MULTIMEDIA COMMUNICATIONS TECHNICAL COMMITTEE
IEEE COMMUNICATIONS SOCIETY
http://www.comsoc.org/~mmc

E-LETTER

Vol. 10, No. 6, November 2015

IEEE COMMUNICATIONS SOCIETY

CONTENTS

Message from MMTC Chair 3
EMERGING TOPICS: SPECIAL ISSUE ON CLOUD-AWARE MULTIMEDIA SYSTEMS 5
Guest Editors: Kuan-Ta Chen, Academia Sinica Ali C. Begen, Cisco Chin-Feng Lai, National Chung Cheng University 5
swc@iis.sinica.edu.tw, acbegen@ieee.org, cinfon@cs.ccu.edu.tw 5
Cloud-based Transcoding and Adaptive Video Streaming-as-a-Service 7
Christian Timmerer†‡, Daniel Weinberger‡, Martin Smole‡, Reinhard Grandl‡, Christopher Müller‡, Stefan Lederer‡ 7
‡bitmovin GmbH, Klagenfurt, Austria, {firsname.lastname}@bitmovin.net 7
†Alpen-Adria-Universität Klagenfurt, Institute of Information Technology (ITEC), Austria christian.timmerer@itec.aau.at 7
Transcoding in the Cloud: Optimization and Perspectives 12
Ramon Aparicio-Pardo, Gwendal Simon° and Alberto Blanc°* 12
** University of Nice, ramon.aparicio-pardo@unice.fr* 12
°Telecom Bretagne, France, firstname.lastname@telecom-bretagne.eu 12
Delay Reduction in Cloud Gaming 16
Shervin Shirmohammadi 16
Distributed and Collaborative Virtual Environments Research (DISCOVER) Lab 16
University of Ottawa, Canada, ... 16
shervin@eecs.uottawa.ca 16
Cloud-based System for Large-Scale Video Analysis from Camera Networks 20
Wei-Tsung Su[1] and Yung-Hsiang Lu[2] 20
[1] Aletheia University, Taiwan (R.O.C.), au4451@au.edu.tw 20
[2] Purdue University, USA, yunglu@purdue.edu 20
INDUSTRIAL COLUMN: SPECIAL ISSUE ON CLOUD GAMING 24
Guest Editors: Gwendal Simon[1] and Adlen Ksentini[2] 24
[1]Télécom Bretagne, France, gwendal.simon@telecom-bretagne.eu 24
[2]University of Rennes 1, France, adlen.ksentini@irisa.fr 24
QoE for Cloud Gaming 26
Tobias Hoßfeld[1], Florian Metzger[1], Michael Jarschel[2] [1]University of Duisburg-Essen, Modeling of Adaptive Systems, Germany 26
{tobias.hossfeld, florian.metzger}@uni-due.de 26

[2]Nokia, Munich, Germany, michael.jarschel@nokia.com 26
Enhancing Cloud Gaming with Software Defined Networking 30
Shervin Shirmohammadi 30
Distributed and Collaborative Virtual Environments Research (DISCOVER) Lab 30
University of Ottawa, Canada, shervin@eecs.uottawa.ca 30
Optimizing Cloud Gaming Experience and Profits with Virtual Machine Placement Policy 33
Hua-Jun Hong[1], De-Yu Chen[2], Chun-Ying Huang[3], Kuan-Ta Chen[2], and Cheng-Hsin Hsu[1] 33
[1]Department of Computer Science, National Tsing Hua Univiersity, Hsin Chu, Taiwan. 33
[2]Institute of Information Science, Acdemia Sinica, Taipei, Taiwan. 33
[3]Department of Computer Science and Engineering, National Taiwan Ocean University, Kee Lung, Taiwan. 33
Uniquitous: an Open-source Cloud-based Game System in Unity 37
Meng Luo and Mark Claypool 37
Computer Science and Interactive Media & Game Development 37
Worcester Polytechnic Institute, Worcester, MA 01609, USA 37
{mluo2,claypool}@wpi.edu 37
Advanced GPU Pass-through and Cloud Gaming Performance: A Reality Check 41
Ryan Shea and Jiangchuan Liu 41
Simon Fraser University 41
{rws1, jcliu}@sfu.ca 41
Position Paper 45
An Overview of Recent Research in Content-Centric Networking 45
Anand Seetharam[1] and Shiwen Mao[2] 45
[1]Computer Science Program, California State University Monterey Bay, Seaside, USA 45
[2]Department of Electrical and Computer Engineering, Auburn University, Auburn, USA ... 45
aseetharam@csumb.edu, smao@ieee.org 45
Call for Papers 48
Quality of Experience-based Management for Future Internet Applications and Services (QoE-FI 2016) 48
23rd International Conference on Telecommunications (ICT 2016) 49
MMTC OFFICERS (Term 2014 — 2016) 50

Message from MMTC Chair

Dear MMTC friends and colleagues,

Time flies! It is my turn again to provide a message for the November issue of E-Letter, which reminds me that I have already served nearly three quarters of my term. It is a great honour and pleasure to serve as vice Chair-Letters & Member Communications for 2014 ~ 2016. In the past one and half years, I deeply enjoyed working with the MMTC officers, the E-Letter, R-Letter, and Membership boards, the IGs, and MMTC members to serve the MMTC community and to continue the past success of MMTC. Thank you all for your collaboration and support!

I would like to take this opportunity to provide an update of the E-Letter, R-Letter and Membership boards. The E-Letter and R-Letter boards, led by Drs. Periklis Chatzimisios and Christian Timmerer, respectively, have been working diligently with the IGs to publish special issues on hot related topics and review the top papers in the field. I am pleased to announce the 2015 MMTC Excellent Editor Awards awardees:

- Dr. Kan Zheng, Beijing University of Posts & Telecommunications, Bejing, China, , E-Letter Editor
- Dr. Pradeep Atrey, State University of New York, Albany, NY, USA, R-Letter Editor

Please join me to congratulate Drs. Zheng and Atrey for this well-deserved recognition and thank them for their hard work.

In addition, the Letter Boards are working on collaborations with MMTC sponsored conferences such as ICME and CCNC, and with related journals including IEEE Transactions on Circuits and Systems for Video Technology (CSVT) and IEEE Multimedia. To beef up the impact of E-Letter, we are soliciting original position papers. If you are organizing a panel at a conference, or a workshop on a related topic, we encourage you to submit a short paper summarizing the dynamics and discussions at the event, and get it published at E-Letter. If you have any suggestions on how to promote the letters, please do not hesitate to contact me or the Letter Directors.

A new initiative, with help from the executive team, in particular, Drs. Yonggang Wen and Fen Hou, is to create a Newsletter Editor position, whose job is to collect MMTC related news items, such as call for papers, job openings, nominations, and announcements, and edit these into a weekly newsletter to distribute to all MMTC members. This way, the MMTC related email traffic will be greatly reduced, and it is also much easier for our members to check for such information from the Newsletter page at the MMTC website. It is my great pleasure to introduce our first Newsletter Editor, Dr. Mugen Peng, to you. Dr. Peng is a Full Professor at Beijing University of Posts & Telecommunications, Beijing, China. In the past month, you probably have already received the weekly newsletters edited and sent by Dr. Peng. Let's thank him for the excellent work done!

Our Membership Board, led by Drs. Zhu Liu, Lifeng Sun, and Laura Galluccio, have been working on streamlining the membership subscription procedure. The past procedure was quite cumbersome and has not been helpful to attract more members. With help from Dr. Dalei Wu, the new subscription site is working now.

A new member can click the link "click here" in the Membership Board page at: http://committees.comsoc.org/mmc/membership.asp, and then enter his/her name and email address in the next page to become an MMTC member.

Note that to become an MMTC member, no IEEE or IEEE Communications Society membership is required. In other words, anyone who is interested in multimedia communications can subscribe and become an MMTC member. Please spread the word and encourage your friends, colleagues, and more important, students to subscribe. I am sure your students will greatly benefit from the interaction with the MMTC community and participation in MMTC events.

I hope you enjoy reading this E-Letter issue, and strongly encourage you find the IG of interest to get involved and to contribute to future E-Letter special issues. If you have any suggestions or comments on improving the E-Letter, R-Letter and Membership boards, please do not hesitate to contact me or the Board Directors.

Sincerely,

Shiwen Mao
Vice Chair—Letters & Member Communications
Multimedia Communications Technical Committee, IEEE ComSoc

EMERGING TOPICS: SPECIAL ISSUE ON CLOUD-AWARE MULTIMEDIA SYSTEMS

Guest Editors: Kuan-Ta Chen, Academia Sinica
Ali C. Begen, Cisco
Chin-Feng Lai, National Chung Cheng University

swc@iis.sinica.edu.tw, acbegen@ieee.org, cinfon@cs.ccu.edu.tw

Multimedia cloud computing is an emerging area that involves multimedia communications, applications, and services using / on the cloud. Here by cloud we refer to to a shared pool of configurable computing resources that can allow ubiquitous, convenient, on-demand access and helps with multimedia computing in different aspects. For example, a straightforward use is to rely the powerful, scalable computing power of cloud to faciliate multimedia content encoding and processing; also, the cloud can be used to speed up and/or improve the quality of multimedia communications by serving as a relay node on the transmission path. Due to much use of the cloud in multimedia computing, the provisioning of cloud infrastructure, resource allocation, network routing, and QoE management are also important issues in this emerging field.

This special issue of E-Letter focuses on the recent progresses of cloud-aware multimedia applications and systems. It is the great honor of the editorial team to have four leading research groups, from both academia and industry laboratories, to report their innovations for developing novel methodologies and solutions in addressing the challenges in providing quality cloud-aware multimedia systems.

In the first article titled, "Cloud-based Transcoding and Adaptive Video Streaming-as-a-Service", Timmerer *et. al* presented research that led to the deployment of *bitcodin*, a live transcoding and streaming-as-a-service platform using the MPEG-DASH standard which is used for both live 24/7 and event-based temporary services and *bitdash*, an adaptive client framework. They have shown that with the proposed live transcoding and streaming-as-a-service – *bitcodin* – is able to exploit the flexibility and elasticity of the cloud to provide scalability on demand for both live 24/7 services and event-based streaming for a limited time period. In addition, the presented streaming client offers a high average media throughput without stalling when operating in fluctuating network environments and provides instant playback for live services with a low start-up delay as well as high Quality of Experience for the actual end user.

Aparicio-Pardo, Simon and Blanc from University of Nice and Telecom Bretagne authored the second article, "Transcoding in the Cloud: Optimization and Perspectives". The authors describe research activities on the global management of a live video streaming service running in the cloud and show that a management policy that takes into account all the inter-dependencies among technologies can bring significant advantages. In particular, they address the problem of preparing the adequate video package (the set of representations) taking into account multiple constraints. The studies show that significant gains can be obtained by implementing smart strategies for the transcoding of the videos in the context of adaptive streaming.

The third article is contributed by Shervin Shirmohammadi from University of Ottawa, and the title is *"*Delay Reduction in Cloud Gaming*"*. Cloud gaming or Gaming as a Service is one of the newest entries in the online gaming world, leverages the well-known concept of cloud computing to provide real-time gaming services to players. Despite advancing at a rapid rate, Cloud Gaming is fundamentally challenged by two main obstacles: high bandwidth requirement and strict sensitivity to network delay. In this article, the author gave an overview of delay points in a cloud gaming system, and also described some techniques for reducing delay in the cloud side, including optimzing the routing path within a data center using SDN.

Su and Lu presented a cloud-based system, Continuous Analysis of Many Cameras (CAM2, in short) in the fourth article, *"*Cloud-based System for Large-Scale Video Analysis from Camera Networks*"*. The multimedia data generated from these network cameras can be a type of big data because (1) at multiple frames per second, multimedia data pass through networks at high velocity; (2) multimedia data have wide variety; (3) and storing multimedia data requires large volume of capacity. These features indicate that cloud computing can be an appropriate solution to process multimedia big data. Thus, a cloud-based system, Continuous Analysis of Many Cameras (CAM2, in short), was proposed by the authors for harvesting valuable information embedded in multimedia data from multiple network cameras. This article reviews and reveals the research issues on cloud resource management for performance improvement of CAM2 for users and researchers, respectively.

While this special issue is far from delivering a complete coverage on this exciting research area, we hope that the four invited letters give the audiences a taste of recent activities in this area, and provide them an opportunity to explore and collaborate in the related fields. Finally, we would like to thank all the authors for their great contribution and the E-Letter Board for making this special issue possible.

Kuan-Ta Chen is a Research Fellow at the Institute of Information Science of Academia Sinica. He was an Assistant Research Fellow from 2006 to 2011 and an Associate Research Fellow from 2011 to 2015 at the Institute of Information Science, Academia Sinica. He received his Ph.D. in Electrical Engineering from National Taiwan University in 2006, and his B.S. and M.S. in Computer Science from National Tsing Hua University in 1998 and 2000, respectively.

His research interests span various areas in multimedia computing and social computing with an emphasis on user experience, multimedia systems and networking, crowdsourcing, and computational social science. He received the Best Paper Award in IWSEC 2008 and K. T. Li Distinguished Young Scholar Award from ACM Taipei/Taiwan Chapter in 2009. He also received the Outstanding Young Electrical Engineer Award from The Chinese Institute of Electrical Engineering in 2010, the Young Scholar's Creativity Award from Foundation for the Advancement of Outstanding Scholarship in 2013, and IEEE ComSoc MMTC Best Journal Paper Award in 2014. He has been an Associate Editor of *ACM Transactions on Multimedia Computing, Communications, and Applications (ACM TOMM)* since 2015. He is a Senior Member of ACM and a Senior Member of IEEE.

Ali C. Begen is with the Video and Content Platforms Research and Advanced Development Group at Cisco. His interests include networked entertainment, Internet multimedia, transport protocols and content delivery. Ali is currently working on architectures and protocols for next-generation video transport and distribution over IP networks. He is an active contributor in the IETF and MPEG, and has given a number of keynotes, tutorials and guest lectures in these areas.

Ali holds a Ph.D. degree in electrical and computer engineering from Georgia Tech. He received the Best Student-paper Award at IEEE ICIP 2003, the Most-cited Paper Award from *Elsevier Signal Processing: Image Communication* in 2008, and the Best-paper Award at Packet Video Workshop 2012. Ali has been an editor for the *Consumer Communications* and *Networking series in the IEEE Communications Magazine* since 2011 and an Associate Editor for the *IEEE Transactions on Multimedia* since 2013. He served as a general co-chair for ACM Multimedia Systems 2011 and Packet Video Workshop 2013. He is a senior member of the IEEE and a senior member of the ACM. Further information on Ali's projects, publications, presentations and professional activities can be found at http://ali.begen.net.

Chin-Feng Lai is an Associate Professor at Department of Computer Science and Information Engineering, National Chung Cheng University since 2014. He received the Ph.D. degree in department of engineering science from the National Cheng Kung University, Taiwan, in 2008. He received Best Paper Award from IEEE 17th CCSE, 2014 International Conference on Cloud Computing, IEEE 10th EUC, IEEE 12th CIT. He has more than 100 paper publications. He is an Associate Editor-in-Chief for Journal of Internet Technology and serves as Editor or Associate Editor for *IET Networks*, *International Journal of Internet Protocol Technology*, *KSII Transactions on Internet*, *Information Systems*, and *Journal of Internet Technology*. He is TPC Co-Chair for FCST 2014, ICS 2014, ICESS 2013, FC 2013, EmbeddedCom 2012, CIT 2012 and the Interest Group on Multimedia Services and Applications over Emerging Networks of the IEEE Multimedia Communication Technical Committee during 2012 and 2017. His research focuses on Internet of Things, Body Sensor Networks, E-healthcare, Mobile Cloud Computing, Cloud-Assisted Multimedia Network, Embedded Systems, etc. He is an IEEE Senior Member since 2014.

Cloud-based Transcoding and Adaptive Video Streaming-as-a-Service

Christian Timmerer[†‡], Daniel Weinberger[‡], Martin Smole[‡], Reinhard Grandl[‡], Christopher Müller[‡], Stefan Lederer[‡]

‡bitmovin GmbH, Klagenfurt, Austria, {firsname.lastname}@bitmovin.net

†Alpen-Adria-Universität Klagenfurt, Institute of Information Technology (ITEC), Austria
christian.timmerer@itec.aau.at

1. Introduction

Real-time entertainment services such as streaming video and audio are currently accounting for more than 60% of the Internet traffic, e.g., in North America's fixed access networks during peak periods [1]. Interestingly, these services are all delivered over-the-top (OTT) of the existing networking infrastructure using the Hypertext Transfer Protocol (HTTP) which resulted in the standardization of MPEG Dynamic Adaptive Streaming over HTTP (DASH) [2]. The MPEG-DASH standard enables smooth multimedia streaming towards heterogeneous devices and commonly assumes the usage of HTTP-URLs to identify the segments available for the clients [3].

More and more services are getting deployed adopting the MPEG-DASH standard and we see an increasing offer of various live events – 24/7 or special events (e.g., operas, festivals, sports) for a limited time – which are solely delivered over the open Internet without any quality guarantees. Most of these services are offered for free including advertisements, which provide service providers means for monetization. In this paper we present research that led to the deployment of *bitcodin*, a live transcoding and streaming-as-a-service platform using the MPEG-DASH standard which is – at the time of writing this paper – used for both live 24/7 and event-based temporary services and bitdash, our adaptive client framework. The system architecture is described in Section 2 and Section 3 provides details about our live transcoding and streaming-as-a-service. Quality of Experience (QoE) and client support mechanisms for live streaming are described in Section 4 and Section 5 concludes the paper. Please note that this paper has been published within IEEE ICME 2015 Industry Track [4].

2. System Architecture

The high-level system architecture bitcodin and bitdash is depicted in Figure 1. It comprises the following components (blue-rimmed modules are developed by us):

a) the actual *transcoding and streaming-as-a-service* deployed on standard cloud infrastructure (e.g., Google, Amazon, Windows, etc.) taking the live source as input and providing multiple representations (e.g., resolution, bitrate, etc.) according to the MPEG-DASH standard as output;
b) the integration within the *customer Web portal* for the actual deployment;
c) the streaming utilizing *standard delivery infrastructure* over a content distribution network (CDN); and
d) the *DASH client* implementation integrated within *heterogeneous* devices.

The transcoding and streaming-as-a-service takes the live multimedia content as an input and transcodes it to multiple content representations in real-time on standard infrastructure-as-a-service (IaaS) cloud environments according to the requirements of the customer in terms of resolutions, bitrates, etc. These requirements are expressed through an application programming interface (API) exposed to the customer. The resulting manifest describing the individual content representations and primary input for the streaming client is incorporated within the customer's Web portal offering the service to the actual clients (end users). The streaming is conducted utilizing standard CDN infrastructure. The heterogeneous clients request the multimedia segments based on the manifest received prior to the streaming and adapt themselves to the context conditions such as fluctuating network bandwidth.

Please note that our approach is explicitly targeting MPEG-DASH as the primary multimedia format representation taking into account guidelines provided by the DASH Industry Forum (DASH-IF: http://www.dashif.org). In practice, however, there is also a need to support Apple's HTTP Live Streaming (HLS) for iOS-based devices such as iPhone, iPad, and AppleTV. This is a requirement for iOS apps submitted for distribution in their App Store. Thus, we support also HLS in addition to MPEG-DASH but we hope that Apple will relax this requirement in the near future.

3. Cloud-based Transcoding and Streaming

Figure 1: High-Level System Architecture for Live Transcoding and Streaming-as-a-Service.

The core of bitcodin is the ability to take multimedia content from a live source and to transcode it in real-time (actually much faster than real-time is possible) into various content representations based on a given configuration (e.g., video profile, video quality, video resolution, audio/video bitrate). The input is typically provided using the proprietary Real-Time Messaging Protocol (RTMP) push, which is a de-facto standard used within the industry to push live content over the Internet. Other open standards such as HTTP/2 push could be a replacement but it is not yet widely available, as it has been only standardized recently [5]. Therefore, we still have to stick with RTMP push for a while.

We support a variety of input formats in terms of video, audio, and containers as well as subtitles. Our transcoding mechanism utilizes the flexibility and elasticity of existing cloud infrastructure-as-a-service (IaaS) providing scalability on demand when it is needed. In particular, cloud instances are requested and utilized depending on the demand in order to satisfy real-time requirements and even beyond, i.e., transcoding to various content representations ranging from standard definition resolution to ultra high definition resolution multiple times faster than real-time. A screen shot is shown in Figure 2, which reveals the performance of being much faster than real-time. Additionally, a preview of the results while the transcoding is still in progress is possible.

A REST API enables easy integration into existing media workflows as well as support for multiple CDNs depending on the costumer needs.

4. Client Implementation Framework

The MPEG-DASH standard defines the media presentation description (MPD) as well as segment formats and deliberately excludes the specification of the client behavior, i.e., the implementation of the adaptation logic, which determines the scheduling of the segment requests, is left open for competition. In the past, various implementations of the adaptation logic have been proposed both within the research

Figure 2: Screen Shot of Customer Portal showing 98.67x Real-Time Transcoding.

community and industry deployments/products. In any case, the behavior of the adaptation logic directly impacts the Quality of Experience (QoE) which can be defined as *"the degree of delight or annoyance of the user of an application or service. It results from the fulfillment of his or her expectations with respect to the utility and/or enjoyment of the application or service in the light of the user's personality and current state"* [6]. For DASH-based services the main QoE influence factors can be described as **initial/start-up delay**, **buffer underruns** also known as **stalls**, **quality switches**, and **media throughput**.

In order to evaluate our client bitdash we performed an objective evaluation adopting a setup similar to [7] where the bandwidth and delay between a server and client are shaped using a shell script that invokes the Unix program TC with netEM and a token bucket filter. In particular, the delay was set to 80ms and the bandwidth follows real-world bandwidth traces or a predefined trajectory comprising both abrupt and step-wise changes in the available bandwidth. The delay corresponds to what can be observed within long-distance fixed line connections or reasonable mobile networks and, thus, is representative for a broad range of application scenarios.

The test sequence is based on the available DASH dataset [8] where we adopt the Big Buck Bunny

Table 1: Comparison of Results with Commercially available Products with real-world bandwidth traces: *i)* first DASH implementation in VLC with throughput-based adaptation logic, *ii)* improvement due to HTTP persistent connections and pipelined requests, *iii)* buffer-based adaptation logic using AVC (basis for bitdash), *iv)* improvement due SVC.

Name	Average Throughput [kbps]	Switches [Number]	Stalls [s]
Microsoft Smooth Streaming	1,522	51	0
Adobe HTTP Dynamic Streaming (HDS)	1,239	97	64
Apple HTTP Live Streaming (HLS)	1,162	7	0
DASH-TPB [i]	1,045	141	0
DASH-TPB Pipelined [ii]	1,464	166	0
DASH-BB (AVC) [iii]	2,341	81	0
DASH-BB (SVC) [iv]	2,738	101	0

sequence providing representations with a bitrate of 100, 150, 200, 350, 500, 700, 900, 1100, 1300, 1600, 1900, 2300, 2800, 3400, 4500 kbps and resolutions ranging from 192×108 to 1920×1080. The configuration provides a good mix of resolutions and bitrates for both fixed and mobile network environments. We evaluated two versions, one with 2s segment length and one with 10s as these are the most common segment sizes currently adopted by proprietary deployments (i.e., Apple HLS uses 10s whereas others like Microsoft and Adobe use 2s).

A *first evaluation* of our DASH client implementation was based on the average throughput (in kbps), number of switches, and the duration in which the playback was stalled (in seconds). We have compared our implementation utilizing different strategies within real-world bandwidth traces to existing, proprietary solutions deployed on various platforms such as Microsoft Smooth Streaming, Adobe HTTP Dynamic Streaming (HDS), and Apple HTTP Live Streaming (HLS). The results are summarized in Table 1 and demonstrate the smoothness of the solutions offered by Microsoft and Apple but also issues with the implementation from Adobe which produced an increased number of stalls. In contrast, our first implementation of a DASH client adopts a simple throughput-based adaptation logic and provides comparable results to commercially available products, both in terms of throughput and stalls. The number of quality switches is shown to be higher but without impacting the QoE. It is known that QoE is impacted only when switching every second with a high amplitude (e.g. from high-to-low quality representation and vice-versa) [9]. Using HTTP persistent connections and pipelining increases the throughput, making it directly competitive with Microsoft Smooth Streaming. Finally, with a buffer-based adaptation we could even further increase the average throughput by reducing the

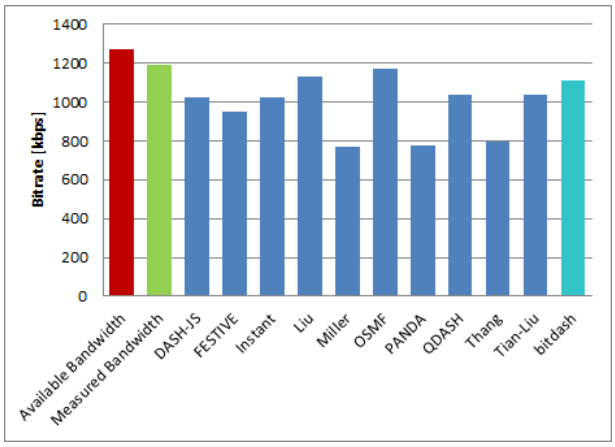


Figure 3: Average Media Throughput/Bitrate of all Adaptation Logics (higher is better).

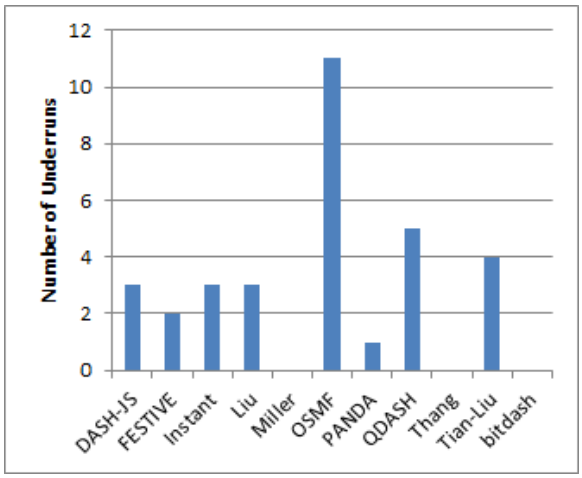


Figure 4: Number of Stalls (lower is better).

number of quality switches resulting in an overall performance much better than proprietary/commercially available solutions. When using scalable video coding (SVC) a much more aggressive adaption logic can be used due the nature of layered video coding that can be exploited during streaming [9].

The *second evaluation* comprises a comparison of our buffer-based adaptation with ten different adaptation approaches proposed in the literature using the same setup as shown before but with a predefined bandwidth trajectory. The average media throughput in terms of bitrate [kbps] is shown in Figure 3. The "Available Bandwidth" on the left side of the figure shows the average bandwidth according to the predefined bandwidth trajectory used in the evaluation. The "Measured Bandwidth" by the clients is shown next to it, which is typically a bit lower than the available bandwidth due to the network overhead. The results of the different adaptation logics are shown subsequently and our implementation – on the very right side of the

figure – is among the top three implementations, namely 1. OSMF (1170.65 kbps), 2. Liu (1129.69 kbps), and 3. bitdash (1109.43 kbps). However, taking into account the average media throughput only is a fallacy when investigating the number of stalls as depicted in Figure 4. Interestingly, **among the top three, only bitdash does not produce any stalls** whereas the client with the best average media throughput produces the highest number of stalls, obviously not good for high QoE.

Finally, for DASH-based live services the initial or start-up delay is an important aspect for which we have developed a specific solution. The initial or start-up delay comprises the time between service/content request and start of the actual playout which typically involves processing time both at the server and client, network time for sending the MPD request and receiving first segments, and initial buffer time before the playout starts. In general, the start-up delay shall be low but it also depends on the use case. For example, the QoE of live streams or short movie clips is more sensitive to start-up delay than full-length video on demand content. Therefore, we propose a solution targeting live services using the live profile which includes a live edge hint within the MPD that allows DASH clients effectively determining the live edge. In particular, the proposed solution comprises an additional attribute to be included within the SegmentTemplate element in combination with the "$Number$" identifier for URL templates. This additional attribute is called *liveEdgeNumber* and provides the latest number of the segment which has been just written by the server upon MPD request from a client. A sequence diagram of the proposed solution is shown in Figure 5.

In contrast to conventional live-edge detection methods (i.e., calculating a starting point and searching the timeline in both directions) with liveEdgeNumber, the start-up delay introduced by the live-edge calculation and/or searching the timeline would decrease to zero. This mechanism can be used for all adaptive streaming systems where segments are using a "$Number$" identifier for URL templates. We have also deployed this solution within bitcodin demonstrating its scalability in real-world deployments.

5. Conclusions

In this paper we have shown a live transcoding and streaming-as-a-service – bitcodin –, which has been specifically designed for dynamic adaptive streaming over the top of the existing infrastructure using the MPEG-DASH standard (also supporting HLS for iOS apps). By using standard infrastructure, we are able to

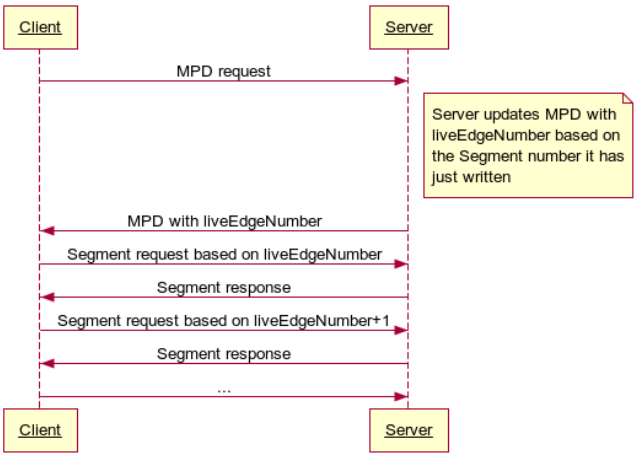


Figure 5: Sequence Diagram for liveEdgeNumber.

exploit the flexibility and elasticity of the cloud to provide scalability on demand for both live 24/7 services and event-based streaming for a limited time period. For the actual delivery we adopt standard content delivery networks without vendor lock-in enabling flexibility and stability for streaming services.

Finally, our streaming client offers a high average media throughput without stalling when operating in fluctuating network environments and provides instant playback for live services with a low start-up delay due to the proposed liveEdgeNumber included within the MPD and, thus, enables high Quality of Experience for the actual end user.

Acknowledgment

This work was supported in part by the EU-FP7-ICT-610370 (ICoSOLE) and Austrian FFG AdvUHD-DASH projects.

References

[1] Sandvine, "Global Internet Phenomena Report 2H 2014", Sandvine Intelligent Broadband Networks, 2014.
[2] I. Sodogar, "The MPEG-DASH Standard for Multimedia Streaming over the Internet", *IEEE Multimedia*, vol. 18, no. 4, pp. 62-67, Oct.-Dec. 2011.
[3] C. Timmerer, C. Mueller, S., Lederer, "Adaptive Media Streaming over Emerging Protocols", *Broadcast Engineering Conference (BEC)*, NAB2014, 2014.
[4] C. Timmerer, D. Weinberger, M. Smole, R. Grandl, C. Muller, S. Lederer, "Live transcoding and streaming-as-a-service with MPEG-DASH," *in 2015 IEEE International Conference on Multimedia & Expo Workshops (ICMEW)*, Jun.-Jul. 3, 2015.
[5] M. Belshe, R. Peon, M. Thomson, (eds.), "Hypertext Transfer Protocol version 2", Feb. 2015.
[6] P. Le Callet, S. Möller, A. Perkis, (eds), "Qualinet White Paper on Definitions of Quality of Experience", European Network on Quality of Experience in *Multimedia Systems*

and Services (COST Action IC 1003), Lausanne, Switzerland, Version 1.2, Mar. 2013.
[7] C. Mueller, S. Lederer, C. Timmerer, "An Evaluation of Dynamic Adaptive Streaming over HTTP in Vehicular Environments", *Proc. of ACM MoVid'12*, 2012.
[8] S. Lederer, C. Mueller, C. Timmerer, "Dynamic Adaptive Streaming over HTTP Dataset", *Proc. ACM MMSys'12*, 2012.
[9] M. Seufert, S. Egger, M. Slanina, T. Zinner, T. Hoßfeld, P. Tran-Gia, "A Survey on Quality of Experience of HTTP Adaptive Streaming," *IEEE Communications Surveys & Tutorials*, vol.17, no.1, pp. 469-492, 2015.

Christian Timmerer is an Associate Professor at Alpen-Adria-Universität Klagenfurt, Austria and his research focus is on immersive multimedia communication, streaming, adaptation, and quality of experience. He was general chair of WIAMIS 2008, QoMEX 2013, and ACM MMSys 2016 and has participated in several EC-funded projects, notably DANAE, ENTHRONE, P2P-Next, ALICANTE, SocialSensor, and the COST Action IC1003 QUALINET. Dr. Timmerer also participated in ISO/MPEG work for several years – notably, in the area of MPEG-21, MPEG-M, MPEG-V, and MPEG-DASH. He is a co-founder of bitmovin and CIA | Head of Research and Standardization. Follow him on http://www.twitter.com/timse7 and subscribe to his blog http://blog.timmerer.com.

Daniel Weinberger received a Bachelor's Degree in Computer Science from the Alpen-Adria-Universität Klagenfurt, Austria, in 2012. He is now product manager of bitdash, a Web-based adaptive streaming client, of bitmovin, Inc., a YCombinator-backed startup. His current research interests include adaptive HTTP streaming, video coding, and Web standards.

Martin Smole received his M.Sc. (Dipl.-Ing.) in 2009 from the Alpen-Adria-Universität Klagenfurt, Austria. He has gained professional experience in software development and managing software teams working in various companies including Infineon Technologies and Hewlett Packard. In 2014 he joined bitmovin where he is the product manager of bitcodin, bitmovin's cloud-based encoding service.

Reinhard Grandl received his M.Sc. (Dipl.-Ing.) from the Alpen-Adria-Universität Klagenfurt, specializing on Networked and Embedded Systems, in 2014. He joined bitmovin in 2013 as part of the player department. Now he is product manager of the bitdash player solution, focusing of adaptive streaming technologies. His current research interests include novel Internet video approaches and user-generated content streaming.

Christopher Müller is co-founder of bitmovin and CTO | Head of Technology. He received his M.Sc. (Dipl.-Ing.) from the Alpen-Adria-Universität Klagenfurt with distinction. His research interests are multimedia streaming, networking, and multimedia adaptation; he has published more than 20 papers in these areas and currently holds six U.S. patents in the area of DASH. He participated in the MPEG-DASH standardization, contributed several open source tools (VLC plugin, libdash) and participated in several EC-funded projects (ALICANTE, SocialSensor, ICoSOLE).

Stefan Lederer is co-founder of bitmovin and CEO | Head of Business. He received his his M.Sc. (Dipl.-Ing.) in Computer Science and M.Sc. (Mag.) in Business Administration from the Alpen-Adria-Universität Klagenfurt. He gained practical expertise in various companies (IBM, McKinsey&Company, Dolby, etc.) and has a strong business focus in marketing and international management. His research topics include transport of modern/rich media, multimedia adaptation, QoS/QoE and Future Media Internet architectures. He participated in several EC- funded projects (ALICANTE, SocialSensor, ICoSOLE).

Transcoding in the Cloud: Optimization and Perspectives

Ramon Aparicio-Pardo, Gwendal Simon° and Alberto Blanc°*
** University of Nice, ramon.aparicio-pardo@unice.fr*
°Telecom Bretagne, France, firstname.lastname@telecom-bretagne.eu

1. Introduction

The most popular online video services are now hosted "in the cloud". This now mainstream idiom indicates a set of hardware and software technologies. On the hardware side, the servers in the data-centers offer computing resources for the preparation of the video content, while the *edge-servers* in the Content Delivery Network (CDN) store and deliver the video streams. On the software side, the encoders transform a flow of data depicting images into a compressed, transportable stream. Software also implements the rate-adaptive streaming strategies and is in charge of managing the CDN resources, taking into account various technical and business constraints. The coordination between all software and hardware technologies is a key requirement, affecting both the Quality of Experience (QoE) of the end-users, who consume the video streams, and the operational costs of the service provider.

In this letter, we describe research activities on the global management of a live video streaming service running in the cloud. We focus on the coordination of these technologies and we show that a management policy that takes into account all the inter-dependencies among technologies can bring significant advantages. In particular, we address the problem of preparing the adequate video *package* (the set of representations) taking into account multiple constraints.

We first describe the main elements of a cloud streaming system. We then introduce an optimization framework, which has been presented in more details in [1,2]. This framework can be tuned to reflect the objectives and constraints of different actors. We then present some results of trace-driven performance evaluations to illustrate the different results that can be obtained by the framework. Finally, and most importantly, we discuss the perspectives of the research on cloud optimization from a global standpoint, taking into account both business and scientific concerns.

2. Cloud Streaming Chain

Figure 1 shows a simplified view of the main components, and corresponding actors, of a typical cloud streaming chain. Each one is briefly described in the following.

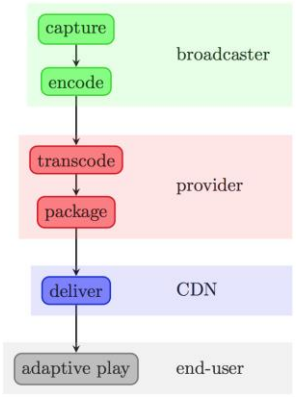


Figure 1: Cloud Streaming Chain

2.1. Broadcaster

The broadcaster is the entity that generates the original video. It can be either a professional broadcaster, such as a traditional TV provider, or, more recently, an individual source of video, such as people broadcasting while playing a videogame [3,4] or while attending all sort of events [5].

In the case of professional broadcaster, the video is captured, and then encoded by so-called *contribution encoders*. The main goal of video contribution solutions is to ensure that the raw video is encoded in a high-quality high-fidelity manner, but at a bit-rate that respects the network condition between the server that host the contribution encoder (usually enclosed or near the camera) and the *entrypoint server* of the video provider. For user-generated live streaming systems, where individuals capture and upload the scene, a new generation of software has been developed, such as Open Broadcaster Software (OBS)[1] and screencasters [5]. The main idea here is to find trade-offs between the quality of the compression and the capabilities of the machine hosting the encoders (typically a smartphone for services like Meerkat and Periscope and a machine that is almost fully utilized for running games in the case of Twitch).

2.2. The Cloud

The term cloud is used to describe all the operations that are done "somewhere" in the Internet, neither at the broadcaster side, nor at the end-user side. We

[1] https://obsproject.com/

distinguish the roles of video providers and CDNs, the former being the main provider of the service while the latter is an intermediary, but key, actor. Some video companies play both roles but this is not always the case.

The video provider gets as input the "original video" (more precisely the video that is the output of the contribution encoders). Its role is to prepare the video so that end-users will be able to eventually play it on their device. In the recent years, the heterogeneity of end-users' device has grown significantly, from smartphones to connected TVs, and even new generations of Virtual Reality (VR) headsets. To address this heterogeneous population of end-users, the video providers have adopted rate-adaptive streaming systems such as Apple's HTTP Live Streaming (HLS) and the MPEG standard Dynamic Adaptive Streaming over HTTP (DASH).

The implementation of rate-adaptive streaming first requires a transcoding operation. For each video, the video provider should generate *k* different *representations*, each of them characterized by a different bit-rate, resolution, sometimes frame rate and key frame period. Then, the video provider should package the video by aggregating the set of representations, by segmenting the representations and by creating the manifest file, which is the file that describes the playlist.

The cost of transcoding these videos can strain the computing infrastructure of video providers. Typically in Twitch, more than 6,000 videos need to be transcoded in average [3]. Furthermore video providers increase the number of representations per video so that each end-user can find a good match among the available representations. To meet this demand, the video providers manage large data-centers with thousands of servers [8].

The CDN gets in input the package of videos that has been prepared by the video provider. Its role is to deliver the content. The literature related to live streaming in CDN provides details about the processes that are implemented in CDN to make sure that an end-user that requests a given segment of a given representation can find a nearby edge-server that stores the said segment [6,7].

The number of edge-servers in a CDN (more than 200,000 for the biggest CDN players) is expected to enable large-scale delivery in good conditions. However, the costs of transporting the packages of video representations from the CDN's origin server to the subset of edge-servers that must deliver this content have grown. Indeed, the set of representations includes many Ultra-High-Definition (UHD) and High-Definition (HD) videos, which have a large bit-rate. CDNs need to reserve a large bandwidth in the core network to deliver the segments to the edge servers.

2.3. The End-Users

The endpoint of the chain is the video player of the end-user. With the adoption of adaptive streaming, the role of the video player is no longer passive; on the contrary, it is responsible for selecting the right representation for every segment of the video. Various strategies have been devised to efficiently choose the representation whose bit-rate is closest to the currently available network capacity, while avoiding to change representation too often. Recent studies have highlighted the relations between the engagement of users in a service and the QoE of the video streaming [9]. These studies have also revealed that many parameters impact the QoE: of course the video bit-rate, but also the delay to start the video playback, and most importantly the interruptions due to video re-buffering. Another aspect that has not been extensively studied, to the best of our knowledge, is the relation between the QoE and the device that is used to play the video. The size of the display screen is typically one of the parameters that the video provider should also consider for the transcoding of the representations.

3. Optimization Models

We briefly introduce in this Section two models that we have described in [1,2]. Both models aim at helping the video provider when deciding which representations to generate.

3.1. Video Type, Popularity and CDN Budget

To the best of our knowledge, the first paper to study encoding choices for adaptive streaming is [1]. The main idea is that every ingested video coming from the contribution encoders can be transcoded as many times as necessary. However, the transcoder impacts all the following modules in the delivery chain. In particular, the more video representations are created, the larger the CDN budget to transport the full package to the edge-servers. Moreover, the load on the packager also depends on the number of representations. To simplify the management and the operational cost of the video service, it is thus preferable to limit the overall number of representations *K*.

In the industry, the norm is to transcode every video into the same number of representations with the same encoding parameters. In [1], we show that this solution is far from optimal. Instead, it is better to decide for each video the number of representations to transcode and the profile of these representations. In particular, we emphasize the benefits that one can expect from following some intuitive rules related to the popularity of the video, the nature of the video, and the devices of the target population.

To illustrate our claim, we first showed that the recommendations for transcoding given by the main video players are sub-optimal. To validate this claim,

we used an Integer Linear Program (ILP) to compute the optimal set of representations for any given overall number of representations *K*.

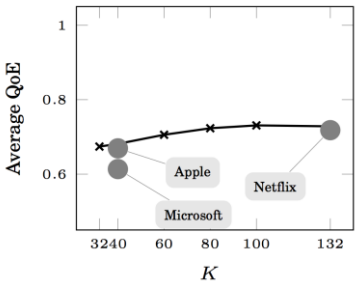


Figure 2: Average QoE for a given overall number of representations K [1]

Figure 2 shows the average QoE of 500 users watching four different videos. The circles correspond to the recommendations by Microsoft, Apple and Netflix and the line with the crosses to the optimal solution obtained with the ILP." This figure shows not only that the representation recommended by Apple and Microsoft are sub-optimal and use too few representations but also that with half the representations recommended by Netflix it is possible to almost achieve the same QoE.
More details about simulations settings, as well as many other simulation results can be found in [1].

3.2. Transcoders in Data-Centers

In [2], we take into account the processing power that is required to transcode each given video stream in real time. If we refer to the delivery chain of Figure 1, we considered the packaging load and the CDN budget as the two main constraints in [1], while in [2] we add the transcoding load as an additional constraint. This latter constraint is likely to be the preponderant one in many cases, and in particular for services like Twitch, where a large number of video streams have to be transcoded.
To study this problem, we collected two large datasets, which are also publicly available. The first one is the result of a of four month study of Twitch [3]. We highlight two main characteristics of the ingested streams: the number of streams can vary significantly within a day, and the diversity of the quality of ingested video is large. Our second dataset is a comprehensive set of measurements that we realized on a series of systematic transcoding of various videos, from any resolution and bit-rate to any other resolution and bit-rate. We focus on the number of CPU cycles that are required to make these transcoding operations and the quality of the transcoded videos compared to the quality of the original videos.
The main claim of [2] is similar as the one in [1], that is, the video provider can obtain better performance if all videos are not transcoded in the same way. We used an ILP to find the optimal number of representations and the corresponding parameters (e.g., resolution, bit-rate) taking into account constraints on the number of CPU cycles available. Figure 3 shows the result of our tests. Again, the optimal set of packages provides a far better quality when compared to the standard choices in the market.

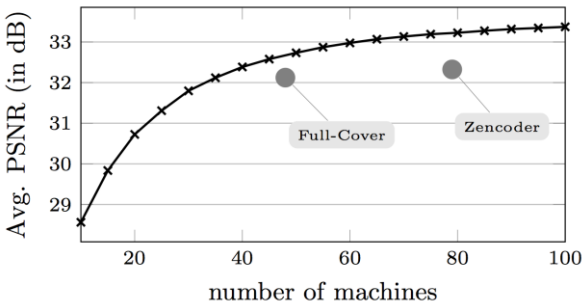


Figure 3: Quality of the videos as a function of the number of machines used.

We then propose and implement a heuristic, which is not optimal but can be easily and efficiently implemented, to decide the number of representations and their profiles, for every ingested stream. We apply this heuristic to various snapshots of the Twitch system, and compare both the number of CPU cycles that the transcoding operations consumed in the datacenter and the average quality of the videos with respect to the original video. We show the result in Figure 4.

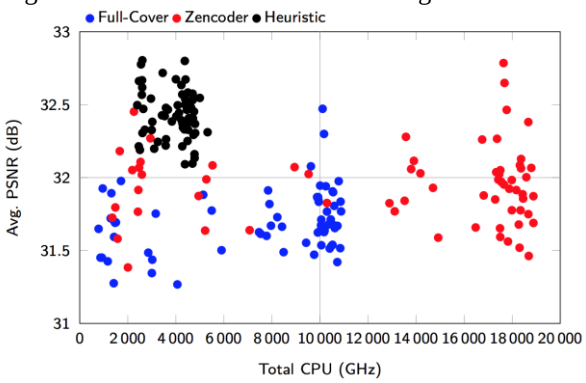


Figure 4: The average quality and the number of consumed CPU cycles for various snapshots of the Twitch system

We especially emphasize in Figure 4 that the traditional approaches cannot accommodate variable load in inputs. The Zencoder solution requires sometimes the reservation of a large-scale datacenter while it requires a small one at other moments of the day. On the contrary a smart management can make sure that a given amount of resources are always fully exploited.

4. Discussion

The studies in [1,2] have shown that significant gains can be obtained by implementing smart strategies for the transcoding of the videos in the context of adaptive streaming. However, to turn our solutions into practical implementations, the actors involved in the delivery chain have to share more information than they currently do. In particular, the set of servers in the datacenter, the packager load, the overall charge on the CDN and the characteristics of the population of users (or at least the size of the population consuming a given video in real-time) are key information that are needed at several stages in the chain. Today, the video provider and the CDN are not tightly coupled (with the exception of vertical integrated companies like Google and Netflix, which manage all combined video services, datacenters and CDNs).

Unsurprisingly, two leading companies specialized in video transcoding in datacenters (namely Envivio and Elemental) have both been recently acquired by larger companies (Ericsson and Amazon respectively). Such events prove that the vertical integration of a maximum number of key actors in the delivery chain is seen as a way to both generate significant savings in operational costs, but also to improve the performance of the video services. Important optimization processes can be put in place when one has a global view of the system.

In the near future, the collaboration between the CDN and the datacenter in charge of transcoding the video should be further improved. The transcoder needs information from the CDN about the end-users who consume the videos and about the network conditions that the CDN has to deal with. On its side, the CDN can optimize the delivery and significantly reduce the operational costs if it knows the underlying structure of the streams that it should transport from the origin server to the edge-servers. All this calls for a better collaboration, possibly by the mean of standard APIs, between these actors.

References

[1] Laura Toni, Ramon Aparicio, Alberto Blanc, Gwendal Simon, and Pascal Frossard. "*Optimal Set of Video Representations in Adaptive Streaming*", in ACM MMSys, 2014.

[2] Ramon Aparicio, Karine Pires, Alberto Blanc and Gwendal Simon. "*Transcoding Live Video Streams at a Massive Scale in the Cloud*" in ACM MMSys, 2015.

[3] Karine Pires and Gwendal Simon. "*DASH in Twitch: Adaptive Bitrate Streaming in Live Game Streaming Platforms*" in ACM VideoNext Conext Workshop, 2014.

[4] Ryan Shea, Di Fu, and Jiangchuan Liu "*Towards bridging online game playing and live broadcasting: design and optimization*", in ACM Nossdav, 2015.

[5] Chih-Fan Hsu, Tsung-Han Tsai, Chun-Ying Huang, Cheng-Hsin Hsu, and Kuan-Ta Chen. "*Screencast dissected: performance measurements and design considerations*", in ACM MMSys, 2015.

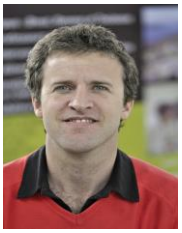
[6] Matthew Mukerjee, David Naylor, Junchen Jiang, Dongsu Han, Srinivasan Seshan, Hui Zhang. "*Practical, Real-time Centralized Control for CDN-based Live Video Delivery*", in ACM Sigcomm, 2015

[7] Jiayi Liu, Gwendal Simon, Géraldine Texier, and Catherine Rosenberg. "*User-centric discretized delivery of rate-adaptive live streams in underprovisioned CDN networks*", IEEE Journal in Selected Areas in Communications, 2014.

[8] Luiz A. Barroso, Jimmy Clidaras, and Urs Hölzle. "*The datacenter as a Computer: An Introduction to the Design of Warehouse-scale Machines*". Morgan Claypool, 2013.

[9] S. Shunmuga Krishnan and Ramesh Sitaraman "*Video Stream Quality Impacts Viewer Behavior: Inferring Causality using Quasi-Experimental Designs*". in ACM Internet Measurement Conference (IMC), 2012.

Ramon Aparicio-Pardo received a Ph.D. Degree in Information and Communication Technologies from Universidad Politécnica de Cartagena (UPCT), Spain, in 2011. After that, he completed a postdoctoral fellowship with Orange Labs in 2012. He has then worked as a postdoc researcher at Telecom Bretagne from 2013 to 2015. He is now Associate Professor at University of Nice. His research interests include planning, design and evaluation of communication networks by means of mathematical optimization.



Gwendal Simon graduated from University Rennes (France). During his PhD, he worked at Orange Research Labs. From 2004 to 2006 he was a researcher at Orange Labs. Since 2006, he has been Associate Professor at Telecom Bretagne, a graduate engineering school within the Institut Mines-Telecom. He was a visiting researcher at University of Waterloo from September 2011 to September 2012. His research interests include multimedia delivery systems (video and gaming) and network management.

Alberto Blanc is an associate professor at Telecom Bretagne since 2010. He received a "laurea" in computer engineering in 1998 from the "Politecnico di Torino," Italy, and a Ph.D. in electrical engineering from the University of California, San Diego, U.S., in 2006. His research interests include performance evaluation of computer networks and cloud computing.

Delay Reduction in Cloud Gaming

Shervin Shirmohammadi

Distributed and Collaborative Virtual Environments Research (DISCOVER) Lab
University of Ottawa, Canada,
shervin@eecs.uottawa.ca

1. Introduction

Cloud gaming or Gaming as a Service [1], the newest entry in the online gaming world, leverages the well-known concept of cloud computing to provide real-time gaming services to players. The idea in cloud gaming is to capture the game events from players and transmit them to the cloud, process those events and run the game logic in the cloud, render the game scene as video in the cloud, and stream that video to the players. The advantage is that as long as the client can display video, which pretty much all smartphones, tablets, game consoles, desktops, laptops, and mobile devices today do, the user can play the game without installing it locally, and without needing to have a machine with high-grade 3D graphics rendering and powerful computing hardware and software.

Cloud Gaming is already available as commercial products, such as Sony's PlayStation Now , Ubitus's GameNow , G-Cluster , Crytek's GFACE , PlayGiga , and LiquidSky , to name a few. There are also many efforts concentrating specifically on the underlying technology behind Cloud Gaming, such as NVIDIA's Grid, OTOY, CiiNow, Kalydo and Gaming Anywhere [2], the latter being the only open source and free technology. Microsoft is also exploring cloud gaming technologies, with recent successes such as its Kahawai project [3].

Despite advancing at a rapid rate, Cloud Gaming is fundamentally challenged by two main obstacles: first, the required video bitrate to achieve acceptable playing quality is quite high. For example, for 720p video resolution at 50 fps, the former OnLive system (whose technology patents were recently bought by Sony) requires a network connection with at least 5 Mbps of bandwidth [4]. Second, cloud gaming is very sensitive to network latencies which impair the interactive experience of a video game [5], especially in multiplayer mode. In addition to the above two main obstacles, Cloud Gaming faces other challenges such as the mobility of today's players and the heterogeneity of players' devices (tablets, smartphones, game consoles, PCs, laptops, etc.) which requires the server to adapt the game content to the characteristics and limitations of the client's underlying network or end device, and the challenge of configuration, deployment, and maintenance of the game in the cloud, including the required resource allocation and virtualization [6].

Previously, we discussed the bandwidth and the adaptation to mobile client issues [7]. In this article, we will take a quick look at the delay issue.

2. Delay in Cloud Gaming

Delay can occur at various points in a Cloud Gaming system, as shown in Figure 1: the cloud, the transport network, the home network, and the client device. The cloud itself consists of game engine, rendering, and cloud resource management, while the client side includes the decoder and user interaction capture. Assuming we have no control over the transport network itself (it's controlled by the ISP), to reduce delay, researchers have been investigating mostly the

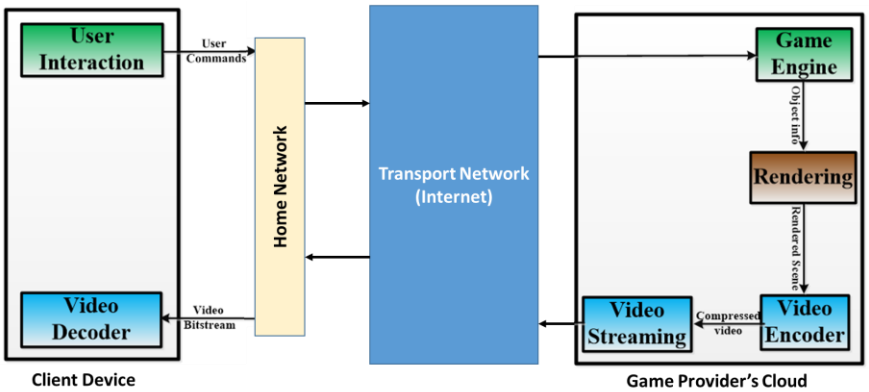


Figure 1. Cloud Gaming system components

cloud and the client side, and some also the home network side [8]. In this paper we focus on the cloud side, and describe some techniques to reduce delay in the cloud.

3. Reducing video encoding delay

In [2], the computational steps of a cloud gaming system is decomposed into four: packetization, format conversion, video encoding and memory copy. Among these four processes, video encoding contributes up to 52% of the processing time. This means that a reduction in video encoding delay would lead to a significant reduction in overall delay at the cloud side.

In [9] we can see an example of a technique to reduce video encoding delay, specifically for cloud gaming. In this approach, the design shown in Figure 2 replaces the game provider's cloud side in Figure 1. The core idea here is the introduction of an interface between the game engine and the video encoder, as shown in Figure 2. This interface collects some information from the game engine and reshapes them to be understandable by the video encoder. Having this information, the video encoder will be able to skip or simplify some encoding operations, especially the motion estimation process, for some blocks in the frame, leading to faster encoding. Skipping or simplifying the motion estimation operation achieves up to 39% speed up in the motion estimation process, leading to a 24% acceleration in the total video encoding process [9].

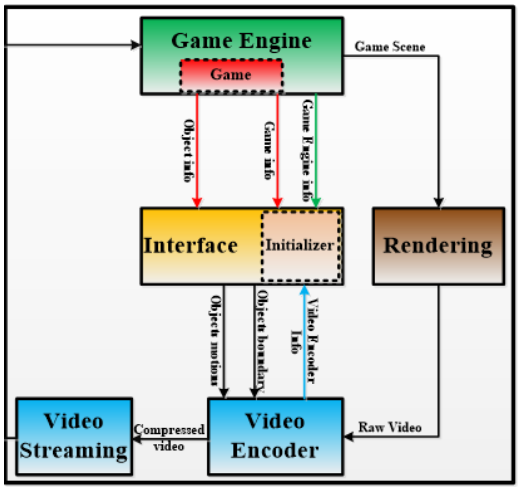


Figure 2. Game provider's cloud side in [9]

The information that the *Interface* needs are:

- Object info: the location and orientation of an object
- Game info: The number of objects within the game, size (e.g. width and length) of each object, and the size of the game frame.
- Game engine info: The game engine's coordination system.
- Video encoder info: The setup of the video encoder including video frame size, video frame coordination system and number of coded frames per second.

The interface uses the above information and performs several tasks to provide the Motion Vectors of each MacroBlock of the game frame to the video encoder. Readers are encouraged to study [9] for details.

4. Reducing delay by optimizing the cloud infrastructure

Another point in a cloud gaming system that causes delay is the cloud infrastructure itself, which include data centers consisting of a core switch that dispatches client requests among aggregation switches, each of which further dispatches the request to a Top of Rack (ToR) switch, which finally dispatches the request to a processing node running multiple Virtual Machines (VM). In cloud gaming systems, since large amounts of video data are transmitted, the core switches are prone to congestion. So, some auxiliary network routes are employed as soon as the main paths are congested. Even though there are several methods to determine the optimal paths in the network, these methods usually select the best routes based on the link metrics provided at the setup time, and they often fail to take into consideration the current status of links (such as current available bandwidth) or the requirements of data flows passing through the network.

The work in [10] reduces the delay in the above-mentioned infrastructure by applying the recent paradigm of Software Defined Networks (SDNs) to Cloud Gaming, and by proposing an SDN controller that adaptively disperses the game traffic load among different network paths according to their corresponding end-to-end delays. The controller works as follows:

When the core switch receives a packet without any matched entry in its own flow table, it sends the packet to the SDN controller, which then finds the optimum path; i.e., the path more suitable to forward this packet. Once the path is chosen, a new entry is created in the core switch's flow table for future packets of the same flow. Each player's session with a game is associated with one flow, and vice-versa. Conventional SDN controllers usually find the optimum path in terms of different criteria (e.g., shortest delay, least hop-count etc...) to create a forwarding rule. Conversely in [10],

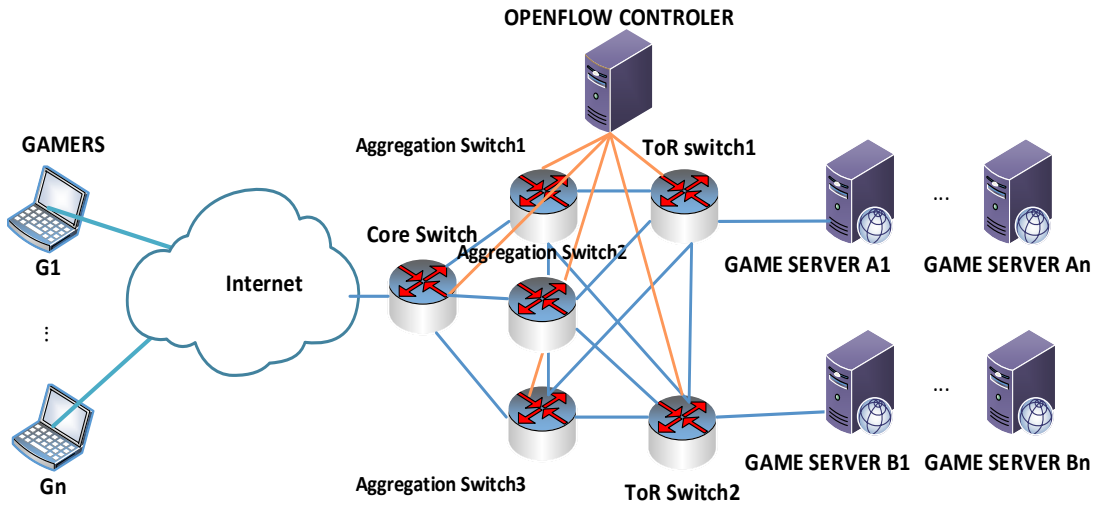


Figure 3. Cloud Gaming Architecture Using SDN

first, all possible paths, from the requesting core switch to the Top of Rack (ToR) switches are identified by the controller. While this is an NP-problem, since the graphs constructed based on the current architectures of data centers are simple (e.g. Fat-tree and VL2), and the source (i.e. core switch) is fixed for all flows, the problem of finding all possible paths can be solved using Dijkstra's algorithm.

Next, the controller collects QoS statistics from the switches along the extracted paths using the OpenFlow protocol messages (STATISTICS_REQUEST, STATISTICS_REPLY,…), defined in the OpenFlow specification. The controller then splits the incoming packets, belonging to a flow, dynamically among different available paths directly proportional to the associated measured end-to-end delays. Also, since the aforementioned computation is conducted only once (when a new flow is detected by the controller), the proposed controlling method does not impose high computational costs on the controller. The controller first picks a path among the possible paths randomly to serve as a starting point, and then changes the path in a weighted round-robin fashion. It is worth noting that the random selection of the first path can reduce the creation of bias among different flows. The number of packets forwarded to each path is proportional to the weight factor which is computed for that path. Thus, the incoming traffic is fairly distributed among the alternative paths so that the network utilization is increased and consequently, traffic congestion is minimized. The cloud gaming architecture using the above SDN controller is shown in Figure 3.

Experimental results show that the proposed controller reduces end-to-end delay and delay variation by almost 9% and 50% respectively without engendering additional packet loss, compared to a representative conventional method: Open Shortest Path First (OSPF). Readers are encouraged to study [10] for details.

5. Conclusion

As cloud gaming become more popular, offering a high quality gaming experience to the players becomes more crucial in the mass adoption of cloud gaming as a De Facto gaming platform. Hence, more research is needed to overcome the two main obstacles in cloud gaming: bandwidth, and delay. In this article, we gave an overview of delay points in a cloud gaming system, and we also described some techniques for reducing delay in the cloud side.

References

[1] W. Cai, M. Chen, and V. Leung, "Toward Gaming as a Service," IEEE Internet Computing, vol. 18, no. 3, pp. 12–18, 2014.

[2] C.Y. Huang et al., "GamingAnywhere: The first open source cloud gaming system", ACM Transactions on Multimedia Computing, Communications, and Applications, Vol. 10, Issue 1s, January 2014, Article No. 10.

[3] E. Cuervo et al., "Kahawai: High-Quality Mobile Gaming Using GPU Offload," Proc. ACM Annual International Conference on Mobile Systems, Applications, and Services (MobiSys), Florence, Italy, May 18-22 2015, pp. 121-135.

[4] M. Claypool et al., "Thin to win? Network performance analysis of the OnLive thin client game system," Proc. ACM/IEEE Annual Workshop on Network and Systems Support for Games (NetGames), Venice, Italy, November 22-23 2012, pp. 1-6.

[5] K. T. Chen et al., "Measuring the latency of cloud gaming systems," Proc. ACM international Conference on Multimedia, November 28 - December 1 2011, Scottsdale, Arizona, USA, pp. 1269-1272.

[6] S. Shirmohammadi, M. Abdallah, D.T. Ahmed, K.T. Chen, Y. Lu, and A. Snyatkov "Guest Editorial for the Special Section on Cloud Gaming and Virtualization", IEEE Trans. o Circuits and Systems for Video Technology, 2015. (to appear)

[7] S. Shirmohammadi, "Adaptive Streaming in Mobile Cloud Gaming", IEEE COMSOC Multimedia Communications Technical Committee E-Letter, Vol. 8, No. 5, September 2013, pp. 20-23.

[8] A. Bujari, M. Massaro, and C. Palazzi, "Vegas over Access Point: Making Room for Thin Client Game Systems in a Wireless Home", IEEE Trans. o Circuits and Systems for Video Technology, 26 June 2015, DOI:10.1109/TCSVT.2015.2450332

[9] M. Semsarzadeh, A. Yassine, and S. Shirmohammadi, "Video Encoding Acceleration in Cloud Gaming", IEEE Trans. on Circuits and Systems for Video Technology, July 6 2015, DOI:10.1109/TCSVT.2015.2452778

[10] M. Amiri, H. Al-Osman, S. Shirmohammadi, and M. Abdallah, "An SDN Controller for Delay and Jitter Reduction in Cloud Gaming", Proc. ACM Multimedia, October 26-30 2015, Brisbane, Australia, 4 pages.

Shervin Shirmohammadi received his Ph.D. degree in Electrical Engineering from the University of Ottawa, Canada, where he is currently a Full Professor at the School of Electrical Engineering and Computer Science. He is Director of the Distributed and Collaborative Virtual Environment Research Laboratory (DISCOVER Lab), and member of the Multimedia Communications Research Laboratory (MCRLab), conducting research in multimedia systems and networking, specifically in gaming systems, video systems, and their application in multimedia-assisted biomedical engineering. The results of his research have led to more than 250 publications, over 20 patents and technology transfers to the private sector, and a number of awards and prizes. He is Associate Editor-in-Chief of *IEEE Transactions on Instrumentation and Measurement*, Senior Associate Editor of *ACM Transactions on Multimedia Computing, Communications, and Applications*, and was Associate Editor of Springer's Journal of Multimedia Tools and Applications from 2004 to 2012. Dr. Shirmohammadi is a University of Ottawa Gold Medalist, a licensed Professional Engineer in Ontario, a Senior Member of the IEEE, and a Professional Member of the ACM.

Cloud-based System for Large-Scale Video Analysis from Camera Networks

Wei-Tsung Su[1] and Yung-Hsiang Lu[2]

[1] Aletheia University, Taiwan (R.O.C.), au4451@au.edu.tw

[2] Purdue University, USA, yunglu@purdue.edu

1. Introduction

Since the Internet became widely accessible to the general public in 1990s, user-created multimedia materials have become a major source of content. Every day, millions of images are shared through social networks, and thousands of video clips are uploaded to sharing sites. In addition, public cameras are connected to the Internet for various purposes: environmental studies [1, 2], transportation planning [3], and so on. A report [4] estimates that 28 million network cameras will be sold in 2017, at 27.2% growth over the five year from 2012 to 2017.

The multimedia data generated from these network cameras can be a type of big data because (1) at multiple frames per second, multimedia data pass through networks at high *velocity*; (2) multimedia data have wide *variety*; (3) and storing multimedia data requires large *volume* of capacity [5]. These features indicate that cloud computing can be an appropriate solution to process multimedia big data. Thus, in our previous work, a cloud-based system, Continuous Analysis of Many Cameras (CAM2, in short), is proposed for harvesting valuable information embedded in multimedia data from multiple network cameras [6]. This paper will review and reveal the research issues on cloud resource management for performance improvement of CAM2 for users and researchers, respectively.

This paper has the following contributions: (1) It will review CAM2 from users' point of view. Readers can register at https://cam2.ecn.purdue.edu/ for using CAM2. (2) A cloud resource manager is proposed for aiming at reducing the overall cost and improving performance. CAM2 uses the proposed resource manager to allocate and scale cloud resources. The experiment result shows that the proposed resource manager can lead to a 13% reduction in cost [12]. (3) The problem of handling multi-camera to multi-cloud video streams according to different analysis requirements is preliminary studied.

2. Review of CAM2

Users who need to process on-line videos streaming from multiple network cameras will face the following problems: (1) Network cameras provide different data formats. (2) Different brands of network cameras have different methods to retrieve the data. Retrieving data from these network cameras requires additional effort. (3) A lot of resources are required to store and analyze the multimedia big data. For example, it requires 16,200TB per day to store the videos at 3840×2160 [5]. Consequently, massive computing resources are required to process these data. To the best of our knowledge, CAM2 is the first common platform to solve the above problems.

The features of CAM2 includes [7]

a) CAM2 supports OpenCV library, which implements many algorithms for computer vision [8].
b) CAM2 currently connects over 70,000 public network cameras such that users can easily select the live images from these network cameras.
c) CAM2 provides an event-driven API (application programming interface) which alleviates users from the need of interfacing with various brands of network cameras.
d) CAM2 automatically allocates cloud instances to execute the analysis programs for meeting the analysis requirements, such as higher computing performance, higher frame rates, and lower costs.

Procedure to use CAM2.

The procedure of using CAM2 is described below:

a) Select the network cameras for analysis according to users' need, such as location and time zone via web user interface.
b) Set the execution configuration, such as the desired frame rate and the duration.
c) Upload an analysis program. CAM2 provides 16 pre-written analysis programs as examples for corner detection, motion detection, sunrise detection, etc. Users can write their own analysis programs with OpenCV-Python in CAM2.
d) Execute the analysis program.
e) Download the execution results.

Writing analysis program with CAM2 APIs.

The users can write and upload their own analysis programs using OpenCV-Python based on proposed CAM2 APIs [9]. An analysis program may import the `FrameMetadata` class, which allows users to obtain the information of a frame captured from each selected camera, such as date, time, and camera ID and the `CameraMetadata` class, which allows users to obtain the information of a camera, such as latitude and longitude.

Each analysis program must extend the `Analyzer` class, which has three methods as shown in Figure 1.

a) The method `initialize` is called once at the beginning of the execution. The parameters or variables could be initialized in this method.
b) The method `on_new_frame` will be called every time a new frame is retrieved from the selected camera. The main computer vision algorithm must be implemented in this method.
c) The method `finalize` is called once after all frames are analyzed based on the configuration. The final calculation (such as summarizing the information from all frames) could be done and the final results can be saved as text files or images in this method.

```
from analyzer import Analyzer
from frame_metadata import FrameMetadata
from camera_metadata import CameraMetadata

import datetime
import numpy as np
import cv2
import cv2.cv as cv

class MyAnalyzer(Analyzer):

  def initialize(self):
  """ Called once at the beginning """

  def on_new_frame(self):
  """ Called when a new frame arrives """

  def finalize(self):
  """ Called once in the end """
```

Figure 1. The structure of an analysis program in CAM2 [5].

3. Challenges of allocating cloud resources in CAM2.

One of the most important challenges in CAM2 is to reduce the overall cost and improve performance by appropriately allocating cloud resources. Cloud vendors offer many instance types with different capabilities in terms of numbers of cores, memory sizes, network performance, storage capacities, geographical locations, etc. With these options, the following questions arise,

a) How many instances does one analysis program need?
b) How many data streams can one cloud instance analyze?
c) What is the most cost-effective cloud instance to use for a given analysis program?

In our previous work [10, 11], the experiment result shows that different types of cloud instances can incurs different performance. Therefore, CAM2 uses the resource manager proposed in [12] to allocate and scale cloud sources in order to meet the CPU and memory requirements of the analysis program. The resource manager continuously monitors the resource utilization of the cloud instances and automatically scales the cloud resources as needed. Moreover, if the same user executes multiple analysis programs at different times, the resource manager can reuse the running instances to reduce the overall analysis cost.

Performance evaluation

Six types of cloud instances and four analysis programs are used for evaluating the resource manager [12]. The cloud instances have different CPU and memory capabilities, and the analysis programs represent different workloads in terms of CPU and memory: image archival, motion estimation, moving objects detection, and human detection.

Experimental setup.

Table 1 compares the six Amazon EC2 cloud instance types that are used in our experiments: two general purpose instances (`m3.xlarge` and `m3.2xlarge`), two compute optimized instances (`c4.xlarge` and `c4.2xlarge`), and two memory optimized instances (`r3.xlarge` and `r3.2xlarge`). The processor of the compute optimized instances is Intel Xeon E5-2666 v3 clocked at 2.9 GHz, and it is Intel Xeon E5-2670 v2 clocked at 2.5 GHz for all the other instances.

TABLE 1: The CPU, memory, and hourly price of different Amazon EC2 cloud instances [12].

Instance	Cores	Memory (GB)	Hourly Price
m3.xlarge	4	15.0	$0.266
m3.2xlarge	8	30.0	$0.532
c4.xlarge	4	7.5	$0.220
c4.2xlarge	8	15.0	$0.441
r3.xlarge	4	30.5	$0.350
r3.2xlarge	8	61.0	$0.700

Kaseb et al. [12] use four analysis programs implemented using OpenCV [13]. These analysis programs are used in the experiments for both image analysis at 0.2 FPS (Frames Per Second) and video analysis at 10 FPS.

a) **IA - Image Archival:** This program downloads the individual images of an image or video stream, without any further analysis.
b) **ME - Motion Estimation:** This anlaysis program estimates the amount of motion in an image or video stream using the background subtraction method proposed by KaewTraKulPong and Bowden [14].
c) **MOD - Moving Objects Detection:** This analysis program detects the moving objects in an image or video stream using the background subtraction method proposed by Zivkovic [15].
d) **HD - Human Detection:** This analysis program detects humans in the individual images of an image stream using the human detection method proposed by Dalal and Triggs [16]. The program saves the input images and the corresponding images annotated with the detected humans.

Effective cost of different cloud instances.
Figure 2 shows the effective cost of different cloud instances for executing different analysis programs. The figure shows the following:

a) Different cloud instances are more cost-effective than the others for some analysis programs. Choosing the right cloud instance for an analysis program can save half on the analysis cost.
b) For image analysis at 0.2 FPS, compute optimized cloud instances (`c4.xlarge` and `c4.2xlarge`) are more cost-effective for moving objects detection. Memory optimized cloud instances (`r3.xlarge` and `r3.2xlarge`) are more cost-effective for motion estimation.
c) For video analysis at 10 FPS, compute optimized cloud instances are always more cost-effective than the other instances. That's because video analysis consumes CPU resources much more than memory resources as we showed earlier.
d) Although the `xlarge` instances provide half the CPU and memory resources of the `2xlarge` instances for half the price as shown in Table 1, the `xlarge` instances are often more cost-effective than the `2xlarge` instances. This recommends using smaller instances instead of larger ones [11].

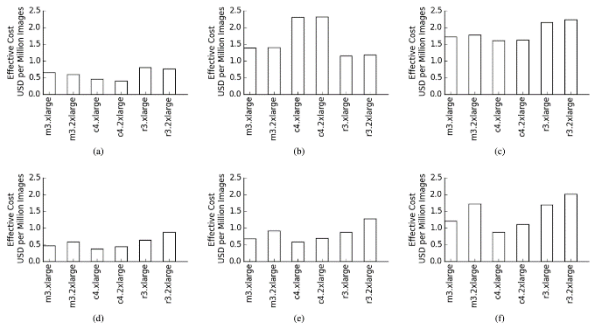


Figure 2. The effective cost as defined in (6) of different cloud instances for executing different analysis programs: (a-c) at 0.2FPS. (d-f) at 10 FPS. (a, d) Image archival. (b, e) Motion estimation. (c, f) Moving objects counting [12].

TABLE 2. The analysis programs in the experiment [5].

Program	Start Time	Duration	Cameras	Frame Rate
ME	0:00	4.50 hours	1000	0.2
HD	1:15	4.75 hours	10	0.2
MOD	1:30	4.50 hours	16	10.0

Cloud Resource Allocation Management.
A 6-hour large-scale experiment that uses CAM2 to analyze the data from 1026 cameras using different analysis programs at different frame rates as shown in Table 2. The experiment analyzes 5.5 million images (260GB data).

Based on the lifetime of the cloud instances in Figure 3 and their prices, the experiment costs $12.77. If the proposed resource manager is not used, and the general-purpose `m3.xlarge` instances are used for all the analysis programs, this experiment needs five, one, and five `m3.xlarge` instances to handle the three analysis programs respectively. The overall analysis cost is $14.63 in this case. This means that our resource manager leads to a 13% reduction in the overall analysis cost [12].

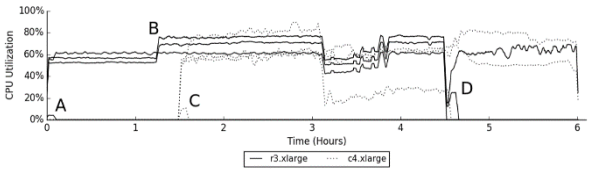


Figure 3. The CPU utilization of the cloud instances while analyzing the data from 1026 cameras using different analysis programs at different frame rates as shown in Table 2 [12].

4. Multi-camera to multi-cloud video streaming
Because the network cameras and the data center of cloud vendors are both geographically distributed, the video streaming path can affect the performance. For example, the execution time for detecting lanes in videos streamed from 100 network cameras using CAM2 is observed. As shown in Figure 4, the execution time is varying if the 100 network cameras are located in different continents. Thus, there is an optimization problem of streaming videos from multiple network cameras to multiple data centers for further performance improvement in CAM2.

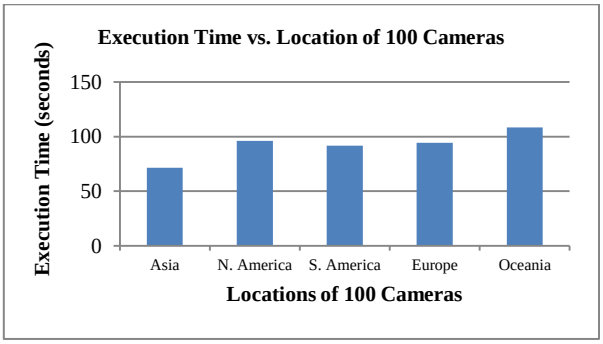


Figure 4. The execution time for detecting lanes in videos from 100 network cameras located in different continents using CAM2.

To simplify the description of this problem, we take the single-camera to single-cloud as an example. The prices of AWS compute-optimized cloud instance, Linux on `c4.large`, in N. Virginia, Singapore, and Tokyo are $0.11, $0.152, and $0.14 per hour, respectively. Streaming the video from one camera located in Singapore to the AWS data center located in Singapore has lower latency. On the contrary, the lowest cost can be obtained if the video is streamed to N. Virginia. Even though round-trip-time (RTT) is not

a linear function of the geographical distances, longer geographical distances usually have longer RTT. Thus, the network latency can be significantly increased. However, the cost can be reduced while keeping low network latency if streaming the video to Tokyo. The problem of handling multi-camera to multi-cloud video streams will be more complex than the problem described above and can be the next step to further improve performance of CAM2.

5. Conclusion

This paper presents the opportunities and challenges of proposed CAM2. From users' point of view, CAM2 is a platform to ease the procedure to process on-line videos streaming from multiple network cameras. From researchers' point of view, the challenge is to reduce cost and improve performance using cloud resources. CAM2 uses the proposed resource manager to allocate and scale cloud resources in order to meet the analysis requirements. The experiments show that the proposed resource manager can lead to a 13% reduction in the overall analysis cost. In addition, we preliminary study the problem of handling multi-camera to multi-cloud video streams in CAM2. This can be the next step to improve the performance of CAM2. At last, readers interested in using CAM2 are welcome to register as the users at https://cam2.ecn.purdue.edu/.

References

[1] T. E. Gilmore et al., "Source and magnitude of error in an inexpensive image-based water level measurement system," in *Journal of Hydrology*, vol. 496, pp. 178–186, Jul., 2013.

[2] W. Nijland et al., "Monitoring plant condition and phenology using infrared sensitive consumer grade digital cameras," in *Agricultural and Forest Meteorology*, vol. 184, no. 15, pp. 98–106, Jan., 2014.

[3] V. Kastrinaki et al., "A survey of video processing techniques for traffic applications," in *Image and Vision Computing*, vol. 21, no. 4, pp. 359–381, 2003.

[4] Marketandmarkets.com. Network camera and video analytics market. September 2012. Report Code: SE 1238.

[5] W.-T. Su et al., "Harvest the Information from Multimedia Big Data in Global Camera Networks," in *Proc. of IEEE BigMM*, pp. 184–191, 2015.

[6] A. S. Kaseb et al., "Multimedia Content Creation using Global Network Cameras: The Making of CAM2," in Proc. of GlobalSIP, 2015.

[7] W.-T. Su et al., "Teaching Large-Scale Image Processing over Worldwide Network Cameras," in *Proc. of IEEE DSP*, pp. 726–729, 2015.

[8] http://opencv.org/

[9] E. Berry et al., "Using Global Camera Networks to Create Multimedia Content," in *Proc. of CCBD*, 2015.

[10] W. Chen et al., "Analysis of large-scale distributed cameras using cloud," in *IEEE Cloud Computing*, 2016.

[11] W. Chen et al., "Adaptive Cloud Resource Allocation for Analysing Many Video Streams," in *Proc. of IEEE CloudCom*, 2015.

[12] A. S. Kaseb et al., "Cloud Resource Management for Image and Video Analysis of Big Data from Network Cameras," in *Proc. of CCBD*, 2015.

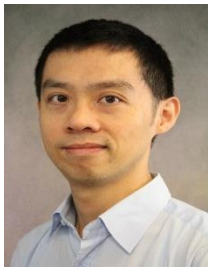
[13] G. Bradski, "The OpenCV library," in *Dr. Dobb's Journal of Software Tools*, vol. 25, no. 11, pp. 120, 122–125, 2000.

[14] P. KaewTraKulPong et al., "An improved adaptive background mixture model for real-time tracking with shadow detection," in *Video-Based Surveillance Systems*, pp. 135–144, Springer US, 2002.

[15] Z. Zivkovic, "Improved adaptive gaussian mixture model for background subtraction," in *Proc. of International Conference on Pattern Recognition*, vol. 2, pp. 28–31, 2004.

[16] N. Dalal et al., "Histograms of oriented gradients for human detection," in *Proc. of IEEE Computer Vision and Pattern Recognition*, vol. 1, pp. 886–893, 2005.

Wei-Tsung Su received B.S. degree in Information and Computer Science from Chung Yuan Christian University, Taiwan in 2000. In 2008, he received Ph.D. degree in Computer Science and Information Engineering from National Cheng Kung University, Taiwan. He is an assistant professor in the Department of Computer Science and Information Engineering of Aletheia University. His research interests include cloud computing, pervasive computing, wireless sensor networks, and embedded system.



Yung-Hsiang Lu is an associate professor in the School of Electrical and Computer Engineering of Purdue University. He is an ACM distinguished scientist and ACM distinguished speaker. His research areas include computer systems, mobile and cloud computing, image processing, and robotics. He is an organizing member of the IEEE Rebooting Computing Working Group. He is the chair of the Multimedia Communication Systems Interest Group in IEEE Multimedia Communications Technical Committee He obtained the Ph.D. from the Department of Electrical Engineering at Stanford University.

INDUSTRIAL COLUMN: SPECIAL ISSUE ON CLOUD GAMING

Guest Editors: Gwendal Simon[1] and Adlen Ksentini[2]

[1]Télécom Bretagne, France, gwendal.simon@telecom-bretagne.eu

[2]University of Rennes 1, France, adlen.ksentini@irisa.fr

Video streaming has changed the way to consume multimedia content. Nowadays, billions of people use their (mobile) devices (ex. smartphone, smartTV, Tablet, PC) to access to streaming platforms like Youtube, Dailymotion, Pandora, Spotify, Deezer, etc. Cloud gaming combine streaming techniques with cloud computing to allow gamers to play sophisticated games (typically 3D games with rich textures) on their favorite device regardless of their computing capacity. The cloud gaming system does all the computing tasks by hosting the game engine "in the cloud", and streams the results to end-user device. The only requirement is to be connected to a network with low latency. However, several issues remain, particularly the challenge to offer the same quality of experience (QoE) to gamers as in the traditional solution where the game engine is hosted in a dedicated console.

This special issue of E-letter focuses on recent progress on cloud gaming, by covering different research topics. It is the great honor of the editorial team to have in this special issue some famous experts in the cloud gaming field, who report their solutions towards better cloud gaming solutions.

In the first paper entitled, "*QoE for Cloud Gaming*", Tobias et al. address the challenges of ensuring QoE for Cloud gamers. The authors begin by presenting a comprehensive survey on the QoE requirements and needs for cloud gaming. Then, they discussed several challenges and issues related to the identified requirements. This paper is a very good warm-up for the remaining papers.

The second paper "*Optimizing Cloud Gaming Experience and Profits with Virtual Machine Placement Policy*", authored by H-J. Hong et al, tackles the server (Virtual Machien) placement issue in Cloud gaming. Indeed, server placement has an important impact on gamers' QoE, since higher latency (i.e. server placed far from users) considerably decreases quality. In this work, the authors study the most cost-effective service placement in Cloud Gaming, which increases the total profit for operators while ensuring just-good-enough QoE to gamers.

Managing the cloud network is also important to reduce latency. In the third paper, "*Enhancing Cloud Gaming with Software Defined Networking*", S. Shirmohommadi investigates the use of Software Defined Networking (SDN) in order to improve the performance of cloud gaming systems. The author proposes a novel way to use SDN in order to steer the traffic in the cloud to reduce latency.

In the fourth paper "*Uniquitous: an Open-source Cloud-based Game System in Unity*", M. Luo and M. Claypool introduce *Uniquitous*, an open-source system for cloud gaming. The main difference between Uniquitous and the previous open-source proposals is that Uniquitous embeds the popular game engine, Unity, and thus enables a better exploration of the game code and the cloud system.

The last paper of this issue is "*Advanced GPU Pass-through and Cloud Gaming Performance: A Reality Check*" authored by Ryan Shea and Jiangchuan Liu. This paper deals with the problem of virtualization on GPU. Since game engines heavily consume GPU, cloud gaming systems should implement efficient techniques for the management of GPU when several Virtual Machines concurrently run on a server. This paper provides a reality check of current performances.

While this special issue is far from a complete coverage on this exciting research area, we hope that the five invited papers give the audiences a taste of the main recent activities in this area, and provide them an opportunity to discuss, explore and collaborate in the related fields. Finally, we would like to thank all the authors for their great contribution and the E-Letter Board for making this special issue possible.

Gwendal Simon is graduated from University Rennes 1 (France). During his PhD, he worked at Orange Research Labs. From 2004 to 2006 he was a researcher at Orange Labs. Since 2006, he has been Associate Professor at Telecom Bretagne, a graduate engineering school within the Institut Mines-Telecom. He was a visiting researcher at University of Waterloo from September 2011 to September 2012. His research interests include multimedia delivery systems (video and gaming) and network management.

Adlen Ksentini (SM'14) received the M.Sc. degree in telecommunication and multimedia networking from the University of Versailles Saint-Quentin-en-Yvelines, and the Ph.D. degree in computer science from the University of Cergy-Pontoise, in 2005, with a dissertation on QoS provisioning in the IEEE 802.11-based networks. He is currently an Associate Professor with the University of Rennes 1, France. He is a member of the Dionysos Team with INRIA, Rennes. He is involved in several national and European projects on QoS and QoE support in future wireless and mobile networks. He has co-authored over 60 technical journal and international conference papers. His other interests include future Internet networks, mobile networks, QoS, QoE, performance evaluation, and multimedia transmission. He received the best paper award from the IEEE ICC 2012 and ACMMSWiM 2005. He will be the TPC Chair of the Wireless and Mobile Symposium of the IEEE ICC 2016. He was a Guest Editor of the IEEE Wireless Communication Magazine and the IEEE Communication Magazine. He has been on the Technical Program Committee of major IEEE ComSoc, ICC/Globecom, ICME, WCNC, and PIMRC conferences.

QoE for Cloud Gaming

Tobias Hoßfeld[1], Florian Metzger[1], Michael Jarschel[2]
[1]University of Duisburg-Essen, Modeling of Adaptive Systems, Germany
{*tobias.hossfeld, florian.metzger*}*@uni-due.de*
[2]Nokia, Munich, Germany, michael.jarschel@nokia.com

1. Introduction

Cloud Gaming combines the successful concepts of Cloud Computing and Online Gaming. It provides the entire game experience to the users by processing the game in the cloud and streaming the contents to the player. The player is no longer dependent on a specific type or quality of gaming hardware, but is able to use common devices. However, at the same time the end device needs a broadband internet connection and the ability to display a video stream properly. While this may reduce hardware costs for users and increase the revenue for developers by leaving out the retail chain, it also raises new challenges for Quality of Service (QoS) in terms of bandwidth and latency for the underlying network. In particular, there is a strong interest in the player's Quality of Experience (QoE) by the involved stakeholders, i.e., the game providers and the network operators. Given similar pricing schemes, players are likely to be influenced by expected and experienced quality. Thus, a provider is interested to understand QoE and to react on QoE problems by managing or adapting the service.

There is also a strong academic interest, since QoE for cloud gaming as well as managing QoE for cloud gaming addresses a multitude of fascinating challenges in QoE. One might think that the topic of online video games is equally popular in research, but efforts are often solely focused on cloud gaming and its subjective QoE through user studies. Compared to plain video streaming, the inner properties of video games are not that straight-forward to observe from the outside. But to conduct proper measurements, it is essential to understand them.

In this positioning paper, we discuss some of those open issues on QoE for cloud gaming and postulate some promising research directions.

2. QoE Influence Factors of Cloud Gaming

A widely accepted definition of QoE is provided in [12] which we adopt here to cloud gaming. QoE is the degree of delight or annoyance of a game player, i.e., the user of a cloud gaming service. QoE results from the fulfillment of the player's expectations with respect to the enjoyment of the game in the light of the user's personality and current state. Thereby, user expectations are often coming from the experience with local games. For commercial cloud gaming services, those expectations are further shaped by the prize for the game and the in-game payments.

There are four different kinds of influence factors on cloud gaming QoE, that are addressing 1) user level, 2) system level, 3) content/game level, and 4) context level. A taxonomy of gaming QoE aspects is provided in [7] which also discusses gaming QoE influence factors in detail. We additionally differentiate here the 3) content factors reflecting the game itself, its mechanics and rules, etc. and the 2) system factors considering the technical influence factors like networking delays, the realization of the game in the cloud or concrete mechanisms towards improving QoS and QoE, e.g. adaptive streaming in cloud gaming [8].

2.1 User Level. For cloud gaming QoE, an important issue is the user itself. In particular, the experience of the user with the game (hardcore vs. casual gamer) is a relevant QoE influence factor. While a hardcore gamer may be very sensitive to, e.g., network delays, the casual gamer may not recognize those delays.
For cloud gaming QoE, it is necessary to investigate the different player types. In QoE studies, it is necessary to characterize the player types and to consider those player types in the analysis.

2.2 System Level. For understanding QoE and to improve QoE, we need to take a closer look at the model and interacting components of cloud gaming as illustrated in Figure .
At their core video games are essentially feedback-directed real-time simulators. The simulator's main loop consists of three central parts: reading input, updating the game state, and rendering the output. Every render-call means putting out a new video frame. As this framerate is usually not limited and variable, the game logic has to update its state on a time-scale operating independent of the current frame. Some games also update parts of the game on a fixed frequency, the so-called tick-rate, for example a non-game-influencing physics effect updating at a lower 30 Hz rate.
Online video games complicate this update logic a bit. In online games, the client is not the final authority over its game state any more. Instead, interpreted input commands are sent to the server and a preliminary game state is calculated locally. When the authoritative update from the server is received the two states can be

once again be synchronized. A further layer is added by cloud gaming, as the video output of the game is captured, encoded and then redirected to an additional computer.

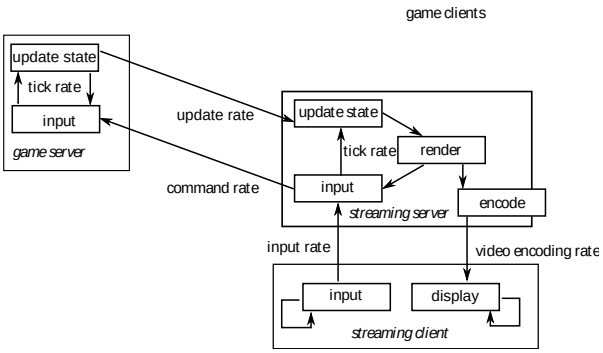


Figure 1. Model and interacting components of an online video game that is streamed through a Cloud Gaming service.

To assess cloud gaming QoE, the interaction between these different technical components needs to be understood in order to derive the end-to-end delay as perceived by the users purely from networking delays. Finding proper means for measuring end-to-end QoS is a challenging [14] but fundamental task in QoE management.
In subjective user studies, often the networking delay is changed and players are asked to rate QoE on a given rating scale. However, in those studies it is often unclear how the full end-to-end-delay was determined which in addition strongly depends on the system factors like the tick rates but also the game's implementation.
Properly conducted subjective studies must report the end-to-end delay in order to produce verifiable and comparable results. Likewise, QoE models need to take the end-to-end delay into account and a mapping of the networking delay to the end-to-end delay is required.

2.3 Content Level. The variety of games and their requirements is manifold. QoE studies typically consider only a few concrete game examples which do not allow to generalize the results. Is a classification of cloud games concerning QoE possible?
It is quite difficult to find games that are representative for certain input and latency demands. For example, the traditional game genre categorization is not a good starting point as games from the same category can be vastly different in terms of game speed and necessary reaction times. As a solution to this dilemma, proper metrics are required which characterize games and their QoE requirements.
For example, the following metrics might be helpful to assess a game's applicability for measurements.

- Required number of decisions or actions in a certain time span.
- Maximum successful reaction time to in-game actions.
- (Un-)Predictability of actions. A game with no surprising events will be much less influenced by a higher end-to-end latency.
- Accuracy and precision of input actions. Accuracy can be both in terms of temporal as well as spatial aspects which can be influenced by both the image quality and the frame rate.

The key idea is that games need to be characterized with respect to key QoE influence factors like interactivity, delay sensitivity, responsiveness, frequency of inputs, frequency of events, etc. QoE studies need to report those quantities. In QoE management, those factors need to be known to meet the QoS and QoE demands. A self-description of games is desired.
It can be seen that there the various influence factors on system, content but also user level are interacting: an experienced player of a certain game (or game type) may have higher QoS demands; memory effects on user level are related to predictable input actions and allow to compensate technical influence factors like lags.

2.4 Context Level. The context level considers aspects like the environment in which the user is consuming the service, the socio-cultural background, but also economic aspects like prices for the games. It is often challenging to directly take into account context factors in QoE modeling, although there is a clear influence.
The question arises which are appropriate QoE metrics to take into account context factors like prices.
In subjective studies, the users are often asked to rate the QoE on an absolute category rating scale from 1 (bad) to 5 (excellent). A typical QoE measure from such a study is the mean opinion score (MOS) averaging over all subjective ratings for a certain test condition. From a service provider's point of view, however, MOS values are not sufficient. Higher order statistics of subjective studies, acceptance tests, or user engagement metrics are required to truly understand QoE and business-related aspects of cloud gaming, such as the ratio of unsatisfied users, which is useful for predicting churn rates. Going beyond the MOS allows service providers to better provision their services so satisfy the QoE demands of the players [13].
In [18], a novel framework for jointly modeling QoE and user behavior is proposed, where user behavior is treated as one of the framework dimensions along with system performance and user state. For cloud gaming, the user engagement may be quantified in terms of playing time or in-game payments.

3. Results on Cloud Gaming QoE and QoS

We have outlined so far some limitations of subjective studies on cloud gaming QoE which may be as follows: missing description and characterization of the game and the players, missing accurate measurement of e2e delays as perceived by the end users, higher order analysis of results, different means like acceptance tests or user behavior analysis, neglecting influence factors, e.g., on context level. Nevertheless, we highlight here some relevant QoE results which focus on the networking perspective.

In [4] relevant QoE influence factors are identified for certain games and three different game categories (slow, medium, fast games) that have been subjected to worsening QoS parameters. Downstream packet loss and delay was noted be especially problematic for achieving a good quality.

Regarding the subjective quality in first person shooters, [5] finds a strong impact of the delay and packet loss on the experienced quality. Work in [3] uses an fEMG approach to examine individual gamers' reaction to various cloud games and measure the quality they are experiencing in terms of real-time strictness and MOS.[10] states that cloud games are more sensitive to latency than online games because game graphics are rendered on cloud servers and thin clients do not possess game state information that is required by delay compensation techniques. [5] studied QoE degradation resulting from network delay and packet loss. As a result, the test game was unplayable for 200 ms RTT and 1% packet loss.

Although the key technical influence factors are determined in several studies, there is currently no complete understanding of QoE. [15] investigates the relationship between content factors (game genre, video characteristics), input characteristics, and resulting system factors (network characteristics). Still, the interaction of the different influence factors needs to be investigated and related subjective studies need to be conducted to understand QoE for cloud gaming. The network characteristics transform in a complex way on user perceived influence factors (delay, artifacts, delayed interactions) which need to be analyzed.

4. Rising to the QoE Challenge

Commercial, internet-based cloud gaming has so far failed to establish itself in the gaming market. One of the main reasons for this failure is the inability of cloud gaming providers to deliver the same kind of QoE the players are accustomed to from their local gaming platforms via the internet at a competitive price point. This raises the question how and if the problem of low QoE can be solved in cloud gaming while still keeping the costs in check.

As those services have to compete with the visual and interactive fidelity of locally-run games, the willingness-to-pay [11] of customers will come into question, if the QoE becomes much lower than this point of reference.

As research has shown (cf. [4]), packet loss is a critical factor and has to be tackled, e.g., with application layer reliability schemes. Likewise, latency is generated at a multitude of points in the game streaming pipeline, which all have to be controlled individually, making the issue of latency a critical one in cloud gaming. Latency compensation techniques can also be an option, but might additionally increase the required computing power at the client's side, which would put the whole idea of cloud gaming in question, cf. [1].

Finally, providing sufficient bandwidth to the streaming process is one of the key factors. As many temporal coding features cannot be used in a real-time environment, very high bandwidths are required for an artifact-free transmission. This will overwhelm many consumer dial-up links. Other proposed adaptation schemes adapt the graphics settings of the game to reduce the visual complexity of the scenes [2], i.e., a trade-off between video and graphics quality. The impact of this in terms of QoE is however not yet fully understood.

Therefore, for the time being, commercial cloud gaming offerings are limited to local game streaming between two nodes in the same local network. This eliminates many of the latency and loss sources and enables very high streaming bandwidths. Current consumer products like Valve's Steam Link or NVIDIA Shield TV use this method rather than internet-based streaming. For the internet solutions to become viable, the QoE challenges have to be met.

5. Conclusions and Discussions

QoE for cloud gaming attracts attention in research and is an important criterion for the success of cloud gaming providers. There is an open source implementation which allows to investigate the behavior of the system and to implement own QoE management schemes [16]. However, QoE for cloud gaming has many different facets. So far, there is no common methodology to investigate QoE for cloud gaming – in contrast to well standardized tests, e.g., for speech quality. An ITU-T Recommendation [10] concerning subjectively measuring video game QoE is in preparation, which discusses game-relevant QoS-metrics as well as the selection of players and games.

A key point is the diversity of games. Proper metrics are required to describe and characterize games (content level) and players (user level). A self-description of games is desired which may also be utilized by QoE management. To fully understand QoE, more sophisticated QoE metrics beyond the MOS are required, as well as other concepts like acceptance tests or engagement metrics [13][17].

Game on!

References

[1] S. Wang and S. Dey. Addressing response time and video quality in remote server based internet mobile gaming. In WCNC, pages 1–6, 2010.

[2] Wang, Shaoxuan, and Sujit Dey. "Adaptive mobile cloud computing to enable rich mobile multimedia applications." IEEE Transactions on Multimedia 15.4 (2013): 870-883.

[3] Y.-T. Lee, K.-T. Chen, H.-I. Su, and C.-L. Lei, "Are all games equally cloud-gaming-friendly? an electromyographic approach," in Network and Systems Support for Games (NetGames), 2012 11th Annual Workshop on, Nov 2012, pp. 1–6.

[4] M. Jarschel, D. Schlosser, S. Scheuring, and T. Hossfeld, "An evaluation of qoe in cloud gaming based on subjective tests," in Innovative Mobile and Internet Services in Ubiquitous Computing (IMIS), 2011 Fifth International Conference on, 2011, pp. 330–335.

[5] V. Clincy and B. Wilgor, "Subjective evaluation of latency and packet loss in a cloud-based game," in Information Technology: New Generations (ITNG), 2013 Tenth International Conference on. IEEE, 2013, pp. 473–476.

[6] Möller, Sebastian, Dennis Pommer, Justus Beyer, and Jannis Rake-Revelant. "Factors influencing gaming qoe: Lessons learned from the evaluation of cloud gaming services." In Proc. 4th Int. Workshop on Perceptual Quality of Systems (PQS 2013), Wien, pp. 2-4. 2013.

[7] Möller, Sebastian, Signe Schmidt, and Justus Beyer. "Gaming taxonomy: An overview of concepts and evaluation methods for computer gaming qoe." In Quality of Multimedia Experience (QoMEX), 2013 Fifth International Workshop on, pp. 236-241. IEEE, 2013.

[8] Shirmohammadi, Shervin. "Adaptive streaming in mobile cloud gaming." E-LETTER (2013).

[9] Beyer, Justus, and Sebastian Möller. "Assessing the Impact of Game Type, Display Size and Network Delay on Mobile Gaming QoE." PIK-Praxis der Informationsverarbeitung und Kommunikation 37, no. 4 (2014): 287-295.

[10] Möller, Sebastian, Jan-Niklas Antons, Justus Beyer, Sebastian Egger, Elena Núñez Castellar, Lea Skorin-Kapov, and Mirko Sužnjević. "Towards a New ITU-T Recommendation for Subjective Methods Evaluating Gaming QoE." In QoMEX 2015.

[11] Sackl, Andreas, et al. "The QoE alchemy: turning quality into money. Experiences with a refined methodology for the evaluation of willingness-to-pay for service quality." *QoMEX 2012*

[12] Qualinet White Paper on Definitions of Quality of Experience (2012). European Network on Quality of Experience in Multimedia Systems and Services (COST Action IC 1003), Patrick Le Callet, Sebastian Möller and Andrew Perkis, eds., Version 1.2, March 2013.

[13] Hoßfeld, Tobias, Poul E. Heegaard, and Martin Varela. "QoE beyond the MOS: Added Value Using Quantiles and Distributions." In QoMEX 2015.

[14] Beyer, Justus, Richard Varbelow, Jan-Niklas Antons, and Steffen Zander. "A Method For Feedback Delay Measurement Using a Low-cost Arduino Microcontroller." In QoMEX 2015.

[15] Suznjevic, Mirko, Justus Beyer, Lea Skorin-Kapov, Sebastian Moller, and Nikola Sorsa. "Towards understanding the relationship between game type and network traffic for cloud gaming." In Multimedia and Expo Workshops (ICMEW), 2014 IEEE International Conference on, pp. 1-6. IEEE, 2014.

[16] Huang, Chun-Ying, Kuan-Ta Chen, De-Yu Chen, Hwai-Jung Hsu, and Cheng-Hsin Hsu. "GamingAnywhere: The first open source cloud gaming system." ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM) 10, no. 1s (2014): 10

[17] Streijl, Robert C., Stefan Winkler, and David S. Hands. "Mean opinion score (MOS) revisited: methods and applications, limitations and alternatives." Multimedia Systems (2014): 1-15.

[18] Reichl, Peter, et al. "Towards a comprehensive framework for QoE and user behavior modelling." Proc. 7th International Workshop on Quality of Multimedia Experience (QoMEX'15), Greece. 2015.

Tobias Hoßfeld is full professor and head of the Chair "Modeling of Adaptive Systems" at the University of Duisburg-Essen, Germany, since 2014. He finished his PhD in 2009 and his professorial thesis (habilitation) in 2013 at the University of Würzburg, Chair of Communication Networks. He has published more than 100 research papers in major conferences and journals, receiving 5 best conference paper awards, 3 awards for his PhD thesis, and the Fred W. Ellersick Prize 2013 (IEEE Communications Society) for one of his articles on QoE.

Florian Metzger received his doctorate degree in computer science at the University of Vienna, Austria, in 2015. Between 2010 and 2015 he was a researcher in the Future Communication group at the University of Vienna, Austria. Since 2015 he is a postdoctoral researcher at the Chair "Modeling of Adaptive Systems" at the University of Duisburg-Essen, Germany. His research interests include video streaming in mobile networks, video game modeling, as well as computer network transport protocols.

Michael Jarschel is working as a research engineer in the area of Network Softwarization at Nokia in Munich, Germany. He finished his Ph.D. thesis, titled "An Assessment of Applications and Performance Analysis of Software Defined Networking", at the University of Würzburg in 2014. His main research interests are in the applicability of SDN and NFV concepts to next generation mobile networks.

Enhancing Cloud Gaming with Software Defined Networking

Shervin Shirmohammadi
Distributed and Collaborative Virtual Environments Research (DISCOVER) Lab
University of Ottawa, Canada, shervin@eecs.uottawa.ca

1. Introduction

As explained in [5], Cloud gaming leverages the well-known concept of cloud computing to provide gaming services to players. Cloud gaming works by capturing the game events from players and transmit them to the cloud, processing those events and running the game logic in the cloud, rendering the game scene as video in the cloud, and streaming that video to the players. The advantage is that as long as the client can display video, which pretty much all end user devices today do, the user can play the game without installing it locally and without needing to have high-grade 3D graphics rendering and powerful computing hardware and software [5]. Cloud Gaming is already available as commercial products, such as Sony's PlayStation Now, Ubitus's GameNow, G-Cluster, Crytek's GFACE, PlayGiga, and LiquidSky, to name some. There are also efforts concentrating on the underlying technology behind Cloud Gaming, such as NVIDIA's Grid, OTOY, CiiNow, Kalydo, and GamingAnywhere[1], the latter as the only open source and free technology. Microsoft is also exploring cloud gaming technologies, with recent successes such as its Kahawai project [2].

We recently published a "Special Section on Cloud Gaming and Virtualization" in *IEEE Transactions on Circuits and Systems for Video Technology* [3]. The special section's 12 papers cover a wide variety of topics related to Cloud Gaming. Also previously, we discussed the issue of delay in cloud gaming, where and why it happens, and how we can reduce it [5]. In this article, we add to the above body of knowledge by focusing on Software Defined Networks (SDN) and how they can enhance the performance of cloud gaming systems.

2. SDN for Cloud Gaming

SDN has recently made much headway in both research and practice arenas. Because SDN separates the forwarding and the routing functionalities of the network and centralizes the routing part [6], it can lead to many benefits for networked applications such as cloud gaming. The cloud infrastructure in cloud gaming consists of a core switch that dispatches client requests among aggregation switches, each of which further dispatches the request to a Top of Rack (ToR) switch, which finally dispatches the request to a processing node running multiple Virtual Machines (VM). SDN can be utilized in such infrastructure to make routing and traffic engineering decisions in a centralized and hence more optimized manner, leading to a higher Quality of Experience (QoE) for gamers. An example is the system proposed in [4], which uses SDN to reduce the delay, jitter, and packet loss within the cloud infrastructure in a cloud gaming data center, leading to higher quality game play for the gamers. Figure 1 shows the cloud configuration proposed for cloud gaming in [4].

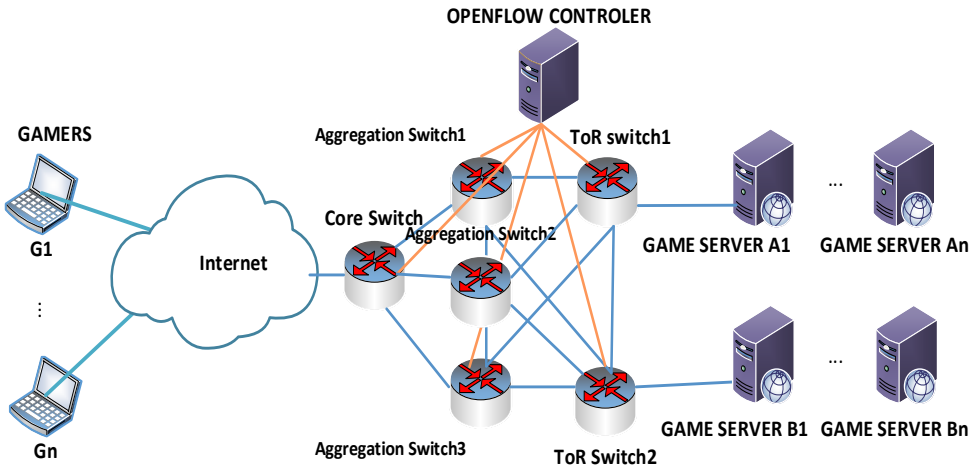


Figure 1. Cloud Gaming Architecture Using SDN [5]. G1 to Gn are the gamers.

3. Optimized Server and Path Selection Using SDN

Cloud gaming suffers from 1.7 times higher latency compared to game consoles and 3 times higher latency compared to PCs [8]. This negatively impacts the QoE of players. Moreover, only 70% of end-users are able to meet the required latency threshold of 100 msec to match the experience of console or PC games [9]. SDN can help reduce these negative impacts, by enabling us to make a more optimized selection of servers and paths within the cloud gaming data center. SDN's centralized architecture provides a global and complete view of the network and its resources, leading to optimized solutions. For example, in [7], we employ a Linear Programing (LP) optimization technique that considers the type of requested games, network information pertaining to link status, and current server loads, to select servers and paths, as described next.

Figure 2 shows the proposed method. The SDN controller periodically monitors the latency and available bandwidth on each link using the OpenFlow protocol. In [4], we described an algorithm to measure the delay of each path between the core switches and the ToR switches. The game server's performance analysis module monitors and analyzes the performance of the game servers in terms of available processing resources. Also, the predefined information and requirements of games are stored in an auxiliary database. Therefore, network-related information, server-related information, and games' requirements are fed into the proposed game-aware optimization method by the SDN controller, performance engine, and game database respectively. Afterwards, the proposed game-aware optimization method makes decisions on the sever selection and the corresponding paths, which are optimized due to the global perspective provided by SDN.

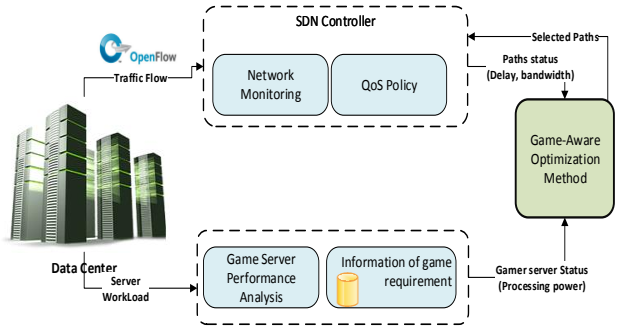


Figure 2. Proposed Game-Aware Optimization Method [7].

We mathematically formulate the optimization problem using LP, and we define an objective function to determine which path and server can minimize the overall delay associated with all gaming sessions running on the datacenter. The objective function consists of the weighted average of the total network and processing delay. Readers are referred to [7] for details.

The results of our emulation, using Ubuntu version 14.4 box and a Mininet emulator on the Oracle virtual box version 4.3, which allowed us to create a realistic network experiment with OpenFlow and SDN, show that using our proposed method, the average delay variation experienced by players is almost 14% and 8% less than the overall delay experienced in the traditional approaches of server-centric and network-centric methods, respectively, where the server-centric method refers to methods that try to improve utilization of network resources, and network-centric method refers to methods that try to improve utilization of server resources.

4. Conclusion

As cloud gaming become more popular, offering a high quality gaming experience to the players becomes more crucial in the mass adoption of cloud gaming. SDN can help, by optimizing the cloud infrastructure to reduce the two main obstacles in cloud gaming: bandwidth limits, and delay. In this article, we saw how SDN can be used to reduce the delay caused at the cloud side.

References

[1] C.Y. Huang et al., "GamingAnywhere: The first open source cloud gaming system", *ACM Transactions on Multimedia Computing, Communications, and Applications*, Vol. 10, Issue 1s, January 2014, Article No. 10.

[2] E. Cuervo et al., "Kahawai: High-Quality Mobile Gaming Using GPU Offload," *Proc. ACM Annual International Conference on Mobile Systems, Applications, and Services (MobiSys)*, Florence, Italy, May 18-22 2015, pp. 121-135.

[3] S. Shirmohammadi, M. Abdallah, D.T. Ahmed, K.T. Chen, Y. Lu, and A. Snyatkov "Guest Editorial for the Special Section on Cloud Gaming and Virtualization", *IEEE Trans. on Circuits and Systems for Video Technology*, 2015. (to appear)

[4] M. Amiri, H. Al-Osman, S. Shirmohammadi, and M. Abdallah, "An SDN Controller for Delay and Jitter Reduction in Cloud Gaming", *Proc. ACM Multimedia*, October 26-30 2015, Brisbane, Australia, 4 pages.

[5] S. Shirmohammadi, "Delay Reduction in Cloud Gaming", *IEEE COMSOC Multimedia Communications Technical Committee E-Letter*, Vol. 10, No. 6, November 2015.

[6] A. Yassine, H. Rahimi, and S. Shirmohammadi, "Software Defined Network Traffic Measurement: Current Trends and Challenges", *IEEE Instrumentation and Measurement Magazine*, Vol. 18, No. 2, April 2015, pp. 42-50.

[7] M. Amiri, H. Al-Osman, S. Shirmohammadi, and M. Abdallah, "SDN-based Game-Aware Network Management for Cloud Gaming", Proc. ACM/IEEE Network and Systems Support for Games, Zagreb, Croatia, December 3-4, 2015.

[8] http://www.geforce.com/whats-new/articles/geforce-grid.

[9] Choy, S., Wong, B., Simon, G. and Rosenberg, C. A hybrid edge-cloud architecture for reducing on-demand gaming latency. *Multimedia Systems*, 2014, 503-519.

Shervin Shirmohammadi received his Ph.D. degree in Electrical Engineering from the University of Ottawa, Canada, where he is currently a Full Professor at the School of Electrical Engineering and Computer Science. He is Director of the Distributed and Collaborative Virtual Environment Research Laboratory (DISCOVER Lab), and member of the Multimedia Communications Research Laboratory (MCRLab), conducting research in multimedia systems and networking, specifically in gaming systems, video systems, and their application in multimedia-assisted biomedical engineering. The results of his research have led to more than 250 publications, over 20 patents and technology transfers to the private sector, and a number of awards and prizes. He is Associate Editor-in-Chief of *IEEE Transactions on Instrumentation and Measurement*, Senior Associate Editor of *ACM Transactions on Multimedia Computing, Communications, and Applications*, and was Associate Editor of Springer's Journal of Multimedia Tools and Applications from 2004 to 2012. Dr. Shirmohammadi is a University of Ottawa Gold Medalist, a licensed Professional Engineer in Ontario, a Senior Member of the IEEE, and a Professional Member of the ACM.

Optimizing Cloud Gaming Experience and Profits with Virtual Machine Placement Policy

Hua-Jun Hong[1], De-Yu Chen[2], Chun-Ying Huang[3], Kuan-Ta Chen[2], and Cheng-Hsin Hsu[1]

[1]Department of Computer Science, National Tsing Hua Univiersity, Hsin Chu, Taiwan.

[2]Institute of Information Science, Acdemia Sinica, Taipei, Taiwan.

[3]Department of Computer Science and Engineering, National Taiwan Ocean University, Kee Lung, Taiwan.

1. Introduction

Cloud gaming attract gamers to play without expensive equipment and game providers to offer on-demand gaming services. Cloud gaming services providers, such as Gaikai [2], Ubitus [3], and OnLive [4] run their games on powerful servers hosted in cloud and stream the game scenes to gamers. Without running heavy game programs, the gamers only need to install a simple application on their heterogeneous devices, such as desktops, laptops, and smartphones to receive and play games. A market report [5] shows that the staggering growth of cloud gaming market will be 8 billion USD by 2017. The potential of cloud gaming service attracts more and more game providers [6].

Providing the cloud gaming service is challenging because of the tradeoff between gaming Quality-of-Experience (QoE) and provider's profits. More specifically, providing high QoE requires expensive hardware installed on cloud servers, which may lead to severe financial burden [4]. However, saving the costs of building the cloud servers may not lead to higher profits because lower QoE may drive the gamers away from the service. Moreover, different type of games needs different equipment, while different games require different levels of gaming experience. If cloud gaming service use high-end hardware to serve a user who requests for low QoE to play an out-of-date 2D game, it wastes lots resources and the profit. These diverse requirements make the cloud gaming providers harder to find the best tradeoff.

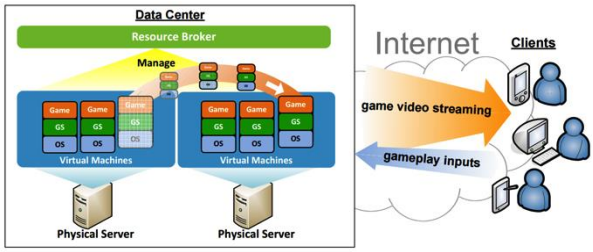


Figure 1. The architecture of cloud gaming services, where GS denotes cloud gaming server.

Due to the diverse requirements of gamers and corresponding games, different games on the same cloud machine lead to different degree of consolidation overhead. In this paper, we study the virtual machine placement problem to find the most cost-effective consolidation decision. As illustrated in Figure 1, we consider the VM placement problem to maximize the total profit while providing the just-good-enough QoE to gamers.

2. Measurement Study

We conduct measurement studies to model the implications of consolidating multiple cloud gaming servers on a physical machine. We adopt VMware and VirtualBox to create VMs, which run on physical machines and the GamingAnywhere (GA) [10] servers run in the VMs. We choose three games in different genres: Limbo, Sudden Strike: Normandy (Normandy), and Police Supercars Racing (PSR), and measure their performance over 5-min game sessions. We consider four metrics relevant to the VM placement problem: (i) CPU utilization: the average CPU load measured on the physical server, (ii) GPU utilization: the average GPU load measured on the physical server, (iii) frame rate: the average number of frames streamed per second, and (iv) processing delay: the average time for the GA server to receive, render, capture, encode, and transmit a frame.

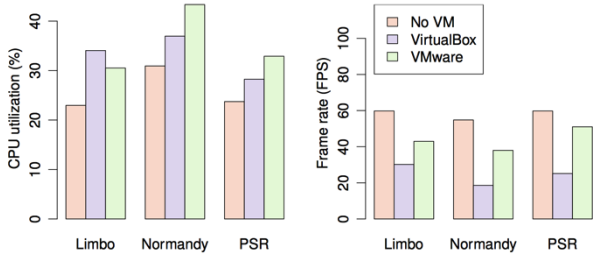


Figure 2. Virtualization overhead depends on game and VM implementations.

Figure 2 gives some sample results, which reveals that: (i) VMs lead to nontrivial overhead, (ii) different VMs result in different amount of overhead, and (iii) different games incur different workloads that may have distinct performance implications on different VMs. Hence, more extensive measurements are required to derive the prediction model of GA performance in each game/VM pair. The details of the extensive measurements are presented in [1] and we adopt sigmoid functions of the number of VMs on a physical machine to model CPU utilization, GPU utilization, frame rate, and processing delay.

3. VM Placement Problem

System overview. Figure 1 illustrates the system architecture of the cloud gaming platform, which consists of S physical servers, P gamers, and a broker. Each physical server hosts several VMs, while every VM runs a game and a game server (GS). Physical servers are distributed in several data centers at diverse locations. The gamers run game clients on desktops, laptops, mobile devices, and set-top boxes to access cloud games via the Internet.

The broker is the core of our proposal. The broker consists of a resource monitor and implements the VM placement algorithm. It is responsible to: (i) monitor the server workload and network conditions, and (ii) place the VMs of individual gamers on physical servers to achieve the tradeoff between QoE and cost that is most suitable to the cloud gaming service. In particular, for public cloud gaming services, the provider's profit is more important, while for closed cloud gaming services, the gaming QoE is more critical. The games may have diverse resource requirements, including CPU, GPU, and memory [7]. The paths between gamers and their associated servers have heterogeneous network resources, such as latency and bandwidth. Moreover, gamers can tolerate different QoE levels for different game genres [8]. Last, we note that the broker can be a virtual service running on a server or a server farm for higher scalability.

Notations and formulation. We use $u_s(v)$ and $z_s(v)$ to model the CPU and GPU utilizations of server s running v VMs. Details are given in [1]. We denote g_p as the hourly fee paid by gamer p. We let $w_s(v) = c_s(u_s(v)+z_s(v))$ be the operational cost of imposing CPU and GPU utilization $u_s(v)$ and $z_s(v)$ on s, where c_s is a cost term consisting of various components, such as electricity, maintenance, and depreciation. We also use sigmoid function to model frame rate $f_p(v)$ and processing delay $d_p(v)$. We let $x_{s,p} \in \{0,1\}(1 \le p \le P, 1 \le s \le S)$ be the decision variables, where $x_{s,p} = 1$ if and only if gamer p is served by a VM on server s. With the notations defined above, we formulate the provider-centric problem as:

$$\max[\sum_{p=1}^{P}\sum_{s=1}^{S} x_{s,p} g_p - \sum_{s=1}^{S} c_s(u_s(v) + z_s(v))]$$

The objective function maximizes the provider's net profit, i.e., the difference between the collected fee and cost. There are several constraints given in [1] and the most important one is to make sure that the gaming QoE degradation is lower than the user-specified maximal tolerant level. In summary, the formulation maximizes the provider's profit while serving each gamer with a (user-specified) just-good-enough QoE level.

The aforementioned problem formulation is provider-centric, and is suitable to public cloud gaming services. For closed cloud gaming services, e.g., in hotels, Internet cafes, and amusement parks, maximizing the overall QoE is more important as the network bandwidth is dedicated to cloud gaming. Therefore, with same constraints of provider-centric problem, we give the the gamer-centric objective function:

$$\min[\sum_{p=1}^{P} \gamma_{p,1} - \sum_{p=1}^{P} \gamma_{p,2} d_p]$$

The objective function minimizes the total QoE degradation. In particular, the QoE degradation is reduced when f_p increases or dp decreases as the empirically derived $\gamma_{p,1}$ is negative and $\gamma_{p,2}$ is positive, where $\gamma_{p,1}$ and $\gamma_{p,2}$ are model parameters that can be empirically derived.

To solve the provider-centric problem, we propose an efficient algorithm, called Quality-Driven Heuristic (QDH_L), and compare with optimal solution (OPT_L) computed by CPLEX in next section. We also make an adaptive version, call QDH'_L to solve the gamer-centric problem, and compare with the optimal solution (OPT'_L). The details of the algorithms are presented in [1].

4. Testbed

Implementation. We have implemented a complete cloud gaming system consisting of a broker, physical servers, and GA servers/clients, as illustrated in Figure 3. We adopt VMWare ESXi 5.1 as the virtualization software on physical servers. ESXi allows us to create VMs on physical servers, and each VM hosts a GA server and a game chosen by the corresponding gamer. We employ VMware vCenter 5.1 as the platform for our broker, which is comprised of Single-Sign-On for user authentication and Inventory Service for managing/monitoring the VMs on ESXi servers. The Inventory Service comes with different APIs, and we use its Java API to interface with the vCenter on the broker so as to control ESXi servers on all physical servers.

Figure 3 shows the flow of our system. We integrate the GA client and server with VMware ESXi and vCenter. In particular, the GA client provides an interface for gamers to send their accounts and passwords to the broker (1). Upon being authenticated (2), the GA client sends the user-specified game to the broker, and the broker determines where to create a new VM for that game based on the status of all physical servers and networks (3). The broker then instructs the chosen physical server to launch a VM (4)

and sends the VM's IP address to the GA client (5, 6). Last, the GA client connects to the GA server (7), instructs the GA server to run the user-specified game (8), and sends the stream of game to GA Client (9). This starts a new GA game session.

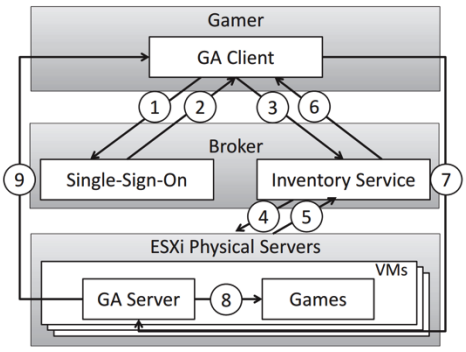
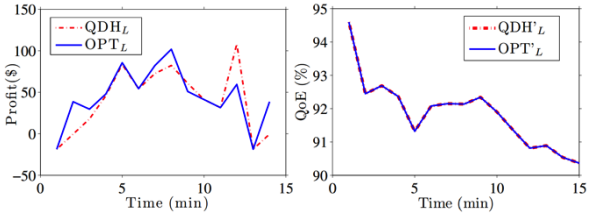


Figure 3. The implemented prototype system.

Experiment setup. To quantify the QDH_L/QDH'_L algorithms, we employ a testbed with 9 physical servers, 15 gamers, and 3 games. In every minute, each gamer joins (leaves) a game session with a probability of D% (1 − D%), where D is a system parameter. Each simulation lasts for T minutes. We assume that each physical server can serve up to two VMs and each VM launches a randomly selected game. In each simulation, we measure the fps and processing delay, and use them in the quality model. Also, we measure the CPU and GPU utilizations, and use them in the profit model. We inject realistic network latency using Dummynet [9]. Last, we set D = 90%, T = 15 minutes and consider the two performance metrics:

- *Net profit.* The total provider profit in every minute.
- *Quality of Experience.* The gaming QoE normalized in the range of [0%, 100%].

Results. We compare the QDH_L/QDH'_L algorithms against the optimal solution that exhaustively checks all servers for each new gamer. We refer to the optimal solutions as OPT_L/OPT'_L. Figure 4 reports the average performance over time. Figure 4 shows that QDH_L and OPT_L result in similar net profit. More specifically, the OPT_L algorithm outperforms the QDH_L algorithm in the first half of the experiment, but the QDH_L occasionally performs better in the second half. A closer look indicates that once game sessions start, they will be executed until the gamers leave. Therefore, even though OPT_L selects the best VM placements for the incoming gamers, it cannot foresee the future (e.g., when will the gamers leave), and thus its profit may be lower than that of the QDH_L algorithm. Nonetheless, the overall profit of QDH_L is still 10% lower than the optimum.



(a) (b)

Figure 4. Comparisons between QDH_L/QDH'_L and OPT_L/OPT'_L : (a) net profits and (b) quality.

5. Conclusion

We study the VM placement problems to (i) maximize the profits with good-enough QoE and (ii) minimize the total gaming experience degradation. We have implemented a testbed using an open-source cloud gaming system, Gaminganywhere [10], to conduct measurement study and evaluate our system. With the measurement study, we derive various system models. We formulated and solved the VM placement problems. The evaluation results show that, compared to the optimal algorithms, our algorithms achieve almost optimal net profit or QoE.

References

[1] H. Hong, D. Chen, C. Huang, K. Chen, and C. Hsu, "Placing virtual machines to optimize cloud gaming experience," IEEE Transactions on Cloud Computing, vol. 3, no. 1, pp. 42–53, January 2015

[2] "Gaikai," http://www.gaikai.com/

[3] "Ubitus," http://www.ubitus.net/

[4] "Onlive," http://www.onlive.com/

[5] "Distribution and monetization strategies to increase revenues from cloud gaming," http://www.cgconfusa.com/report/documents/Content-5minCloudGamingReportHighlights.pdf

[6] "Cloud gaming adoption is accelerating and fast!" http://www.nttcom.tv/2012/07/09/cloud-gaming-adoption-is-acceleratingand-fast/

[7] M. Claypool, "Motion and scene complexity for streaming video games," in Proc. of the International Conference on Foundations of Digital Games (FDG'09), Port Canaveral, FL, April 2009, pp. 34– 41.

[8] C. Mark and C. Kajal, "Latency and player actions in online games," Communications of the ACM, vol. 49, no. 11, pp. 40–45, November 2006.

[9] "DummyNet," http://info.iet.unipi.it/~luigi/dummynet/

[10] C. Huang, C. Hsu, Y. Chang, and K. Chen, "Gaminganywhere: An open cloud gaming system," in Proc. of the ACM Multimedia Systems Conference (MMSys'13), Oslo, Norway, February 2013, pp. 36–47.

Hua-Jun Hong received the B.S. and M.S. degrees in computer science from National Tsing Hua University, Hsinchu, Taiwan, where he is currently working toward the Ph.D. degree. His research interests include fog computing, cloud computing, cloud gaming, and multimedia networking.

De-Yu Chen is a research assistant at the Institute of Information Science of Academia Sinica. He received his M.S. in Computer Science from National Taiwan University in 2009, and his B.B.A. in Business Administration from National Taiwan University in 2006. His research interests include cloud computing, distributed computing, and network traffic analysis.

Chun-Ying Huang (S'03–M'08) received the B.S. degree in computer science from National Taiwan Ocean University, Keelung, Taiwan, in 2000; the M.S. degree in computer information science from National Chiao Tung University, Hsinchu, Taiwan, in 2002; and the Ph.D. degree from the Department of Electrical Engineering, National Taiwan University, Taipei, Taiwan, in 2007. He is an Associate Professor with the Department of Computer Science and Engineering, National Taiwan Ocean University. His research interests include computer network and network security issues, including traffic measurement and analysis, malicious behavior detection, and multimedia networking systems. Dr. Huang is a member of the Association for Computing Machinery, the Chinese Cryptology and Information Security Association, and the Institute for Information Systems and Computer Media.

Kuan-Ta Chen (S'04–M'06–SM'15) received the B.S. and M.S. degrees in computer science from National Tsing Hua University, Hsinchu, Taiwan, in 1998 and 2000, respectively, and the Ph.D. degree in electrical engineering from National Taiwan University, Taipei, Taiwan, in 2006.

He is a Research Fellow with the Institute of Information Science and the Research Center for Information Technology Innovation (joint appointment), Academia Sinica, Taipei. His research interests include quality of experience, multimedia systems, and social computing.

Dr. Chen is a Senior Member of the Association for Computing Machinery. He is an Associate Editor of *ACM Transactions on Multimedia Computing, Communications, and Applications* in 2015.

Cheng-Hsin Hsu (S'09–M'10) received the B.Sc. degree in mathematics and the M.Sc. degree in computer science and information engineering from National Chung Cheng University, Minhsiung, Taiwan, in 1996 and 2000, respectively, the M.Eng. degree in electrical and computer engineering from University of Maryland, College Park, MD, USA, in 2003, and the Ph.D. degree in computing science from Simon Fraser University, Burnaby, BC, Canada, in 2009. He is an Associate Professor with National Tsing Hua University, Hsinchu, Taiwan. His research interests include multimedia networking and distributed systems. Dr. Hsu is an Associate Editor of ACM Transactions on Multimedia Computing, Communications, and Applications.

Uniquitous: an Open-source Cloud-based Game System in Unity

Meng Luo and Mark Claypool

Computer Science and Interactive Media & Game Development
Worcester Polytechnic Institute, Worcester, MA 01609, USA
{mluo2,claypool}@wpi.edu

1. Introduction

Cloud gaming is an emerging service based on cloud computing technology which allows games to be run on a server and streamed as video to players on a lightweight client. Cloud gaming is estimated to grow from $1 billion in 2010 to $9 billion in 2017 [1], a rate much faster than either international or online boxed game sales.

Figure 1 depicts how games can be run in the cloud. The game computation, normally done on the client's computer or game console is instead done on one of many cloud servers, on the right. The server maintains the game world and computes the game scene images that the player sees on the screen, sending these game frames down to the client as streaming video. The client, on the left, can be "thin" since it no longer needs to do the heavy-weight computation of updating the game world and rendering the game scene – the client only needs to display the game frames as video and manage the player input. The client captures the player input in the form of mouse, keyboard and/or game controller actions, and sends it upstream to the game server in the cloud where the server incorporates the input into the game world as if the entire game was played locally on a traditional, non-cloud-based client.

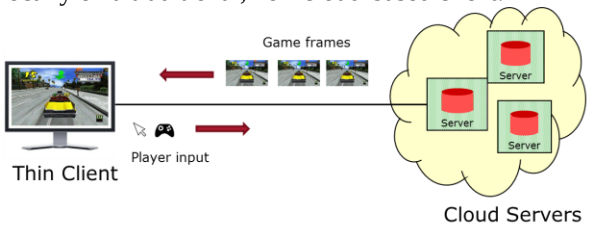


Figure 1. Cloud gaming

Cloud gaming provides benefits to players, developers and publishers over traditional gaming. Cloud gaming allows players to access games streamed as video through mobile devices, such as laptops, tablets and smart phones, which provides the potential to play the same game everywhere without constraints on hardware. Game developers only need to develop one version for the cloud server platform instead of developing a version for each client type, thus reducing game development time and cost. Publishers can more easily protect against piracy since cloud games are only installed in the cloud, thus limiting the ability of malicious users to make illegal copies.

Despite some recent successes, before cloud gaming can be deployed widely for all types of games, game devices and network connections, cloud gaming must overcome a number of challenges: 1) network latency, inherent in the physical distance between the server and the client, must be mitigated; 2) high network capacities are needed in order to stream game content as video down to the client; and 3) processing delays at the cloud gaming server need to be minimized in order for the game to be maintained, rendered and streamed to the client effectively for playing. Research and development required to overcome these challenges need cloud gaming testbeds that allow identification of performance bottlenecks and exploration of possible solutions.

Several commercial cloud gaming systems, such as OnLive [2] and StreamMyGame [3] have been used for cloud gaming research. Although these commercial services can be readily accessed, their technologies are proprietary, providing no way for researchers to access their code. This makes it difficult for researchers to explore new technologies in cloud-based gaming and makes it difficult for game developers to test their games for suitability for cloud-based deployment. While GamingAnywhere [4] provides an open source cloud gaming system, it remains separated from the game itself, not supporting integration and simultaneous exploration of game code and the cloud system. For instance, researchers cannot easily explore latency compensation techniques that require both the game code and the networking code to monitor player lag if the game code is separate from the cloud system.

In order to provide a more flexible and easily accessed platform for cloud gaming researchers and game developers, we present *Uniquitous* [5], a cloud gaming system implemented using Unity [6]. Unity is a cross-platform game creation system with a game engine and integrated development environment. Uniquitous is exported as an independent Unity package, which can blend seamlessly with existing Unity projects, making it especially convenient for

Unity developers, one of the largest and most active developer communities in the world – the Unity community increased from 1 million registered developers in 2012 to over 5 million in 2015 [7].

Uniquitous is open source, allowing modification and configuration of internal cloud gaming structures, such as frame rate, image quality, image resolution and audio quality, in order to allow exploration of system bottlenecks and modifications to meet client-server requirements. In addition to enabling system modifications, by being in Unity, Uniquitous enables game adjustments for further exploring the relationship between the game itself and cloud gaming performance. For example, game objects can be adjusted to study the effect of scene complexity on network bitrates, or camera settings can be altered to study the effect of perspective on cloud gaming frame rates. Although Uniquitous was developed on a desktop, Unity can build to both iOS and Android with full networking support, allowing Uniquitous to provide interactive streaming game video to mobile devices.

This article provides a brief introduction of the design and evaluation of Uniquitous. For details see other publications [8, 9] with source code and system documentation available online [5].

2. Architecture

Uniquitous' architecture is shown in Figure 2. It is composed of three entities: Unity Project, Uniquitous Server and Uniquitous Thin Client. The Uniquitous Server and the Uniquitous Thin Client run on two separate computers connected by an Internet connection while the Unity Project runs on the same computer as the Uniquitous Server. Figure 2 shows three types of data flows in Uniquitous, illustrated with different shades/colors: the red image flow carries data for the game frames; the green audio flow carries data for the game audio; and the blue input flow carries data for user input. Flows within components on the same machine are represented with dashed lines while flows across the network are shown with solid lines.

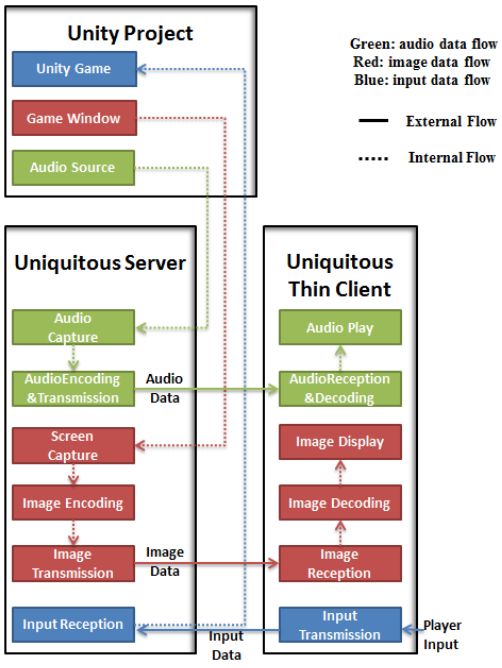


Figure 2. Uniquitous architecture

3. Experiments

We conducted micro experiments to evaluate the performance of the Uniquitous server components, focusing on processing time for bottleneck analysis, and macro experiments to evaluate the Uniquitous system as perceived by the player, focusing on game image quality and frame rate since they are among the most important to the player. Select results are presented in this section, with full results available in [9].

All experiments were run on PCs with Intel 3.4 GHz i7-3770 processors, 12 GB of RAM and AMD Radeon HD 7700 series graphic cards, each running 64-bit Windows 7 Enterprise. The PCs were connected by a 100 Mbps network LAN. The games tested, the Car Tutorial [10] and Angry Bots (version 4.0) [11], are provided by Unity Technologies.

Since Unity by default can do JPEG decoding, the Image Encoding component is implemented using a JPEG encoder [12]. While future work could use inter-frame compression common in video systems (e.g., H.264), JPEG encoding is sufficient to implement and evaluate our Uniquitous prototype.

To evaluate frame rates achievable in Uniquitous, 44 configurations for Car Tutorial and 37 configurations for Angry Bots were tested. Each configuration varied the JPEG encoding quality factor and resolution. To compute the frame rate, the time differences between frames provided the average frame intervals, and the inverse provided the frame rates. Figure 3 shows the frame rate results for Angry Bots. The x axis is the JPEG quality factor, and the y axis is the frame rate. Each point is the framerate average during a predefined period with trendlines grouping the different screen resolutions. Note, the higher resolution images are not tested at higher JPEG quality factors since RPC limits prevent images larger than 64 Kbytes from being transmitted.

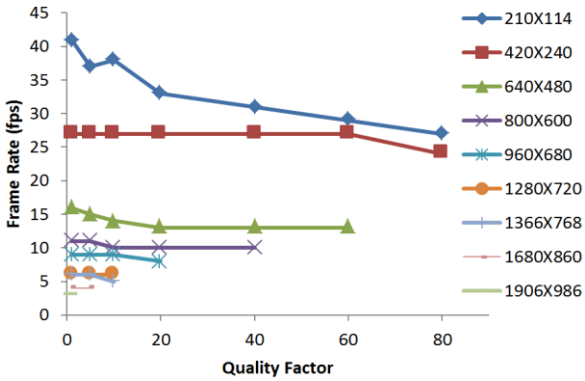


Figure 3. Frame rate versus JPEG quality factor for Angry Bots

From Figure 3, Angry Bots can achieve a maximum frame rate of 41 fps at a 210×114 resolution and 1 JPEG encoding. With the exception of this smallest image resolution, decreasing the JPEG quality factor does little to change the frame rate. However, increasing the frame resolution has a pronounced effect on decreasing the frame rate for both games. Based on previous results [13], frame rate is more important to players than resolution and a game system needs to provide a minimum of 15 fps for reasonable player performance. Both games tested can achieve 15 fps at a resolution of 640×480, hence is the recommended resolution setting for Uniquitous on this hardware setup. Based on player pilot tests, under these settings, both games are quite playable.

4. Frame Rate Predicting Model

With all the data collected from the experiments, we made a model to predict Uniquitous frame rate for configurations not yet tested. In order to build the model, we used a Weka classifier [14] with a 10-fold cross validation to make a linear regression model for both games:

$F_{CarTutorial} = 1 / (0.1348 \times R + 0.118 \times Q + 21.0)$
$F_{AngryBots} = 1 / (0.1361 \times R + 0.1224 \times Q + 22.5)$

where F is the predicted frame rate, R is the total pixel resolution divided by 1000, and Q is the JPEG quality factor. In order to validate our model, new R and Q values that had not been tested before were chosen, 35 for the Car Tutorial and 30 for Angry Bots, and the actual frame rates measured. The results are show in Figure 4.

The x axis is the predicted frame rate and the y axis is the actual frame rate as measured. Each point is the average frame rate over the experimental run. The diagonal line shows what would be perfect prediction. Generally, most of the data points are near this line,

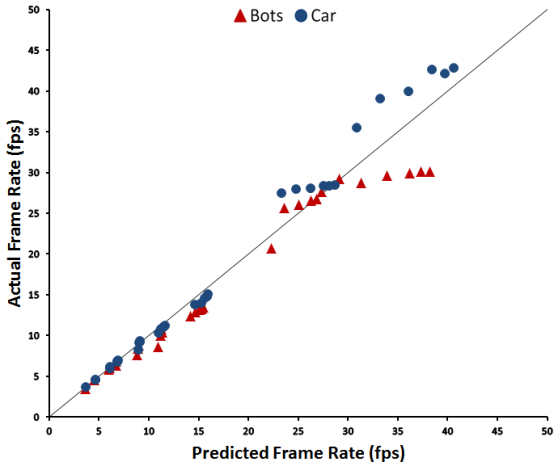


Figure 4. Actual versus predicted frame rate

showing that the model is generally quite accurate. The points are somewhat closer to the line for frame rates under 20 fps than for frame rates over 25 fps, probably due to unaccounted for processing that accumulates more with more frames per second. The actual and predicted frame rates have a correlation of 0.995 for Car Tutorial and 0.981 for Angry Bots.

5. Conclusion

Realizing the potential for cloud gaming requires testbed systems for researchers and developers. This article introduces Uniquitous [5], an open source cloud gaming system in Unity, providing a prototype that can be used for evaluating cloud gaming performance tradeoffs. Uniquitous seamlessly blends with Unity game development, providing control not only over the game system but also over the game content in a cloud-based environment. Micro experiments provide performance evaluation of the Uniquitous components, macro experiments evaluate game quality of the Uniquitous system, and a model predicts Uniquitous frame rates for games and hardware not yet tested. Validation of the model shows effectiveness for predicting frame rate over a range of configuration parameters. The evaluation shows the Unity Project running the game is the most time consuming component on the server when the game image quality and resolution are both low, but Image Encoding becomes the bottleneck for higher resolutions. For our system testbed, JPEG quality factors below 35 and resolutions below 640×480 pixels provide a configuration suitable for game play.

Future work can seek to increase Uniquitous frame rates and/or support higher resolutions and image qualities by addressing the identified bottlenecks. In addition, more game genres can be tested, exploring the relationship between the game genre and cloud

gaming performance. Lastly, since Unity IOS and Android are fully supported by Unity, Uniquitous can be extended and evaluated on mobile devices, helping research and development of cloud-based games on wider range of clients.

References

[1] "Distribution and Monetization Strategies to Increase Revenues from Cloud Gaming," goo.gl/YnlE9V, 2012, accessed 01-May-2014.

[2] OnLive: http://onlive.com/

[3] StreamMyGame: http://streammygame.com/

[4] C. Huang, Y. C. C. Hsu, and K. Chen, "GamingAnywhere: An Open Cloud Gaming System," in Proceedings of ACM MMSys, Oslo, Norway, Feb. 2013.

[5] Uniquitous: http://uniquitous.wpi.edu/

[6] Unity3d: http://unity3d.com/

[7] "Unity Company Facts": http://unity3d.com/public-relations, accessed 09-Jun-2015.

[8] Meng Luo and Mark Claypool. "Uniquitous: Implementation and Evaluation of a Cloud-based Game System in Unity", In *Proceedings of the IEEE Games, Entertainment, Media Conference (GEM)*, Toronto, Canada, October 2015.

[9] M. Luo, "Uniquitous: Implementation and Evaluation of a Cloud-Based Game System in Unity 3D," Master's Thesis, Worcester Polytechnic Institute, 2014, adv: M. Claypool.

[10] Car Tutorial:
https://www.assetstore.unity3d.com/en/#!/content/1012

[11] Angry Bots:
https://www.assetstore.unity3d.com/en/#!/content/12175

[12] A. Broager, "JPEG Encoder Source for Unity in C#," goo.gl/FwOfOW, accessed 06-May-2014.

[13] M. Claypool and K. Claypool, "Perspectives, Frame Rates and Resolutions: It's all in the Game," in *Proceedings of Foundations of Digital Games (FDG)*, FL, USA, Apr. 2009.

[14] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten, "The WEKA Data Mining Software: An Update," *SIGKDD Explorations*, vol. 11, no. 1, 2009.

Meng Luo received a B.Eng. degree in Telecommunications Engineering with Management from Beijing University of Posts and Telecommunications in 2012. He received an M.S. degree in Interactive Media and Game Development from Worcester Polytechnic Institute in 2014. Currently he is working as a software engineer at Advanced Visual Systems, Inc.

Mark Claypool is a Professor in Computer Science and Interactive Media & Game Development at Worcester Polytechnic Institute in Massachusetts, USA. Dr. Claypool earned M.S. and Ph.D. degrees in Computer Science from the University of Minnesota in 1993 and 1997, respectively. His primary research interests include multimedia networking, congestion control, and network games.

Advanced GPU Pass-through and Cloud Gaming Performance: A Reality Check

Ryan Shea and Jiangchuan Liu

Simon Fraser University

{rws1, jcliu}@sfu.ca

1. Introduction

Existing cloud gaming platforms have mainly focused on private, non-virtualized environments with proprietary hardware. Modern public cloud platforms heavily rely on *virtualization* for efficient resource sharing, the potentials of which have yet to be fully explored. Migrating gaming to a public cloud is non-trivial however, particularly considering the overhead for virtualization. Further, the use GPUs for game rendering has long been an obstacle in virtualization. Cloud gaming, in its simplest form, renders an interactive gaming application remotely in the cloud and streams the scenes as a video sequence back to the player over the Internet. A cloud gaming player interacts with the application through a *thin client*, which is responsible for displaying the video from the cloud rendering server as well as collecting the player's commands and sending the interactions back to the cloud.

This new paradigm of gaming services brings immense benefits by expanding the user base to the vast number of less-powerful devices that support thin clients only, particularly smartphones and tablets [1]. Extensive studies have explored the potential of the cloud for gaming and addressed challenges therein [2] [3] [4]. Open-source cloud gaming systems such as *GamingAnywhere* for Android OS [5] have been developed.

This letter takes a first step towards bridging the online gaming system and the public cloud platforms. We closely examine the technology evolution for GPU virtualization and pass-through, and measure the performance of both the earlier and the advanced solutions available in the market.

2. GPU Virtualization and Pass-through

Recent hardware advances have enabled virtualization systems to perform a one-to-one mapping between a device and a virtual machine guest, allowing hardware devices that do not virtualize well to still be used by a VM, including GPU. Both Intel and AMD have created hardware extensions for such device *pass-through*, namely VT-D by Intel and AMD-Vi by AMD. They work by making the processor's *input/output memory management unit* (IOMMU) configurable, allowing the systems hypervisor to reconfigure the interrupts and *direct memory access* (DMA) channels of a physical device, so as to map them directly into one of the guests [6].

As illustrated in Figure 1a, data flows through DMA channels from the physical device into the memory space of the VM host. The hypervisor then forwards the data to a virtual device belonging to the guest VM. The virtual device interacts with the driver residing in the VM to deliver the data to the guest's virtual memory space. Notifications are sent via interupts and follow a similar path. Figure 1b shows how a 1-1 device pass-through to a VM is achieved. As can be seen, the DMA channel can allow data to flow directly from the physical device to the VMs memory space. Also, interrupts can be directly mapped into the VM through the use of remapping hardware, which the hypervisor configures for the guest VM.

The advanced pass-through grants a single VM a one-to-one hardware mapping between itself and the GPU. These advances have allowed the cloud platforms to offer virtual machine instances with GPU capabilities. For example, Amazon EC2 has added a new instance class known as the GPU Instances, which have dedicated NVIDIA GPUs for graphics and general purpose GPU computing. There have also been recent studies on enabling multiple VMs to access CUDA-enabled GPUs [7][8], analyzing the performance of CUDA applications using a GPU pass-through device in Xen [9].

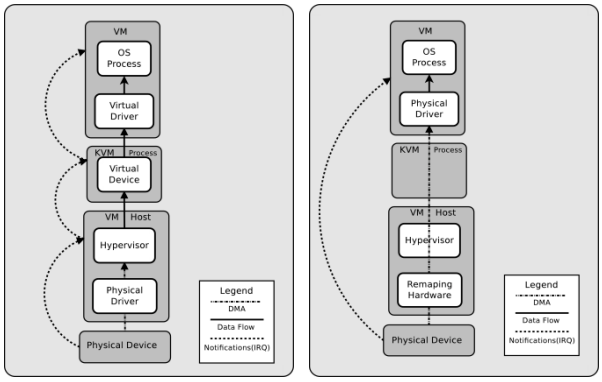


Figure 1: Shared vs Pass-through Device Simplified Architecture

3. Advanced GPU Pass-through and Gaming Performance: A Reality Check

We will now compare an older more primitive implementation of virtualized device pass-through from 2011 to a newer more optimized version from 2014. For brevity, we will refer to these two platforms as "earlier" and "advanced", respectively. With direct access to the hardware, these local virtualized platforms facilitate the measurement of virtualization overhead on game applications Since we are interested in the impact virtualization overhead has on gaming performance, we also compare these system to their optimal non-virtualized performance (referred to as "bare-metal" performance).

Earlier GPU pass-through Platform (2011)

Our first test system is a server with an AMD Phenom II 1045t 6-core processor running at 2.7 Ghz. The motherboard's chipset is based on AMD's 990X. The server is equipped with 16 GB of 1333 MHz DDR-3 SDRAM. The GPU is an AMD-based HD 5830 with 1 GB of GDDR5 memory. The Xen 4.0 hypervisor is installed on our test system and the host and VM guests used Debian as their operating system. We configure Xen to use the HVM mode, since the GPU pass-through requires hardware virtualization extensions. The VM is given access to 6 VCPUs and 8048 MB of RAM.

Advanced GPU pass-through Platform (2014)

Our second system is a state-of-the-art server with an Intel Haswell Xeon E3-1245 quad core (8 threads) processor. The motherboard utilizes Intel's C226 chipset, which is one of Intel's latest server chipsets, supporting device pass-through using VT-D. The server has 16 GB of 1600 MHz ECC DDR-3 memory installed. We have also installed an AMD based R9-280x GPU with 3 GB of GDDR5 memory. The Xen 4.1 hypervisor is installed and the VM guests again use Debian as their operating system.

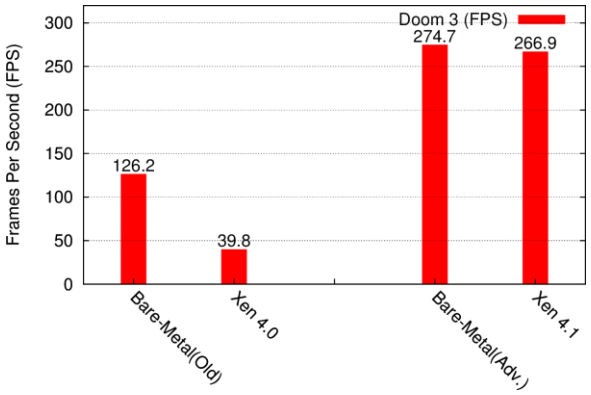


Figure 2: Doom 3 Performance

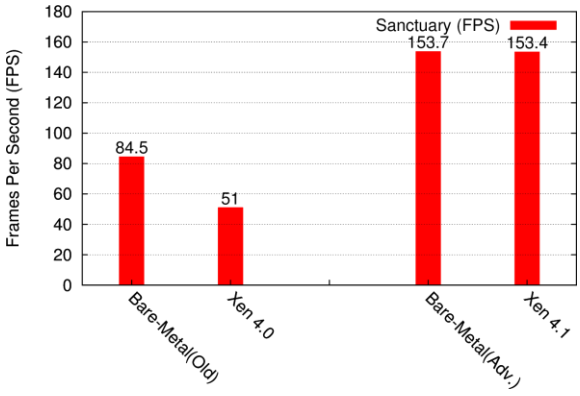


Figure 3: Unigine Sanctuary Performance

Comparison and Benchmarks

As the optimal baseline for comparison, for both systems we run each test on a bare-metal setup with no virtualization, i.e., the system has direct access to the hardware. The same drivers, packages and kernel were used as in the previous setup. This particular configuration enabled us to calculate the amount of performance degradation that a virtualized system can experience.

To compare the pass-through performance, we have selected two game engines, both of which have cross-platform implementation, and can run natively on our Debian Linux machines. The first is Doom 3, which is a popular game released in 2005, and utilizes OpenGL to provide high quality graphics. The second is the Unigine's Sanctuary benchmark, which is an advanced benchmarking tool that runs on both Windows and Linux. The Unigine engine uses the latest OpenGL hardware to provide rich graphics that are critical to many state-of-the-art games. For each of the following experiments, we run each benchmark three times and depict the average. For Doom3 and Sanctuary, we give the results in *frames per second* (fps).

Game Performance

For Doom 3, we utilize the built in time demo, which loads a predefined sequence of game frames as fast as the system will render them. In Figure 2, we show the results of frame rates, running on our bare metal systems as well as on the Xen virtualized systems. We start the discussion with our older 2011 server. This bare metal system performs at 126.2 fps, while our virtualized system dramatically falls over 65% to 39.2 fps. Our newer 2014 system processing the frames at over 274 fps when running directly on the hardware and falling less than 3% when run inside a virtual machine. This first virtualization experiment makes it clear that the device pass-through technology has come a long way in terms of performance. The advanced

platform performs within 3% of the optimal bare-metal result.

To confirm the performance implications with newer and more advanced OpenGL implementations, we next run the Unigine Sanctuary benchmark. The results are given in Figure 3. Once again, we see that our earlier virtualized system shows significant signs of performance degradation when compared to its bare-metal optimal. The earlier system drops from 84.4 fps to 51 fps when virtualized, i.e., nearly 40%. The advanced system has near identical performance when the game engine is running in a virtualized environment or directly on the hardware.

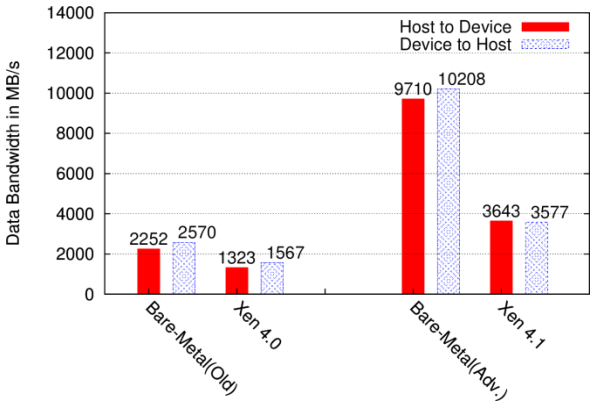


Figure 4: Memory Bandwidth by System

GPU Memory Bandwidth

Memory transfer from the system's main memory to the GPU can be a bottleneck for gaming, which can be even more severe when the transfer is performed in a virtualized environment [10] . This would affect the game's performance as well, because both Doom 3 and the Unigine engine must constantly move data from the main memory to the GPU for processing. To understand the impact, we ran a simple memory bandwidth experiment written in OpenCL by NVIDIA. We tested three different copy operations, from host's main memory (DDR3) to the GPU device's global memory (GDDR5), from the device's global memory to the host's main memory and finally from the devices global memory to another location on the devices global memory. We give the results for host-to-device and device-to-host experiments in Figure 4.

For both systems, the bare-metal outperforms the VMs in terms of memory bandwidth across the PCI-E bus to the GPU. The earlier system loses over 40% of its memory transfer to and from the host memory when compared to the bare-metal system. The newer more advanced platform degrades even more, losing over 60% of its memory transfer speed when virtualized. Interestingly, all virtualized systems sustain a nearly identical performance to their bare-metal counterpart for device-to-device copy: the earlier platform at 66,400 MB/s while, the more advanced platform at 194,800 MB/s. This indicates that, once the data is transferred from the virtual machine to the GPU, the commands that operate exclusively on the GPU are much less susceptible to the overhead incurred by the virtualization system.

Our results indicate that, even though the gaming performance issues have largely disappeared in the more modern advanced platform, the memory transfer from main memory across the PCI-E bus remains a severe bottleneck. It can become an issue during the gaming industry's transition from 1080p to the newer high memory requirements of 4k (4096 x 2160) UHD resolution.

4. Future Work and Conclusion

Cloud gaming has attracted significant interest from both academia and industry, and its potentials and challenges have yet to be fully realized, particularly with the latest hardware and virtualization technology advances. We closely examined the performance of modern virtualization systems equipped with virtualized GPU and pass-through techniques. Our results showed virtualization for GPU has greatly improved and is ready for gaming. Although there is degradation of memory transfer between a virtualized system's main memory and its assigned GPU, the game performance at the full HD resolution of 1080p was only marginally impacted.

References

[1]W. Cai, M. Chen, and V. Leung, "Toward gaming as a service," IEEE Internet Computing, vol. 18, no. 3, pp. 12–18, 2014.

[2]R. Shea and J. Liu, "Cloud gaming: Architecture and performance," IEEE Network, vol. 27, no. 4, pp. 16–21, 2013.

[3]M. Claypool and D. Finkel, "On the performance of onlive thin client games," Springer Multimedia Systems Journal, Special Issue on Network Systems Support for Games, vol. PP, no. 9, pp. 1–14, 2014.

[4]K. Lee, D. Chu, E. Cuervo, A. Wolman, and J. Flinn, "Demo: Delorean: using speculation to enable low-latency continuous interaction for mobile cloud gaming," in Proceedings of ACM MobiSys, 2014.

[5]C.-Y. Huang, C.-H. Hsu, Y.-C. Chang, and K.-T. Chen, "Gaminganywhere: An open cloud gaming system," in Proceedings of the 4th ACM Multimedia Systems Conference, ser. MMSys '13. New York, NY, USA: ACM, 2013, pp. 36–47.

[6]N. Amit, M. Ben-Yehuda, and B.-A. Yassour, "Iommu: Strategies for mitigating the iotlb bottleneck," in Computer Architecture. Springer, 2012, pp. 256–274.

[7]L. Shi, H. Chen, J. Sun, and K. Li, "vcuda: Gpu-

accelerated high-performance computing in virtual machines," Computers, IEEE Transactions on, vol. 61, no. 6, pp. 804–816, 2012.

[8]C. Reaño, A. J. Peña, F. Silla, J. Duato, R. Mayo, and E. S. Quintana-Orti, "Cu2rcu: Towards the complete rcuda remote gpu virtualization and sharing solution," in High Performance Computing (HiPC), 2012 19th International Conference on., 2012, pp. 1–10.

[9]C.-T. Yang, H.-Y. Wang, and Y.-T. Liu, "Using pci pass-through for gpu virtualization with cuda," in Network and Parallel Computing. Springer, 2012, pp. 445–452.

[10] R. Shea and J. Liu, "On gpu pass-through performance for cloud gaming: Experiments and analysis," in Network and Systems Support for Games (NetGames), 2013 12th Annual Workshop on., 2013, pp. 1–6.

Ryan Shea received Bachelor of Science degree in computer science from Simon Fraser University in 2010. He is currently a Ph.D. candidate in the Network Modelling lab at Simon Fraser University, where he also completed the Certificate in University Teaching and Learning. In 2013 Ryan received the Natural Sciences and Engineering Research Council of Canada (NSERC) Alexander Graham Bell Canada Graduate Scholarship. His research interests include computer and network virtualization, and performance issues in Cloud Computing. In 2012 Ryan received the best student paper award at IEEE/ACM 21st International Workshop on Quality of Service for his paper Understanding the Impact of Denial of Service Attacks on Virtual Machines.

Jiangchuan Liu is a Professor in the School of Computing Science, Simon Fraser University, British Columbia, Canada, and an NSERC E.W.R. Steacie Memorial Fellow. He is an EMC-Endowed Visiting Chair Professor of Tsinghua University, Beijing, China (2013-2016). From 2003 to 2004, he was an Assistant Professor at The Chinese University of Hong Kong.

He received the BEng degree (cum laude) from Tsinghua University, Beijing, China, in 1999, and the PhD degree from The Hong Kong University of Science and Technology in 2003, both in computer science. He is a corecipient of the inaugural Test of Time Paper Award of IEEE INFOCOM (2015; for the INFOCOM'05 paper on CoolStreaming, which has been cited 2000+ times); ACM TOMCCAP Nicolas D. Georganas Best Paper Award (2013), ACM Multimedia Best Paper Award (2012), IEEE Globecom Best Paper Award (2011), and IEEE Communications Society Best Paper Award on Multimedia Communications (2009). His students received the Best Student Paper Award of IEEE/ACM IWQoS twice (2008 and 2012).

His research interests include multimedia systems and networks, cloud computing, social networking, online gaming, big data computing, wireless sensor networks, and peer-to-peer and overlay networks. He has served on he editorial boards of IEEE Transactions on Big Data, IEEE Transactions on Multimedia, IEEE Communications Surveys and Tutorials, IEEE Access, IEEE Internet of Things Journal, Computer Communications, and Wiley Wireless Communications and Mobile Computing. He is the Steering Committee Chair of IEEE/ACM IWQoS from 2015 to 2017.

Position Paper

An Overview of Recent Research in Content-Centric Networking

Anand Seetharam[1] and Shiwen Mao[2]

[1]Computer Science Program, California State University Monterey Bay, Seaside, USA

[2]Department of Electrical and Computer Engineering, Auburn University, Auburn, USA

aseetharam@csumb.edu, smao@ieee.org

Abstract: In this paper, we provide an overview of the research papers published in the recently concluded content-centric networking workshop (CCN) held with IEEE MASS 2015. Our goal is to provide the reader a summary of the state-of-the-art research in the area of CCN, discuss some open problems and explore avenues of future research.

1. Introduction

In this paper, our goal is to provide the networking community a quick review of the state-of-the-art research in content-centric networking[2]. Specifically, we present a summary of the papers published in the recently concluded IEEE MASS 2015 Content-Centric Networking (CCN) Workshop and discuss some of the open challenges. We encourage the reader to refer the workshop proceedings [2] for detailed understanding of the papers. We next describe the aims and scope of the CCN workshop, quoted from [2] with minor modifications.

"With the exponential growth of content in recent years (e.g., videos) and the availability of the same content at multiple locations (e.g., same video being hosted at Youtube, Dailymotion), users are interested in obtaining a particular content and not concerned with the host housing the content. Also, the ever-increasing numbers of mobile devices that lack fixed addresses call for a more flexible network architecture that directly incorporates in-network caching, mobility and multipath routing, to ease congestion in core networks and deliver content efficiently. By treating content as first-class citizen, content-centric networking (CCN) aims to evolve the current Internet from a host-to-host communication based architecture to a content-oriented one where named objects are retrieved in a reliable, secure and efficient manner.

The CCN workshop provided researchers and practitioners an opportunity to meet and discuss the latest developments in this field. The outcomes of the workshop included 1) investigating and understanding some of the challenges in CCN, 2) fostering collaboration among researchers interested in CCN."

2. CCN Research Overview

The workshop facilitated some interesting discussion in CCN. Specifically, the participants explored and discussed issues related to routing and caching, fragmentation, security and multimedia streaming in CCN. We next present a summary of the published workshop papers.

Routing and Caching.

Unlike prior work that has mainly investigated routing and cache management techniques in CCN [5, 14, 15], papers presented at the CCN workshop explored deeper issues related to the routing overhead, energy consumption and distributed file sharing, that points to the increasing maturity of the field. *Hemmati et. al* compare the performance (i.e., overhead) of two name-based content routing protocols, namely the Named-data Link State Routing (NLSR) protocol and Distance-based Content Routing (DCR) protocol [9]. The performance metrics used for comparison purposes are the number of control messages sent, number of events processed, and number of operations performed by routers after the protocols are initialized. Their simulation results indicate that there is no clear winner; while NLSR incurs lower control overhead than DCR to react to name prefixes changes when the number of replicas is very small, DCR incurs less overhead than NLSR as the number of replicas increases.

In his keynote presentation [6] at the workshop, Prof. J.J. Garcia-Luna-Aceves also discussed the issue of overhead in CCN. He points out in his paper that *"the number of FIB entries stating name prefixes required for NDN and CCN to operate at Internet scale today is likely $O(10^8)$. By comparison, the size of routing tables maintained by high-end routers today is $O(10^6)$"* (quoted verbatim). He reexamines the mechanisms used in the forwarding plane in CCN architectures and proposes CCN-GRAM (Gathering of Routes for

[2] Other names for content-centric networking include information-centric networking or named-data networking. There are multiple research efforts currently underway involving academic institutions and industry aimed at designing and implementing CCN prototypes [1, 3].

Anonymous Messengers), an approach that reduces overhead by operating with a stateless forwarding plane. *Kurihara et. al* propose a strategy for grouping interest packets with similar information into a single one so as to decrease the computational overhead needed to look up large number of names of incoming interests in the FIB/PIT/CS entries in routers [10]. They demonstrate that their proposed scheme can reduce the computational overhead in routers to approximately 40% of a standard CCN implementation that works on individual packets.

The authors [16] analyze the energy consumption in CCN and demonstrate that in-network caching alone does not significantly reduce energy consumption. They then demonstrate that in-network caching compliments the energy-aware routing protocol proposed in [19] and enhances the energy reduction gains. In [4], the authors first showcase the advantages of the NDN architecture that supports the seamless integration of secure and distributed file sharing applications. They then present Chronoshare, a mobile-friendly distributed file-sharing applications that allow users to seamlessly share files regardless of device type, mobility (i.e., stationary, mobile) or connectivity patterns (i.e., constantly or intermittently connected).

Fragmentation.
Another important issue in CCN that received significant attention in the workshop is fragmentation. CCN disseminate data using hierarchical names for the different data chunks; if these data chunks exceed the maximum transmission unit (MTU) for Ethernet, then they need to be fragmented. In recent times, multiple hop-by-hop, end-to-end, and mid-to-end fragmentation schemes for CCN have been proposed. Instead of proposing a new fragmentation protocol, *Ueda et. al* analyze the performance of end-to-end fragmentation in terms of cache hit ratio and header overhead [17].

Secure fragmentation is investigated in [18], where the authors propose Named Network Fragments (NNF), an approach that improves upon the existing FIGOA protocol [8]. One of the main drawbacks of FIGOA is that signature verification is delayed until the last fragment is received as the signature is dependent on the hash computed based on the entire message (content object chunk). Additionally, the hash-based content retrieval in FIGOA is dependent on the message name; as content names in CCN may be potentially unbounded, this presents a significant challenge. The NNF protocol not only overcomes the above-mentioned challenges of FIGOA, it also allows users or routers to selectively request and retransmit fragments of a content object chunk.

Multimedia Streaming over Wireless Networks.
The workshop also included interesting papers related to multimedia streaming over wireless CCN. Via a small scale measurement study over a WiFi media streaming testbed comprising of five nodes [13], the authors show that *"bandwidth consumption between a content publisher and its forwarder (i.e., access point) over Wi-Fi can be effectively and dramatically reduced by NDN, offering much better scalability than IP"* (quoted verbatim). However, their experimental study also indicates that CPU utilization in NDN can be significantly higher in comparison to IP networks, indicating a need for further exploration. *Liu et. al* [12] combine caching and software-defined networking techniques in the design of CloudEdge, a computation-capable and programmable wireless access network architecture that maintains a good Quality of Experience (QoE) of multiple videos streams, while taking wireless transmission capacity and in-network computation power constraints into consideration.

4. Conclusion
The papers presented at the workshop provided preliminary solutions to important research problems related to routing overhead, security and improving quality of experience of video streams in CCN. The primary factors impeding real-world implementation and adoption of CCN architectures are related to the overhead incurred in maintaining the data structures needed in CCN. Future research should focus on analyzing and reducing the overhead incurred in CCN.

Security and privacy in CCN have received limited attention so far. Cybersecurity has been identified by NSF as one of the research areas requiring immediate attention [11]. Hence the success of CCN hinges on identifying attacks exclusive to CCN architectures and implementations and proposing effective countermeasures. In-network caching opens up the possibility of a plethora of new denial of service (DoS) and timing attacks that exploit cache characteristics to degrade performance and steal private information. An overview of possible DoS attacks in CCN is provided in [7], but in-depth investigation of these attacks is an important area of future research.

References
[1] CCNx project. https://www.ccnx.org/
[2] IEEE MASS CCN 2015 Workshop. http://www.eng.auburn.edu/~szm0001/ccn2015/
[3] Named-data networking project. http://named-data.net/
[4] A. Afanasyev, Z. Zhu, and L. Zhang. "The story of chronoshare, or how NDN brought distributed secure file sharing back." In IEEE MASS CCN Workshop, 2015.
[5] M. Badov, A. Seetharam, J. Kurose, V. Firoiu, and S. Nanda. "Congestion-aware caching and search in information-centric Networks." In ACM ICN, pages 37–

46, 2014.

[6] J. J. Garcia-Luna-Aceves. "New directions in content centric networking." (Invited Paper), In IEEE MASS CCN Workshop, 2015.

[7] P. Gasti, G. Tsudik, E. Uzun, and L. Zhang. "Dos & DDos in named data networking." In IEEE ICCCN, 2013.

[8] C. Ghali, A. Narayanan, D. Oran, G. Tsudik, and C. Wood. "List interest: Packing interests for reduction of router workload in ccn1.0." In arXiv preprint arXiv:1405.2861, 2014.

[9] E. Hemmati and J. J. Garcia-Luna-Aceves. "A comparison of namebased content routing protocols." In IEEE MASS CCN Workshop, 2015.

[10] J. Kurihara, K. Yokota, K. Ueda, and A. Tagami. "List interest: Packing interests for reduction of router workload in ccn 1.0." In IEEE MASS CCN Workshop, 2015.

[11] J. Kurose. "The expanding cyber threat." 2015. https://www.nsf.gov/about/congress/114/jk_cyber_150127.pdf

[12] H. Liu and K. Smith. "Improving the expected quality of experience in cloud-enabled wireless access networks." In IEEE MASS CCN Workshop, 2015.

[13] S. Mohammed and M. Xie. "A measurement study on media streaming over WiFi in named data networking." In IEEE MASS CCN Workshop, 2015.

[14] I. Psaras, W. K. Chai, and G. Pavlou. "Probabilistic in-network caching for information-centric networks." In ACM ICN, pages 55–60, 2012.

[15] L. Saino, I. Psaras, and G. Pavlou. "Hash-routing schemes for information centric networking." In ACM ICN, pages 27–32, Aug 2013.

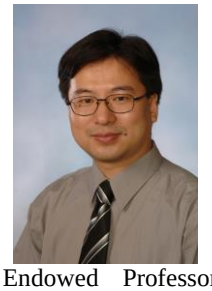
[16] J. Takemasa, Y. Koizumi, T. Hasegawa, and I. Psaras. "On energy reduction and green networking enhancement due to in-network caching." In IEEE MASS CCN Workshop, 2015.

[17] K. Ueda, K. Yokota, J. Kurihara, and A. Tagami. "Performance analysis of end-to-end fragmentation in content-centric networking." In IEEE MASS CCN Workshop, 2015.

[18] C. A. Wood and M. Mosko. "Secure fragmentation for content centric networking." In IEEE MASS CCN Workshop, 2015.

[19] M. Zhang, C. Yi, B. Liu, and B. Zhang. "GreenTE: power-aware traffic engineering". In IEEE ICNP, 2010.

Anand Seetharam received his Ph.D. from the University of Massachusetts Amherst. He is currently an assistant professor at California State University Monterey Bay. His research interests encompass various aspects of computer networking including information-centric networks, wireless networks, cellular networks and mobile computing. He has served as technical program committee member for a number of international networking conferences: ICCCN, IEEE WoWMoM and as reviewer for a number of international journals: IEEE Transactions on Mobile Computing, IEEE Transactions on Wireless Communications.



Shiwen Mao (S'99-M'04-SM'09) received Ph.D. in electrical and computer engineering from Polytechnic University, Brooklyn, NY in 2004. He was the McWane Associate Professor in the Department of Electrical and Computer Engineering from 2012 to 2015, and is the Samuel Ginn Endowed Professor and Director of the Wireless Engineering Research and Education Center (WEREC) since 2015 at Auburn University, Auburn, AL, USA. His research interests include wireless networks, multimedia communications, and smart grid. He is a Distinguished Lecturer of the IEEE Vehicular Technology Society in the Class of 2014, and Vice Chair—Letters and Member Communications of IEEE Communications Society Multimedia Communications Technical Committee. He is on the Editorial Board of IEEE Transactions on Multimedia, IEEE Internet of Things Journal, IEEE Communications Surveys and Tutorials, and IEEE Multimedia, among others, and was Associate Editor of IEEE Transactions on Wireless Communications from 2010 to 2015. He serves as Steering Committee Member for IEEE ICME and AdhocNets, Area TPC Chair of IEEE INFOCOM 2016 and Technical Program Vice Chair for Information Systems of IEEE INFOCOM 2015, symposium co-chairs for many conferences, including IEEE ICC, IEEE GLOBECOM, ICCCN, et al. He received the 2013 IEEE ComSoc MMTC Outstanding Leadership Award and the NSF CAREER Award in 2010. He is a co-recipient of the IEEE WCNC 2015 Best Paper Award, the IEEE ICC 2013 Best Paper Award, and the 2004 IEEE Communications Society Leonard G. Abraham Prize in the Field of Communications Systems.

Call for Papers

Quality of Experience-based Management for Future Internet Applications and Services (QoE-FI 2016)

collocated with

IEEE ICC 2016, May 23-27, Kuala Lumpur, Malaysia

Recent technological advances have enabled a constant proliferation of novel immersive and interactive services that pose ever-increasing demands to our communication ecosystem. Examples are: social TV, immersive environments, mobile gaming, UHD (4K/8K), 3D virtual worlds, just to cite a few. Furthermore, the ongoing migration of end-to-end multimedia communication ecosystem to the cloud requires improved dynamic resource provisioning and parallelization of media processing tasks that considers the end-user and application-related QoS /QoE requirements. Using multiple independent multimedia cloud services that may compete for the resource poses additional challenges to provide high quality-of-experience (QoE) for the aggregated service.

In this dynamic context, network and service providers are struggling to achieve higher levels of user satisfaction through new and better multimedia experiences. This will be also accelerated by adopting evolution on Future Internet and 5G Communications. Future Internet has been designed to overcome current limitations and to address emerging trends that impact on multiple aspects including: network architecture, content and service mobility, diffusion of heterogeneous nodes and devices, new forms of user centric/user generated content-aware provisioning and Communications (M2M, IoT).

To address these issues, the QoE-FI workshop aims at bringing together researchers from academia and industry to identify and discuss the following topics:

- QoE evaluation methodologies and metrics
- Frameworks and testbeds for QoE evaluation (crowd-sourcing, field testing, etc.)
- QoE studies & trials in the context of Smart Cities
- QoE models, their applications and use cases
- QoE-aware cross-layer design
- QoE-driven media processing and transmission over the cloud
- QoE for emerging applications (3D, OTT, Immersive, Gaming, Haptics)
- Datasets for QoE validation and benchmarking
- QoE control, monitoring and management strategies
- QoE in community-focused interactive systems
- KPI and KQI definition for QoE optimization in emerging environments (5G, IoT, Cloud)
- Integration of QoE in infrastructure and service quality monitoring solutions
- Media analytics from QoE Big Data
- QoE-based adaptive media services
- From Quality of Experience to Quality of Life

Submission:

Papers can be submitted using the following URL: http://edas.info/newPaper.php?c=21692&track=77342
Submitted papers must represent original material which is not currently under review in any other conference or journal and has not been previously published. Paper length should not exceed the six-page standard IEEE conference two-column format. Please see the author information page for submission guidelines on the ICC 2016 website http://icc2016.ieee-icc.org/cfw

Important Dates:

Paper submission deadline: **December 18, 2015**
Acceptance notification: **February 21, 2016**
Camera-ready papers: **March 13, 2016**

Workshop Co-Chairs:

Raimund Schatz, FTW, Austria
Tasos Dagiuklas, Hellenic Open University, Greece
Pedro Assuncao, Institute of Telecommunications/IPL, Portugal

23rd International Conference on Telecommunications (ICT 2016)

http://ict-2016.org

The 23rd International Conference on Telecommunications (ICT 2016) will be held in Thessaloniki, a modern metropolis bearing the marks of its stormy history and its cosmopolitan character, known for its hospitality and cuisine.

This year's theme "Expansion to Small" aims to draw research community's attention to the enormous anticipated expansion of communication systems through small architectures and devices. Small cells, short wavelengths, small sensors, small scale communications going down to molecular level are expected to boost next generation communications leading to Big Data networking, massive Internet of Things and green, energy efficient applications. Additionally, high quality papers on all aspects of contemporary research and applications of telecommunications are welcome.

Prospective authors are invited to submit high-quality original technical papers reporting original research of theoretical or applied nature. All submitted papers will be peer-reviewed. The manuscripts must be prepared in English with a maximum paper length of five (5) printed pages (following the standard IEEE 2-column format) without incurring additional page charges (maximum 2 additional pages with over length page charge of USD100 for each page if accepted). Please note that for every accepted contribution, at least one person must register for the conference and present. Accepted papers not presented in the conference will be excluded from the proceedings.

All papers for ICT 2016 should be submitted via EDAS using http://www.edas.info/newPaper.php?c=21703

The Organizing Committee invites proposals for Workshops/Special Sessions/Tutorials/Demos to be held in ICT 2016. More information can be found in http://ict-2016.org (under "Submissions").

Important Dates:

Paper submission: **January 31, 2016**
Paper Acceptance Notification: **March 1, 2016**
Camera-Ready Papers: **March 10, 2016**
Workshop/Special Session Proposals: **December 20, 2015**
Tutorial Proposals: **December 20, 2015**
Demo Proposals: **January 15, 2016**

Organizing Committee

General co-Chairs:

Mischa Dohler
King's College London, UK

George Karagiannidis
Aristotle University of Thessaloniki, Greece

Technical Program Chairs:

Periklis Chatzimisios, Alexander TEI of Thessaloniki, Greece
Athanasios C. Iossifides, Alexander TEI of Thessaloniki, Greece
Shiwen Mao, Auburn University, USA
Kan Zheng, Beijing University of Posts & Telecomm., China
Vasilis Friderikos, King's College London, UK

MMTC OFFICERS (Term 2014 — 2016)

CHAIR

Yonggang Wen
Nanyang Technological University
Singapore

STEERING COMMITTEE CHAIR

Luigi Atzori
University of Cagliari
Italy

VICE CHAIRS

Khaled El-Maleh (North America)
Qualcomm
USA

Liang Zhou (Asia)
Nanjing University of Posts &Telecommunications
China

Maria G. Martini (Europe)
Kingston University,
UK

Shiwen Mao (Letters & Member Communications)
Auburn University
USA

SECRETARY

Fen Hou
University of Macau, Macao
China

E-LETTER BOARD MEMBERS (Term 2014—2016)

Periklis Chatzimisios	Director	Alexander TEI of Thessaloniki	Greece
Guosen Yue	Co-Director	Broadcom	USA
Honggang Wang	Co-Director	UMass Dartmouth	USA
Tuncer Baykas	Editor	Medipol University	Turkey
Tasos Dagiuklas	Editor	Hellenic Open University	Greece
Chuan Heng Foh	Editor	University of Surrey	UK
Melike Erol-Kantarci	Editor	Clarkson University	USA
Adlen Ksentini	Editor	University of Rennes 1	France
Kejie Lu	Editor	University of Puerto Rico at Mayagüez	Puerto Rico
Muriel Medard	Editor	Massachusetts Institute of Technology	USA
Nathalie Mitton	Editor	Inria Lille-Nord Europe	France
Zhengang Pan	Editor	China Mobile	China
David Soldani	Editor	Huawei	Germany
Shaoen Wu	Editor	Ball State University	USA
Kan Zheng	Editor	Beijing University of Posts & Telecommunications	China