

# Efficient Multimedia Transmission over Scalable IoT Architecture

**Vasileios A. Memos**

Department of Applied Informatics, School of Information Sciences, University of Macedonia, Thessaloniki, Greece  
E-mail: mail6016@uom.edu.gr

Received: 24 January 2018; Accepted: 18 April 2018; Published: 08 June 2018

**Abstract**—Today the new technology concept, called “Internet of Things” (IoT), presents a significant field of research, due to the fact that it promises many advantages to the users worldwide. Besides this, multimedia file streaming tends to be a daily user habit, as a result of the evolution of the internet and the capabilities of devices, such as mobile phones and computers. This tendency has established Cloud Computing (CC) as a vital technology due to the fact that it reduces economic costs, improves accessibility and flexibility, while it increases the total network performance too. This paper demonstrates a new technological approach for IoT architecture, where a layered implementation with the use of many cloud servers can offer optimized streaming and download, because of the significant reduction of the transfer time costs and the storage space of the Main Cloud Server (MCS). Moreover, users will have the ability to choose or adapt to their devices’ restrictions, between low - up to - Ultra-High Definition (UHD) quality for multimedia access and downloading, thanks to a new quality scalability feature which is introduced and is based on the new High-Efficiency Video Coding (HEVC) compression standard. Experimental results show the efficiency of the proposed approach with math formulas and graphs.

**Index Terms**—Cloud Computing, HEVC, IoT, Layered Architecture, Mathematical Model, Multimedia Streaming, Quality Scalability.

## I. INTRODUCTION

In recent years, many architectures and mathematical models have been proposed to provide more efficient use of the cloud distributed data centers to eliminate server consolidation problems [1], reduce energy consumption [2], [3] and improve total cloud performance [4]. Other research papers emphasize in the virtualization of the cloud geo-distributed data centers for optimized performance too [3], [5]. In addition, there have been proposed data storage [6] and cloud quality [7] models, and cost optimization approaches for geo-distributed [8] and public [9] clouds. However, the most models emphasize in the storage cost reduction, but not in the delivery time and how it could be minimized.

This paper emphasizes in media files transmission and an optimization approach for minimizing their buffer and

download time from cloud servers to the end users is proposed. Specifically, a new scheme for cloud architecture is presented which ensures optimized streaming and downloading of media files, compressed with HEVC standard. Moreover, a new quality scalability feature which improves QoS and QoE performance is introduced.

The proposed method is based on layered cloud architecture, where there is not one cloud, but locally sub-clouds, geographically dispersed around the main cloud server, such as in [10], [11]. Each sub-cloud consists of many servers which are in constant communication with the Main Cloud Server (MCS) so as to respond to the users’ access requests of the geographical area which they cover.

Moreover, the paper adopts the storage caching [12] of the sub-cloud servers for popular (frequently) requested files [13] by the connected users, and the quality scalability [14] and bitstream division [15] of the media files, so as to give the capability to the users to access them in quality of Class A, B, C and D, as they are presented in the Scalable HEVC software SHM provided by JCT-VC [16], due to their preferences or the quality restrictions of their internet navigation devices. In addition, the media files which are stored in the MCS will be compressed with the new generation compression technology, H.265 or High-Efficiency Video Coding (HEVC) standard, as it is well-known.

The paper is organized as follows: In Section II, the proposed cloud architecture for optimized media files transmission is described. Section II presents the related works already done. Section III provides mathematical models to explain the proposed approach and conduct experimental analysis. Section IV includes the experimental results in comparison with other current related models. Section V concludes the paper.

## II. RELATED WORKS

Many relevant studies have already been done to present the benefits of using the High Efficiency Video Coding (HEVC) compression standard for media files compared to the predecessor H.264/MPEG-4 Advanced Video Coding (AVC) standard, such as in [17], [18], [19], [20], converging to the fact that HEVC standard offers about 50% bitrate decrease for about equal video quality

in comparison with the H.264 standard.

Other research papers focused on HEVC complexity and implementation [21] and its integration into multimedia systems and protocols [22], demonstrating the benefits of the new compression standard against its predecessor, H.264. A benchmarking in real-time streaming of video files, encoded with HEVC standard, in loss-prone network environments, presented in [23], while a rate control algorithm proposed in [24] and the experimental results demonstrated more effective rate-distortion (R-D) performance. A new scheme which offers mixing capability of multiple videos into a single video stream at a low complexity using HEVC standard presented in [25] and can be applied in real-time cloud-based video-mixing applications.

Scalable Extension HEVC (SHVC) as part of the second version of HEVC [26], [27] offers many advantages compared to other standards. An overview of SHVC and performance comparisons for SHVC versus simulcast HEVC and versus the SHVC to H.264/advanced video coding, are presented in [28]. Scalable video multicast by using expanding window fountain codes (EWF-based FEC) was proposed in [29] and the experimental results demonstrate flexibility and efficiency of this scheme.

In addition, there are many studies and developed models based on cloud computing and its contribution to users' experience - Quality of Experience (QoE) - in mobile cloud gaming video [30], cloud-based video transmission [31], [32], cloud multimedia streaming [33], and cloud-assisted video conferencing [34]. The future of cloud-based experience can be extended to proliferation and mutation of internet-connected devices, Cloud-resident content, Cloud-resident identity, Cloud-resident personal agents, Cloud-resident computation and Cloud-resident video streaming, according to [35].

The QoE maximization with a low complexity of an individual content for the end users is investigated in [36]. Several algorithms are analyzed and compared for this scope. It should be noted that the scalability of the proposed algorithms depends on the parameters of the QoE model for a specific video and the allowable limited cache storage budget [36]. Moreover, the balance of monetary cost and user QoE by an effective algorithm is examined in [37].

Other studies focus on Quality of Service (QoS) of video [38], [39], [40], and multiview video streaming over cloud infrastructures [41]. QoS/QoE mapping and adjustment scheme for cloud-based multimedia infrastructures are proposed in [39]. Efficient performance improvements on request blocking probability, bandwidth utilization, packet delay, packet loss rate, and video playback quality, are achieved by several multi-path provisioning algorithms for cloud-assisted Scalable Video Coding (SVC) streaming in heterogeneous networks, which are proposed in [40].

Both scalable video storage in the cloud [42], and robust streaming of scalable video over packet lossy networks [43] promise significant unequal error protection, by offering reduced storage and reconstruction

costs, and higher average video peak signal-to-noise ratio compared to conventional content independent systems, respectively.

An algorithm which configures storage and virtual machine (VM) resources for video stream services in multi-cloud environments is proposed in [38] and it demonstrates lower service costs, while maintains high QoS. Other studies similar to the above but mainly based on mobile streaming over cloud infrastructures are presented in [44], [45], [46] and [47], which demonstrate corresponding advantages of QoS and QoE indicators.

### III. DESCRIPTION OF THE PROPOSED APPROACH

#### A. General Description of the Architecture

This section presents the proposed cloud architecture model for optimized media files delivery and minimized transfer time costs. The approach of this architecture is presented in Fig.1. As it is shown in this figure, there are several smaller clouds – the “sub-clouds” – instead one cloud, which are geographically dispersed in several large areas. All of these sub-clouds are connected to the Main Cloud Server (MCS), where it includes all the media files. This server has the full management and control of the sub-clouds and is responsible for their proper operation and collaboration each other. Moreover, each sub-cloud consists of many servers around in response to the users' access requests. Each sub-cloud server covers users of a particular geographical area, a factor that is determined by users' IP address criterion. Thus, the main cloud architecture approach can be represented as a weighted undirected graph of the form:

$$\text{Cloud} = (\text{Main Cloud Server}, \text{Sub-Clouds}, \text{Servers}, \text{Devices}, \text{Users}) \quad (1)$$

In other words, the new approach includes three layers:

1<sup>st</sup> layer: the connected users and the devices they use to access the cloud.

2<sup>nd</sup> layer: the Sub-Clouds that are extended worldwide and include many servers.

3<sup>rd</sup> layer: the Main Cloud Server (MCS) in which are connected all the sub-clouds.

This project with layered cloud architecture is used to improve performance and flexibility. Furthermore, this infrastructure is used to apply various methods for optimized media transmission, due to the minimization of the transfer time that is required for media files located in the cloud servers so to be available for real-time access or download to the connected devices by their users.

In addition, by using sub-clouds and many servers, instead of one global main server, problems such as server overload or collapse are reduced and can be confronted immediately due to the existence of many servers which are collaborative. If a server is overloaded or collapsed, another server in the surrounding area of the

sub-cloud will contribute to users' requests until the collapsed server will be up again.

### B. Cache Storage

Assume that a user from a particular geographical area requests an access in real time to a media file. In this case, if the user is located in a long distance from the Main Cloud Server where is located the media file, as all the files too, there will be needed much more time to access the file than the case of shorter corresponding distance, due to network delays. The same applies when the user wants to download and save the file to his terminal.



Fig.1. The Proposed Sub-cloud Infrastructure

Thus, in the case of popular media files which are requested by a lot of users of a geographical area who are connected to a sub-cloud, as proposed in [13], an additional storage area of all sub-cloud servers is adopted so as to be used as cache storage, where the sub-cloud servers will copy locally the requested popular media files from the MCS to their cache databases (but with a difference which is about the quality scalability factor which is analyzed below).

### C. Scalability

In this subsection, it is adopted and introduced the meaning of quality scalability [14] to the proposed model, which allow users to have access to media files in various qualities, depending on their preference or their device quality restrictions (mobile phone, tablet, netbook etc). These qualities are classified in classes and are based on HEVC compression standard, which it is described in detail in the next subsection. By using quality scalability, a media file bitstream can be adapted to the preferences of the end users as well as to the capabilities and restrictions of the devices they use to access the cloud, by removing properly specific parts of the bitstream [14].

Thus, in the proposed model, the MCS will store all the media files in Ultra-High Definition (UHD) quality. Each server of a sub-cloud will store some part – specific bitstream – of each popular media file to its cache area. Assume that each multimedia file – popular or not – will be divided properly to equal size bitstreams, as many as the number of servers in the sub-cloud, as it is depicted in

Fig.2. If a user requests to have access to a media file in UHD quality, all the connected servers of the sub-cloud will contribute to providing the media file in this requested quality. If he requests lower quality, a specific percentage of the sub-cloud servers will contribute to the response to this request. This percentage will be determined by the proportional percentage of the requested quality as to the UHD quality.

### D. High Efficiency Video Coding (HEVC)

H.265/MPEG-H or High-Efficiency Video Coding, known as HEVC, is the latest compression standard, which was officially approved in April 2013 [48], and became the successor of H.264/MPEG-4 or AVC

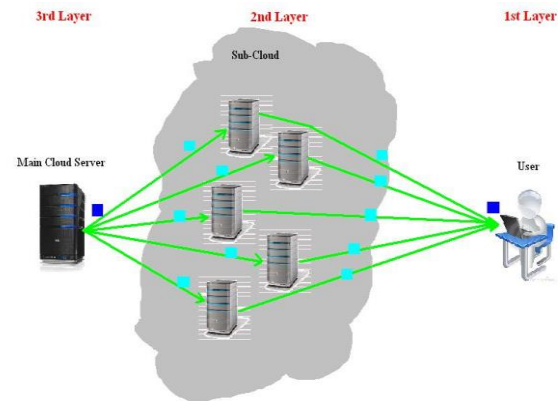


Fig.2. Media File Transfer based on Quality Scalability

(Advanced Video Coding) standard [17]. The HEVC standard design has the features to be easily adaptable to about all the current existing H.264/MPEG-AVC applications and emphasizes mainly on the capability of Ultra-High-Definition (UHD) video view [17] without much bandwidth consumption.

The basic achieve of the HEVC compression standard is the fact that presents significantly better compression performance in comparison with the current existing standards, in the range of 50% bit-rate reduction for about the same video quality, compared to its predecessor, the H.264/MPEG-AVC standard [18], [19]. In addition, it is designed to provide high-quality multimedia streaming, even on low-bandwidth networks, due to the fact that it consumes only the half bandwidth, compared to AVC.

Therefore, in this proposed architecture, it is adopted the HEVC compression standard for the media files which will be stored in the cloud servers for benefits, such as less bandwidth consumption and fewer files' size occupation in the servers.

### E. Scalable Extension HEVC (SHVC)

SHVC is the scalable video coding extension of HEVC standard technology and was finalized in July 2014 as part of the second version of HEVC [26], [27]. While the first version of HEVC standard provides temporal scalability, the SHVC provides spatial, signal-to-noise ratio, bit depth, and color gamut scalability functionalities, as well as various combinations of any of these [28], such

as multiview extensions of HEVC (MV-HEVC) and range format extensions of HEVC (RExt) [49].

The SHVC test model (SHM 10) document and the latest reference software (SHM 12) can be found in [50] and [51], respectively. The common conditions under which the performance of SHVC is tested can be found in [52]. The experimental tests were conducted with Scalable HEVC software SHM provided by JCT-VC ([16], [51]).

#### IV. MATHEMATICAL MODEL ANALYSIS

In this section, the scalable IoT architecture is described and analyzed with the use of math formulas for both the default and the proposed method.

##### A. General Mathematical Model Description

Based on the above Section, let's make the following assumptions with the use of math formulas:

*For Sub-clouds:*

Let  $m$  sub-clouds connected to the Main Cloud Server. Thus, the main cloud server consists of  $m$  sub-clouds:

$$MCS = (SC_1, SC_2, \dots, SC_m) \quad (2)$$

*For Sub-cloud Servers:*

Let  $k$  be servers in each sub-cloud:

$$SC_j = \{S_{j1}, S_{j2}, \dots, S_{jk}\} \quad (3)$$

where  $j=1 \dots m$  (sub-clouds).

*For Users:*

Let  $n$  be users in the world:

$$U = \{u_1, u_2, \dots, u_n\} \quad (4)$$

To define each unique user of the world, the  $n$  users are divided to  $m$  user groups (different geographical areas), as many as the sub-clouds, as follows:

$$U_j = \{U_{j1}, U_{j2}, \dots, U_{jx}\} \quad (5)$$

where  $x$ : the maximum number of users connected to each sub-cloud.

Each user group is divided into  $k$  user subgroups as many as the servers of each sub-cloud:

$$U_{jh} = \{U_{jh1}, U_{jh2}, \dots, U_{jhx/k}\} \quad (6)$$

where  $h=1 \dots k$  (servers)

Let  $U_{jhy}$  be each user of each sub-cloud' server, therefore unique user in the world,

where  $y=1 \dots x/k$ ,

where  $x/k$ : the maximum number of users connected to each sub-cloud' server.

*For Files:*

Let  $\lambda$  media files stored in the Main Cloud Server:

$$MCS = (f_1, f_2, \dots, f_\lambda) \quad (7)$$

Let  $f_i$  each file stored in the main cloud server:

$$f_i = \{f_{i1}, f_{i2}, \dots, f_{i\lambda}\} \quad (8)$$

where  $i=1 \dots \lambda$

Let  $S_{f_i}$  be the size of each  $f_i$  media file.

*For Bandwidth:*

Let  $B_{U_{jhy}}$  be the user's  $U_{jhy}$  bandwidth when he is connected directly to the main server (in the case of non-intermediate sub-cloud servers – existence).

##### B. Default Method (based on HEVC):

In this case, each file  $f_i$  is stored in the main server in four quality versions:

- Class A: p30 in 2560X1600 resolution (Ultra HD)
- Class B1: p24 in 1920X1080 resolution (HD)
- Class B2: p50-60 in 1920X1080 resolution (HD)
- Class C: p30-60 in 800X480 resolution (WVGA)
- Class D: p30-60 in 400X240 resolution (WQVGA)

for proper play in different devices (Internet TVs, desktops, laptops, netbooks, tablets, smartphones etc).

$$\text{Let } f_{iQ} = \{f_{iA}, f_{iB1}, f_{iB2}, f_{iC}, f_{iD}\} \quad (9)$$

each quality version of  $f_i$  media file, and:

$$S_{f_{iQ}} = \{S_{f_{iA}}, S_{f_{iB1}}, S_{f_{iB2}}, S_{f_{iC}}, S_{f_{iD}}\} \quad (10)$$

their corresponding sizes, where:

$$Q = (A, B1, B2, C, D) \quad (11)$$

The total size (TS) of the files which are stored in the main server is:

$$TS_{MCS} = \sum_{i=1}^{\lambda} \sum_{Q=A}^D S_{f_{iQ}} \quad (12)$$

In this method the user downloads his desirable media file in the proper quality, due to his preferences or his device restrictions, immediately from the main server, in which where is stored the media file. Note that internet speed is minimized because of network delays.

Thus, the required download time ( $t_d$ ) for each requested quality of  $f_{iQ}$  is:

$$t_{df_{iQ}} = t_{MCS \rightarrow U_{jhy}} \quad (13)$$

$$t_{dfiq} = s_{fiQ}/B_{U_{jhy}} \quad (14)$$

The bit rate of each quality version of the media file is:

$$R_{fiQ} = s_{fiQ}/t_{lfiQ} \quad (15)$$

where  $t_{lfiQ}$  the time length of each quality version of  $f_i$  media file.

The required time to load the media file (buffering time) in a specific desirable time point ( $t_{dp}$ ), so as to be properly viewable by the user, is:

- When  $t_{dp} > t_{cp}$ ,  $t_{bfIQ} = (t_{dp} - t_{cp}) \cdot \frac{R_{fiQ}}{B_{U_{jhy}}}$
- When  $t_{dp} < t_{cp}$ ,  $t_{bfIQ} = 0$ \*
- When  $t_{dp} = t_{cp}$ ,  $t_{bfIQ} = 0$

In conclusion:

$$t_{bfIQ} = \begin{cases} (t_{dp} - t_{cp}) \cdot R_{fiQ}/B_{U_{jhy}}, & t_{dp} > t_{cp} \\ 0, & t_{dp} < t_{cp} \\ 0, & t_{dp} = t_{cp} \end{cases} \quad (16)$$

where  $t_{cp}$  the current time point of the media streaming and  $t_{bfIQ}$  the buffering time.

\*Assume that the response time of the video player application is negligible.

Therefore, the buffering time of the whole media file, from the start to the end, apart from the possible network delays, is:

$$\begin{aligned} t_{bfIQ} &= (t_{dp} - t_{cp}) \cdot \frac{R_{fiQ}}{B_{U_{jhy}}} \\ t_{bfIQ} &= t_{lfiQ} \cdot \frac{R_{fiQ}}{B_{U_{jhy}}} \\ t_{bfIQ} &= t_{lfiQ} \cdot \frac{s_{fiQ}}{t_{lfiQ}} \cdot \frac{1}{B_{U_{jhy}}} \\ t_{bfIQ} &= s_{fiQ}/B_{U_{jhy}} \end{aligned} \quad (17)$$

### C. Proposed Method:

The proposed method is divided into two cases: popular and non-popular media files. In the first case, frequently requested files, the files will be copied and stored in the sub-cloud servers of the geographical area where they are popular and thus, it will be needed only the transfer from the closest server – in which will be connected the user – to the user's device. This method has been proposed in [13]. However, it is proposed additionally for MCS to maintain media files only in

Ultra High-Definition (UHD), Class A, while they will be divided, due to scalability [14], [15] to equally sized bitstreams (file parts) in the  $k$  servers of the sub-cloud of the area where they are popular, as it was described previously.

In the second case, for non-popular files, the process will be the same, but it will be needed firstly the transfer from MCS to the proper sub-cloud server nearby the user who requests the media file. Thus:

In the MCS will be stored only the qualities of class A of each media file.

$$MCS = (f_{1A}, f_{2A}, \dots, f_{\lambda A}) \quad (18)$$

So, the total size ( $TS'$ ) of the files which are stored in the main server is:

$$TS'_{MCS} = \sum_{i=1}^{\lambda} s_{fiA} \quad (19)$$

Let  $f_{ij}$  be the  $f_{iA}$  media file which is transferred to the  $j$  sub-cloud, and  $f_{ijh}$  be each divided file, where  $h=1 \dots k$  (servers).

Each segment  $f_{ijh}$  will have size:

$$s_{fijh} = s_{fiA}/k \quad (20)$$

The bit rate of each segment in this case will be:

$$R'_{fijh} = s_{fijh}/t_{lfiA} \quad (21)$$

Table 1. Test Classes and Bit Rates

| Class | Bit Rate | Percentage | $p_Q$ |
|-------|----------|------------|-------|
| A     | 14 Mbps  | 100%       | 1.00  |
| B1    | 6 Mbps   | 43%        | 0.43  |
| B2    | 10 Mbps  | 71%        | 0.71  |
| C     | 2 Mbps   | 14%        | 0.14  |
| D     | 1,5 Mbps | 11%        | 0.11  |

Assume the quality classes of Table I with the corresponding bit rates as they are presented in [53].  $p_Q$  is the percentage (%) of each quality class, as follows:

$$p_Q = \{p_A, p_{B1}, p_{B2}, p_C, p_D\} \quad (22)$$

Let  $B'_{U_{jhy}}$  be the user's  $U_{jhy}$  bandwidth when he is connected to a nearby sub-cloud server, subject to:

$$B'_{U_{jhy}} > B_{U_{jhy}} \quad (23)$$

due to less network (propagation and queuing) delays.

This  $B'_{U_{jhy}}$  user bandwidth will be distributed equally for transmitting the proper parts of the media file for each requested quality class. In other words, the total simultaneous downloads will be:  $p_Q \cdot k$ , and the available bandwidth for each download will be:

$$B''_{U_{jhy}} = B'_{U_{jhy}}/p_Q k \quad (24)$$

The total transfer time of the media file is:

$$t'_{fiQ} = t_{MCS \rightarrow SC_j} + t_{SC_j \rightarrow U_{jhy}} \quad (25)$$

In the case of popular files the  $t_{MCS \rightarrow SC_j}$  time transfer is 0, due to the fact that the segments of the file has already been copied and stored to the  $SC_j$  sub-cloud servers.

In the case of non-popular files the  $t_{MCS \rightarrow SC_j}$  time transfer can be considered to be negligible, due to the fact of many sub-clouds existences and the high server performance.

Thus, in both cases, the required download time ( $t'_d$ ) for each requested quality of  $f_{iQ}$  is:

$$\begin{aligned} t'_{dfiQ} &= t_{MCS \rightarrow U_{jhy}} \\ t'_{dfiQ} &= s_{fijh} / B''_{U_{jhy}} \\ t'_{dfiQ} &= \frac{s_{fijh}}{B'_{U_{jhy}} / p_Q k} \\ t'_{dfiQ} &= \frac{p_Q k s_{fijh}}{B'_{U_{jhy}}} \end{aligned} \quad (26)$$

The required time to load the media file (buffering time) in a specific desirable time point ( $t_{dp}$ ), so as to be properly viewable by the user, is:

a. When  $t_{dp} > t_{cp}$ ,

$$\begin{aligned} t'_{bfIQ} &= (t_{dp} - t_{cp}) \cdot \frac{R'_{fijh}}{B''_{U_{jhy}}} \\ t'_{bfIQ} &= (t_{dp} - t_{cp}) \cdot \frac{p_Q k R'_{fijh}}{B'_{U_{jhy}}} \end{aligned}$$

b. When  $t_{dp} < t_{cp}$ ,

$$t_{bfIQ} = 0^*$$

c. When  $t_{dp} = t_{cp}$ ,

$$t_{bfIQ} = 0$$

In conclusion:

$$t'_{bfIQ} = \begin{cases} (t_{dp} - t_{cp}) \cdot p_Q k R'_{fijh} / B'_{U_{jhy}} & t_{dp} > t_{cp} \\ 0^* & t_{dp} < t_{cp} \\ 0 & t_{dp} = t_{cp} \end{cases} \quad (27)$$

where  $t_{cp}$  the current time point of the media streaming.

\*Assume that the response time of the video player application is negligible.

Therefore, the buffering time of the whole media file, from the start to the end, apart from the possible network delays, is:

$$t'_{bfIQ} = (t_{dp} - t_{cp}) \cdot \frac{R'_{fijh}}{B''_{U_{jhy}}}$$

$$t'_{bfIQ} = t_{lfijh} \cdot \frac{p_Q k R'_{fijh}}{B'_{U_{jhy}}}$$

$$t'_{bfIQ} = t_{lfijh} \cdot \frac{p_Q k s_{fijh} / t_{lfiA}}{B'_{U_{jhy}}}$$

$$t'_{bfIQ} = \frac{p_Q k s_{fijh}}{B'_{U_{jhy}}} \quad (28)$$

#### D. Default vs Proposed Method: Mathematical Comparison

##### 1) Main Server Storage:

Each file in the proposed method occupies space:

$$s_{fiA} = p_A s_{fi} = s_{fi} \quad (29)$$

In the default method:

$$\sum_{Q=A}^D s_{fiQ} = s_{fiA} + s_{fiB1} + s_{fiB2} + s_{fiC} + s_{fiD}$$

$$\sum_{Q=A}^D s_{fiQ} = p_A s_{fi} + p_{B1} s_{fi} + p_{B2} s_{fi} + p_C s_{fi} + p_D s_{fi}$$

$$\sum_{Q=A}^D s_{fiQ} = (p_A + p_{B1} + p_{B2} + p_C + p_D) s_{fi}$$

$$\sum_{Q=A}^D s_{fiQ} = (1 + 0.43 + 0.71 + 0.14 + 0.11) s_{fi}$$

$$\sum_{Q=A}^D s_{fiQ} = 2.39 s_{fi} \quad (30)$$

##### 2) Downloading:

From the previous subsection's equations:

$$t_{dfiQ} = \frac{s_{fiQ}}{B_{U_{jhy}}} = \frac{p_Q s_{fi}}{B_{U_{jhy}}}$$

$$t'_{dfiQ} = \frac{p_Q k s_{fijh}}{B'_{U_{jhy}}} = \frac{p_Q s_{fi}}{B'_{U_{jhy}}}$$

$$s_{fijh} = s_{fi} / k \quad (31)$$

$$s_{fiQ} = p_Q s_{fi} \quad (32)$$

Therefore:

$$\frac{t_{dfiQ}}{t'_{dfiQ}} = \frac{p_Q s_{fi} / B_{U_{jhy}}}{p_Q s_{fi} / B'_{U_{jhy}}}$$

$$t'_{dfiQ} = \frac{B_{U_{jhy}}}{B'_{U_{jhy}}} t_{dfiQ} \quad (33)$$

3) *Buffering*:

In the same way for buffering a media file:

$$\frac{t_{bfIQ}}{t'_{bfIQ}} = \frac{s_{fiQ}/B_{U_{jhy}}}{p_Q k s_{fiQ}/B'_{U_{jhy}}} = \frac{p_Q s_{fi}/B_{U_{jhy}}}{p_Q s_{fi}/B'_{U_{jhy}}}$$

$$t'_{bfIQ} = \frac{B_{U_{jhy}}}{B'_{U_{jhy}}} t_{bfIQ} \quad (34)$$

For example, by taking into consideration that the bandwidth is reduced by 20% when a user connected directly to the remote cloud server,  $B_{U_{jhy}} = 0,8B'_{U_{jhy}}$

$$t'_{dfIQ} = 0,8t_{dfIQ} \text{ and } t'_{bfIQ} = 0,8t_{bfIQ}$$

This means that the transfer time of a media file for download or streaming depends on the user bandwidth. As much (%) the user bandwidth is increased to the proposed method, as much is decreased the time. Thus, there is a time optimization of:

$$\text{Time Optimization} = \left(1 - \frac{B_{U_{jhy}}}{B'_{U_{jhy}}}\right) 100\% \quad (35)$$

## V. EXPERIMENTAL RESULTS

Experiments based on the above mathematical models were conducted with Scalable HEVC software SHM

provided by JCT-VC [16], so as to present the effectiveness of the proposed model, compared to the currently used methods for multimedia streaming and downloading by users via cloud infrastructure. HEVC supports low-delay coding structures that usually provide an improved coding efficiency [18].

Table II indicates the test sequences, retrieved by JCT-VC [16], which are used for the following tests. As shown in this Table, the sequences differ from each other in terms of frame count, frame rate (fps) and bit rate (Mbps), and as a result of them, they have different size (Mb) too.

Sequences used in the experiments are classified into five classes based on their resolution (class A, B1, B2, C, D). Class A sequences correspond to ultra high definition (UHD) sequences with a resolution of 2560x1600. Class B1 and B2 sequences correspond to full high definition (HD) sequences with a resolution of 1920x1080. Class C and Class D sequences correspond to WVGA and WQVGA resolutions of 800x480 and 400x240 respectively.

For the experiments, Class A includes the Traffic, PeopleOnStreet, Nebuta and SteamLocomotive sequences; Class B1 includes the Kimono, ParkScene sequences; Class B2 includes the Cactus, BQTerrace and BasketballDrive sequences; Class C includes the RaceHorses, BQMall, PartyScene and BasketballDrill sequences; and Class D includes the RaceHorses, BQSquare, BlowingBubbles and BasketballPass sequences.

Table 2. Test Sequences

| <i>sn</i> | <i>Class</i> | <i>Sequence name</i> | <i>Frame count</i> | <i>Frame rate (fps)</i> | <i>Bit Rate (Mbps)</i> | <i>Size (Mb)</i> |
|-----------|--------------|----------------------|--------------------|-------------------------|------------------------|------------------|
| 1         | A            | Traffic              | 150                | 30                      | 14                     | 70               |
| 2         | A            | PeopleOnStreet       | 150                | 30                      | 14                     | 70               |
| 3         | A            | Nebuta               | 300                | 60                      | 14                     | 70               |
| 4         | A            | SteamLocomotive      | 300                | 60                      | 14                     | 70               |
| 5         | B1           | Kimono               | 240                | 24                      | 6                      | 60               |
| 6         | B1           | ParkScene            | 240                | 24                      | 6                      | 60               |
| 7         | B2           | Cactus               | 500                | 50                      | 10                     | 100              |
| 8         | B2           | BQTerrace            | 600                | 60                      | 10                     | 100              |
| 9         | B2           | BasketballDrive      | 500                | 50                      | 10                     | 100              |
| 10        | C            | RaceHorses           | 300                | 30                      | 2                      | 20               |
| 11        | C            | BQMall               | 600                | 60                      | 2                      | 20               |
| 12        | C            | PartyScene           | 500                | 50                      | 2                      | 20               |
| 13        | C            | BasketballDrill      | 500                | 50                      | 2                      | 20               |
| 14        | D            | RaceHorses           | 300                | 30                      | 1,5                    | 15               |
| 15        | D            | BQSquare             | 600                | 60                      | 1,5                    | 15               |
| 16        | D            | BlowingBubbles       | 500                | 50                      | 1,5                    | 15               |
| 17        | D            | BasketballPass       | 500                | 50                      | 1,5                    | 15               |

The above video sequences were compressed with Scalable HEVC software SHM, provided by JCT-VC [16]. The Random Access (RA) configuration was selected for the tests since it gives better results than the

Low Delay (LD) configuration. The Group of Pictures (GOP) size was set to 8 pictures, while the Intra Period was set to 24 and 32 pictures for 24 and 30 fps contents, respectively. Hierarchical B-pictures were used, with a

Quantization Parameter (QP) increase of 1 between each Temporal Level. The Coding Order was set to 0 8 4 2 1 3 6 5 7. The configuration parameters for HEVC were selected based on Scalable HEVC software SHM provided by JCT-VC [16]. Moreover, all sequences were progressively scanned and used the YUV (YCBCR) 4:2:0 color format with 8 bit per color sample.

For each Class (A, B1, B2, C, D), the download time costs are calculated both for default and for the proposed method. Due to the fact that a user's bandwidth varies from region to region because of network propagation delays and his distance from the server, useful information about internet connection speeds are retrieved from the Speed Test site [54].

Assume that the main server is located in New York City, United States. The tests are classified in three scenarios using Speed Test site, so as to calculate approximately with real data, the difference between bandwidths for each case. For the research purposes, an internet connection of 6 Mbps maximum bandwidth is used.

Table III, indicates for each scenario: the real internet connection speed tested by selecting automatically the closest and best server based on ping; the IP address and the ISP; the download speed, calculated by the transfer of a test file from the main server in New York City to each location, for the default method test; the download speed from the nearest server to the location of the end-user of each scenario.

Table 3. The Download Speeds for Each Scenario

| <i>Scenario</i>                       | <i>1st</i>   | <i>2nd</i>                            | <i>3rd</i>               |
|---------------------------------------|--|---------------------------------------|--------------------------|
| <i>Location</i>                       | Thessaloniki, Greece                               | Miami, Florida, USA                   | Tokyo, Japan             |
| <i>ISP</i>                            | Greek Research and Technology Network S.A., Athens | Serverpronto Dedicated Servers, Miami | GMO Internet Inc., Tokyo |
| <i>IP Address</i>                     | 37.32.218.25                                       | 64.251.15.49                          | 157.7.65.21              |
| <i>Real internet connection speed</i> | 5.12 Mbps  | 5.12 Mbps                             | 4.18 Mbps                |
| <i>Default method</i>                 | 3.79 Mbps  | 4.99 Mbps                             | 2.61 Mbps                |
| <i>Proposed method</i>                | 5.14 Mbps  | 5.08 Mbps                             | 3.82 Mbps                |

Table 4. The Download Speed for Default and Proposed Method

| <i>Location</i>                          | Greece     | USA        | Japan      |
|--|------------|------------|------------|
| <i>Average internet connection speed</i> | 10.20 Mbps | 33.50 Mbps | 60.70 Mbps |
| <i>Default method</i>                    | 7.55 Mbps  | 32.65 Mbps | 37.90 Mbps |
| <i>Proposed method</i>                   | 10.24 Mbps | 33.24 Mbps | 55.47 Mbps |
| <i>Optimization</i>                      | +35,63%    | +1,81%     | +46,36%    |

Specifically, assume that in the case of 1<sup>st</sup> scenario, the closest sub-server is located in Rome, Italy; in the case of 2<sup>nd</sup> scenario, the closest sub-server is located in Houston, Texas, USA; and in the case of 3<sup>rd</sup> scenario, the closest sub-server is located in Beijing, China. As it is clearly

shown in this Table, the download speed varies by a specific percentage for each scenario.

Three tests for three locations are considered in this point, as they are shown in Table IV: Greece, USA, and Japan. For each of them, it is taken into account as real download speed the average speed in each country based on a rolling 30-day broadband average taken from Net Index site [55]: for Greece 10.20 Mbps; for USA 33.50 Mbps; and for Japan 60.70 Mbps.

Based on the data of Table III they are calculated proportionally the corresponding download speeds for the default and the proposed method for each location, as they are shown in Table IV. Moreover, Table IV indicates the optimization percentage (%) of the internet connection speed for each case, by applying the proposed method. Note that the increase of download speed implies corresponding minimization of transfer time costs too.

Thus, based on the download speeds of Table IV, the downloading time costs of each class of Table II are calculated. As it is shown in Table II, the test sequences of Class A have 10Mb size; the test sequences of Class B1 have 60Mb size; the test sequences of Class B2 have 100Mb size; the test sequences of Class C have 20Mb size, and the test sequences of Class D have 15Mb size. The results of these calculations are depicted graphically in Figs.3-7.

Specifically, Fig.3 indicates the corresponding size of a Class A sequence which is downloaded every second from Greece, USA, and Japan, with default (from the main server) and proposed method (from a nearby server) respectively; Fig.4 indicates the corresponding size of a Class B1 sequence which is downloaded every second from Greece, USA, and Japan, with default (from the main server) and proposed method (from a nearby server) respectively; Fig.5 indicates the corresponding size of a Class B2 sequence which is downloaded every second from Greece, USA, and Japan, with default (from the main server) and proposed method (from a nearby server) respectively; Fig.6 indicates the corresponding size of a Class C sequence which is downloaded every second from Greece, USA, and Japan, with default (from the main server) and proposed method (from a nearby server) respectively; and Fig.7 indicates the corresponding size of a Class D sequence which is downloaded every second from Greece, USA, and Japan, with default (from the main server) and proposed method (from a nearby server) respectively. As it is shown clearly in these diagrams, the proposed method causes notable transfer time costs minimization, and thus, improves the relevant QoS and QoE indicators.

In addition, as it is analyzed with mathematical equations in the previous Section, in the case of the proposed method with HEVC compression, each media file will occupy about 2.4 times less space (29) (30) in the MCS than the standard method with one Cloud Server and HEVC compression for the files. In comparison with the standard method and H.264 compression of the media files, the proposed method saves about 4.8 times more storage space in the MCS. This applies due to the fact that by using HEVC compression standard for media files,



there are about the half of bitrate (50%) savings, without degrading the quality [18], [19], [56]. Therefore, a media file compressed with H.264 standard has about 2 times more size than the other one with HEVC compression standard. Scalable Distributed Data Structure (SDDS) – based architectures for real-time data store in the MCS may be applied additionally, so as to improve the storage QoS [57].

Fig. 8 shows the disk space occupation impact in the MCS of each of the 17 samples of Table II, considering that they are stored inside it with a numerical order (s.n.) one by one. In the case of the proposed method, all the samples are compressed with HEVC and are stored in quality of Class A (2560x1600 resolution at 30fps), while in the standard method with HEVC, each sample is stored in all the available quality versions (classes A-D).

By using the proposed method, the total bandwidth usage of the MCS is reduced due to the fact that the frequently requested files by users of a specific geographical area are copied and stored locally in the corresponding sub-cloud which outspreads the area. Fig. 9 shows the MCS load, based on the bandwidth usage and the users' requests. For this test, it is considered that the average users' requests are ranged to quality classes of about 7Mbps bitrate. Fig. 10 depicts the total CPU consumption for the default and the proposed method. As it is shown in Fig. 9 and 10, over 80% bandwidth usage or CPU consumption, there is server overload. Over 100% the server falls down.

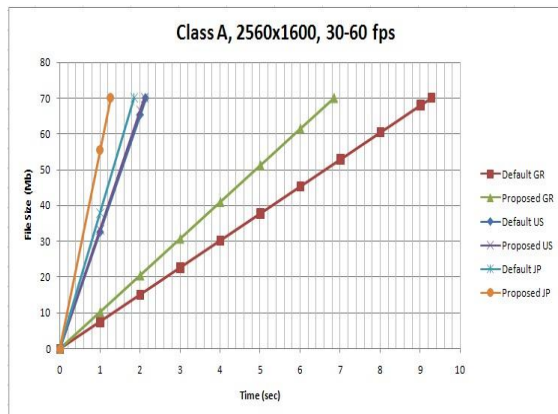


Fig.3. Default and Proposed Method Of Downloading Test Sequences of Class A from Greece, USA, and Japan



Fig.4. Default and Proposed Method of Downloading Test Sequences of Class B1 from Greece, USA, and Japan.



Fig.5. Default and Proposed Method of Downloading Test Sequences of Class B2 from Greece, USA, and Japan.

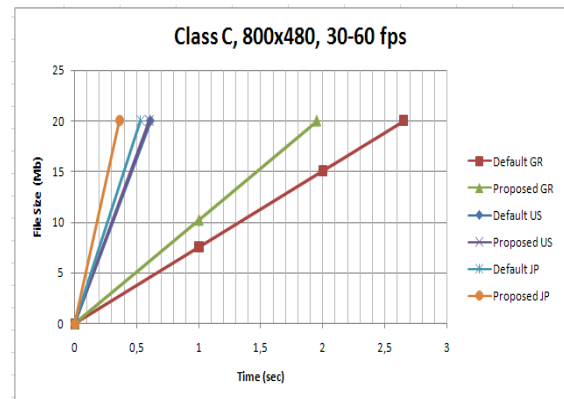


Fig.6. Default and Proposed Method of Downloading Test Sequences of Class C from Greece, USA, and Japan.

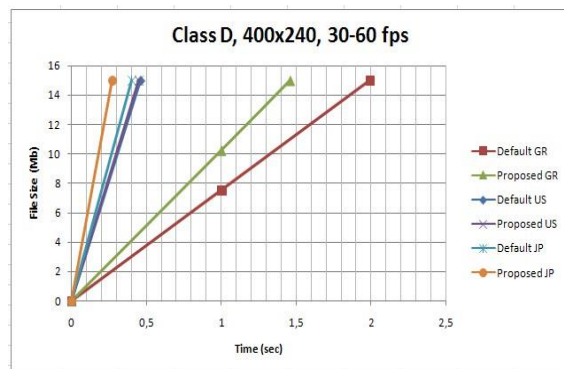


Fig.7. Default and Proposed Method of Downloading Test Sequences of Class D from Greece, USA, and Japan.

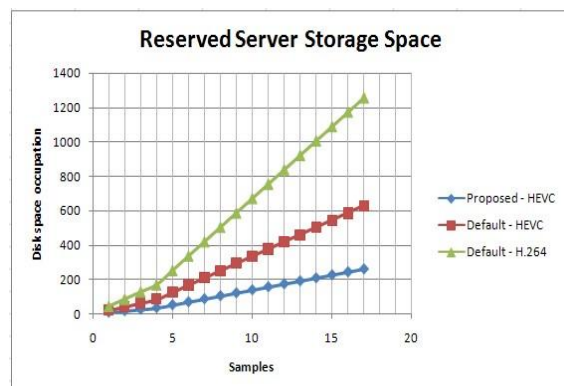


Fig.8. The Reserved Server Storage Space of the test Sequences for Each Method.

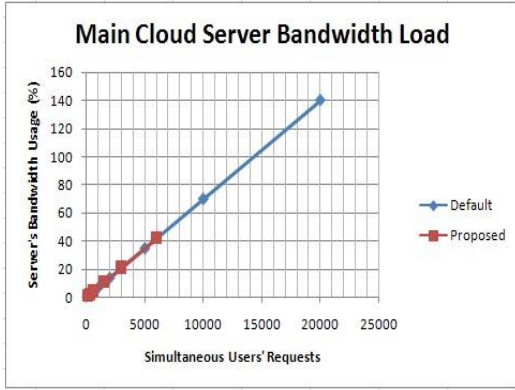


Fig.9. The Main Cloud Server Load Per Simultaneous Users' Requests for Each Method.

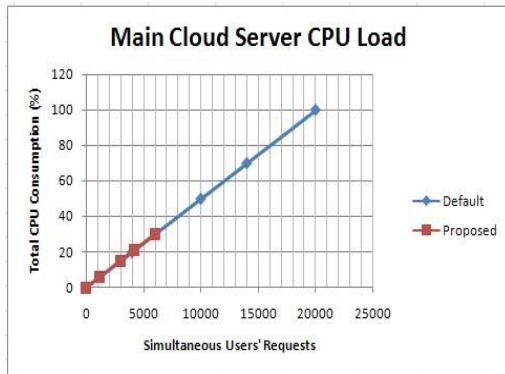


Fig.10. The Percentage CPU of The Main Cloud Server Per Simultaneous Users' Requests for Each Method

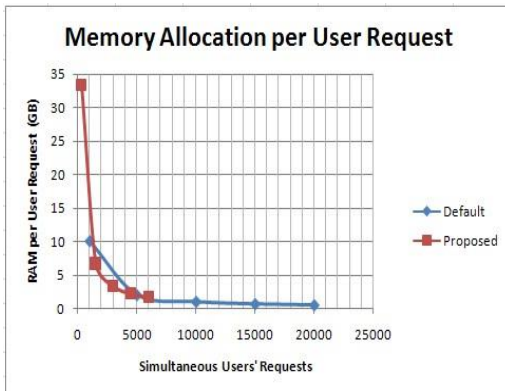


Fig.11. Memory Allocation of the Main Cloud Server per User Request for Each Method

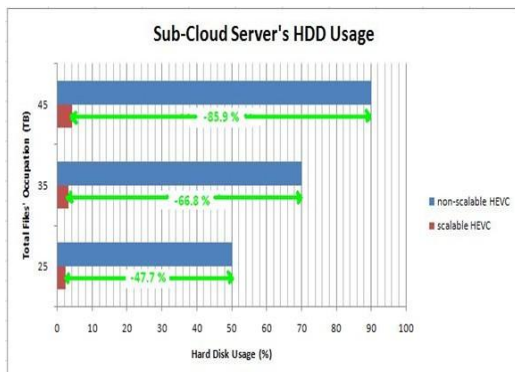


Fig.12. The Storage Space Usage (%) For Scalable and Non-Scalable HEVC Transmission Method in Each Sub-Cloud Server.

Finally, Fig. 11 depicts the memory allocation of the Main Cloud Server per each user-request job. As it is clearly shown, due to the fewer users' requests, the available Main Cloud Server's memory is much more per each user-request job. Note that for the experimental results of Fig. 9, 10 and 11, it is considered that the 70% of the media files which are requested by the users of a specific geographical area are popular, so they are located in the sub-cloud, and only 30% are the requested media files which are non-popular.

In the default method for popular HEVC media files, each file is available in all quality versions from Class A to D, and thus it occupies in each sub-cloud server, according to (30), disk space  $DSO(def)_{f_i}$ :

$$DSO(def)_{f_i} = \sum_{Q=A}^D s_{fiQ} = 2.39 s_{fi} \quad (36)$$

In the proposed method for popular HEVC media files, each file is in its lowest quality, and thus it occupies in each sub-cloud server, disk space  $DSO(pro)_{f_i}$ :

$$DSO(pro)_{f_i} = s_{fiD} = p_D s_{fi} = 0.11 s_{fi} \quad (37)$$

From the above equations (36) and (37):

$$DSO(def)_{f_i} = 21.73 DSO(pro)_{f_i} \quad (38)$$

$$DSO(def) = 21.73 DSO(pro) \quad (39)$$

which means that  $DSO(pro)$  is about 22 times less than  $DSO(def)$ . In other words, the proposed method saves about 22 times more space in each sub-cloud server, making each server more lightweight and optimizing its performance.

Fig. 12 depicts the percentage (%) storage savings in each sub-cloud server of the use of the scalable HEVC transmission method, compared to non-scalable one. As it is clearly shown in this figure, it is observed a significant reduction of 47.7, 66.8, and 85.9 percentage (%) of the usage server's storage, respectively for 50, 70, and 90 percentage (%) occupation of the files which are stored in the server.

## VI. CONCLUSIONS

An optimization approach for minimization of delivery time costs for streaming and download media files from a remote cloud server by users from various geographical areas was presented. The approach is based on layered cloud infrastructure where there are several smaller sub-clouds (rather than a global big one) consisting of many intermediate servers for better resource allocation and quicker response to the users' requests of surrounding areas.

Cache available storage of each sub-cloud server was adopted to keep copies of popular media files requested from the Main Cloud Server by the connected users for real-time access or download. Users have the choice to access or download a media file in any quality scalability

they prefer, from low up to ultra-high definition quality. Furthermore, this model is easily adaptable for users' devices which support specific quality accessibility due to their manufacturing restrictions.

Finally, a mathematical model analysis was presented so as to describe the layered cloud architecture for media files streaming and downloading by users worldwide. Experimental results demonstrate the efficacy and optimization of transfer time costs, and about 2.4 and 4.8 times minimization of storage space of the Main Cloud Server compared to the existing conventional methods with HEVC and H.264 media files compression standard respectively. Moreover, the proposed scalable HEVC-media transmission scheme saves about 22 times more space in each sub-cloud server compared to traditional non-scalable methods.

Future work will include more experimentation to determine the percentage improvement of the proposed layered cloud model on real systems and cloud environments. Moreover, an accurate estimation of Quality of Experience (QoE) parameters (subjective assessment) from Quality of Service (QoS) parameters (objective assessment) is also a potential research direction. Finally, new techniques for secure multimedia transmission [58], [59] with higher Signal to Noise Ratio (SNR) and low information loss [60] may be integrated in the proposed method.

#### REFERENCES

- [1] B. Speitkamp and M. Bichler, "A mathematical programming approach for server consolidation problems in virtualized data centers," *IEEE Transactions on Services Computing*, vol. 3, no. 4, pp. 266–278, October–December 2010.
- [2] A. Beloglazov and R. Buyya, "Energy efficient resource management in virtualized cloud data centers", in *Proceedings of the 10th IEEE/ACM International Conference on Cluster, Cloud and Grid Computing*, pp. 826–831, 2010.
- [3] Carlo Mastroianni, Michela Meo and Giuseppe Papuzzo, "Probabilistic Consolidation of Virtual Machines in Self-Organizing Cloud Data Centers", *IEEE Transactions on Cloud Computing*, Vol. 1, No. 2, December 2013.
- [4] C. Papagianni, A. Leivadeas, S. Papavassiliou, V. Maglaris, C. Cervell'ó-Pastor, and A'. Monje, "On the optimal allocation of virtual resources in cloud computing networks", *IEEE Transactions on Computers*, vol. 62, no. 6, pp. 1060–1071, June 2013.
- [5] Fei Xu, Fangming Liu, Hai Jin and Athanasios V. Vasilakos, "Managing Performance Overhead of Virtual Machines in Cloud Computing: A Survey, State of Art and Future Directions", Submitted to *Proceedings of the IEEE*, 2013.
- [6] Arkaitz Ruiz-Alvarez and Marty Humphrey, A Model and Decision Procedure for Data Storage in Cloud Computing, *Proceedings of the IEEE/ACM International Symposium on Cluster, Cloud, and Grid Computing (CCGrid'12)*, May 13–16, 2012.
- [7] Xianrong Zheng, Patrick Martin, Kathryn Brohman and Li Da Xu, "CLOUDQUAL: A Quality Model for Cloud Services", *IEEE Transactions on Industrial Informatics*, February 2014.
- [8] L. Jiao, J. Li, T. Xu, and X. Fu, "Cost optimization for online social networks on geo-distributed clouds," in *Network Protocols (ICNP)*, 20th IEEE International Conference on, pp. 1–10, 2012.
- [9] Zhonghong Ou, Hao Zhuang, Andrey Lukyanenko, Jukka K. Nurminen, Pan Hui, Vladimir Mazalov and Antti Ylä-Jääski, "Is the Same Instance Type Created Equal? Exploiting Heterogeneity of Public Clouds", *IEEE Transactions on Cloud Computing*, Vol. 1, No. 2, October 2013.
- [10] Nawsher Khan, Noraziah Ahmad, Tutut Herawan and Zakira Inayat, *Cloud Computing: Locally Sub-Clouds instead of Globally One Cloud*, *International Journal of Cloud Applications and Computing*, 2(3), 68–85, July–September 2012.
- [11] Guangna Lv, Zhenmin Zhao, Xiangming Wen and Zhicai Zhang, *New Cloud Computing Network Architecture Directed At Multimedia*, 2nd International Conference on Information Communication and Management (ICICM 2012), *IPCSIT* vol. 55, 2012.
- [12] G. Chockler, G. Liden, and Y. Vigfusson, "Design and implementation of caching services in the cloud", *IBM Journal of Research and Development*, vol. 55, no. 6, pp. 9:1–9:11, 2011.
- [13] Efstratios Rappos, Stephan Robert and Rudolf H. Riedi, "A Cloud Data Center Optimization Approach Using Dynamic Data Interchanges", in *IEEE 2nd International Conference on Cloud Networking (CloudNet)*, pp. 175 – 179, 11–13 Nov. 2013.
- [14] Philipp Helle, Haricharan Lakshman, Mischa Siekmann, Jan Stegemann, Tobias Hinz, Heiko Schwarz, Detlev Marpe, and Thomas Wiegand, "A Scalable Video Coding Extension of HEVC", *Data Compression Conference*, 2013.
- [15] Wenwu Zhu, Chong Luo, Jianfeng Wang, and Shipeng Li, "Multimedia Cloud Computing: An emerging technology for providing multimedia services and applications", *Signal Processing Magazine, IEEE*, Vol. 28, Issue: 3, pp. 59 – 69, May 2011.
- [16] Scalable HEVC software SHM, Joint Collaborative Team on Video Coding (JCT-VC), <http://hevc.kw.bbc.co.uk/git/w/jctvc-shm.git>
- [17] Dan Grois, Detlev Marpe, Amit Mulyoff, Benaya Itzhaky and Ofer Hadar, "Performance Comparison of H.265/MPEG-HEVC, VP9, and H.264/MPEG-AVC Encoders", 30th Picture Coding Symposium (PCS), December 2013.
- [18] Jens-Rainer Ohm, Gary J. Sullivan, Heiko Schwarz, Thiow Keng Tan and Thomas Wiegand, "Comparison of the Coding Efficiency of Video Coding Standards—Including High Efficiency Video Coding (HEVC)", *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 22, No.12, December 2012.
- [19] Gary J. Sullivan, Jens-Rainer Ohm, Woo-Jin Han and Thomas Wiegand, "Overview of the High Efficiency Video Coding (HEVC) Standard", *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 22, No.12, December 2012.
- [20] Jarno Vanne, Marko Viitanen, Timo D. Hamalainen and Antti Hallapuro, "Comparative Rate-Distortion-Complexity Analysis of HEVC and AVC Video Codecs", *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 22, No.12, December 2012.
- [21] Frank Bossen, Benjamin Bross, Karsten Suhling and David Flynn, "HEVC Complexity and Implementation Analysis", *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 22, No.12, December 2012.

- [22] Thomas Schierl, Miska M. Hannuksela, Ye-Kui Wang and Stephan Wenger, "System Layer Integration of High Efficiency Video Coding", IEEE Transactions on Circuits and Systems for Video Technology, Vol. 22, No.12, December 2012.
- [23] James Nightingale, Qi Wang and Christos Grecos, "Benchmarking Real-Time HEVC Streaming", Real-Time Image and Video Processing, Proceedings of SPIE, Vol. 8437 84370D-1, 2012.
- [24] Shanshe Wang, Siwei Ma, Shiqi Wang, Debin Zhao and Wen Gao, "Rate-GOP Based Rate Control for High Efficiency Video Coding", IEEE Journal of Selected Topics in Signal Processing, Vol. 7, No.6, December 2013.
- [25] Y. Sanchez, R. Globisch, T. Schierl, and T. Wiegand, "Low Complexity Cloud-video-Mixing Using HEVC", IEEE 11th Consumer Communications and Networking Conference (CCNC), 2014.
- [26] J. Chen, J. Boyce, Y. Ye, et al., "High efficiency video coding (HEVC) scalable extension draft 7", Joint Collaborative Team on Video Coding, JCTVC - R1008\_v7, Sapporo, Japan, 2014.
- [27] J. Boyce, J. Chen, Y. Chen, et al., "Draft high efficiency video coding (HEVC) version 2, combined format range extensions (RExt), scalability (SHVC), and multi-view (MV-HEVC) extensions", Joint Collaborative Team on Video Coding, JCTVC-R1013\_v6, Sapporo, Japan, 2014.
- [28] J. Boyce, Y. Ye, J. Chen, and A. Ramasubramanian, "Overview of SHVC: Scalable extensions of the high efficiency video coding standard", IEEE Trans. Circuits Syst. Video Technol., vol. 26, no. 1, pp. 20–34, 2016.
- [29] D. Vukobratovic, V. Stankovic, D. Sejdinovic, L. Stankovic, and Z. Xiong, "Scalable video multicast using expanding window fountain codes", IEEE Trans. Multimedia, Vol. 11, no. 6, pp. 1094-1104, Oct. 2009.
- [30] Jiyan Wu, Chau Yuen, Ngai-Man Cheung, Junliang Chen, and Chang Wen Chen, "Streaming Mobile Cloud Gaming Video over TCP with Adaptive Source-FEC Coding", IEEE Transactions on Circuits and Systems for Video Technology, Vol.27, I.1, pp. 32-48, January 2017.
- [31] Kai Lin, Jeungeun Song, Jiming Luo, Wen Ji, M. Shamim Hossain, Ahmed Ghoneim, "Green Video Transmission in the Mobile Cloud Networks Sign In or Purchase", IEEE Transactions on Circuits and Systems for Video Technology, Vol.27, I.1, pp. 159-169, January 2017.
- [32] Xili Dai, Xiaomin Wang, and Nianbo Liu, "Optimal Scheduling of Data Intensive Applications in Cloud Based Video Distribution Services", IEEE Transactions on Circuits and Systems for Video Technology, Vol.27, I.1, pp. 73-83, January 2017.
- [33] Ashiq Anjum, Tariq Abdullah, M. Tariq, Yusuf Baltaci, and Nick Antonopoulos, "Video Stream Analysis in Clouds: An Object Detection and Classification Framework for High Performance Video Analytics", IEEE Transactions on Cloud Computing, Vol. PP, I. 99, pp. 1, January 2016.
- [34] Richard G. Clegg, Raul Landa, David Griffin, Miguel Rio, Peter Hughes, Ian Kegel, Tim Stevens, Peter Pietzuch and Doug Williams, "Faces in the Clouds: Long-Duration, Multi-User, Cloud-Assisted Video Conferencing", IEEE Transactions on Cloud Computing, Vol. PP, I. 99, pp. 1, March 2017.
- [35] Kilroy Hughes, "The Future of Cloud-Based Entertainment", Proceedings of the IEEE, Vol. 100, May 2012.
- [36] Weiwen Zhang, Yonggang Wen, Zhenzhong Chen and Ashish Khisti, "QoE-Driven Cache Management for HTTP Adaptive Bit Rate Streaming Over Wireless Networks", IEEE Transactions on Multimedia, Vol. 15, No. 6, October 2013.
- [37] Jian He, Yonggang Wen, Jianwei Huang and Di Wu, "On the Cost-QoE Tradeoff for Cloud-Based Video Streaming Under Amazon EC2's Pricing Models", IEEE Transactions on Circuits and Systems for Video Technology, Vol. 24, No. 4, April 2014.
- [38] Wei Chen, Junwei Cao and Yuxin Wan, "QoS-Aware Virtual Machine Scheduling for Video Streaming Services in Multi-Cloud", Tsinghua Science and Technology, pp. 308-317 Vol.18, No. 3, June 2013.
- [39] Wu-Hsiao Hsu and Chi-Hsiang Lo, "QoS/QoE Mapping and Adjustment Model in the Cloud-based Multimedia Infrastructure", IEEE Systems Journal, Vol. 8, No. 1, March 2014.
- [40] Zuqing Zhu, Suoheng Li and Xiaoliang Chen, "Design QoS-Aware Multi-Path Provisioning Strategies for Efficient Cloud-Assisted SVC Video Streaming to Heterogeneous Clients", IEEE Transactions on Multimedia, Vol. 15, No. 4, June 2013.
- [41] Jacob Chakareski, "Adaptive Multiview Video Streaming: Challenges and Opportunities", Interactive 3D Video Streaming in IEEE Communications Magazine, May 2013.
- [42] X. Song, X. Peng, J. Xu, G. Shi, and F. Wu, "Unequal error protection for scalable video storage in the cloud", in IEEE Int. Conf. on Multimedia and Expo (ICME), pp. 1–6, 2015.
- [43] E. Maani and A. Katsaggelos, "Unequal error protection for robust streaming of scalable video over packet lossy networks", IEEE Trans. Circuits Syst. Video Technol., vol. 20, no. 3, pp. 407–416, March 2010.
- [44] Chin-Feng Lai, Honggang Wang, Han-Chieh Chao and Guofang Nan, "A Network and Device Aware QoS Approach for Cloud-Based Mobile Streaming", IEEE Transactions on Multimedia, Vol. 15, No. 4, June 2013.
- [45] Chun-Han Lin, Pi-Cheng Hsiu and Cheng-Kang Hsieh, "Dynamic Backlight Scaling Optimization: A Cloud-Based Energy-Saving Service for Mobile Streaming Applications", IEEE Transactions on Computers, Vol. 63, No. 2, February 2014.
- [46] Xiaofei Wang, MinChen, Ted Taekyoung Kwon, Laurence T. Yang and Victor C. M. Leung, "AMES-Cloud: A Framework of Adaptive Mobile Video Streaming and Efficient Social Video Sharing in the Clouds", IEEE Transactions on Multimedia, Vol. 15, No. 4, June 2013.
- [47] Xiaofei Wang, Ted Taekyoung Kwon, Yanghee Choi, Haiyang Wang and Jiangchuan Liu, "Cloud-Assisted Adaptive Video Streaming and Social-Aware Video Prefetching for Mobile Users", Mobile Computing in IEEE Wireless Communications, June 2013.
- [48] Information technology - High efficiency coding and media delivery in heterogeneous environments - Part 2: High efficiency video coding, ISO/IEC 23008-2:2013 ITU-T Rec. H.265, April 2013.
- [49] Yan Ye, Yong He, Ye-Kui Wang, and Hendry, "SHVC, the Scalable Extensions of HEVC, and Its Applications", Special Topic in ZTE Communications, Vol. 14, No. 1, February 2016.
- [50] J. Chen, J. Boyce, Y. Ye, and M. M. Hannuksela, "SHVC Test Model 10 (SHM 10) Introduction and Encoder Description", Joint Collaborative Team on Video Coding, JCTVC-U1007, Warsaw, Poland, Jun. 2015.
- [51] SHM-12.0 Reference Software [Online], April 2016. [https://hevc.hhi.fraunhofer.de/svn/svn\\_SHVCSoftware/tags/SHM-12.0/](https://hevc.hhi.fraunhofer.de/svn/svn_SHVCSoftware/tags/SHM-12.0/)
- [52] V. Seregin and Y. He, "Common SHM test conditions and software reference configurations", JCT-VC, Joint

- Collaborative Team on Video Coding (JCT-VC) Document JCTVC-Q1009, Valencia, Spain, April 2014.
- [53] O. Le Meur, "Video compression beyond H.264, HEVC", November 2011, [http://www.academia.edu/9456527/Table\\_of\\_Content\\_Video\\_compression\\_Beyond\\_H.264\\_HEVC](http://www.academia.edu/9456527/Table_of_Content_Video_compression_Beyond_H.264_HEVC)
- [54] OOKLA Speed Test, The Global Broadband Speed Test, <http://www.speedtest.net/>
- [55] OOKLA Net Index, Global Broadband and Mobile Performance, <http://www.netindex.com/download/map>
- [56] Alaa F. Eldeken, and Gouda I. Salama, "An Adaptive Deblocking Filter to Improve the Quality of the HEVC Standard", International Journal of Image, Graphics and Signal Processing (IJIGSP), I. 3, pp. 9-15, Modern Education and Computer Science (MECS), February 2015.
- [57] Maciej Lasota, Stanisław Deniziak, and Arkadiusz Chrobot, "An SDDS-Based Architecture for a Real-Time Data Store", International Journal of Information Engineering and Electronic Business (IJIEEB), I. 1, pp. 21-28, Modern Education and Computer Science (MECS), January 2016.
- [58] A. D. Senthil Kumar, T. S. Anandhi, and Ranganath Muthu, "Secure Video Compression and Integration Technique for Data Transmission", International Journal of Computer Network and Information Security (IJCNIS), I. 11, pp. 63-70, Modern Education and Computer Science (MECS), November 2017.
- [59] Rasha Subhi Ali and Alaa Kadhim F., "Security Protocol of Keys Management System for Transmission Encrypted Data", International Journal of Computer Network and Information Security (IJCNIS), I. 1, pp. 10-17, Modern Education and Computer Science (MECS), January 2018.
- [60] Sunil Kumar, C. Rama Krishna, and A. K. Solanki, "Error Prone Transmission System to Resist Data Loss in a Wireless Sensor Network", International Journal of Computer Network and Information Security (IJCNIS), I. 11, pp. 17-26, Modern Education and Computer Science (MECS), November 2017.

### Authors' Profiles



**Vasileios A. Memos** received the B.Sc. degree in Technology Management and the M.Sc. degree in Applied Informatics (with specialization in Computer Systems and Network Technologies) from School of Information Sciences of University of Macedonia (Thessaloniki, Greece), in 2012 and 2017 respectively. His main research interests include computer networks, wireless and mobile communications, cloud computing, wireless sensor networks, internet of things, network and computer security, cryptography, privacy and security software testing.

**How to cite this paper:** Vasileios A. Memos, "Efficient Multimedia Transmission over Scalable IoT Architecture", International Journal of Computer Network and Information Security(IJCNIS), Vol.10, No.6, pp.27-39, 2018.DOI: 10.5815/ijcnis.2018.06.03