A DASH-based Performance-oriented Adaptive Video Distribution Solution

Lejla Rovcanin and Gabriel-Miro Muntean
Performance Engineering Laboratory, Network Innovations Centre, RINCE Institute
School of Electronic Engineering, Dublin City University, Ireland
lejla.rovcanin2@mail.dcu.ie and munteang@eeng.dcu.ie

***Abstract*— Nowadays increasing amounts of video-based content is stored and distributed across networks for diverse use, including learning. These networks also carry multiple other traffic types, all of which are growing in intensity. In the context of limited bandwidth resource networks, content adaptation and in particular video adaptation is seen as a very promising solution. As the newly introduced Dynamic Adaptive Streaming over HTTP (DASH) standard supports video adaptation, but does not introduce any adaptation mechanism, this paper proposes a DASH-based performance oriented Adaptive Video distribution solution (DAV) which improves video delivery mechanism to ensure better viewing experience. DAV's goal is to increase user Quality of Experience (QoE) by considering various factors, including characteristics of the links connecting video providers and the local network, quantity of requested content available locally, and device and user profiles. This solution dynamically selects best performing sources (among remote hosts and local nodes) for the delivery of video segments. Preliminary evaluation shows that DAV increases the quality of delivered video in comparison with a classic DASH video distribution approach.**

***Index Terms*—Video distribution, networking, device-orientation, Quality of Experience.**

I. Introduction

THE global community of online video viewers is expanding rapidly with the proliferation of low cost video acquisition and multimedia-enabled mobile devices. Video delivery over the Internet is becoming ubiquitous (i.e. the video-on-demand traffic will triple by 2016 [1]), while a sharp increase (25-fold) is predicted for mobile video in the near future [2]. Among many other areas in which the online distribution of video content is expanding, education is one of the most important. Production of educational video content is growing cheaper and faster (e.g. lecture recordings, student videos, etc.) and is easily made available online. There are already many offerings of free educational video content, including Coursera (www.coursera.org), Udacity (www.udacity.com) and edX (www.edx.org). At the same time, university classes are getting larger and more interactive, and students frequently use university-provided computers and personal portable devices (e.g. smart phones, netbooks, tablets, etc.) to access multimedia-based learning content.

The high data rate of the video content and the large number of viewers (e.g. class, university campus, etc.) **impose high demands on the delivery network,** which may result in long delays, high loss, etc. potentially affecting viewer Quality of Experience (QoE) levels. At the same time, as the number of videos available for download or streaming is growing rapidly and their quality is increasing in response to user demand, not all viewing devices (e.g. handheld devices) are capable of receiving, storing and playing the same (large) amount of video content at highest quality. However the users expect steady non-interrupted streaming of video data and short pre-buffering periods, regardless of viewing device type and network characteristics. There is a clear need to adjust video content selection and delivery both to network conditions and to device characteristics in order to improve user-perceived quality levels and make a positive impact on the overall viewing experience. This is particularly important in an educational setting, as video viewing is a part of the learning process. Various adaptive delivery solutions have already been proposed for different media [3] [4] but majority focus on video delivery adaptation [5] [6] [7], which is most difficult due to both size and timing constraints of video context. Recently, Dynamic Adaptive Streaming over HTTP (DASH) [8] has been shown to be best suited for content originating at standard HTTP web servers.

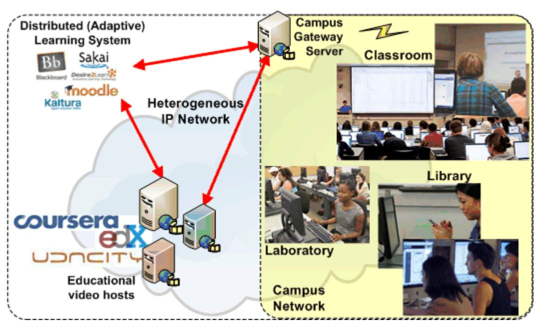

Fig. 1. Video Delivery in a University Setting

In this context, a **DASH-based performance oriented Adaptive Video distribution system (DAV)** is proposed to enhance the performance of existing systems for personalised distribution of learning content. DAV considers both device and network issues and improves the content delivery process thereby increasing the video viewing experience. User QoE levels are increased by considering a number of factors, including characteristics of the links connecting video providers/hosts and the local network, amount of content already available locally, user device characteristics and user profile (e.g. student learning profile). DAV is DASH-based as it assumes that the Web servers already host video content of varying quality in segmented form, enabling video delivery adaptation. DAV involves two stages. First, it clusters local nodes based on user profile information provided by a personalised system. Second, it considers current network and local device characteristics, as well as data collected over time to select the most suitable host for segment delivery.

The proposed solution is illustrated in a university campus setting in Fig. 1, but it could be easily applied to situations where a coincidence of demands for similar video content emerges in a large group of users interacting with a personalised system, such as a personalized video retrieval system in a corporate network or at a public performance.

This paper continues with a related works section, followed by a description of the principles behind the proposed solution and its architecture in Section III, Section IV describes the algorithm employed by DAV, whilst Section V presents and analyses simulation results. Section VI concludes the paper.

II. Related Works

A. Device-aware Video Delivery for Learning

Today's learners are using a wide range of devices to obtain and interact with learning material. However, learners report discontent with the limitations of handheld devices in terms of: size and weight; inadequate memory; short battery life [9]; limited storage space; and slow connectivity [10]. While the need for end user device adaptation is recognized [11], very few adaptive hypermedia learning systems consider it. Exceptions include APeLS [11] which tailors both the appearance and navigational structure of the learning experience to match the current environment of the learner and MAS-SHAAD [12] which dynamically generates XHTML pages considering device characteristics. In [13] the content authoring process is enriched by considering the end-user device characteristics, such as display resolution, battery power, colour depth, CPU power and multimedia support.

B. QoE-aware Video Delivery for Learning

QoEAHA [14] is a QoE-aware system, that infers information about the learner perception of the delivery performance (a stereotype-based technique) to provide suggestions on properties of the content to be served to enhance viewing experience. Although it improves perceived system usability and overall experience, it does not consider video transmission. The Mobile Mathematics Tutoring (MoMT) system performs contextual content adaptation using transcoding taking into account learner and device characteristics in the process, but does not involve video content adjustment [15].

C. Local Content Aware Video Delivery for Learning

The agent-based P2P system proposed in [16] divides multimedia data into fragments managed by assigned agents. Each agent is aware of the locations of agents managing both the preceding and trailing media fragments. However, unlike the proposed DAV, this system pushes fragments of multimedia content to all nodes joining the network.

D. Streaming over HTTP

Many deployed adaptive multimedia streaming solutions are based on HTTP [17], which traverses firewalls and NAT devices and supports progressive download, allowing play out of incompletely downloaded videos using simple players or HTML5-enabled browsers. While video servers can provide multiple versions of a video (e.g. bit rate, resolution, colour depth, level of detail; ranging from lower quality renditions for 3G connections, up to HD quality), thus meeting the requirements of heterogeneous viewing devices.

The Dynamic Adaptive Streaming over HTTP (DASH) [8] supports video streaming using consecutive downloads of short video segments and addresses the problems with traditional approaches to web streaming such as RTP/RTSP-based streaming and progressive download. Unlike progressive download, DASH supports dynamic bitrate switching and live media services. It leverages existing HTTP based multimedia content delivery infrastructure, such as Web Servers, HTTP caches and Content Delivery Networks (CDNs) without the need for specialised servers (e.g. Flash Media Server).

DASH Media Presentation is a sequence of one or more *Periods* (temporal sections) containing one or more *Adaptation Sets*. *Representations* (content alternatives) are grouped into Adaptation Sets and consist of media segments of predefined duration (e.g. 6 seconds). At most one Representation within an Adaptation Set is selected to compose a presentation. DASH *Media Presentation Description (MPD)* is an XML document which describes media content and among other information provides the relationship between the segments which form media presentations [18] and the media-component locations in the network (e.g. remote servers). MPDs are provided by Media Presentation Servers and can be downloaded in multiple steps. DASH video streaming is controlled by the client, which requests content quality that matches initial delivery conditions (i.e. available bandwidth, buffer size, remaining battery life) without the need to negotiate with the streaming server. Similarly, no negotiation is required when the client seeks to seamlessly change the content rate on the fly. After a segment is received, the client simply requests (via HTTP) the next segment of the quality that matches changes in the device state, network traffic or user preferences [18]. The DASH standard specifies media description formats, but not client behavior and adaptation mechanisms.

Examples of similar vendor-specific implementations include Adobe Dynamic HTTP Streaming [19], Apple HTTP Live Streaming [20] and Microsoft Smooth Streaming [21].

E. Modified MPD DASH Streaming

The peer-assisted DASH system proposed in [22] modifies

MPD files, giving the client an option to download parts of segments (chunks) from randomly selected peers which have the segments cached. However, unlike this system, DAV considers peer hosts inside a local network, and uplink characteristics do not have to be considered. Furthermore, DAV selects the best performing hosts for inclusion in the modified MPD, based on the local host ranking. Apart from simplifying the decision making process at the client, limiting the number of alternative hosts listed per segment also reduces the size of the MPD file.

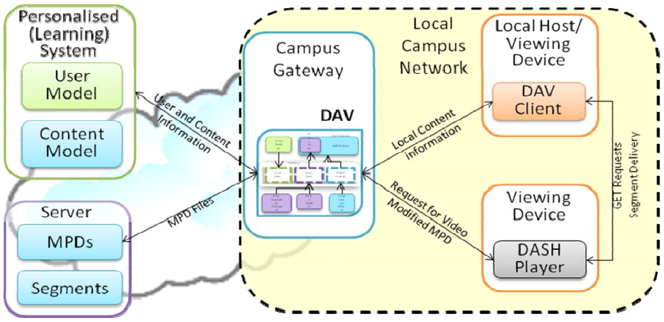

Fig. 2. DAV System Architecture

III. DAV System Architecture

The high level DAV system architecture illustrated in Fig. 2. is composed of diverse networked *Client Devices* which request video streams, a *Campus Network* to which these users are connected, a *Campus Gateway* at which level the DAV adaptive solution is deployed, distributed *Servers* which store and serve video content and *the Internet* which enables connectivity between users and hosts. DAV operates in conjunction with a Personalized Learning System (PLS) that provides learner/student specific information, such as WHURLE 2.0 [23]. DAV is proposed to enhance user QoE levels by performing DASH-based adaptive video delivery by selecting the best performing source for the delivery of each segment. The source can either be one of the servers storing video segments belonging to the requested video stream or other client devices located within the campus network.

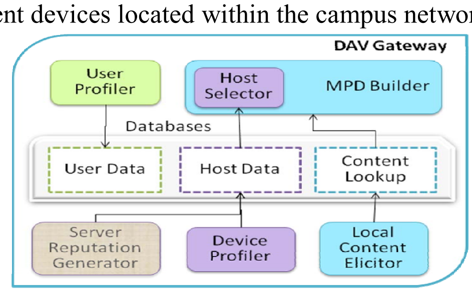

Fig. 3. Detailed DAV System Architecture

As DAV is located at the level of the campus gateway and deals with all the requests coming from the campus users for video-based learning content, it has access to a variety of information, including **network-related data** (performance characteristics related to the links connecting the remote hosts and the campus network), **video content-related data** (e.g. the number of segments available locally), and **user context-related information** (i.e. device characteristics and user preferences). The DAV components illustrated in Fig. 3 are outlined below.

A. *Server Reputation Generator*

The Server Reputation Generator component is based on the Performance Oriented Adaptation Agent (POAA) [24], built under the assumption that at any point in time, a number of videos (learning objects) with the same learning objectives exist and that they are stored on distributed media servers. The hosting repositories differ in response time and availability (varying performance, load, etc.) as well as the quality of connection (e.g. throughput, delay). This module generates the Server Reputation, which is inferred from historic performance information gathered over a number of recent sessions with the servers that store the requested video content. This reputation is used to select the best performing server where requested content is available from multiple sources.

B. *User and Device Profilers*

Group adaptation takes into account actions undertaken by other users belonging to the same (manually or automatically created) group [25]. Here, partial group based adaptation is performed. The User Profiler performs the user clustering based on the (a) enrolling course and (b) learner's profile as provided by the PLS. Each user cluster CU_x, $x \in [0, M]$ groups users from the same course and having the same learning profile. PLSs maintain user/learner profiles in User Model components, where user profiles are mainly grouped into classes based on their characteristics, such as learning style, goals, background knowledge, etc. Students having similar learning profiles are interested in similar learning content, while students doing the same course will be more likely to request/store the same video content at the same time. This node filtering reduces the number of local nodes that are considered for potential content retrieval, leads to faster decision making and increases overall solution performance.

DAV identifies capabilities of the current viewing devices (storage, memory, display resolution) using the Wireless Universal Resource File (WURFL) Device Description Repository [26]. This is a static approach, where the user agent string provided with the request is used to query the database to obtain the device capabilities. Well resourced nodes (e.g. PCs and laptops) in the campus network act as video segment providers to other devices (e.g. smart phones and tablets).

C. *Local Content Elicitor*

The gateway-located Local Content Elicitor is aware of nodes that recently downloaded video segments. It monitors well resourced nodes currently in the network and recently or currently downloading segments of the requested video. This data together with the bitrate and the time of segment download, as provided by DAV clients, are stored in the Content Lookup database.

Whilst DAV facilitates sharing content among peers, it is not a classical peer-to-peer system [27] and peers are not aware of each other in the sense that there is no distributed hash table, nor do nodes implement gossip/flood mechanisms. The Content Elicitor could be extended with an external solution such as the one proposed in [28] that uses a grouping-based storage strategy distributing the video segments along a Chord overlay which would allow DAV to use peer devices outside the campus network.

D. Media Presentation Description Builder – MPD Builder

The MPD Builder module dynamically creates a new MPD for the requested video based on MPDs provided by the remote servers hosting the video and on the host performance data. When content is available from multiple sources, the best performing server according to Server Reputation Generator is selected as the BaseURL at MPD level. For each segment within a video, the MPD Builder takes the *n* best performing local hosts to add Representation level BaseURLs to the new MPD file (using one period per segment). When a segment is unavailable locally the client reverts to the server. MPD Builder operation is presented in Section IV.

E. DAV Databases

DAV Databases store all data necessary for the MPD Builder. Some of the information originates from the external personalised system (i.e. PLS), while other information is inferred or collected by DAV components. PLS provides information about the video that matches the current learner request (e.g. URLs of external servers hosting the video based on the PLS Content Model) and learner profile (required for user profile clustering). Other systems, such as [29] maintain user information, but this increases computational complexity and adaptation time. DAV does not maintain detailed information about user profiles, as it changes over time (e.g. when a user learns a new topic). Instead, DAV makes use of the sophisticated user models already implemented within current PLSs and keeps only information necessary for user clustering in the *User Data database*.

The *Host Data database* maintains information about the local active nodes (both well and limited resourced ones). Remote server-related information such as inferred throughput and delay are dynamically updated after each interaction with the remote servers and stored here as well.

The Content Lookup component contains information about the video segments stored locally and information about currently viewed videos.

F. DAV Client

The DAV client side module (a) accepts requests and sends the requested content to another node in the network (i.e. acting as a simple Web server) (b) reports information on locally stored segments to the DAV Gateway after each segment download and (c) plays video content. The DAV client player is aware of content location through the MPD and chooses content from local hosts when available. For each segment, it parses the MPD file, to find the best quality available locally (local URL). If a given segment is not found locally, the client will request it from the remote server, using the bandwidth estimation formula given in equation (1) to determine the bitrate of the next requested segment - B_n. T_M denotes the throughput for the last downloaded segment, whilst Tp_X denotes the average throughput over the X previous requests, and is introduced to make the client less sensitive to short lived fluctuations in the server connection throughput.

TABLE I
ALGORITHM OUTLINE FOR MPD GENERATION

Input: original MPD file and collated host statistics

Output: new MPD file

1. If multiple servers exist, select the best performing one according to Server Reputation Generator to set the BaseURL at MPD level
2. For each segment in the video sequence
   For each bitrate supported in the original MPD
      Select best performing *local host* within the cluster according to equation (2) if not available locally specify server
      Add the corresponding URL to the MPD.

$$B_n \approx w \cdot Tp_X + (1-w) \cdot T_M \qquad 0 < w < 1 \tag{1}$$

Users without a DAV Client installed interact with the DAV Gateway when they request learning content from the PLS. They receive a version of the relevant MPD file that depends on the time of their request. However, their cached content cannot be used (added to the modified MPDs) by other nodes without DAV Client functionality. A problem may arise where a typical DASH player bases a request to a remote server on the latter local throughput estimate. To alleviate the problem, the DAV Gateway introduces a bitrate ceiling based on the current estimated throughput to the remote servers, so that the player may not request content from the server at a bitrate higher than this ceiling. This is implemented in the modified MPD file.

IV. MEDIA PRESENTATION DESCRIPTION BUILDER

A. DAV Functionality

DAV enhances user interaction with a PLS, by intercepting requests to the PLS.

(1) During **log in**, DAV identifies user's *device capabilities* and *connection type*, and then it forwards the request to the PLS. The PLS provides information on *user's enrolling course* and *profile* (maintained by the PLS User Model) which are used for user-based clustering.

(2) On the **initial content request**, the PLS provides information about the *content* – the relevant and suitable video based on: student learning objective, enrolling course and profile. All videos recommended by the PLS could reside on a number of remote servers and this is reflected in the list of URLs provided by the PLS. Once DAV receives the list of URLs for the recommended video, it obtains the relevant MPDs from the hosting servers. For initial requests for a particular video, the MPDs are forwarded to requesting nodes, and the content is requested from the remote servers. For the subsequent requests for the same video, DAV MPD Builder

constructs a *new MPD* using the best performing nodes chosen by the Host Selector. The *new MPD* is then forwarded to the user, so that the user can request video segments based on this latest information, as indicated in Fig. 4.

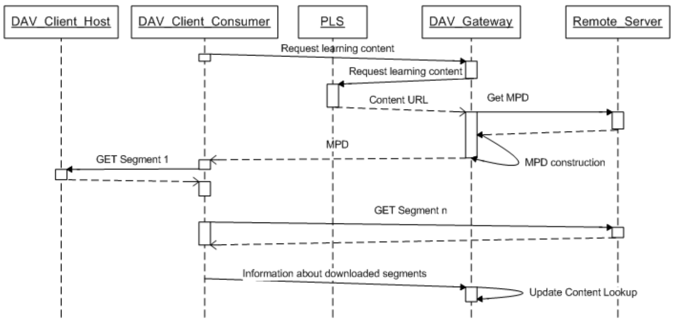

Fig. 4. DAV Sequence Diagram

DAV Gateway uses MPDs provided by the remote servers to build a new MPD based on the information stored in Content Lookup and Host Data databases.

B. *DAV Host Selection*

DAV Gateway composes new MPD constituents, selecting video content from the hosts that are currently rated as the most efficient providers of the requested segments. The Host Selector uses a utility function-based approach to calculate scores for local hosts - L_j (equation (2)) and its functionality can be extended with the scores generated by Server Reputation Generator, when content resides on multiple remote servers.

$$L_j = nC_j \cdot dT_j$$

$$nC_j = \begin{cases} 0 & n > dL_j \\ \frac{1}{n} & n \le dL_j \\ 1 & n = 0 \end{cases} \qquad dT_j = \begin{cases} 1 & |t_c - t_d| \le \varepsilon \\ \frac{\varepsilon}{t_c - t_d} & |t_c - t_d| > \varepsilon \end{cases} \tag{2}$$

In equation (2), nC_j is the utility function component for node j ($1 \le j \le M$), and reflects the estimated number of current connections – *n*. It depends both on the type of device (based on the device profile and related to the overall processing power of the user device) and on the device connection type. The devices are grouped into three classes as proposed in [13]: i.e. *Large Screen*, *Portable* and *Handheld* and the connection types considered are wired and wireless. The two attributes are combined into five classes: $dC_j \in$ {*Handheld-Wireless*, *Portable-Wireless*, *Portable-Wired*, *LargeScreen–Wireless*, *LargeScreen-Wired*} and the limiting number of simultaneous connections is assigned to each class, $dL_j = f(dC_j)$. dT_j is a utility function component that considers the freshness of the content stored at the node j. The value of dT_j ranges from 1 for nodes downloading the segment within the past ε seconds, and it reduces as time passes. The local hosts with the highest scores are used for the MPD composition as given in Table I.

V. Simulation-based DAV Evaluation

A. *Simulated Environment*

The simulations in this paper consider a wired campus (local network) and remote server connections, as illustrated in (Fig. 5). The NS-3 network simulator [30] is used for modeling and simulations. Data transmission performance during the simulations is measured in terms of: the *Join time* (the time that lapses from the initiation of the connection until the client buffer reaches playout level - the segment length of data); *Buffering ratio* (the relative time spent in re-buffering, calculated as the total time of buffer starvation over the total length of playout including pauses for re-buffering); *Rate of buffering events* (relative frequency of induced interruptions calculated as the number of buffering events over the playout time); and *Average bitrate* (the average of bitrates played), as proposed in [31]. The readings are collected for two scenarios.

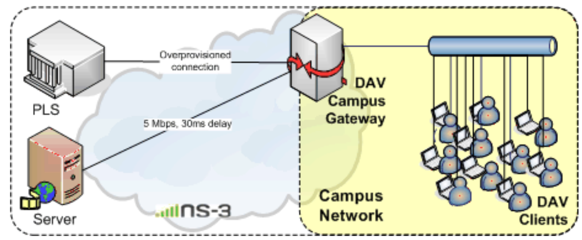

Fig. 5. Test Setting

We simulate a session where during a lecture, a group of 100 students was asked to watch a 2 minute video clip. The PLS assigns the video based on student learning profile, and a group of 20 students need to watch the same video clip using their laptops or university provided PCs (dL level used is *f*(*LargeScreen-Wired*) = 10, ε = 20 seconds). The segment bitrates range from 400, 700, 1200, 2000 to 3000Kbps. A bottleneck link of 5Mbps and 30 ms between Remote Server and DAV Gateway is imposed. Simulations were performed with 20 clients, each requesting 20 video segments. These segments are 6 seconds long. All simulated client buffers were monitored to determine interruptions (when the buffer level drops below the level of 0.2 seconds of video data). The clients use the bandwidth estimation formula given in equation (1). Specific values used in the simulation are w = 0.33 and X=5.

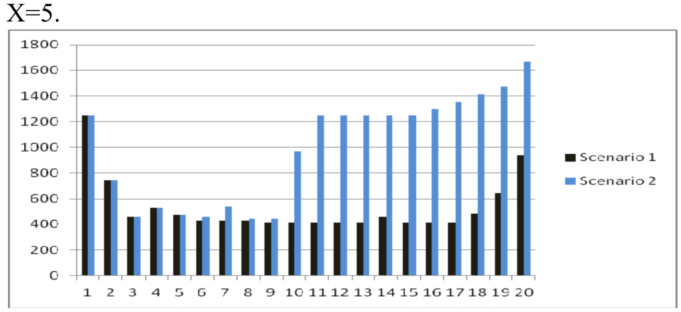

Fig. 6. Average Bitrate per Client in Kbps

Scenario 1 involved all clients requesting all content from the remote server and represents the classical DASH situation. Scenario 2 considered utilization of local nodes as content providers by 10 of the 20 clients. In both scenarios, the clients

TABLE II
SIMULATION RESULTS

Scenario	Join Time (seconds)	Rate of Buffering Events	Buffering Ratio	Average Bitrate (Kbps)
1	8.63	0.192	0.067	530
2	3.42	0.035	0.014	987

are sending their initial request at 10 seconds intervals.

The results of this evaluation involving one remote server, and hence without the use of Server Reputation Generator, are given in Table II. It can be observed that in Scenario 2, where local content is made available to clients, the overall performance is enhanced: join time has decreased on average with 5 seconds per client, rate of buffering events and buffering ratio are reduced to one fifth as compared to Scenario 1, whilst the average bitrate almost doubled (Fig. 6). In Scenario 1, all segments are requested from the Server, so each segment is requested 20 times (once for each client), while in Scenario 2, the average number of requests per segment is reduced to 12. Fig 8. indicates that the average bitrate per segment requested is improved for most of segments.

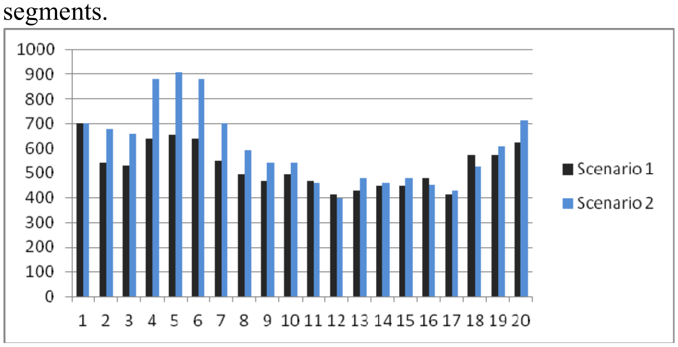

Fig. 7. Average bitrate (Kbps) per segment requested from the Server

VI. CONCLUSION

The latest diversification of video viewing device characteristics and network connectivity can be addressed by making use of adaptive HTTP streaming technologies such as DASH that introduce client controlled delivery of video segments of different quality. Communities of video service consumers that have similar demands (e.g. the same video) and that are in close geographical proximity (e.g. campus LAN), might introduce needless demands on the video hosting servers and the communication link between the servers and users. In this case, identical/similar video content would be requested and delivered multiple times to users. The DASH-based solution proposed in this paper, DAV, enhances both user experience and the performance of existing personalized distributed video delivery systems. DAV considers viewer preferences (i.e. learner profiles), viewing device capabilities and utilizes content available locally, recruiting groups of active (watching) learners within the campus to share their partial copies of the video stream with other peers in the local network. This solution requires no modification of the HTTP servers hosting video content. Furthermore, both DAV client enabled and devices without a DAV client installed can benefit from the proposed approach.

As future work, the authors are planning to perform tests that employ real video streaming traffic using differently coded videos.

REFERENCES

[1] "Cisco Visual Networking Index: Forecast and Methodology, 2011-2016," *Cisco*. [Online]. Available: http://www.cisco.com/en/US/solutions/collateral/ns341/ns525/ns537/ns705/ns827/white_paper_c11-481360_ns827_Networking_Solutions_White_Paper.html [Accessed: 8-May-2013].

[2] "Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update, 2011–2016," *Cisco*. [Online]. Available: http://www.cisco.com/en/US/solutions/collateral/ns341/ns525/ns537/ns705/ns827/white_paper_c11-520862.html [Accessed: 8-May-2013].

[3] C. H. Muntean, J. McManis, and J. Murphy, "The Influence of Web Page Images on the Performance of Web Servers," in in *Networking — ICN 2001*, P. Lorenz, Ed. Springer Berlin Heidelberg, 2001, pp. 821–828.

[4] C. H. Muntean and J. McManis, "A QoS-aware adaptive Web-based system," in *2004 IEEE International Conference on Communications*, 2004, vol. 4, pp. 2204 – 2208.

[5] G.-M. Muntean and L. Murphy, "Adaptive pre-recorded multimedia streaming," in *IEEE Global Telecommunications Conference, 2002. GLOBECOM '02*, 2002, vol. 2, pp. 1728–1732.

[6] G.-M. Muntean, G. Ghinea, and T. N. Sheehan, "Region of Interest-Based Adaptive Multimedia Streaming Scheme," *IEEE Transactions on Broadcasting*, vol. 54, no. 2, pp. 296–303, 2008.

[7] G.-M. Muntean, P. Perry, and L. Murphy, "A Comparison-Based Study of Quality-Oriented Video on Demand," *IEEE Transactions on Broadcasting*, vol. 53, no. 1, pp. 92 –102, Mar. 2007.

[8] "ISO/IEC 23009-1:2012 - Information technology -- Dynamic adaptive streaming over HTTP (DASH) -- Part 1: Media presentation description and segment formats." [Online]. Available: http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=57623 [Accessed: 8-May-2013].

[9] D. Corlett, M. Sharples, S. Bull, and T. Chan, "Evaluation of a mobile learning organiser for university students," *Journal of Computer Assisted Learning*, vol. 21, no. 3, pp. 162–170, 2005.

[10] S. So, "A Study on the Acceptance of Mobile Phones for Teaching and Learning with a group of Pre-service teachers in Hong Kong," *Journal of Educational Technology Development and Exchange*, vol. 1, no. 1, pp. 81–92, 2008.

[11] A. Brady, O. Conlan, and V. Wade, "Dynamic composition and personalization of PDA-based eLearning–Personalized mLearning," in *Proceedings of world conference on e-learning in corporate, government, healthcare, and higher education*, 2004, pp. 234–242.

[12] D. Mérida, R. Fabregat, X. Prat, D. Huerva, and J. Velez, "A Dynamic Content Generator for Adaptation in Hypermedia Systems," in in *Adaptive Hypermedia and Adaptive Web-Based Systems*, vol. 5149, W. Nejdl, J. Kay, P. Pu, and E. Herder, Eds. Springer Berlin / Heidelberg, 2008, pp. 320–323.

[13] C. H. Muntean, G.-M. Muntean, J. McManis, and A. I. Cristea, "Quality of Experience-LAOS: create once, use many, use anywhere," *International Journal of Learning Technology*, vol. 3, no. 3, pp. 209–229, Jan. 2007.

[14] C. H. Muntean, "Improving learner quality of experience by content adaptation based on network conditions," *Computers in Human Behavior*, vol. 24, no. 4, pp. 1452–1472, Jul. 2008.

[15] X. Zhao, T. Ninomiya, F. Anma, and T. Okamoto, "A context-aware prototype system for adaptive learning content in ubiquitous environment," in *IT in Medicine and Education, 2008. ITME 2008. IEEE International Symposium on*, 2008, pp. 164–168.

[16] T. Hayakawa, M. Higashino, K. Takahashi, T. Kawamura, and K. Sugahara, "Management of Multimedia Data for Streaming on a Distributed E-learning System," in *2012 26th International Conference on Advanced Information Networking and Applications Workshops (WAINA)*, 2012, pp. 1282 –1285.

[17] A. Zambelli, "IIS smooth streaming technical overview," *Microsoft Corporation*, 2009.

[18] T. Stockhammer, "Dynamic Adaptive Streaming over HTTP–Design Principles and Standards," in *Proceedings of the second annual ACM*

conference on Multimedia systems (MMSys11), ACM, New York, NY, USA, 2011.

[19] "Adobe HTTP Dynamic Streaming * Using Adobe HTTP Dynamic Streaming." [Online]. Available: http://help.adobe.com/en_US/HTTPStreaming/1.0/Using/index.html [Accessed: 8-May-2013].

[20] R. Pantos, "HTTP live streaming," 2011. [Online]. Available: http://tools.ietf.org/html/pantos-http-live-streaming-07). [Accessed: 8-May-2013].

[21] "Microsoft Smooth Streaming." [Online]. Available: http://www.iis.net/downloads/microsoft/smooth-streaming. [Accessed: 8-May-2013].

[22] S. Lederer, C. Muller, and C. Timmerer, "Towards peer-assisted dynamic adaptive streaming over HTTP," in *Packet Video Workshop (PV), 2012 19th International*, 2012, pp. 161 –166.

[23] M. Meccawy, P. Blanchfield, H. Ashman, T. Brailsford, and A. Moore, "Whurle 2.0: Adaptive learning meets web 2.0," *Times of Convergence. Technologies Across Learning Contexts*, pp. 274–279, 2008.

[24] L. Rovcanin, C. Muntean, and G.-M. Muntean, "Performance Enhancement for Open Corpus Adaptive Hypermedia Systems," in in *Adaptive Hypermedia and Adaptive Web-Based Systems*, vol. 4018, V. Wade, H. Ashman, and B. Smyth, Eds. Springer Berlin / Heidelberg, 2006, pp. 462–466.

[25] E. Knutov, P. De Bra, and M. Pechenizkiy, "AH 12 years later: a comprehensive survey of adaptive hypermedia methods and techniques," *New Review of Hypermedia and Multimedia*, vol. 15, no. 1, pp. 5–38, Apr. 2009.

[26] L. Passani, "WURFL - Mobile Device Database by ScientiaMobile." [Online]. Available: http://wurfl.sourceforge.net/. [Accessed: 13-Dec-2012].

[27] Y. Liu, Y. Guo, and C. Liang, "A survey on peer-to-peer video streaming systems," *Peer-to-Peer Netw. Appl.*, vol. 1, no. 1, pp. 18–28, Mar. 2008.

[28] C. Xu, X. Xia, J. Guan, H. Zhang, and G.-M. Muntean, "Disorder Analytic Model-Based CMT Algorithms in Vehicular Sensor Networks," *International Journal of Distributed Sensor Networks*, vol. 2013, Mar. 2013.

[29] X. Zhao, T. Ninomiya, F. Anma, and T. Okamoto, "A context-aware prototype system for adaptive learning content in ubiquitous environment," in *IT in Medicine and Education, 2008. IEEE International Symposium on*, 2008, pp. 164 –168.

[30] "Network Simulator ns-3." [Online]. Available: http://www.nsnam.org/. [Accessed: 8-May-2013].

[31] F. Dobrian, V. Sekar, A. Awan, I. Stoica, D. Joseph, A. Ganjam, J. Zhan, and H. Zhang, "Understanding the impact of video quality on user engagement," in *Proceedings of the ACM SIGCOMM 2011 conference*, New York, NY, USA, 2011, pp. 362–373.