 5: Number of unique CDN cache IPs of the European Eyeball ISP measurement.

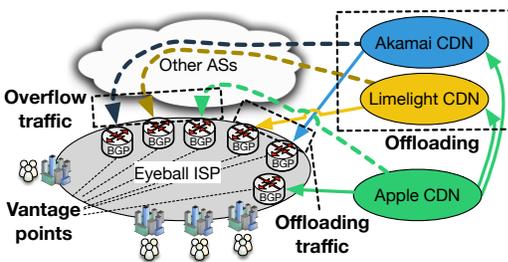


Figure 6: Illustration of CDN offloading and traffic overflow from perspective of an ISP.

Apple’s CDN cannot further increase the number of download cache locations. The steady number of Apple IPs also indicates that the demand could not be satisfied with their infrastructure alone and thus needed to be delegated to third-party CDNs—most notably Akamai. This assumption is supported by the fact that it e.g., takes six hours for Akamai to increase its number of distributed IP addresses to its load-dependent peak. The ISP is thus facing the challenge of the update to cause traffic spikes by third-party CDNs—solely because of Apple’s Meta-CDN service. This motivates us to study consequences for the ISP in the next section.

Takeaway. *The rollout of a major software update causes high traffic demands that are handled by offloading requests to third-party CDNs using the Meta-CDN service. During the update, no party has full control over the entire infrastructure while user assignment and server selection is made by multiple CDNs independently.*

5 ISP PERSPECTIVE ON A META-CDN

The previous section showed that the Apple Meta-CDN handles peak traffic events by involving third-party CDNs. This process, known as *offloading*, is controlled by the Apple operated Meta-CDN service and has effects on ISP traffic engineering and their peering links. These effects motivate us to look at a Meta-CDN service from the perspective of a Tier-1 European Eyeball ISP.

5.1 Definition of offload and overflow Traffic

For ease of presentation, we define the traffic sources as follows.

- **Source AS** is the AS that originates the traffic of a connection, i.e., the AS of the servers’ IP address. For example, the *Source*

AS for an Apple update can be Akamai if the request was delegated there by the Apple Meta-CDN service.

- **Handover AS** is the direct neighbor AS handing traffic to the measured ISP network. This AS can be completely unrelated to any used CDN (e.g., a transit AS).

The terms *offload* and *overflow* are defined as follows.

- **Offload** describes traffic that the Apple Meta-CDN delivers via third-party *CDN servers*, i.e., the third-party CDN is the *Source AS*. In Figure 6, all traffic origination from Akamai and Limelight to the Eyeball ISP is *offload* traffic, as Apple hands that traffic over to third-party CDNs.
- **Overflow** describes traffic received from non-direct neighbors, i.e., the *Source AS* and handover AS differ. In Figure 6 this is all traffic received via “*Other ASes*”.

Note that *offload* and *overflow* traffic are orthogonal. For example, Akamai and Limelight traffic going via *Other ASes* is both, *offload* and *overflow* traffic. Apple traffic going via *Other ASes* is *overflow* traffic only.

Consequences for ISPs. Both traffic types, *offload* and *overflow*, pose a significant strain on the ISPs network. For example, *offload* makes it harder for ISPs to predict how much traffic to expect on what links. This is because it is unclear *i)* which CDNs are selected by the Meta-CDN and *ii)* which CDN is serving how much traffic. Overflow further challenges traffic flow prediction, since seemingly unrelated peering links change their traffic volume. And finally, *offload* is controlled by the Meta-CDN operator while *overflow* is handled by the individual CDNs’s load balancers. As these are two independent control loops, it becomes tough to predict how the traffic is going to behave.

5.2 ISP Measurement Setup

To quantify the effect of offloading and *overflow*, we gather BGP, Netflow and SNMP data directly on all border routers of a Tier-1 European Eyeball ISP (see vantage points in Figure 6) between Sep. 15 and Sep. 23. In total, we collect ~300 billion Netflow records as well as ~350 Million SNMP measurements while actively keeping track of ~60 million BGP routes in ~300 active sessions. We also get information about all active peering links and their respective *Handover AS*. Furthermore, we verify that that internal cache links are handled as direct connections to the CDN controlling the cache.

5.3 Offload Impact

Offloading happens when traffic from Apple is served via a third-party CDN. To quantify *offloading*, we select all CDN server IPs observed in RIPE Atlas DNS measurements to the Apple Meta-CDN located within the ISP (see Section 4) and cross-correlate them with Netflow for traffic flows and BGP for finding the *Source AS*. Finally, we scale the Netflow traffic on the peering links by the byte counters from SNMP to minimize Netflow sampling errors. Thereby, we estimate the traffic caused by Apple and its *offload* CDNs due to the iOS update.

Figure 7 shows the development of *offload* traffic as its *ratio* relative to the average peak traffic three days before the update for each CDN. Here, a ratio of 100% reflects the maximum traffic rate seen for a CDN over the course of three days before the update. A ratio of > 100% indicates traffic spikes, which we are interested

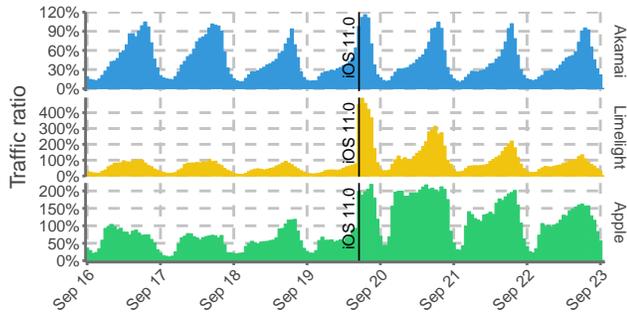


Figure 7: Update Traffic by Source AS during the iOS update.

in studying. We observe the maximum traffic from the Apple AS to peak at 211%, from Limelight at 438% and Akamai at 113% on Sep. 19. When comparing the excess volume of extra traffic caused by the iOS update between the CDNs, we find that for, Sep. 19, 33% come from Apple, 44% from Limelight and 23% from Akamai. For Sep. 20 and 21 the bulk of the additional traffic is transported by Apple itself (~60%) and Limelight (~40%) with no additional Akamai traffic during this time.

With Akamai being the biggest CDN traffic-wise, the assumption of Akamai carrying the bulk of the *offload* traffic could be assumed. However, this is not the case, as Akamai is only used on the Sep. 19, while all other days are shared between Apple and Limelight only. When looking at the traffic of Apple and Limelight, it also becomes visible that Apple runs at high capacity all of Sep. 20, while the other CDNs show a diurnal traffic pattern. This leads to the conclusion that Apple uses its own CDN first before *offloading*.

Takeaway. *The observed traffic patterns show that knowledge about how CDNs possibly deliver the update says little about the distribution of how the traffic is split between them. Predictions become even harder when the traffic amounts are not stable and change on a daily basis.*

5.4 Overflow Impact

Finally, we turn our attention to *overflow*. This phenomenon is particularly hard for ISPs to handle as it is extremely difficult to predict this traffic. We extend the *overflow* analysis by enhancing the traffic flows with handover AS information by leveraging SNMP for handover AS classification and determining the capacity of the of the peering links. Figure 8 depicts the traffic delivered by Limelight via indirect links (i.e., via *Other* ASes as in Figure 6). Limelight carries a significant amount of update traffic and is, at the same time, an *offload* CDN. Note that we grouped ~40 smaller *Handover* AS into “other” as they do not change their behavior significantly.

We found the traffic to be stable regarding its *Handover* AS distribution in the days before and after the update. However, on Sep. 19, AS **A** spikes in *overflow* traffic. We assume that this is the pre-cache fill to distribute the content. From an ISP’s perspective, this tempts the assumption that AS **A** might carry a bulk of the update traffic. However, when the actual update is delivered, AS **D**—an AS not seen before—spikes to deliver more than 40% of the *overflow* traffic due to Limelight’s sudden use of servers in or behind AS **D**. This AS is connected to the ISP via four direct connections, two of which

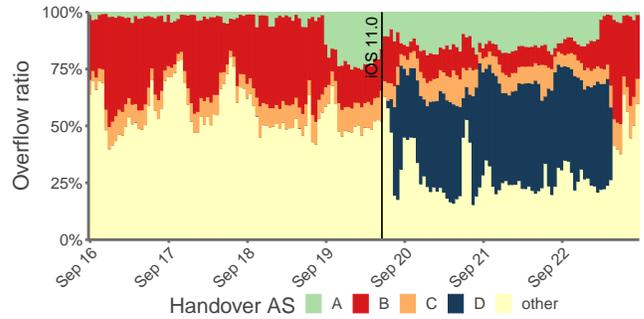


Figure 8: Overflow by Handover AS during the iOS update.

become entirely saturated at peak times. After three days, Limelight decides to no longer use these caches, and the normal traffic pattern returns. From the perspective of AS **D**, this could mean a multifold increase of their monthly bill, because the prevalent 95/5 billing [12] is affected by the traffic spike induced by Limelight.

Takeaway. *Due to offload and overflow, traffic predictions during high-stress situations are becoming volatile, which leads to unexpected traffic behaviors and overloads on unrelated peering links*

6 CONCLUSION

This paper assessed self-operated Meta-CDN deployments, exemplified by the first characterization of the Apple Meta-CDN. We shed light on Apple’s DNS-based request mapping infrastructure by tracing it from over 800 vantage points. We found Apple to use two major CDNs, Akamai and Lighthouse, as well as its own content delivery infrastructure. Our findings are two-fold: First, we detected 34 sites cache-sites, and second, we determined the internal structure of the cache sites by analyzing HTTP headers.

We performed detailed measurements of the Apple Meta-CDN’s behavior before, during, and after a flash crowd event caused by a major iOS software update in Sep. 2017. Our analysis shows that Apple handles the flash crowd on a per continent basis. We found that in Europe, Apple offloads traffic to third-party CDNs. For the analysis of our traces from a European Eyeball ISP we define two behaviors, *offload* and *overflow*. Through these, we show the difficulty of an ISP in predicting which CDN of the Meta-CDN will be delivering the content due to unknown offload strategies. Furthermore, we show that overflow traffic has a significant impact on unrelated peering connections, fully saturating links not expected to be affected. Since this traffic peak is unpredictable for an ISP, we argue that studying Meta-CDNs and their consequences on ISP traffic pose an interesting perspective for future work.

ACKNOWLEDGMENTS

This work has been funded by the German Research Foundation (DFG) as part of project B1 and C2 within the Collaborative Research Center (CRC) 1053 – MAKI. The authors would like to thank the RIPE Atlas team, Marc Werner, Matthias Hollick, and David Hausheer for their valuable support during the measurement campaign. We further thank the reviewers and our shepherd Philipp Richter for their valuable input and feedback.

REFERENCES

- [1] [n. d.]. Cedexis. <https://www.cedexis.com/>.
- [2] [n. d.]. Conviva. <https://www.conviva.com/>.
- [3] [n. d.]. UN/LOCODE Location Codes. http://www.unece.org/cefact/codesfortrade/codes_index.html.
- [4] [n. d.]. *Visible ASNs: Customer Populations (Est.)*. APNIC Labs. <https://stats.labs.apnic.net/aspop/> Accessed: 2018-9-27.
- [5] Vijay Kumar Adhikari, Yang Guo, Fang Hao, Volker Hilt, and Zhi-Li Zhang. 2012. A tale of three CDNs: An active measurement study of Hulu and its CDNs. In *INFOCOM Workshops*.
- [6] Vijay Kumar Adhikari, Sourabh Jain, Yingying Chen, and Zhi-Li Zhang. 2012. Vivisecting YouTube: An active measurement study. In *IEEE INFOCOM*.
- [7] Apple. [n. d.]. iOS SoftwareUpdate File URL. http://mesu.apple.com/assets/com_apple_MobileAsset_SoftwareUpdate/com_apple_MobileAsset_SoftwareUpdate.xml.
- [8] Apple. [n. d.]. iOS UpdateBrain File URL. http://mesu.apple.com/assets/com_apple_MobileAsset_MobileSoftwareUpdate_UpdateBrain/com_apple_MobileAsset_MobileSoftwareUpdate_UpdateBrain.xml.
- [9] Fabrice Bendfeldt, Jeremias Blendin, Ingmar Poese, Boris Koldehofe, Oliver Hohlfeld, and Ripe Atlas. 2017. RIPE Atlas Measurement #9299652: Apple iOS 11 Release Day DNS Resolution Measurements of appldnld.apple.com. <https://atlas.ripe.net/measurements/9299652>.
- [10] Matt Calder, Ashley Flavel, Ethan Katz-Bassett, Ratul Mahajan, and Jitendra Padhye. 2015. Analyzing the Performance of an Anycast CDN. In *Proceedings of IMC*.
- [11] Fangfei Chen, Ramesh K. Sitaraman, and Marcelo Torres. 2015. End-User Mapping: Next Generation Request Routing for Content Delivery. In *ACM SIGCOMM*.
- [12] Xenofontas Dimitropoulos, Paul Hurley, Andreas Kind, and Marc Ph. Stoecklin. 2009. On the 95-Percentile Billing Method. In *Proceedings of PAM*.
- [13] Florin Dobrian, Vyas Sekar, Asad Awan, Ion Stoica, Dilip Joseph, Aditya Ganjam, Jibin Zhan, and Hui Zhang. 2011. Understanding the Impact of Video Quality on User Engagement. In *Proceedings of SIGCOMM*.
- [14] Ashley Flavel, Pradeepkumar Mani, David Maltz, Nick Holt, Jie Liu, Yingying Chen, and Oleg Surmachev. 2015. FastRoute: A Scalable Load-Aware Anycast Routing Architecture for Modern CDNs. In *USENIX NSDI*.
- [15] Benjamin Frank, Ingmar Poese, Yin Lin, Georgios Smaragdakis, Anja Feldmann, Bruce Maggs, Jannis Rake, Steve Uhlig, and Rick Weber. 2013. Pushing CDN-ISP collaboration to the limit. *ACM SIGCOMM CCR* 43, 3 (2013), 34–44.
- [16] Alexandre Gerber and Robert Doverspike. 2011. Traffic Types and Growth in Backbone Networks. In *IEEE OFC/NFOEC*.
- [17] Oliver Hohlfeld, Jan R uth, Konrad Wolsing, and Torsten Zimmermann. 2018. Characterizing a Meta-CDN. In *Proceedings of PAM*.
- [18] Apple Inc. 2017. 2016 Annual Report. <http://files.shareholder.com/downloads/AAPL/3135363200x0xS1628280-16-20309/320193/filing.pdf>.
- [19] Craig Labovitz, Scott Iekel-Johnson, Danny McPherson, Jon Oberheide, and Farnam Jahani. 2010. Internet Inter-domain Traffic. In *Proceedings of SIGCOMM*.
- [20] Hongqiang Harry Liu, Ye Wang, Yang Richard Yang, Hao Wang, and Chen Tian. 2012. Optimizing Cost and Performance for Content Multihoming. In *Proceedings of SIGCOMM*.
- [21] Michael Henriksen. [n. d.]. AQUATONE: A Tool for Domain Flyovers. <https://github.com/michenriksen/aquatone>.
- [22] Matthew K. Mukerjee, Ilker Nadi Bozkurt, Bruce Maggs, Srinivasan Seshan, and Hui Zhang. 2016. The Impact of Brokers on the Future of Content Delivery. In *ACM HotNets*.
- [23] Matthew K. Mukerjee, Ilker Nadi Bozkurt, Devdeep Ray, Bruce M. Maggs, Srinivasan Seshan, and Hui Zhang. 2017. Redesigning CDN-Broker Interactions for Improved Content Delivery. In *ACM CoNEXT*.
- [24] Erik Nygren, Ramesh K. Sitaraman, and Jennifer Sun. 2010. The Akamai Network: A Platform for High-performance Internet Applications. *SIGOPS OS Rev.* 44, 3 (Aug. 2010), 2–19.
- [25] John S. Otto, Mario A. S nchez, John P. Rula, and Fabi n E. Bustamante. 2012. Content Delivery and the Natural Evolution of DNS: Remote Dns Trends, Performance Issues and Alternative Solutions. In *Proceedings of IMC*.
- [26] Ingmar Poese, Benjamin Frank, Bernhard Ager, Georgios Smaragdakis, and Anja Feldmann. 2010. Improving Content Delivery Using Provider-aided Distance Information. In *Proceedings of IMC*.
- [27] Ripe Network Coordination Centre. [n. d.]. RIPE Atlas Internet Data Collection System. <https://atlas.ripe.net/>.
- [28] Ao-Jan Su, David R. Choffnes, Aleksandar Kuzmanovic, and Fabi n E. Bustamante. 2009. Drafting Behind Akamai: Inferring Network Conditions Based on CDN Redirections. *IEEE/ACM ToN* 17, 6 (2009), 1752–1765.
- [29] Matthias Wichtlhuber, Jan Kessler, Sebastian B cker, Ingmar Poese, Jeremias Blendin, Christian Koch, and David Hausheer. 2017. SoDA: Enabling CDN-ISP Collaboration with Software Defined Anycast. In *IFIP Networking*.
- [30] Jing'an Xue, David Choffnes, and Jilong Wang. 2017. CDNs Meet CN An Empirical Study of CDN Deployments in China. *IEEE Access* 5 (2017), 5292–5305.