Cooperative Active Distribution of Videos in Telco-CDNs

Khaled Diab and Mohamed Hefeeda
School of Computing Science, Simon Fraser University, Burnaby, BC, Canada

ABSTRACT

Telco-CDNs are ISP-managed CDNs deployed inside ISP networks. We propose a new content distribution system inside telco-CDN called CAD. In CAD, Content Provider and ISP collaborate to distribute multimedia content to users inside the ISP network. CAD manages both the overlay and underlay of the network to reduce the ISP interdomain traffic, improve the service latency, and minimize the intradomain link utilization. CAD achieves these goals by allowing caching servers to fetch content from other caching servers, and create videos on-demand inside the telco-CDN. We propose an algorithm to calculate the overlay and underlay of the CAD-managed telco-CDN in polynomial time. Compared against the closest approach in the literature, our initial results showed that CAD achieves up to 30% reduction in the interdomain traffic and up to 230% improvement in the service latency, while not increasing the intradomain link utilization.

CCS CONCEPTS

• **Networks** → **Traffic engineering algorithms**; In-network processing; • **Information systems** → *Multimedia streaming*;

KEYWORDS

Telco-CDN, Content Distribution, TE

ACM Reference Format:
Khaled Diab and Mohamed Hefeeda. 2017. Cooperative Active Distribution of Videos in Telco-CDNs. In *Proceedings of SIGCOMM Posters and Demos '17, Los Angeles, CA, USA, August 22–24, 2017,* 3 pages.
https://doi.org/10.1145/3123878.3131972

1 OVERVIEW

In current video streaming systems, Content Providers such as Netflix create multiple copies of the same video at different bitrates and formats to accommodate user heterogeneity. These transcoding operations usually take place at the Content Provider side. Since users and Content Providers often exist in different geographical locations, this increases the *service latency* and the *interdomain traffic* between multiple ISPs. To solve these two issues, Content Providers contract Content Distribution Networks (CDNs) to deliver these copies to users who are close to CDN servers. However, traditional CDNs can only select the videos to be stored at specific servers based on content popularity and server capacity. When a user requests a video, the request is redirected to the selected server. Since CDNs do not *know* the ISP network *underlay* [4], server selection algorithms may increase link utilization and service latency. In addition to link increased utilization, if the video does not exist in CDN, the request is redirected to the Content Provider servers resulting in interdomain traffic.

Recently, major ISPs such as AT&T start managing CDNs inside their networks [1]. These new distribution networks, called telco-CDNs, enable ISPs to manage the overlay (e.g., server selection) and underlay (e.g., traffic flows) of their distribution network. Thus, telco-CDNs can select servers based on content popularity, server capacity and inter- and intra-domain traffic. Typically, ISPs seek to minimize interdomain traffic and the maximum link utilization (MLU) inside their networks [9]. On the other hand, Content Providers seek to minimize service latency to increase user engagement [3].

We present a new distribution system for multimedia content in telco-CDNs called **C**ooperative **A**ctive **D**istribution (CAD). ISP and Content Provider communicate periodically to produce the server mapping and traffic flows. The goal of CAD is to minimize the interdomain traffic and service latency while not overloading the intradomain links. To achieve these goals, CAD keeps as many video requests as the telco-CDN resources allow *inside the network*. CAD is *cooperative* where caching servers fetch content from other caching servers, and *active* such that caching servers create versions of videos on-demand by utilizing available processing resources at ISP. Specifically, each unique multimedia content has a *parent version*, which in turn has a set of *leaf versions* created by transcoding the parent version. The parent and leaf versions are the videos that users request during streaming sessions.

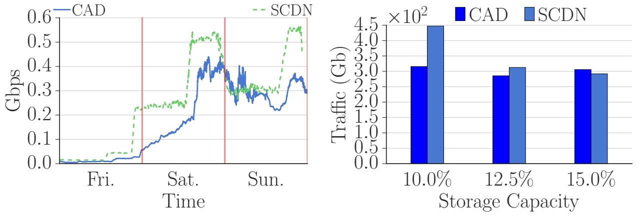
We study the following cooperative active distribution problem: Given a telco-CDN, expected demands $\widehat{D_{iv}}$ of version v at region i, storage and CPU resources per caching server, storage and processing requirements for each version, the maximum allowed processing latency and the capacity of each link, calculate server selection and intradomain traffic flows, in order to minimize the interdomain traffic without exceeding telco-CDN resources and the processing latency.

CAD algorithm runs at the ISP side to manage its storage, network and processing resources. It periodically receives from the Content Provider the video library metadata, the maximum allowed processing latency, and the expected demands for the next period. It uses this information in addition to ISP topology and resources to jointly produce: server selection and traffic flows of telco-CDN. CAD then sends the

---

SIGCOMM Posters and Demos '17, August 22–24, 2017, Los Angeles, CA, USA

ACM ISBN 978-1-4503-5057-0/17/08.
https://doi.org/10.1145/3123878.3131972



(a) Interdomain Traffic **(b) Avg. Interdomain Traffic**

Figure 1: Sample results: CAD vs SCDN.

mappings to the corresponding servers, and the traffic flows to the SDN controller managing the traffic in the ISP network. Our algorithm consists of two main steps to solve the cooperative active distribution problem. First, CAD *initializes* its internal data structures by calculating version-server values and all the simple paths in the network. For each server, CAD sorts versions according to their values. The second step is to *traverse* these versions to calculate the overlay and underlay of the network. CAD traverses the versions in a round-robin fashion across the servers. In some cases, CAD cannot handle a version during the traversal because its dependencies were not yet handled. So, CAD re-traverses these *un-handled versions* to set their mappings in next server rounds. Finally, since fetching and creating versions on-demand produce intradomain flows, the algorithm transforms the problem of finding traffic flows to min-cost flow problem, and uses the cost-scaling algorithm to solve this problem in polynomial time.

2 INITIAL RESULTS

To emulate an ISP network, we implemented a configurable framework on top of Mininet [5]. Each region in the ISP network consists of an Open vSwitch [7], caching server and a set of DASH [10] clients implemented in Go. We manage the ISP network using OpenFlow [6] and implement an SDN controller using Ryu 4.9 [8]. Our controller proactively installs flow entries that correspond to the traffic flows computed by CAD. In CAD, each source-destination flow can have multiple paths. For each multi-path flow, we install flow entries at each virtual switch using OpenFlow *select* group. The weight of each output port is proportional to the traffic it is expected to transmit.

We use the AT&T topology in the East Coast of the US which consists of 16 locations. We choose AT&T because it manages its own CDN [1], hence, our setup is close to a real telco-CDN. We assume that caching servers are deployed to all locations of AT&T in the East Coast. So, there are 16 caching servers, three of them are PoPs representing connections to real Internet eXchange Points. We consider four characteristics when generating user requests and demands: peak weekly demands, peak daily demands, user behavior of quitting streaming sessions, and version popularity distribution. In particular, we generate user requests for three days: Friday, Saturday and Sunday, since they represent peak weekly demands. Also, user arrival rate is not constant during the day, where number of users per time unit increases till its peak value in the evening. Moreover, users may quit their sessions without watching the whole video [2]. Finally, version popularity follows Zipf-Mandelbrot distribution.

We compare CAD against an algorithm similar to [9]. We refer to this algorithm as Simple CDN (SCDN). Like CAD, SCDN knows the expected demands and the network topology. To distribute contents, SCDN pushes the highest demand versions to corresponding caching servers and uses LRU to reactively evict video segments. To route traffic, SCDN uses shortest path routing, where the cost is the end-to-end delay. We compare CAD against SCDN because it was shown in [9] that it outperforms other approaches. In our experiments, we emulate an ISP network that delivers $7,000$ versions at the US east cost. The total size of these versions is 6 TB. We set the maximum allowed processing latency to 5 seconds and the storage capacity to 10%, 12.5% and 15% of the video library size. We repeat each experiment 5 times.

Figure 1 shows samples of our results. Figure 1a shows the interdomain traffic during the three days when storage capacity is 10% of the library size. It shows that CAD reduces the interdomain traffic by 20% during peak hours. In this scenario, the total interdomain traffic for CAD and SCDN is 315.7 and 447.2 Gb, respectively. Thus, CAD outperforms SCDN in terms of interdomain traffic by 29.4%. To assess the importance of *active distribution*, we note that the created on-demand traffic is 46.13 Gb. Without on-demand creation, the created traffic would have been requested from the Content Provider. This means that the on-demand processing contributes to 35% of the improvement in interdomain traffic. In Figure 1b, we show the average interdomain traffic across multiple storage capacity values. CAD outperforms SCDN by up to 29.4% and 8.7% in the first two scenarios, while achieving the same traffic in the highest storage capacity. For SCDN, this is because many segments are kept in the cache when increasing storage, while the demands are constant. For CAD, it is conservative when it stores parents, hence, some versions may not be created on-demand if their parents are not stored. It is however important to note that storage capacity of 10% of the library size is a realistic value in current distribution networks. CAD does not overload the intradomain ISP links, even when fetching parent or leaf versions from inside the network. This is because it jointly manages server mapping and traffic flows. Moreover, CAD reduces service latency up to 2.3X compared to SCDN because CAD keeps many requests inside ISP network (figures omitted).

3 CONCLUSIONS & FUTURE WORK

We presented a new distribution system called CAD to reduce interdomain traffic and service latency by keeping many video requests inside the telco-CDN. We developed an algorithm to jointly calculate the server selection and traffic flows of the telco-CDN in polynomial time. We are currently extending our work to consider the cooperation between multiple telco-CDNs to handle spatio-temporal demand variations.

REFERENCES

[1] AT&T 2017. AT&T-Content Delivery Network Services. http://soc.att.com/1T4sXmB. (2017). [Online; accessed July 2017].

[2] Athula Balachandran, Vyas Sekar, Aditya Akella, and Srinivasan Seshan. 2013. Analyzing the Potential Benefits of CDN Augmentation Strategies for Internet Video Workloads. In *ACM IMC'13*.

[3] 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 *Proc. of ACM SIGCOMM'11*. 362–373.

[4] 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. *SIGCOMM Comput. Commun. Rev.* (July 2013), 11.

[5] Nikhil Handigol, Brandon Heller, Vimalkumar Jeyakumar, Bob Lantz, and Nick McKeown. 2012. Reproducible Network Experiments Using Container-based Emulation. In *Proc. of ACM CoNEXT'12*. 253–264.

[6] Nick McKeown, Tom Anderson, Hari Balakrishnan, Guru Parulkar, Larry Peterson, Jennifer Rexford, Scott Shenker, and Jonathan Turner. 2008. OpenFlow: Enabling Innovation in Campus Networks. *SIGCOMM Comput. Commun. Rev.* 38, 2 (April 2008), 69–74.

[7] Ben Pfaff, Justin Pettit, Teemu Koponen, Ethan Jackson, Andy Zhou, Jarno Rajahalme, Jesse Gross, Alex Wang, Joe Stringer, Pravin Shelar, Keith Amidon, and Martin Casado. 2015. The Design and Implementation of Open vSwitch. In *Proc. of NSDI'15*. 117–130.

[8] Ryu 2017. Ryu SDN framework. https://osrg.github.io/ryu. (April 2017). [Online; accessed April 2017].

[9] Abhigyan Sharma, Arun Venkataramani, and Ramesh K. Sitaraman. 2013. Distributing Content Simplifies ISP Traffic Engineering. In *ACM SIGMETRICS'13*.

[10] Thomas Stockhammer. 2011. Dynamic adaptive streaming over HTTP: standards and design principles. In *Proc. of ACM MMSys'11*. 133–144.