Dynamic and Intelligent SAND-enabled CDN Management

Antti Heikkinen, Janne Vehkaperä, Toni Mäki and Mikko Myllyniemi
VTT Technical Research Centre of Finland Ltd.
Kaitoväylä 1, Oulu, Finland
firstname.lastname@vtt.fi

ABSTRACT

In this demonstration, an advanced Content Delivery Network (CDN) solution for enhancing the delivery of adaptive HTTP-based video streaming is introduced. The demonstration showcases intelligent and scalable CDN testbed which is based on intelligent CDN management functionalities and scalable CDN architecture. The intelligent CDN management functionalities and the associated signaling are based on the upcoming MPEG standard for Server and Network assisted DASH (SAND). The scalable CDN architecture achieved via lightweight service virtualization allows dynamic scaling and balancing of available resources according to the current needs.

The demonstration showcases a scenario where end-users are streaming MPEG-DASH video from the advanced CDN featuring intelligent CDN management and monitoring functionalities to dynamically add or remove virtualized edge servers and reroute end-users based on the resource needs and the location of end-users to achieve more balanced traffic load within the network and better Quality of Experience (QoE) for end-users.

CCS Concepts

•**Information systems** → **Multimedia streaming;** •**Networks** → *Signaling protocols; Network resources allocation; Network control algorithms;*

Keywords

CDN; MPEG-DASH; SAND; Virtualization

1. INTRODUCTION

Content Delivery Networks (CDN) are broadly used for improving the speed, accuracy, and availability of delivered content in the Internet. Today, CDN plays a key role also in video service provisioning, as HTTP has become the defacto standard in delivering Internet-based video for both fixed and mobile end-users.

MMSys'16 May 10-13, 2016, Klagenfurt, Austria

ACM ISBN 978-1-4503-4297-1/16/05…$15.00

DOI: http://dx.doi.org/10.1145/2910017.2910635

The popularity of video services and the need for service providers to ensure high-quality services for their customers are reflected in the demand for CDNs, which has been constantly increasing. The trend has brought also new players into the market, such as Internet Service Providers (ISPs) and telecom operators, in addition to the traditional CDN service providers (e.g. Akamai). ISPs' telco-CDN approaches bring the CDN edge servers closer to the end-users and even into the radio access networks [1]. Also, softwarization and virtualization approaches are emerging into CDNs for additional flexibility in terms of scaling the resources dynamically based on the demand.

Nevertheless, supporting video services efficiently and reliably in CDNs leaves room for optimization. Internet-based video today is largely of streaming type and end-users are demanding good Quality of Experience (QoE) for video services, regardless of the way they access services (e.g. high end TV set vs. mobile handset). Due to the heterogeneous nature of networks, varying network conditions, and diversity of end-user devices, many services implement adaptive video streaming technologies in order to provide good QoE for the end-users with varying requirements regarding the technical quality of the video streams. Adaptive HTTP video streaming in particular requires CDN to maintain multiple representations of the same video content which may easily create storage problems in the cache located in edge servers. In addition, CDN resources need to scale dynamically to cope with peak periods when the load of provided video services are at the highest.

We have presented a solution and testbed implementation for optimized CDN operation available especially for adaptive HTTP video streaming services [2]. We have further developed the testbed by introducing an intelligent load-balancing solution for our CDN which is achieved by utilizing lightweight service virtualization together with intelligent CDN management and monitoring. Proposed new solutions enable to scale the CDN architecture and to perform dynamic load-balancing based on varying video traffic. We have also deployed a simple QoE prediction model into the implementation that allows to better consider the perceived video quality of end-users within the management function. We will showcase the proposed solutions with a demonstration where real and emulated end-user clients are using MPEG-DASH to stream video from our CDN [3]. The virtual edge servers are added (or removed) dynamically based on the traffic load in certain geographical areas (e.g. coverage area of a set of mobile network base stations) and perceived QoE of end-users.

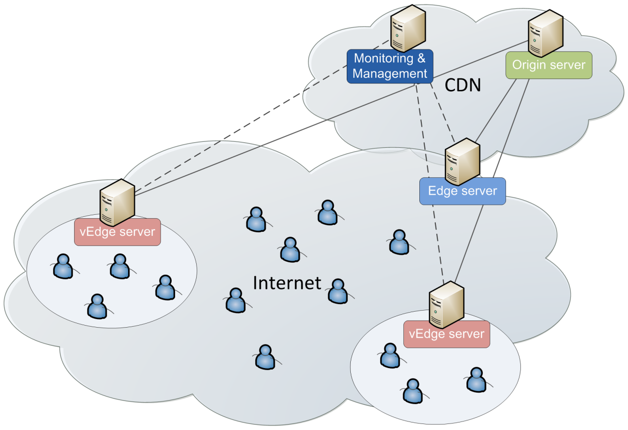

Figure 1: The advanced CDN architecture.

2. SYSTEM OVERVIEW

The demonstrated system consists of our advanced CDN testbed and MPEG-DASH clients with SAND support (SAND is the proposal from MPEG for improved DASH management and signaling) [4]. The overview of the system for video delivery is presented in Fig. 1. The main improvements compared to a traditional CDN are intelligent CDN monitoring and management service (MMS), virtualized edge servers (vEdge) and SAND-based signaling.

2.1 CDN Architecture

The main components of CDN are the origin server, (virtualized) edge servers and intelligent CDN monitoring and management service (MMS). The video content in MPEG-DASH format is stored in the origin server and it is delivered to edge servers based on a request. Edge servers (ES) are usually located close to the edge of network and are in charge of receiving and handling the client requests. When a client requests data from the CDN, the selected ES distributes data to the client. If the requested data is not available in the ES, it requests the data from the origin server and forwards it to the client. The ES stores the requested data for a certain time according its internal caching rules.

We have built our CDN testbed by using off-the-shelf servers and open source software. All servers run Ubuntu 14.04 OS with Nginx HTTP server. In edge servers, Nginx is configured to cache MPEG-DASH content from the origin server while the origin server is configured to act as a HTTP server. We have deployed Docker[1] to virtualize edge server functionalities into the edge server container. These virtualized edge servers can be started and added to CDN or stopped and removed from CDN quickly according the resource demand. Each ES is running an instance of Zabbix[2], an open source monitoring tool for networks and servers.

Monitoring and management service (MMS) is a signaling and control point between the connected video clients and ESs. MMS maintains a database of connected clients and ESs. It can be configured to run different algorithms for routing the clients to the optimal edge server and adding or removing virtual ESs. In addition, MMS collects system status information related to edge servers (ES), for instance, load, congestions or failures. Connected video clients report location and QoE information to the MMS, which then based on the information (and the information from ESs) makes decisions regarding CDN optimization, such as dynamic addition or removal of virtual ESs and directing or redirecting clients to the optimal ES.

2.2 Clients

The testbed comes with two MPEG-DASH client types: a real video client and an emulated video client. The real video client is based on MP4Client from the GPAC [5] enhanced with the implementation of SAND signaling support and a QoE model.

We have developed and implemented a video client emulator called *DASH-Emulator* which emulates real video client(s) and includes same functionalities as the real video client except it does not decode nor render the video. From the network point of view, it behaves as a normal MPEG-DASH video client. First *DASH-Emulator* parses the media presentation description (MPD) file which is a structured collection of references to media content in XML-format. Client uses the network location (baseURL) and timing information in the MPD file in order to request and playback the content. Then it starts the buffering while downloading the video segments as fast as possible. As soon as the initial buffering is done (low threshold reached) the client starts the playback and continues the buffering after the total buffer is full (high threshold reached). Finally, *DASH-Emulator* switches to a steady state phase where the average download rate is the same as the video bitrate. The emulator includes an adaptation algorithm, a SAND signaling support and a QoE model. In addition, we can remotely launch several *DASH-Emulators* according to a predefined pattern (time, network location, number of clients).

We are currently including a full-fledged DASH QoE model (please refer to our in-progress QoE model in [7]) into the testbed. At the first phase the testbed includes a simple J.247 [6] based model that approximates the best available quality for HD content for a given bitrate. This admittedly simplistic model was trained by running the full-reference QoE estimation algorithm on a set of original and degraded video sequences (generated from 4 different kinds of videos and transcoded into desired bitrate spectrum).

2.3 Signaling

The management signaling, based on the proposed SAND specification [4], in the advanced CDN is depicted in Fig. 2. The SAND specification introduces messages between DASH clients and various network elements and attempts to enhance the content delivery, in terms of more accurate adaptation and reaction time with cooperation of different components of the delivery chain. The network-side elements participating into the message exchange are called DASH assisting network elements (DANE). DANEs have at least a minimum intelligence about DASH but may also perform more complex operations that influence DASH content delivery. SAND includes four categories of messages which can provide information about the real-time characteristics of both the network elements and clients:

- Parameters Enhancing Delivery (PED) messages, exchanged between DANEs, including information related to e.g. minimum and maximum required bandwidth

[1]https://www.docker.com/
[2]http://www.zabbix.com/

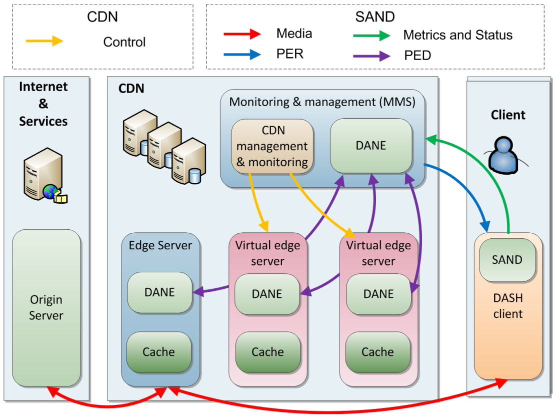

Figure 2: Signaling in the advanced CDN architecture.

- Parameters Enhancing Reception (PER) messages, sent from DANEs to DASH clients, including information e.g. related to content availability, validity, throughput and guaranteed Quality-of-Service (QoS)
- Status messages sent from DASH clients to DANEs, including information e.g. about anticipated segment requests of a client
- Metrics messages sent from DASH clients to Metrics servers, including information e.g. description of the streaming session of a client like QoS/QoE

We used WebSocket (as defined in RFC 6455) to transport SAND messages in the network. MMS contains a WebSocket server that MPEG-DASH clients equipped with a WebSocket client communicate with by transmitting/receiving the SAND messages. When a client opens a connection it will be open during the whole streaming session and enables full-duplex communication between clients and MMS. Metrics information from ES is collected utilizing the Zabbix monitoring tool. Controlling the pool of the virtualized edge servers (adding or removing them) is based on proprietary commands delivered over SSH protocol connections.

The messaging is described in the following in detail:

1. MMS sends periodical PED messages (see Fig. 2) to ESs for a status inquiry (e.g. failure or load status)
2. A client sends Status message to MMS informing its location and requesting ES's address (baseURL)
3. MMS sends PER message with ES's address (baseURL) to the client
4. The client sends Metric message to MMS informing its QoE value
5. MMS sends Control message to add a virtual edge server to CDN and sends PER message with a new ES's address to the client
6. The client sends Status message to MMS informing when it leaves the service
7. MMS sends Control message to remove the virtual edge server from CDN

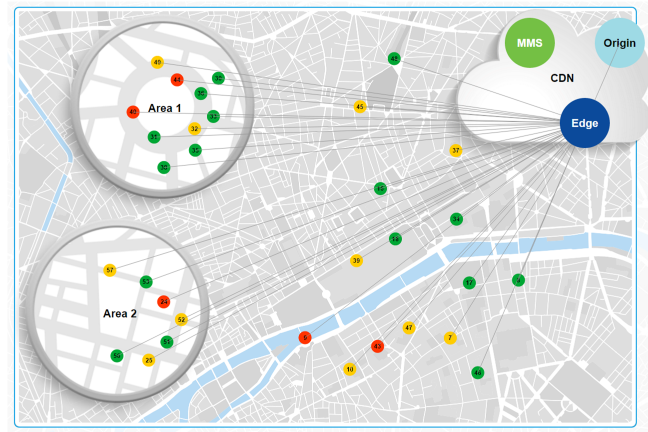

Figure 3: Users in the CDN using the main Edge.

By using a single control point (MMS) in CDN, ESs can be dynamically added or removed from CDN without the need to update the MPD files. However, the proposed architecture is scalable and it permits to use a hierarchical architecture where MMS is distributed in the network.

3. DEMONSTRATION

The components of the demonstration are our advanced CDN, MPEG-DASH clients and visualization of the introduced functionalities. Advanced CDN consists of origin server, monitoring and management service (MMS), one main and three virtualized edge servers. Two types of clients are used: a real video client (MP4Client) with SAND support and a number of emulated MPEG-DASH clients with SAND support (*DASH-Emulator*). The real video client shows a single video stream from the service. The changing topology of CDN, active clients with QoE values and their connections to a edge server together with relevant SAND signaling are visualized using a web browser.

In the demonstrated scenario we use our advanced CDN including one main edge server which serves a large geographical area (e.g. large city). There are two locations in the city with a large crowd of people at the same time (e.g. a football stadium and a convention center). In these areas it is possible to add a virtual edge server (vEdge) if certain preconditions are met. The triggering preconditions are e.g. too large number of users allocated to an ES instance or too high amount of CDN traffic being routed to the area. In addition, we can add another vEdge to serve the whole city if the load of the main edge server increases and QoE of users served by the main ES decreases below a predefined threshold. We demonstrate following two scenarios in a single demonstration. In both scenarios a large group of users start watching videos from the CDN within predefined short time intervals. The users are joining the service from three locations as depicted in Fig. 3 (City, Area 1 and Area 2). The video client first connects to MMS which routes the client to the optimal edge server. The client reports the location and perceived QoE value back to MMS.

3.1 Scenario 1

At some point the number of clients within Area 1 increases above the threshold and MMS decides to add a vir-

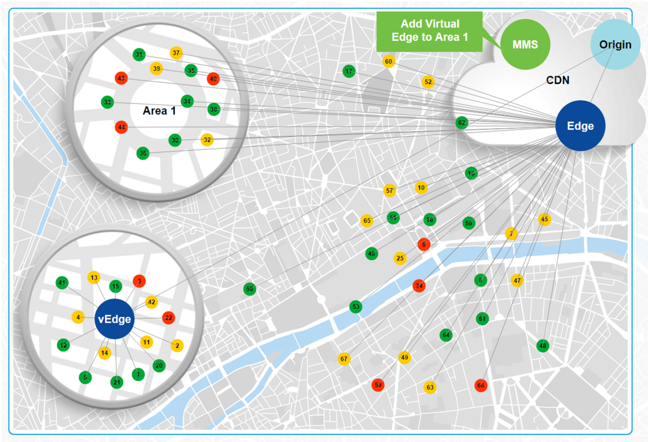

Figure 4: vEdge has been added to Area 2 and MMS decides to add a new virtual Edge to Area 1.

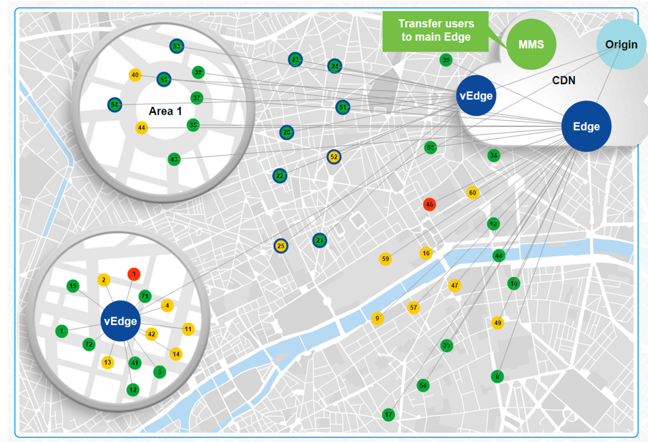

Figure 5: MMS decides to remove vEdge from CDN and to transfer connected users back to the main edge.

tual edge server (vEdge) to Area 1 and routes all the clients from Area 1 to the new vEdge. All new clients that are joining the service in Area 1 are also routed to vEdge. After some time the clients start to leave from the service. When there are only few clients left within Area 1, MMS redirects all the remaining clients from Area 1 to connect the main edge server and removes vEdge server from Area 1.

3.2 Scenario 2

After a while large number of clients are using the same service and the main edge server's outgoing network traffic increases close to the limits. At the same time users are reporting low perceived QoE values. Based on the increased load of the main edge server and clients low QoE values, MMS adds a new virtual edge server to CDN and redirects all clients whose QoE value is below a certain threshold to the new virtual edge server. After the clients have connected to the new edge server they start to report better QoE values. When the main edge server's outgoing network traffic decreases below a certain threshold MMS redirects remaining clients from the virtual edge server back to the main edge server and removes the virtual edge server from CDN.

3.3 Visualization

The visualization of the scenarios are depicted in Fig. 3, Fig. 4 and Fig. 5. The demonstrator shows a map of a city with CDN architecture components and connected clients. The map contains two zoomed areas where virtual edge servers can be added. The color of a client illustrates the perceived QoE value. The visualization illustrates when a virtual edge server is added or removed, when a client joins or leaves the service, when a client reports new QoE value and when MMS makes decisions such as addition or removal of a virtual edge or rerouting users between the edges.

4. SUMMARY

The demonstrated dynamic and intelligent SAND-enabled CDN Management solution enables to scale the CDN architecture and to perform dynamic load-balancing based on varying traffic load and end users perceived QoE. The intelligent CDN management functionalities and the associated signaling are based on SAND. The scalable CDN architecture is achieved via lightweight service virtualization. The users' perveived QoE is approximated by using our simple QoE prediction model. In the demonstration real and emulated clients stream video from our CDN. The virtual edge servers are added dynamically based on the traffic on certain geographical areas and perceived QoE of end-users. The demonstrated solution is one of the first implementations of SAND proposal to assist content delivery within CDN.

5. ACKNOWLEDGMENTS

The demonstration was carried out in CELTIC-Plus NOTTS project and was partially funded by the Finnish Funding Agency of Technology and Innovation (Tekes).

6. REFERENCES

[1] ETSI: Mobile-Edge Computing - Introductory Technical White Paper. (September 2014)

[2] A. Heikkinen, T. Ojanperä and J. Vehkaperä, Dynamic Cache Optimization for DASH Clients in Content Delivery Networks, 13th IEEE Annual Consumer Communications & Networking Conference (CCNC), Las Vegas, 2016

[3] MPEG: Information technology - Dynamic adaptive streaming over HTTP (DASH). International Standard ISO/IEC 23009-1, Second Edition(2015)

[4] MPEG: Information technology - Dynamic adaptive streaming over HTTP (DASH) - Part 5: Server and network assisted DASH (SAND). International Standard ISO/IEC 23009-5, DRAFT(2015)

[5] Le Feuvre, J., Concolato, C., Moissinac, J.C.: GPAC, Open Source Multimedia Framework. ACM Multimedia, Augsburg, Allemagne, September (2007)

[6] International Telecommunication Union, ITU-T recommendation J.247: Objective perceptual multimedia video quality measurement in the presence of a full reference, August, 2008

[7] T. Mäki, M. Varela and D. Ammar, A Layered Model for Quality Estimation of HTTP Video from QoS Measurements, In proceedings of SITIS / QUAMUS, Bangkok, Thailand, 2015