Computer Communications 84 (2016) 1–11

Contents lists available at ScienceDirect

Computer Communications

journal homepage: www.elsevier.com/locate/comcom

Big Data-backed video distribution in the telecom cloud

M. Ruiz[a], M. Germán[a], L.M. Contreras[b], L. Velasco[a,*]

[a] Universitat Politècnica de Catalunya (UPC), Barcelona, Spain
[b] Telefónica Investigación y Desarrollo (TID), Madrid, Spain

ARTICLE INFO

Article history:
Received 23 November 2015
Revised 8 February 2016
Accepted 30 March 2016
Available online 8 April 2016

Keywords:
Live-Television (TV)
Video on Demand (VoD)

ABSTRACT

Telecom operators are starting the deployment of Content Delivery Networks (CDN) to better control and manage video contents injected into the network. Cache nodes placed close to end users can manage contents and adapt them to users' devices, while reducing video traffic in the core. By adopting the standardized MPEG-DASH technique, video contents can be delivered over HTTP. Thus, HTTP servers can be used to serve contents, while packagers running as software can prepare live contents. This paves the way for virtualizing the CDN function. In this paper, a CDN manager is proposed to adapt the virtualized CDN function to current and future demand. A Big Data architecture, fulfilling the ETSI NFV guidelines, allows controlling virtualized components while collecting and pre-processing data. Optimization problems minimize CDN costs while ensuring the highest quality. Re-optimization is triggered based on threshold violations; data stream mining sketches transform collected into modeled data and statistical linear regression and machine learning techniques are proposed to produce estimation of future scenarios. Exhaustive simulation over a realistic scenario reveals remarkable costs reduction by dynamically reconfiguring the CDN.

1. Introduction

Live-TV and Video on Demand (VoD) distribution is in the portfolio of many telecom operators aiming at entering into competition with on-line, over-the-top broadcasters, such as Netflix. To this end, a Content Delivery Network (CDN) is being considered as a suitable option to be deployed by telecom operators internally within their network infrastructure by placing cache nodes in geographically distributed locations covering a territory [1,2]. Forecasts show that 79% of the global IP traffic will be related to video traffic by 2018 [3] thus managing its own CDN allows the network operator to better control and manage the video content injected into the network through predictable traffic sources strategically placed according to a careful network planning to maximize capacity savings. Cloud-based CDNs provide CDN functionalities using cloud resources. Nonetheless, the introduction of cloud imposes additional challenges that have to be addressed. Authors in [4] present a survey on available cloud-based CDNs and identify the open challenges.

In fact, the telecom infrastructure is undergoing a huge transformation since telecom operators are deploying their own cloud infrastructure [5] to prove cloud services and enabling *Software Defined Networking (SDN)* [6] and Network Functions Virtualization (NFV) [7]. The resulting infrastructure is referred to as the *telecom cloud* [8]. NFV decouples network functions (e.g., caching) from proprietary hardware appliances, so they can be implemented in software and deployed on virtual machines (VM) running on commercial off-the-shelf computing hardware. A Virtualized Network Function (VNF) can be functionally decomposed into one or more components and different VNF instances can be placed in geographically distributed locations and communicate among them.

Regarding video delivery, the standardized MPEG Dynamic Adaptive Streaming over HTTP (MPEG-DASH) [9] technique enables media content to be delivered over the Internet. MPEG-DASH requires from a HTTP web server infrastructure to allow users' devices (e.g., Internet-connected televisions, desktop computers, smart phones and tablets, etc.) to consume multimedia content. MPEG-DASH divides contents into a sequence of small file *segments*, each containing a short interval of the content. At the start of a streaming session, the MPEG-DASH client downloads a Media Presentation Description (MPD) file with the resource identifiers (HTTP-URLs) for content's segments. A variety of different qualities (e.g., by changing bitrate and resolution) is made available for each content; while a content is being played back, the MPEG-DASH client automatically selects the segment with the highest quality possible that can be downloaded in time thus, dealing with variable Internet conditions. In addition, client buffer size can be adjusted to ensure a given probability of video re-buffering [10].

* Corresponding author. Tel.: +34934016999.
E-mail address: lvelasco@ac.upc.edu (L. Velasco).

http://dx.doi.org/10.1016/j.comcom.2016.03.026
0140-3664/

MPEG-DASH enables CDN virtualization, where cache nodes are virtualized and be placed in datacenters (DC) (see use case in [7]). Virtualizing caching capabilities facilitates rapid distribution and/or scaling of cache nodes in a cost-efficient and scalable manner. For instance, as a result of using MPEG-DASH for delivery, multimedia contents can be served by HTTP servers. Another cache component must be in charge of generating DASH segments in several qualities and the related MPD files. However, the component that requires more computational effort is video transcoding/transrating, although it can be implemented in software and performed in real-time (see [11] for available software implementations).

To reduce traffic in the core network, cache nodes can be placed as close as possible to the end users. Authors in [12] presented a configurable, efficient and transparent in-network caching service to improve the VoD distribution efficiency by caching video contents as close to the end-user as possible. The solution leverages SDN technology improve network utilization and increasing the Quality of Experience for the end-user. Related to this, authors in [13] proposed a hierarchical *telecom CDN* and a caching algorithm to decide which objects to cache and a cache collaboration strategy to determine how cacheable items are propagated throughout the telecom CDN. In [14] authors studied the performance of distributing caches and the impact of its size and the cache update logic for VoD services, e.g. catch-up programs and movies. They concluded that placing caches in the aggregation network improves the percentage of requested content found in the cache (*Hit Ratio*, HR); in contrast, placing the cache in the access reduces the amount of traffic.

Apart from their right placement, cache nodes are typically dimensioned for peak demand and therefore, greatly underutilized at other times. Aiming at elastically adapt the allocated resources to the current service needs, authors in [15,16] proposed to leverage on the resources of cloud providers to increase capacity when required.

Analyzing video sessions, authors in [17] concluded that a centralized controller could improve user experience, while authors in [18] introduced presented a centralized algorithm for live video optimization providing real-time, fine-grained control. In addition, placing new cache nodes to accommodate spikes in demand and consolidate workload in few cache nodes when the load decreases can also bring benefits. Apart from classical optimization algorithms on conventional content distribution problems, the usage of cloud resources offers a new dimension for optimization that is the IT resource cost (i.e., storage, CPU, etc.) Commercial cloud infrastructure for CDN deployment was reported in [19]. While the concept is applicable to the idea of virtualized CDN, the proposed architecture does not fit to network operator scenarios.

Regarding the interconnection network, connection capacity adaptation is not trivial when it is based on optical technology. Authors in [20,21] proposed a cross-stratum orchestrator architecture to coordinate DC and network elastically.

To detect when resources have to be added or released, the performance and load of cache nodes need to be monitored. Monitoring a *variety* of network elements, servers and applications entails collecting huge *volumes* of data that needs to be transferred and stored assessing *validity*, as well as being analyzed and processed *fast* to achieve near real-time performance. Therefore, Big Data techniques for data collection, pre-processing, and analysis and visualization should be applied. In [22], the ITU-T Study Group 13 proposes a classification of the roles in a Big Data ecosystem. Among the identified roles, the *Big Data application provider* executes a specific set of data life-cycle to meet the requirements of data analysis and visualization as well as the security and privacy requirements. It utilizes the resources from a cloud provider for data analysis and provides analysis result to the Big Data service user. Another role is that of the *Big Data infrastructure provider*, which establishes a computing fabric (computation, storage, and networking resources as well as platforms and processing frameworks) in which certain transformation applications are executed, while protecting the privacy and integrity of data.

A telecom company can take advantage of all the above when deploying its own CDN to provide VoD and live-TV services. In this paper, we assume the hierarchical CDN architecture presented in Section 2 that includes: (i) a Big Data CDN Manager that detects opportunities to minimize operational costs and dynamically serves users from the most proper cache node, while adapting the CDN to the current load by reconfiguring cache nodes (i.e., scaling them by adding new resources), adding and releasing cache nodes, and managing connectivity; (ii) the CDN Admission and Control module responsible for controlling content access and deciding from which cache every user will be served; and (iii) the virtualized leaf cache node with a number of HTTP servers, packagers, storage and a cache manager. Specifically, the contributions of this paper are the following:

(1) A Big Data CDN Manager responsible for adapting the CDN to the current and future load as well as its internal components is presented in Section 2, including: (i) a prediction module to forecast likely scenarios; (ii) a decision maker module to select the most appropriate reconfiguration; and (iii) an optimizer in charge of computing the optimal configuration of the CDN.
(2) To facilitate CDN optimization, three incremental optimization problems are proposed in Section 3; (i) single cache node optimization; (ii) users re-allocation among caches and connectivity re-configuration; and (iii) global CDN re-configuration. Integer Linear Program (ILP) formulations are proposed and heuristic algorithms to solve the problems in real time are devised.
(3) Section 4 targets at making decisions from collected data: (i) data stream mining *sketches* conveniently summarize collected data into modeled data; (ii) a prediction module based on machine learning techniques predicts likely scenarios; and (iii) a simple decision maker module based on threshold violations triggers the most appropriate optimization problem.

The discussion is supported by the results from exhaustive simulation over a realistic scenario in Section 5.

2. Telecom CDN

2.1. CDN architecture

A virtualized hierarchical CDN infrastructure can be deployed in the telecom cloud with some (few) central *Intermediate Cache Nodes* receiving contents from several sources and a number of *Leaf Cache Nodes* placed close to end users (Fig. 1). A centralized *CDN Admission and Control* module implements CDN access policies and redirects users' requests, e.g., based on their geographical location, to the (intermediate or leaf) cache node that will serve them.

Intermediate cache nodes and leaf cache nodes distribute two kinds of contents: VoD and live-TV. VoD contents are prepared in intermediate cache nodes and stored in leaf caches based on its popularity (see e.g., [13]). Nonetheless, in line with [23], live-TV is distributed from intermediate cache nodes and locally prepared in those leaf cache nodes delivering every specific TV channel to the users.

A virtualized leaf cache node would consist of the following components running as software inside VMs deployed in the same DC. The *packager* is in charge of live-TV preparation, including stream transcoding/transrating, segmentation and MPD generation. The *HTTP server* component serves end users' segment requests.