Computer Communications 69 (2015) 79–87

Contents lists available at ScienceDirect

Computer Communications

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

# Using Software Defined Networking to enhance the delivery of Video-on-Demand☆

Panagiotis Georgopoulos [a,*], Matthew Broadbent [b], Arsham Farshad [b], Bernhard Plattner [a], Nicholas Race [b]

[a] Communication Systems Group, TIK, ETH Zurich, Zurich 8092, Switzerland
[b] School of Computing and Communications, Lancaster University, Lancaster, LA1 4WA, United Kingdom

ARTICLE INFO

*Article history:*
Available online 4 July 2015

*Keywords:*
Video-on-Demand (VoD)
Caching
Quality of Experience (QoE)
Software Defined Networking (SDN)
OpenCache

ABSTRACT

High quality online video streaming, both live and on-demand, has become an essential part of many consumers' lives. The popularity of video streaming, however, places a burden on the underlying network infrastructure. This is because it needs to be capable of delivering significant amounts of data in a time-critical manner to users. The Video-on-Demand (VoD) distribution paradigm uses a unicast independent flow for each user request. This results in multiple duplicate flows carrying the same video assets that only serve to exacerbate the burden placed upon the network. In this paper we present OpenCache: a highly configurable, efficient and transparent in-network caching service that aims to improve the VoD distribution efficiency by caching video assets as close to the end-user as possible. OpenCache leverages Software Defined Networking technology to benefit last mile environments by improving network utilisation and increasing the Quality of Experience for the end-user. Our evaluation on a pan-European OpenFlow testbed uses adaptive bitrate video to demonstrate that with the use of OpenCache, streaming applications play back higher quality video and experience increased throughput, higher bitrate, and shorter start up and buffering times.



## 1. Introduction

Online video streaming has seen a huge growth in popularity during recent years. This includes the consumption of both live and on-demand content. In 2013, Internet video traffic represented 66% of all global Internet traffic, and is predicted to increase to 79% by 2018 [2]. At the same time, the popularity of Video-on-Demand (VoD) traffic also continues to increase, with consumer VoD traffic expected to double by 2018: transporting the equivalent of 6 billion DVDs per month [2]. High Definition (HD) video traffic has already surpassed that of Standard Definition (SD) [3], and with the introduction of Ultra-High Definition (UHD) content providers will continue to push expectations in the availability of higher video quality and bitrates. Undoubtedly, high quality online video streaming has become an essential part of many consumers' lives.

In a Video-on-Demand scenario, individuals are able to retrieve content for playback after the initial broadcast. The growth in VoD traffic, coupled with the trend towards content of higher resolution and quality, such as HD and UHD, presents significant challenges. In particular, the evolution in video quality requires networks that are capable of transferring significant amounts of data, in the order of tens or hundreds of Mbps for a single video stream, in a time-sensitive manner. This increase, together with the general growth in traffic, places an additional burden on the underlying network and distribution infrastructure.

Currently, VoD requests are handled individually, leading to *an independent flow* in the distribution network serving *each user's request*. Using such a unicast content delivery paradigm naively ignores that much of the content is identical to transmissions minutes, hours or days earlier. Hence, a very large amount of identical media objects, in the order of gigabytes for a typical HD film, are delivered over the same network segment repeatedly. In order to efficiently support such VoD streaming, the end-to-end capacity of the network must continuously match the increasing number of Internet video users and the growing popularity of higher resolution content. Mechanisms are therefore sought to improve the efficiency of VoD distribution.

In this paper we introduce OpenCache: a transparent, flexible and highly configurable in-network caching service for VoD streaming. OpenCache's contribution is to provide a programmable service that allows any caching strategy, e.g. [4–8], to be easily deployed within

☆ This is an extended version of our previous work appearing in ICCCN 2014 [1].

* Corresponding author. Tel.: +41 446327461.
*E-mail addresses:* panos@tik.ee.ethz.ch, email@panagiotisgeorgopoulos.com (P. Georgopoulos), m.broadbent@lancaster.ac.uk (M. Broadbent), a.farshad@lancaster.ac.uk (A. Farshad), plattner@tik.ee.ethz.ch (B. Plattner), n.race@lancaster.ac.uk (N. Race).

http://dx.doi.org/10.1016/j.comcom.2015.06.015
0140-3664/

the network infrastructure. To achieve this goal, OpenCache uses Software Defined Networking (SDN) to provide *cache as a service* for media content in an efficient and transparent fashion. This is achieved through powerful interfaces, designed to directly benefit last mile environments. By leveraging SDN, and OpenFlow in particular [9], we provide a control plane that orchestrates the caching and distribution functionalities, and transparently pushes the content as close to the user as possible without requiring any changes in the delivery methods or the end-hosts.

Our approach, building an SDN-based in-network caching service, has three important contributions. Firstly, it improves network utilisation and minimises the external link usage on the last mile. Secondly, OpenCache reduces the distribution load from the VoD content provider and all the transient networks along the path of the VoD server to the end-user. Thirdly, by transparently caching the content closer to the user, OpenCache minimises the distance between the VoD streaming server and the user. This provides significant improvements to the Quality of Experience (QoE) of the end-user, as the streaming application observes higher throughput, higher minimum and average streaming bitrate, and smaller start up and buffering times; key QoE differentiators [10–12].

The remainder of the paper is organised as follows. Section 2 provides the background of this work, whilst related work is presented in Section 3. Section 4 introduces the main components and functionality of OpenCache, whereas Section 5 describes the benefits achieved with using SDN. Evaluation is described in Section 6 and finally, Section 7 concludes the paper.

## 2. Background

This section introduces the motivation and problem statement that underpins our work. We then present a new networking approach, called Software Defined Networking, that is key to the solution we provide in this problem space.

### 2.1. Motivation and problem statement

To achieve high quality VoD streaming, a potential solution should be capable of addressing these primary requirements:

(1) *Provide high throughput end-to-end:* High quality video streaming demands quick and reliable transmission of high bitrate encoded content end-to-end. It is often the case that the intermediate networks become the bottleneck for high quality video streaming. It is not sufficient to simply ensure adequate origin server capacity, but adequate network bandwidth must be available in all the intermediate networks between the content server and the end-user [13–15]. Considering the fragmented nature of the Internet, illustrated by the fact that even the largest network worldwide accounts for only 5% of user traffic and needs over 650 networks to reach 90% of access traffic [13,16], this is a stark problem. This fragmentation means that content that is centrally hosted must travel over multiple networks to reach end-users. Therefore, the burden falls on the intermediate networks to ensure that adequate capacity is available to achieve the necessary end-to-end throughput for high quality streaming.

(2) *Minimise distance between VoD server and user:* Large geographical distance between the content server and the end-user introduces the potential for higher latency and packet loss in today's best-effort Internet. High latency and packet loss are particularly important as, when present, the user will observe greater start up and buffering times and may also be subjected to frame drops and playback freezing. Ultimately, these events result in a lower Quality of Experience (QoE) [10–12,14,15,17]. In order to minimise packet loss and benefit from reliable transmission, major VoD content providers (e.g. Netflix, Amazon's Instant Video, YouTube etc.) use TCP to stream VoD content [14–16,18,19]. However, TCP's performance is highly affected by latency and packet loss, which is noticeably present when the VoD server and client are far away from each other. This is because TCP's throughput is inversely related to network latency or RTT [13,14]. Therefore, from both a networking and QoE perspective, the distance between the server and the end-user can become a significant bottleneck in maintaining high quality video streaming.

A potential solution should address the aforementioned challenges and ensure that the media content resides as close to the user as possible. Such an approach would ensure lower latency and higher throughput end-to-end, eventually leading to higher video quality and higher QoE overall [10–12].

### 2.2. Software Defined Networking

Software Defined Networking (SDN) is a new networking approach that facilitates the decoupling of the control plane in a network (i.e. the decision making entity) from the data plane (i.e. the underlying forwarding mechanism). OpenFlow [9], a prominent SDN protocol, defines the communication between the Layer 2 switches and the controller of a network in an open and vendor-agnostic manner. OpenFlow allows experimenters, application developers and network administrators to exploit the true capabilities of a network in an easily deployable and flexible manner. With the centralised network perspective that SDN provides (through its controller), an administrator has an overarching view of the current network status and has the ability to programmatically introduce new network-wide functionality without having to interact with each individual network or user device. OpenCache, our in-network caching service, uses OpenFlow to dynamically cache and distribute media content within a network in a highly efficient and transparent manner.

## 3. Related work

Related work that seeks to improve VoD distribution efficiency spans across multiple domains. These vary from application based solutions, such as Peer-to-Peer, to server based solutions, such as cache and proxy servers or dedicated infrastructure, such as Content Delivery Networks (CDNs). In this section we provide a brief overview of each approach.

The efficiency of a Peer-to-Peer (P2P) based networking solution for video streaming depends heavily upon the participation of users and their willingness to share their limited storage and network resources [20]. P2P pushes the content closer to end-users and can deliver a live video stream to multiple users simultaneously. This is possible because peers can sustain the short-term retention of live video using their own limited resources. However, P2P is much less effective for VoD distribution, as the time between requests for identical content may be in the order of hours, days or even months. This imposes additional resource requirements. In addition, peers may join and leave the service at will, making quality assurance very challenging. Furthermore, the distributed nature of P2P brings a lack of central control, particularly for authentication, authorisation, accounting and security. This prohibits administrators and content providers from making informed decisions and improving the service that they provide. In addition, it prevents these parties from using intelligent caching and distribution techniques [21].

Alternatively, traditional in-network cache and proxy approaches aim to provide additional network and storage support by focusing on delivering the content to users locally. For example, [6] demonstrates the significant benefits of caching YouTube content, where even a very basic caching policy (i.e. a static cache with long-term popular videos) can achieve a 51% cache-hit ratio. Similar benefits are demonstrated in [7], where a simple two hour expiration caching policy yields an aggregated request and byte hit rate of 24% using cache storage of a size less than 2% of the overall data transferred. The benefits of simple (i.e. never flushed) chunk-based transparent caching