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

# *E-LETTER*

Vol. 9, No. 2, March 2014

IEEE COMMUNICATIONS SOCIETY

## CONTENTS

## Message from MMTC Chair

Dear Fellow MMTC Members,

Time flies, and it has been almost two years since I served as the MMTC Chair from June 2012. In this message, I would like to call for nominations of **MMTC officers for 2014-2016** and **MMTC Service Awards of 2014**.

During the past two years, MMTC has continued to flourish through the dedicated services of many volunteers. 13 MMTC Interest Groups have taken lead in organizing numerous special issues in top journals and magazines as well as conferences and workshops on hot multimedia topics. The TC has been keeping providing high quality monthly publications, with E-letters published in odd months and R-letter published in even months. Each issue of E-letter contains a special issue on emerging topics in multimedia as well as an industry column focusing on the recent industry innovations. Each issue of R-letter recommends top recent papers from the community, and includes a new distinguished category through collaborative recommendations with the Interest Groups. MMTC Service & Publicity Boards and Membership Boards work together in providing the best services to the community, through timely updates of the MMTC website and maintaining active MMTC Facebook and LinkedIn pages. MMTC Award Board is playing the key decisive role in selecting MMTC Best Paper Awards (by working with the Review Board) and MMTC Service Awards (by working with the MMTC officers).

What I am particularly proud is the formalization of various voting and selection procedures, regarding the nomination and election of the following important positions/awards:

- IEEE ICC/GLOBECOM Symposium Chairs
- IEEE ICME TPC Co-Chair
- IEEE ICME Steering Committee Member
- IEEE CCNC Steering Committee Member
- IEEE ComSoc Distinguished Lecturers
- Associate Editors for IEEE Transactions on Multimedia
- MMTC Best Paper Award
- MMTC Outstanding Leadership Award
- MMTC Distinguished Service Award
- MMTC Excellent Editor Award

For all the above positions and awards, MMTC has formulated a transparent procedure that usually involves nominations open to MMTC members through the email list, followed by a voting by a committee usually includes the current MMTC officers, and/or IG Chairs/Vice-Chairs as well as Board Directors/Co-Directors. The procedure encourages the maximum participation from the community, and the decisions are made through the collected wisdom of the most active volunteers who are contributing to the success of MMTC on a daily basis.

Moving forward, it is now the time to elect the next edition of the TC leaders. An ad-hoc election committee has been formed, which is chaired by the past TC Chair, Dr. Haohong Wang (haohong@ieee.org) and includes a few past TC Chairs and senior officers. If you are interested in taking a leading role in shaping and serving the MMTC community in the next two years, please send your nomination (including self nomination) to Dr. Wang by April 15, for one of the following positions.

- MMTC Chair
- MMTC Steering Committee Chair
- MMTC Vice Chair - North America
- MMTC Vice Chair - South America
- MMTC Vice Chair - Asia
- MMTC Vice Chair - Europe
- MMTC Vice Chair - Letters and Member Communications
- MMTC Secretary

- MMTC Standard Liaison

The election on the short-listed candidates will be conducted in ICC 2014, Sydney, Australia.

Next, I would like to call for nominations of the MMTC Service Awards 2014, according to the procedures outlined at http://committees.comsoc.org/mmc/awards.asp.

MMTC has two types of service awards that are approved by ComSoc. **The MMTC Distinguished Service Award** is given to an individual who has made significant contributions to MMTC as a core leader over a substantial number of years. **The MMTC Outstanding Leadership Award** is given to an individual who has made significant contributions to MMTC as a current or past IG Chair/Co-Chair or Board Director/Co-Director. Current MMTC officers are not eligible for the awards.

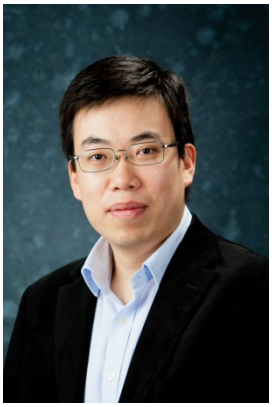
This year, we called for nominations of the service awards through the MMTC email list on February 17, with the deadline of March 22. A nominator should send the nomination to me and copy to MMTC Secretary Dr. Liang Zhou (liang.zhou[at]ieee.org) in a single PDF file in the following format:

- Award name
- Nominator name & affiliation, and contact info
- Nominee name & affiliation, and contact info
- A nomination letter no more than one page

The MMTC officers will discuss the nominations and provide the short-listed candidates to the Award Board for election. The results are again expected to be available by May, and the plaques will be awarded in ICC 2014.

I want to thank all the researchers and volunteers for making MMTC a great community. In particular, I would like to thank all current MMTC officers. It has been a great pleasure working with you and serving the community together.

Regards,



Jianwei Huang

Chair, IEEE ComSoc Multimedia Communication Technical Committee

http://jianwei.ie.cuhk.edu.hk/

# SPECIAL ISSUE ON CODING ALGORITHMS AND TECHNIQUES FOR MULTIMEDIA NETWORKING

## Coding Algorithms and Techniques for Multimedia Networking

*Guest Editor: Zongpeng Li, Chuan Wu*

*zongpeng@ucalgary.ca, cwu@cs.hku.hk*

Multimedia computing and networking is an ever-young research direction that has witnessed sustained interest from both academia and industry in the past decade. Multimedia applications naturally benefit from a number of coding techniques, including source coding, network coding, media trans-coding, to name a few. Source coding such as layered cods and MDS codes conducts encoding at the media source and decoding at the end clients, preparing the media content in a format for in-network transmission that is most appropriate according to the network status. Network coding are more general coding techniques that can be applied to different data transmission tasks in wireline and wireless settings, and have enjoyed particular success in multimedia networking. Media transcoding translates media streams between different playback rates, in response to network conditions and customer demand.

This special issue of E-Letter focuses on the recent progresses of multimedia networking, with a particular accent on networking algorithms and systems that are related to one of the many coding schemes relevant to multimedia.

In the first article titled, "*Optimizing Media Quality in DASH (Dynamic Adaptive Streaming over HTTP)",* Sanjeev Mehrotra from Microsoft Research review solutions for optimal client and/or network adaptation in response to varying network bandwidth constraints, for Internet media streaming applications. A utility maximization model is formulated, whose solution is discussed. The solution framework is known as DASH --- Dynamic Adaptive Streaming over HTTP.

The second article, "*Peer-to-Peer 3D/Multiview Video Distribution with View Synthesis Cooperation",* is from Fei Chen, Yan Ding and Jiangchuan Liu at Simon Fraser University, as well as Yong Cui at Tsinghua University. As evident in its name, this work is on the timely topic of 3D video delivery in a network. New challenges and opportunities in the design and implementation of 3D video streaming systems are discussed.

The third article is contributed by Ralf Globisch, Varun Singh, Juergen Sienel, Peter Amon, Mikko Uitto, and Thomas Schierl, from Germany and Finland. The title is *"Consideration for Distributed Media Processing in The Cloud"*. The authors discuss the challenges in building distributed media processing platforms in the cloud --- a desired direction to go since the cloud provides a rich pool of computing resources suitable for multimedia processing exemplified by transcoding. It was shown that open standards such as the IETF Internet standards can be used to build a distributed media processing pipeline consisting of multiple cloud media processing services.

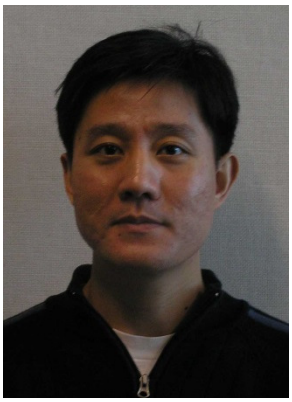
Yaochen Hu, Di Niu from the Univiersity of Alberta and Zongpeng Li from the University of Calgary presented a new work on video multicast optimization using a new mathematical model, in the fourth article, *"Internet Video Multicast via Constrained Space Information Flow"*. Space information flow is a relatively new mathematical model for optimization problems in network design and data networking. In this work, it is applied to Internet video multicasting. A salient feature is that network optimization is formulated and conducted in the Internet delay space instead of in a finite network/graph. It is hoped that the space information flow model may prove useful in future work on information networking and communications, including ones that are related to multicast and network coding.

The fifth article is from Quan Zhou and Liang Zhou at Nanjing University of Posts and Telecommunications, with the title *"Compressive Sensing for Video Coding: A Brief Overview"*. It's nice to see an article that relates compressive sensing, a rather extensively studied theoretical tool for dimension reduction and image processing, connected to multimedia coding. Background on compressive sensing as well as existing and possible future applications are outlined in this succinct article.

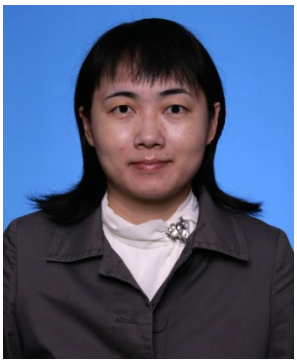
The final article, *"Offloading Policy of Video Transcoding for Green Mobile Cloud"*, is contributed by Weiwen Zhang and Yonggang Wen from Nanyang Technological University. We are glad that one of the articles in this issue addresses green computing, in a time where a growing portion of the general public become concerned in environment and green life styles.

Specifically, this work proposes an energy-efficient offloading policy for TaaS to minimize the energy consumption of transcoding on both mobile devices at the end user and in the cloud, while striving to maintain low delay latency.

This special issue is, of course, not intended to serve as a comprehensive survey in the field of coding for multimedia networking. Nonetheless, we hope that each of the articles presented is of interest to a group of audience in the multimedia community, including members of the MMTC. Finally, we would like to thank all the authors for their great contribution and the E-Letter Board for making this special issue possible.



**Zongpeng Li** received his B.E. degree in Computer Science and Technology from Tsinghua University (Beijing) in 1999, his M.S. degree in Computer Science from University of Toronto in 2001, and his Ph.D. degree in Electrical and Computer Engineering from University of Toronto in 2005. Since 2005, he has been with the Department of Computer Science in the University of Calgary. In 2011-2012, Zongpeng was a visitor at the Institute of Network Coding, Chinese University of Hong Kong. His research interests are in networking science and network coding. Zongpeng is a senior member of IEEE.



**Chuan Wu** received her B.Engr. and M.Engr. degrees in 2000 and 2002 from the Department of Computer Science and Technology, Tsinghua University, China, and her Ph.D. degree in 2008 from the Department of Electrical and Computer Engineering, University of Toronto, Canada. Between 2002 and 2004, She worked in the Information Technology industry in Singapore. Since September 2008, Chuan Wu has been an Assistant Professor in the Department of Computer Science at the University of Hong Kong. Her research is in the areas of cloud computing, online and mobile social networks, peer-to-peer networks and wireless networks. She is a member of IEEE and ACM, and the Chair of the Interest Group on Multimedia services and applications over Emerging Networks (MEN) of the IEEE Multimedia Communication Technical Committee (MMTC) from 2012 to 2014.

# Optimizing Media Quality in DASH (Dynamic Adaptive Streaming over HTTP)

*Sanjeev Mehrotra*
*Microsoft Research, Redmond, WA, USA*
*sanjeevm@microsoft.com*

## 1. Introduction

Media streaming over HTTP is seeing rapid adoption and is becoming the prevalent method for media streaming as it allows for the use of standard web servers and caching architecture (e.g. Content Delivery Networks (CDN)) which is already ubiquitously present for the delivery of other (non-media) web content.

In addition, although media streaming has deadline constraints, those constraints are typically at least an order of magnitude larger than network latencies (which are on the order of hundreds of milliseconds) since the client buffers several seconds of content. Therefore the underlying use of TCP inside HTTP does not affect performance as networking latency is of little concern to such applications.

However, since the internet does not typically have any Quality of Service (QoS) guarantees, the network bandwidth can vary dramatically. Therefore, *Dynamic Adaptive Streaming over HTTP* (DASH) has been developed to allow for client-side adaptation in response to network variation as well as other client considerations (e.g. CPU, memory, battery, device capability) [1],[2],[3].

In this paper, we review existing solutions for optimal client and/or network adaptation in response to varying bandwidth constraints by formulating it as a utility maximization (or equivalently distortion minimization) subject to rate constraints imposed by the network. The optimization can occur over several users sharing the same resource (as in the case of LTE or mobile) or temporally across various segments for a single user.

## 2. DASH Architecture Overview

In DASH, the serving architecture (datacenter, CDN, or edge servers inside the ISP) host the content as files which the client requests using HTTP GET calls. The data files hosted in the serving infrastructure includes the encoded media which is broken into independently decodable segments, with each portion coded at various bitrates and perhaps even using different codecs. In addition to the files containing the actual media encodings, a manifest or *Media Presentation Description* (MPD) is present which is an XML based file containing information regarding the representations or encodings present for each segment along with information regarding the segments, such as bitrate, file size, starting time, duration, codecs used, etc.. An MPD can be dynamically generated, e.g. for live content, and also can be progressively sent to the client so as to avoid a large start-up time caused by downloading the MPD. In addition, quality information (or distortion) can be included in the MPD for each segment encoding. Figure 1 shows the basic architecture for the content present within the serving architecture.

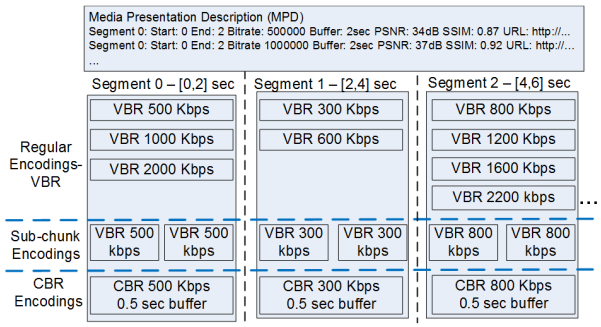



**Figure 1**: Content layout in DASH. The MPD file contains information regarding each encoding for segment. Each encoding shown in a block is a different representation for the content.

## 3. Optimization of Dynamic Adaptive Streaming Across Multiple Users

In [4], the authors formulate the problem of optimizing adaptive streaming of media over a shared resource such as a mobile cellular network. The network may contain media streams as well as data streams. The problem is formulated as that of maximizing the aggregate utility of all streams as

$$\max \sum_{k=1}^{K} \beta_k \left( \frac{\left(\frac{R_{v,k}}{\theta_{v,k}}\right)^{1-\alpha} - 1}{1-\alpha_k} \right) + \sum_{l=1}^{L} \ln\left(\frac{R_{d,l}}{\theta_{d,l}}\right)$$

subject to

$$A_{v,k} \le R_{v,k} \le B_{v,k}$$

$$\sum_{k=1}^{K} \omega_{v,k} R_{v,k} + \sum_{l=1}^{L} \omega_{d,l} R_{d,l} \le \Pi_{tot},$$

where $\alpha$ and $\beta$ are parameters used in the weighted $\alpha$-proportional fairness utility function, $\theta$ is a parameter depending on content complexity, $R_{v,k}$ is the rate of the $k$th video source, $R_{d,l}$ is the rate of the $l$th data source, and $A_{v,k}$ and $B_{v,k}$ are the minimal and maximal

allowable rates of the $k$th video source. $\Pi_{tot}$ is the total resource constraint and for mobile cellular networks the resource consumed is a function of both the rate as well as the Signal-to-Interference Plus Noise Ratio (SINR), given by $\omega R$, where $R$ is the rate and $\omega = [\log_2(1 + SINR)]^{-1}$. For other networks, we could simply use $R$ for the resource consumed, that is, $\omega = 1$.

Using standard techniques by turning the constrained problem into an unconstrained one using Lagrange multipliers, the authors in [4] derive the optimal rate allocation which shows that $R$ is directly proportional to $\theta$ and inversely proportional to $\omega$. This makes sense as a higher $\theta$ (more complex) content is one which has lower utility at the same rate and thus requires more rate in order to operate at the same slope point in the rate-utility curve. Similarly a lower SINR (and thus higher $\omega$) implies a receiver which consumes more resources when operating at the same rate and thus should receive less rate. It is then shown that the rate allocation can be accomplished by traditional proportional fair schedulers with guaranteed bit rate (GBR) constraints already employed at base stations by simply picking the GBR values appropriately for the various flows.

**4. Optimization of Dynamic Adaptive Streaming Across Multiple Segments of Single User**

In many cases, the constrained resource (for example the bottleneck link in the network) is the last hop from the server to the client. For example, most ISPs, such as cable modem and DSL providers throttle the rate available to the user depending on the service level. In such cases, cross-user optimization is not a possibility.

However since the client buffer is usually on the order of several seconds, the client can make optimizations temporally across segments of the content [5]. In [5], we show that each client attempts to minimize distortion or correspondingly maximize the utility across the next $N$ segments, that is,

$$\max \sum_{k=t}^{t+N-1} U_k(R_k)$$

subject to multiple ($N$) rate constraints

$$\sum_{k=t}^{n} R_k \leq C_n \ \forall n = t, \dots, t+N-1$$

where $U_k$ is the concave utility function for the $k$th segment as a function of rate, $R_k$. The multiple rate constraints, $C_n$, are derived from the following five parameters: i) estimated throughput, ii) the current time, iii) the desired playback deadline of segment $k$, iv) the current client buffer size, and v) the desired buffer size after the download of current segment $t$. The playback deadline for each segment is chosen to make sure the client achieves glitch-free playback as well as the desired startup latency. The estimated throughput can either be estimated or computed from the optimization in Section 3 for the multi-user case.

In order to optimize media quality, each segment is typically encoded using variable bit-rate (VBR) encoding, that is the entire file or at least most of the file has to be downloaded in order to attempt playback of the segment. However, in order to further reduce startup latency, smaller sub-segment or constant bit-rate (CBR) encodings with a smaller buffer can be available.

In the optimization presented in [6], the client buffer is assumed to have no constraints, which is typically okay as most clients have sufficient memory or can cache future segments to disk. However, additional constraints can be used to prevent decoder buffer-overflow (or encoder underflow) as $\sum_{k=t}^{n} R_k \leq D_n$, where $D_n$ is a function of the current client buffer size and the maximum client buffer size.

Although the optimization here is a classic convex minimization (concave maximization) optimization problem with multiple linear constraints, it can be readily be solved in $O(n)$ complexity as the constraints are nested [5], [6]. The basic idea is to optimize first for the final rate constraint using the method of Lagrange multipliers. If any earlier constraints are violated, the problem is split into two optimization problems with constraints applied to each segment.

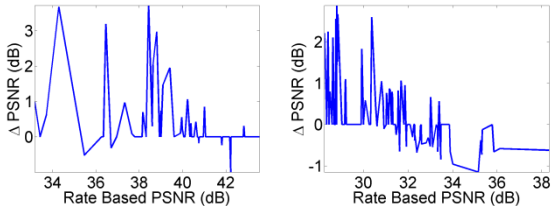
The optimization can also be applied for live encodings by simply limiting the horizon of search to the segments that are already available, which is typically on the order of tens of seconds.

By using this simple optimization, the client can achieve a 3dB to 5dB gain in PSNR for segments which are difficult to encode at the expense of minimal loss for segments which are easy to encode. In Figure 2, we show the gains by plotting the improvement in PSNR for segments as a function of the PSNR the segment would obtain when using a simple non-optimized request strategy. The segments with low PSNR see the most gain, thus making the media more constant quality as opposed to constant bitrate.

**5. Challenges and Future Directions**

Although optimizing for utility and quality across a multi-user shared resource (such as a mobile cellular network with multiple users connected to the same base station) is a noble goal, the practicality of it somewhat difficult as centralized scheduling is needed.

Distributed rate control and scheduling is also possible using distributed utility maximization algorithms, whereby each flow responds to congestion (once the resource is fully utilized) using its own utility scaling [7]. However, even then, the debate about network neutrality [8], [9] makes it difficult since it is difficult to calibrate how one user's utility relates to another. It is much easier to calibrate utility from one flow to another flow of the same user.




(a) (b)

Figure 2. PSNR improvements for segments as function of un-optimized segment PSNR for (a) TV drama clip and (b) Sports clip

By optimizing across various flows for the same user and across time in a given flow, we can avoid the need for iterative distributed scheduling or centralized scheduling.

In addition, one of the key aspects in the optimization across a single user is to estimate the link throughput. Although the application can estimate the throughput using simple metrics (e.g. file size and time to download), lower layers of the networking stack may have additional information -- e.g. SINR, number of users on the network, location, past throughput at the location -- which can further enhance the rate estimation. Alternatively, rate prediction can also be used to predict future throughput to use in the optimization [10].

## 6. Conclusion

In this paper we have shown methods in which adaptive media streaming (such as DASH) can be optimized for media quality. This optimization can be done across multiple users, multiple flows for a given user, as well as multiple segments for a given stream. By taking advantage of multiplexing across users, flows, and segments, and taking advantage of the client buffer, we can significantly enhance quality.

## References

[1] ISO/IEC 23009-1, ISO/IEC 23009-2, ISO/IEC 23009-3, ISO/IEC 23009-4, "Information technology – Dynamic adaptive streaming over HTTP (DASH)" – Parts 1 – 4.

[2] I. Sodagar, "The MPEG-DASH Standard for Multimedia Streaming Over the Internet", IEEE Multimedia, pp. 62-67, Oct.-Dec. 2011.

[3] T. Stockhammer, "Dynamic adaptive streaming over HTTP --: standards and design principles," in Proceedings of ACM MMSys, 2011.

[4] D. De Vleeschauwer, H. Viswanathan, A. Beck, S. Benno, G. Li, and R. Miller, "Optimization of HTTP adaptive streaming over mobile cellular networks," in Proceedings of INFOCOM 2013, pp.989-997, April 2013.

[5] S. Mehrotra and W. Zhao, "Rate-distortion optimized client side rate control for adaptive media streaming," in Proc. of IEEE International Workshop on Multimedia Signal Processing (MMSP), October 2009.

[6] A. Ortega, "Optimal bit allocation under multiple rate constraints," in Proceedings of Data Compression Conference (DCC), pp.349-358, Mar/Apr 1996.

[7] F. P. Kelly, A. K. Maulloo, and D. K. H. Tan, "Rate control for communication networks: Shadow prices, proportional fairness and stability," in Journal Operations Research Society, vol. 49, no. 3, pp. 237–252, Mar. 1998.

[8] Net neutrality: http://en.wikipedia.org/wiki/Network_neutrality.

[9] J. Crowcroft, "Net neutrality: the technical side of the debate: a white paper" in SIGCOMM Computer Communications Rev. 37, 1, pp.49-56, January 2007.

[10] Q. Xu, S. Mehrotra, Z. M. Mao, and J. Li, "PROTEUS: Network Performance Forecast for Real-Time, Interactive Mobile Applications," in Proceedings of ACM MobiSys, June 2013.

**Sanjeev Mehrotra** is a Principal Architect at Microsoft Research in Redmond, WA, USA. He received his Ph.D. in electrical engineering from Stanford University in 2000. He was previously the development manager for the audio codecs and DSP team in the core media processing technology team and led the development of WMA audio codec. He is the inventor of several audio and video codec technologies in use today and helped develop the initial prototype version of Microsoft's Smooth Streaming. His work has shipped in numerous Microsoft products and standard codecs such as H.264 AVC. His research interests include multimedia communications and networking, multimedia compression, and large scale distributed systems.

Dr. Mehrotra is an author on over 30 refereed journal and conference publications and is an author on over 70 US patent applications, out of which 35 have been issued. He was general co-chair and TPC chair for the Hot Topics in Mobile Multimedia (HotMM) workshop at ICME 2012 and is currently an associate editor for the Journal of Visual Communication and Image Representation. He is a senior member of IEEE and a recipient of the prestigious Microsoft Gold Star Award.

# Peer-to-Peer 3D/Multiview Video Distribution with View Synthesis Cooperation

*Fei Chen [1], Yan Ding [1], Jiangchuan Liu[1], and Yong Cui [2]*
*[1]School of Computing Science, Simon Fraser University, Canada*
*[2]Department of Computer Science and Technology, Tsinghua University, Beijing, China*
*{feic, yda15, jcliu}@sfu.ca, cuiyong@tsinghua.edu.cn*

## 1. Introduction

The recent advances in stereoscopic video capture, compression, and display have made 3-Dimensional (3D) video a visually appealing and costly affordable technology. We have witnessed a series of new releases of stereoscopic 3D products in the past years (e.g., Nintendo 3DS, Fuji W3 3D, and HTC Evo 3D), with much more being just announced. The abundant content, together with the dramatically decreasing price, have become driving forces to the vast growth of 3D-capable disc players and LCD/LED TVs, or even tablets and smartphones in the market, which are quickly sweeping away the conventional 2D only devices.

3D video (also referred as stereo video) is perceived by human beings with additional three-dimensional depth, which is resulting from the spatial disparity of two slightly different streams for left and right eyes' viewpoints respectively. Multiview video further extends 3D perception by capturing diverse viewpoints from a camera array, and enabling a user to interactively choose the viewpoints of interest [1]. There have been a series of pioneer academic works on streaming 3D video over the Internet [2] [3] [4]. Most of them, like those commercial products, are client/server based however. This classical communication paradigm has already suffered from streaming the traffic-intensive 2D videos, and the remarkably increased data volume of 3D videos poses even greater challenges, not to mention multi-view videos. The inherent multi-stream nature of 3D video unfortunately makes the streaming delivery more difficult than that of 2D video [5]. This is because, to enable stereoscopic perception at the user side, not only segments in one stream have to arrive before playback deadlines, but also they have to be paired with corresponding segments with the same playback time in the other stream. Poor synchronization between streams prolongs the playback delay, and would even cause spatial displacement of objects in the two views, resulting in false parallax perception. The problem is severe in a peer-to-peer overlay, given the existence of multiple senders and node churns. For multiview video, since a user is not only interested in a subset of the views but can switch over views, the challenge becomes even acuter with such view diversity and dynamics.

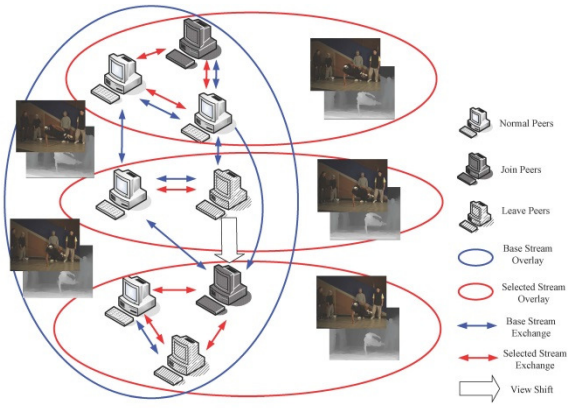



Figure 1. Peer-to-peer overlays for different views.

In this article, we present an initial attempt toward efficient streaming of 3D/multiview videos over peer-to-peer networks. The remainder of the paper is organized as follows. We first outline the structure of peer-to-peer 3D/multiview streaming in Section 2. Then we will present the inherent partner selection strategy in both the time and space dimensions in Section 3. In Section 4, we illustrate the implement issues with view synthesis technology, and the simulation result is presented in Section 5. Finally, we conclude this paper in Section 6.

## 2. Peer-to-peer 3D streaming: system overview

We advocate a mesh-based peer-to-peer overlay design, which has been widely used in state-of-the-art commercial systems [5] [6]. In a mesh overlay, each newly joined client (peer) will be informed with a list of active peers interested in the current video, and it will randomly select a subset of them to establish partnerships. It will then exchange bandwidth and data availability information with these partners and fetch video data through a scheduler, which specifies which partner to transmit which segments. The active peer list will be periodically gossiped through the overlay, so that existing peers can update their partners to accommodate peer and network dynamics.

Fig.1 illustrates the the peer-to-peer overlays with separate streams for different views, where the peers in the same overlay share the data segments from the same viewpoint. Note that a peer from a adjacent overlay may also be able to access the data in a given

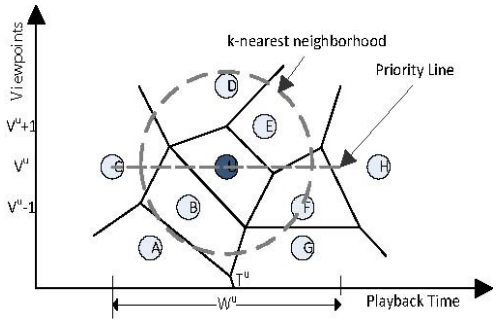



Figure 2. V-Plane Overlay: U has direct partners C in its *priority line*, and indirect partners B,D,E.

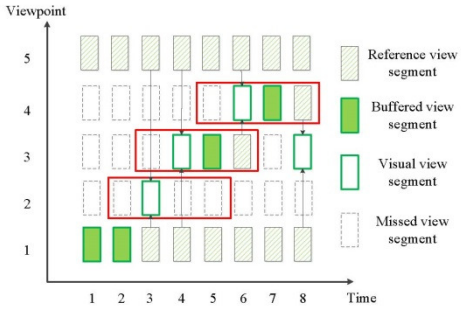



Figure 3. Dynamic view synthesis cooperation with viewpoint changes.

overlay, which facilitates potential interactive view switching in the multiview scenario. Each peer maintains a buffer for the received video segments, and a window slides over the buffer, covering the segments of interest for playback and transmission. The availability of the segments within the sliding window is represented by a peer's buffer bitmap, which is periodically updated and exchanged among partners.

**3. Voronoi diagram in time and space dimentions**

Differently from traditional monoscopic 2D video streaming, two distinct data sequences for a 3D video need be distributed, which are correspondingly produced by the encoded left and right views. Data segments with the same playback time, upon receiving, will be combined to produce a *stereo frame pair*, enabling depth perception. To ensure that the segments of different views arrive simultaneously at a destination, a simple approach would be mixing the segments into one stream for transmission. It is however quite inflexible, and we instead suggest that the segments being delivered through two separate streams. Further, in the multi-view scenario, not only a user might be interested in different part of a video, but also might prefer to watch the video from different views as well. This constrains resource sharing among peers. Besides the view diversity, the users may change viewpoints during watching.

We address these challenges by leveraging clustering to efficiently locate partners with same/overlapped interest. As in Fig.2, we use a Voronoi diagram [7] as a virtual plane (called V-Plane) that extends in time and space dimensions. The space dimension consists of discrete viewpoints, each representing a 3D perception of two adjacent views. The time dimension then corresponds to playback time slots. Let a peer's buffer status be represented by its value of center of interest (*COI*), expressed as *COI* (*time, viewpoint*). The time is set as the middle of the sliding window and the viewpoint represents which two views constitute the stereoscope perception. After the *COI* values are distributed, the peer will find interested partners, and establish partnerships for data exchange. We can see that, although the peers have their individual *COI*, they can still have overlapped content of interest. This is because the peers are usually interested in video content in a period of time, which locates near its *COI* (in the time scale). Further, adjacent peers will have one same view shared (in the space scale).

In the V-Plane, the peers having overlapped interest are partners. Each peer maintains two partners lists, namely, direct partners (DP) and indirect partners (uDP). The DP list consists of partners that are watching at the same viewpoint and are within the interest time period (in peer's *priority line*, Fig.2), while the uDP list has partners locating at farther range (in peer's *k*-nearest neighborhood). Although the indirect partners may not exchange the current video content, they could be possibly promoted to be direct partners, especially when the peer changes viewpoint or when DP has insufficient bandwidth to transmit the segment data of adjacent view stream. The priority of segment transmission can be formulated as the scheduling problem, which is solved by our proposed algorithms of the former work in [8].

**4. Implementation issue with view synthesis**

Even though the V-Plane overlay reduces the view switching delay by quickly re-locating new partners to avoid data outage with low overhead, it always takes time to establish the data connections with new partners. The smoothness of playback continuity suffers from sudden view shift behaviors with wide range, or high frequency. To address this issue, we deploy the view synthesis technology to assist the smooth playback during the view shift applications. The view synthesis refers to utilize the left and the right views with the depth and texture maps to generate the visual views, rather than the real views captured by the camera. Therefore, even though the users do not receive the segment data of all the viewpoints, the view synthesis can still be proceeded to render the visual views with

absent segments. In recent years, it has been extensively explored in the multiview streaming system [2] [9]. Specifically, we propose an online view synthesis algorithm with low computation cost in [1]. Here, we will illustrate how it can be integrated in the peer-to-peer 3D/multiview streaming system.

In Fig. 3, there are five different viewpoints, in which the *view1* and *view5* are the reference views. These two views are shared commonly among the clients for view synthesis, in case the segments for playback are not received. Let the selected views be *View ={1, 1, 2, 3, 3, 4, 4, 3}* for the time series *{1, ..., 8}*. In time slots 1, 2, 5, and 7, the view segments are buffered for the playback of these viewpoint streams. Otherwise, the view synthesis is proceeded to generate the visual views. Specifically, in the time slots 3 and 4, the common reference views *view1* and *view5* are utilized for visual view generation. In the time slots 6 and 8, the buffered view segments (without being viewed) are performed as the reference views for view synthesis. According to the result of our proposed algorithm, the adjacent reference view can provide a more accurate rendering performance.

**5. Performance evaluation**

We evaluate the performance of our 3D/multiview video streaming through simulations. The video source we used for simulation is based on the standard multiview video sequence "Ballet". The resolution is 1024*768 with a frame rate of 15 frames/s. For comparison, we also implement *Rarest First (RF)* scheduling and *Random (Random)* scheduling, which have been widely used in existing P2P streaming.

In Fig. 4 the viewpoint is changed at 30th second and 40 second, respectively. The PSNR is 40.2dB on average after compression. The view synthesis performance deteriorates as the range increases, 25.9dB for 1 visual view, 24.7dB for 3 visual views and 20.3dB for 5 visual views on average. In Fig. 5, we can observe an obviously rendering quality gain when the view synthesis strategy is deployed during the viewpoint changes. If the viewpoint keeps constant, the *RF* strategy has a little bit better performance as it does not need to transmit the reference views as well.

**6. Conclusion and future disscussion**

As 3D video remains in its early stage, many of the existing client video playback platforms support monoscopic 2D video only, even though a client may have sufficient bandwidth to receiving 3D streams. On the other hand, some 3D-capable client may want to disable it and instead have smoother 2D playback with less bandwidth demand. We thus believe that it is necessary to accommodate these clients in our 3D

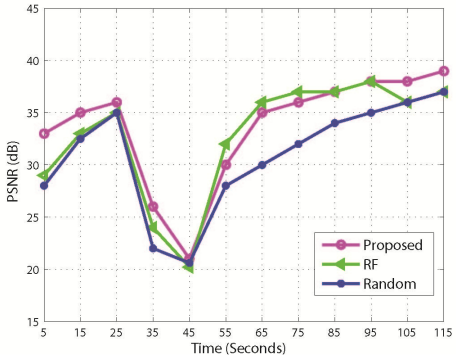



Figure 4. Rendering quality with viewpoint changes.

streaming system, so as to enable a smooth transition toward full 3D streaming. Our solution serves as an initial attempt towards supporting efficient separate view streaming and switching. Given the high degree of view diversity and dynamics, more advanced multi-view design in terms of overlay construction and partnership maintenance are to be developed.

**References**

[1] F. Chen, J. Liu, Y. Zhao, and E. C.-H. Ngai, "Collaborative view synthesis for interactive multi-view video streaming ," in Proc. of ACM NOSSDAV 2012.

[2] G. Park, J. Lee, G. Lee, and K. Kim, "Efficient 3D adaptive HTTP streaming scheme over internet TV," in Proc. of IEEE BMSB 2012.

[3] Z. Zhou, L. Zhuo, J. Zhang, and X. Li, "A user-driven interactive 3D video streaming transmission system with low network bandwidth requirements," in Proc. of ICACT 2012.

[4] S. Savas, C. G. Gurler, A. M. Tekalp, E. Ekmekcioglu, S. Worrall, and A. Kondoz, "Adaptive streaming of multi-view video over P2P networks," Signal Processing: Image Communication, 27(5): 522–531, 2012.

[5] J. Liu, S. G. Rao, B. Li, and H. Zhang, "Opportunities and challenges of Peer-to-Peer Internet video broadcast," Proceedings of the IEEE, 96(1): 11–24, 2008.

[6] PPLive http://www.pplive.com/.

[7] D. Lee, "On k-nearest neighbor Voronoi diagrams in the plane," IEEE Transactions on Computers, 100(6): 478–487, 1982.

[8] Y. Ding and J. Liu, "Efficient stereo segment scheduling in Peer-to-Peer 3D/multi-view video streaming," in Proc. of IEEE P2P 2011.

[9] N.-M. Chueng, A. Ortega, and G. Cheung, "Rate-distortion based reconstruction optimization in distributed source coding for interactive multiview video streaming," in Proc. of IEEE ICIP 2010.

# Considerations for distributed media processing in the cloud

Ralf Globisch[1], Varun Singh[2], Juergen Sienel[3], Peter Amon[4], Mikko Uitto[5], and Thomas Schierl[6]
[1] Technische Universität Berlin (TUB), Berlin, Germany
[2] Aalto University, Espoo, Finland
[3] Alcatel-Lucent, Stuttgart, Germany
[4] Siemens AG, Munich, Germany
[5] VTT, Espoo, Finland
[6] Fraunhofer Heinrich Hertz Institute, Berlin, Germany
rglobisch@mailbox.tu-berlin.de, varun.singh@aalto.fi, juergen.sienel@alcatel-lucent.com, p.amon@siemens.com, Mikko.Uitto@vtt.fi, thomas.schierl@hhi.fraunhofer.de

## Introduction

The cloud infrastructure provides additional network and computational resources to devices which have limited processing or bandwidth. Consequently, the applications offload computationally-intensive tasks to the cloud infrastructure. Currently, cloud-based systems provide multimedia processing services such as such as transcoding [1], [2], speech recognition [3], and deliver live media stream to millions of viewers [4], [5].

Video traffic is expected to account for a significant part of Internet traffic over the next few years. The demand for more capacity is only going to increase with the forthcoming deployments of Web-based Real-Time Communication (WebRTC) [6] and telepresence. In the future Internet it is foreseen that an increasing number of users will not only consume media content, but will also help produce it. For example, a live event is broadcast online using live footage recorded by multiple spectators. The trend towards increased media production by ordinary users is already evident today in the content that users upload onto YouTube [7], and in Google+'s [8] Auto Awesome feature and Hangouts. This is driven by social media trends as well as by mobile phone trends, as illustrated in Figure 2. The sales of mobile devices have already overtaken computer and laptop sales, and these devices typically have built-in recording capabilities. While most of the content recorded today follows a Video-on-Demand model, this may change once the technological challenges have been solved.

Figure 2 ***Global communication over IP***

Consumed content will be customizable according to user preferences e.g. a user in a video conference is able to select which participants they wants to view, and even control aspects of the video such as the position and size of each participant. The recording, sharing and customizability of video content, places new burdens on the network infrastructure, but also opens the door to a new generation of multimedia services and applications.

In this paper, we present a distributed media processing platform in the cloud and discuss the approach taken as well as the challenges that exist.

## Case study: a distributed media processing pipeline

We focus on the following use-case: a user wants to host a multi-party video conference. A traditional multi-party video conferencing uses a Multipoint Control Unit (MCU) that is responsible for mixing the audio, multiplexing video, and sending the resulting stream to the participants of the conference. Leveraging the cloud to provide the function of a Conference Bridge has several advantages including that the user only pays for the time that the bridge is needed, no hardware device is necessary, and the location of the conference bridge can be selected based on locations of available cloud nodes and the participants.

In this paper, we consider the entire media processing pipeline, from video capture at the end-user, to encoding, transport over the network for processing by multiple cloud services, and last mile delivery to the end-user. Further, in our experiments, the content was delivered in real-time to active participants, and with a short delay (2-10 seconds) also available to a larger number of passive participants, i.e. participants that are just consuming the media and not interacting with the other participants.

Our system uses three cloud services to fulfil the features of an MCU in the multi-party video conferencing use-case. In the future such services could be provided by one provider, or by different providers and the main reason is that open standards provide an easy interface for integration. Furthermore, the emergence of WebRTC is resulting in further standardization of discrete components of the media processing pipeline [6], [9].

Our proposed distributed media processing pipeline uses open standards for the purpose of interoperability and communication between the various components. We use H.264/AVC [10] as the video codec, and use the following Internet Engineering Task Force (IETF) protocols; the Real-time Transport Protocol (RTP) [11] for media transport, the H.264 RTP payload format [12] for encapsulating media, the Real-time Streaming Protocol (RTSP) [13] for signaling between the various components. It should be noted that the Session Initiation Protocol (SIP) [14] is typically used for conference call setup. RTSP was used instead to simplify the integration of the various cloud services. In particular, the endpoints use Multipath RTP (MPRTP) protocol [15], [16] to stream media to the conference bridge, mainly to aggregate capacity across available interfaces and provide robustness to failure (fail-over), which is useful in distributed cloud scenarios where it provides load balance without requiring signaling. The cloud nodes take care of playout dejittering and delivery. The use of IETF standards simplified integration between different cloud media processing components and also allows a user with a standard-conformant media player to consume the processed media stream.

The video-mixing function is carried out by two separate real-time transcoding and video-mixing services. Of these, one service uses the Software as a Service (SaaS) approach, while the other is a Platform as a Service (PaaS). Lastly, the conference bridge provides an RTP to Dynamic Adaptive Streaming over HTTP (DASH) [17] protocol translation service to deliver media to passive participants.

**Service 1: Real-time Transcoding/Video-Mixing**

The video stitching service follows the SaaS model. Such a service is typically deployed into the cloud using virtual machines. The service is setup using RTSP and receives the live media from the user (e.g., client devices of video conferencing users or IP cameras) over RTP. The RTP streams are decompressed by an H.264/AVC decoder, scaled to the appropriate resolutions if necessary, and then multiple streams are stitched together in the uncompressed domain in a configurable and flexible layout. The newly-created compound pictures are re-encoded in the H.264/AVC format, packetized in the RTP format and then made available to the other participants via an RTSP server. The media pipeline is shown in Figure 3.The service provider needs to make sure that this stream is available to a wider audience by using a content delivery network.

**Service 2: Distributed Media processing Platform**

MediaCloud [18] realizes a distributed system that exposes multiple physical computing resources as a unified cloud execution environment. In terms of NIST's cloud definition [19] it provides a Platform-as-a-Service (PaaS) approach on top of a cloud infrastructure. Its main objective is to provide scalability beyond a single physical resource e.g. for multimedia and other data intensive real-time applications. Therefore the platform offers capabilities for automated placement of processing elements (realized as self-contained service components) and taking care of their scaling and load-balancing behavior. The framework relieves the application developer to take care of these issues. The MediaCloud framework mixes multiple video streams by plugging in several service components together (RTP-handlers, video stitching, codecs and scaler). The demonstrator shows gains of using the MediaCloud approach by optimizing resource utilization, automated handling of scaling and placement in the infrastructure.

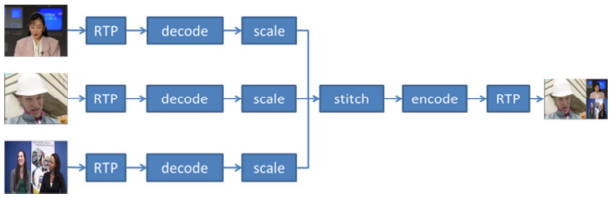



Figure 3 ***Cloud-based real-time stitching media pipeline***

The media streams are coordinated using RTSP for session setup and media is delivered via RTP. In the integrated demonstrator the input streams are received from the RTSP server, and the mixed stream is sent back to the same server. From there the mixed content can be distributed to end-devices.

**Service 3: Translation/Distribution Service**

The RTP to DASH (Dynamic Adaptive Streaming over HTTP [17]) conversion service retrieves the stitched video streams from the other cloud media processing services using RTSP/RTP. Incoming RTP media packets are converted to DASH media segments. The RTP to DASH conversion machine also acts as an HTTP web server that hosts the DASH files and hence provides wider client coverage to the service. In general, the RTP to DASH conversion module outputs the actual video segments and a playlist file representing the order of the video segments.

For the conversion process, we identify live and non-live playlist file types. In the live case, a client can start the playout from the current point of the conversion process with a 2-10 second processing/TCP delay compared to RTP. For the non-live option, the client can start the playout only from the beginning of the playlist, in other words, from the point when the RTP to DASH service was launched. Naturally, the non-live playlist requires extensive disk space for long recordings, because all the converted segments need to be saved.

On the client side, VLC and OSMO4 were used as DASH clients. The RTP to DASH module is currently validated with a Linux OS PC, and the DASH client with an Android OS Tablet client. In the service setup the RTSP/RTP packet losses before the DASH conversion module can decrease the end video quality. Furthermore, e.g. severe network congestion between the DASH server and DASH client(s) can result in playback pauses, which is a result of using TCP.

**Demonstrator**

The entire distributed media pipeline was assembled as a demonstrator. Figure 4 shows the end-to-end jitter variation of a participant streaming from Aalto University, Helsinki to TU Berlin.

Figure **5** shows a screenshot of the media players playing the final streams. The demonstrator showed that it is feasible to use cloud media processing platforms for real-time applications (within tight real-time guarantees), and that the cloud media services can offer added value enabling a new range of media services in the future Internet. Further, the MediaCloud approach demonstrated that scalability can be realized from within the service. Conversely, media services that use the SaaS approach also have to consider scaling and resource allocation, as well as node placement, which are challenging tasks in themselves, and even more challenging for real-time services with continuous media flows and tight end-to-end latency constraints. Further, how can cloud resources be optimized to support real-time multimedia services in the cloud, especially in the case where multiple service providers exist in the cloud?

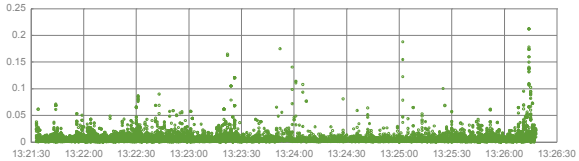



Figure 4 ***Variation in packet jitter (milliseconds)***

Figure 5 **Project demonstrator**

Media service discovery is another challenging topic. How will the user find out what services are available?

**Conclusion**

In this article we have discussed the challenges in building distributed media processing platforms in the cloud. The demonstrator showed how open Standards such as the IETF Internet standards can be used to build a distributed media processing pipeline consisting of multiple cloud media processing services. We have however only taken the first steps to investigate how standardized interfaces enable multi-cloud multimedia services and have outlined the challenges faced in delivering the next-generation multimedia services.

**REFERENCES**

[1] Zencoder. [Online]. http://zencoder.com

[2] Sorenson Media. [Online]. http://www.sorensonmedia.com/video-cloud-solutions

[3] Siri [online]. http://www.apple.com/ios/siri/

[4] R. Sweha, V. Ishakian, and A. Bestavros, "AngelCast: cloud-based peer-assisted live streaming using optimized multi-tree construction," in *ACM Proceedings of the 3rd Multimedia Systems Conference*, 2012.

[5] A. H Payberah, H. Kavalionak, V. Kumaresan, A. Montresor, and S. Haridi, "CLive: Cloud-Assisted P2P Live Streaming," in *IEEE Proc. of the 12th IEEE P2P Conference on Peer-to-Peer Computing (P2P'12)*, Tarragona, Spain, 2012.

[6] C. Jennings, T. Hardie, and M. Westerlund, "Real-time communications for the web," *IEEE Comm. Magazine,* vol. 51, no. 4, April 2013.

[7] YouTube [online] http://www.youtube.com/

[8] Google+ [online] https://plus.google.com/

[9] C. Perkins, M. Westerlund and J. Ott, "Web Real-Time Communication (WebRTC): Media Transport and Use of RTP", October, 2013 http://tools.ietf.org/html/draft-ietf-rtcweb-rtp-usage-10

[10] ITU-T, "Advanced video coding for generic audiovisual services," ITU-T, Standard H.264, 2012.

[11] H. Schulzrinne, S. Casner, R. Frederick and V. Jacobson, "RTP: A transport protocol for real-time applications," IETF, Standard, RFC3550, 2003.

[12] Y.-K. Wang, R. Even, T. Kristensen and R. Jesup, "RTP Payload Format for H.264 Video", IETF, Standard, RFC6184, 2011.

[13] H. Schulzrinne, A. Rao and R. Lanphier, "Real Time Streaming Protocol (RTSP)", IETF, Standard, RFC2326, 1998.

[14] J. Rosenberg, H. Schulzrinne, G. Camarillo, A. Johnston, J. Peterson, R. Sparks, M. Handley and E. Schooler, "SIP: Session Initiation Protocol", IETF, Standard, RFC3261, 2002.

[15] V. Singh, T. Karkkainen, J. Ott, S. Ahsan and L. Eggert, "Multipath RTP (MPRTP)", draft-singh-avtcore-mprtp-06 (work in progress), January 2013.

[16] V. Singh, S. Ahsan and J.Ott, "MPRTP: multipath considerations for real-time media". In *MMSys'13: Proceedings of the 4th ACM Multimedia Systems Conference*, New York, USA, 2013.

[17] T. Stockhammer, "Dynamic Adaptive Streaming over HTTP – Design Principles and Standards," in *MMSys '11: Proceedings of the second annual ACM conference on Multimedia systems,* New York, USA, 2011.

[18] P. Domschitz and M. Bauer, "MediaCloud - a framework for real-time media processing in the network". In *Proceedings of EuroView 2012*,Wuerzburg, Germany, July 2012.

[19] P. Mell and T. Grance, The NIST Definition of Cloud Computing. (800-145), National Institute of Standards and Technology (NIST), Gaithersburg, MD (2011).

**Ralf Globisch** is a Ph.D. candidate and guest researcher at TU Berlin. He is currently working as a developer/researcher in the Real-time Video Coding group of the CSIR Meraka Institute. Since 2009, he has been a guest researcher in the Multimedia Communications group of the Fraunhofer Heinrich Hertz Institute (HHI).

Varun Singh is a final year doctoral student at Aalto University, Finland. His research centres around congestion control and resource management of multimedia applications. He is the co-author of several Internet drafts related, including Multipath RTP (MPRTP) and RTP circuit breakers.

Jürgen Sienel received his diploma degree in Computer Science from the University of Stuttgart in 1992 and is working since then in Alcatel's research branch now being part of Bell Labs. His current interest within the IP Platforms research program is focused on cloud computing platforms for carrier grade multimedia services.

Peter Amon received his diploma (Dipl.-Ing.) degree in electrical engineering from Friedrich-Alexander-Universität Erlangen-Nürnberg, Germany, in 2001. In 2001 he joined Siemens Corporate Technology, Munich, where he is currently working as a Senior Research Scientist in the Imaging and Computer Vision department. His research field include video coding and transmission, image/video processing and analytics, and future internet technologies. In these areas, he published several conference and journal papers and also actively contributed to the standardization at ISO/IEC MPEG and ITU-T VCEG. Currently, he contributes to the standardization of High Efficiency Video Coding (HEVC). He has been and still is actively involved in several European research projects.

Mikko Uitto received his M.Sc. (Tech.) degree in electrical engineering from the University of Oulu, Finland, in 2009. He is currently working as a research scientist in VTT Technical Research Centre of Finland in the Network Resource Management and Control research team and aims for his PhD. The main focus in his research work is adaptive video streaming in mobile networks.

Thomas Schierl received the Dr.-Ing. degree in Electrical Engineering (passed with distinction) from Berlin University of Technology (TUB) in October 2010. He is head of the research group Multimedia Communications in the Image Processing Department at Fraunhofer Heinrich Hertz Institute (HHI), Berlin. Thomas is the co-editor of various IETF RFCs and various MPEG standards. In 2007, Thomas visited the Image, Video, and Multimedia Systems group of Prof. Bernd Girod at Stanford University, CA, USA for different research activities..Thomas' research interests include system integration of video codecs, delivery of real-time media over mobile IP networks such as mobile media content delivery over HTTP, and real-time multimedia processing in cloud infrastructures.

# Internet Video Multicast via Constrained Space Information Flow

*Yaochen Hu[1], Di Niu[1] and Zongpeng Li[2]*
*1. University of Alberta, Canada*
*2. University of Calgary, Canada*
*{yaochen,dniu}@ualberta.ca, zongpeng@ucalgary.ca*

## 1. Introduction

Many multimedia streaming and video webcasting applications can be modeled as a multicast session, where a single source node multicasts the same multimedia content to all the participating terminals, possibly with the help of relay servers. Since video content consumes a lot of network resources, it is desirable to minimize the congestion that the video session imposes on the Internet. The degree of congestion imposed on each link can be modeled by the bandwidth delay product, i.e., the network volume on it. A min-cost video multicast network should be constructed to minimize the sum of bandwidth delay products over all the links in the multicast network.

The min-cost multicast problem is traditionally solved on a given graph formed by the terminals, the source and all the available candidate relay servers in the Internet to find the optimal topology and flow assignments. Such an approach appears to be efficient when the number of candidate relay servers is relatively small. However, recent years have witnessed a rapid growth of the utilizable server pool including CDN nodes and small to medium datacenters, making traditional approaches incapable of handling the large scale of the graph at hand.

An alternative approach is to map the nodes onto a delay space using a network coordinate system, in which the distance between two nodes estimates their pairwise delay [2]. As such, the min-cost multicast network can first be constructed via geometric optimization in a delay space, where we can insert relay servers at arbitrary positions. Such optimal relay positions found in the delay space can then be mapped back to the closest real Internet servers. As long as the servers are densely distributed, this geometric optimization approach can yield a good approximation to the original min-cost multicast problem on a graph.

The remaining geometric problem in the delay space is similar to the *Space Information Flow* (SIF) problem [1], which aims to minimize the sum of bandwidth-distance products in a (geographic) space, allowing network coding and free insertion of relay nodes. The work in [1] presents a heuristic solution to the space information flow problem. However, it has two main drawbacks in practice. *First*, in real applications, introducing more relay server nodes will clearly lead to a higher cost. Neither does [1] consider such cost, nor is it able to solve the problem under a relay number constraint. *Second*, the solution in [1] approximates the geometric problem with a graph version of the problem by dividing the space with grids. This introduces a large number of intermediate variables, as such grids need to be fine-grained to increase accuracy. Although the overall mean complexity of the grid-based approach appears to be polynomial (depending on the optimization accuracy), it still has a large complexity especially when the dimension of the space is high.

In this paper, we propose the *Constrained Space Information Flow* problem, which aims to find the min-cost multicast network under a constraint on the number of relay servers, allowing operators to adjust the cost of using relay servers through such a constraint. We propose an effective EM algorithm to solve the problem *directly* in the geometric space, which can converge to the local optimal solutions with high efficiency.

## 2. Problem Formulation and Algorithms

In this section, we formulate the constrained space information flow problem, show some important properties of the optimal solutions to it, based on which we present our EM algorithm.

**Constrained space information flow problem.**

Although our idea can be extend to a general space, in this letter we focus our work on the min-cost multicast problem in a Euclidean space. Given $N$ terminal nodes $T_1$, $T_2$,..., $T_N$ with coordinates in a space (e.g., in a delay space where the distance between two nodes models their pairwise delay on the Internet) and a multicast session from one source node $S$ to the $N$ terminals as sinks, the objective is to construct a min-cost network in the space, allowing the introduction of at most $M$ extra relay nodes, and allowing any form of coding including network coding to be performed. We define the total cost of the network as

$$\sum_e w(e) f(e),$$

where $f(e)$ is the information flow rate on link $e$, and $w(e)$ is the weight of the link. In a delay space, we set $w(e)$ as the link length $\|e\|$, i.e., the delay on link $e$.

The network cost is determined by two types of

variables, one being the positions of the relay nodes, and the other being the flow assignments on the links. We call these two factors *positions* and *flow assignments*. Note that the flow assignments will also determine the connection topology of all nodes, since a link with a zero rate indicates that the link does not exist. Our problem is to tune these two sets of variables with no more than $M$ relay servers to achieve the minimum cost. Denote $V_R$ as the set of $M$ candidate relay nodes to be found and $V$ as the set of all nodes. Our optimization problem can be stated as

**Minimize** $\sum_{u,v\in V} \| x_u - x_v \| f(\overrightarrow{uv})$ (1)

**Subject to :**

$$\begin{cases} \sum_{v\in V} f_i(\overrightarrow{vu}) = \sum_{v\in V} f_i(\overrightarrow{uv}), & \forall i, \forall u \in V, \quad (2) \\ f_i(\overrightarrow{T_i S}) = r, & \forall i, \quad (3) \\ f_i(\overrightarrow{uv}) \le f(\overrightarrow{uv}), & \forall i, \forall u, v \in V, \quad (4) \\ f(\overrightarrow{uv}) \le c(\overrightarrow{uv}), & \forall u, v \in V, \quad (5) \\ f(\overrightarrow{uv}) \ge 0, f_i(\overrightarrow{uv}) \ge 0, & \forall i, \forall u, v \in V, \quad (6) \\ | V_R | \le M. & \quad (7) \end{cases}$$

In (1), $x_u$ is the position of node $u$. The positions of the source node and the terminal nodes are fixed input vectors, while the positions of the relay server nodes are variables to be optimized. For every network information flow $S \rightarrow T_i$, there is a *conceptional* flow $f_i(uv)$. We call it *conceptional* because different conceptional flows share the bandwidth on the same link. As stated in (4), the final flow rate $f(uv)$ of a link $uv$ should be no less than the maximum conceptional rate, which will directly affect the total cost. The constraint (2) guarantees the conceptional flow equilibrium property for every node and every conceptional flow $i$. The assigned "feedback" flow in (3) characterizes the desired receiving rate at each terminal. (5) is the trivial link capacity constraint. For every pair of nodes, we have both $f_i(uv)$ and $f_i(vu)$ to indicate the flows in two directions, so that (6) gives another trivial bound. Finally, (7) indicates the constraint on the maximum number of relay server nodes.

We need to solve this problem over both the variables $x_u$ (*relay positions*) for $u$ in $V_R$ and all the conceptional flow assignments on all the links (*flow assignment*). Note that any feasible flow assignment satisfying (2)-(7) can be achieved with linear network coding in a single multicast session [3].

**Properties of the Solutions.**

It is hard to simultaneously obtain the optimal values for both relay positions and flow assignments, since it is not hard to verify that the problem (1)-(7) is non-convex. However, we have some good properties for the problem once we fix one set of variables.

When the *positions* of the relay server nodes are fixed, the proposed optimization problem (1)-(7) is reduced to a simple *linear program* (LP). The number of variables is $N+1$ times the number of links, i.e., $O(N(M+N)^2)$. The number of linear constraints is also $O(N(M+N)^2)$. Therefore, we can solve it efficiently with common LP solvers.

When the *flow assignment* of the network is fixed, the cost function in (1) is the sum of norms and all the constraints in (2) to (6) are irrelevant to the position variables. The optimization problem now reduces to a convex optimization problem. There are many efficient algorithms to solve such kind of problems. More specifically, for the sum of norms in this problem, an *Equilibrium* method has been proposed in [1], which can efficiently converge to the optimum.

Based on these observations, we propose our EM heuristic algorithm. In the EM algorithm, the above two local optimizations for relay positions and flow assignments are alternately performed.

**An EM Algorithm.**

Our proposed EM algorithm is shown in **Algorithm 1**. Initially, Step 1 randomly assigns the positions of the relay server nodes in the smallest box region containing all the terminals and the source. The following steps are iterative operations. In each iteration, there are three major steps. We first solve the LP in (1)-(7) with the relay positions fixed to obtain the flow rate assignments. Then with these flow rates fixed, we solve a convex optimization problem for the relay server node positions. Finally, for each relay node that has no throughput on it, we randomly reassign a new position to it and repeat the iterations. The $\varepsilon$ in Step 5 is a small positive threshold to exclude the fake non-zeros, since in our LP solver there is always a small non-zero value on an actually zero-valued variable.

As for the *termination condition* in Step 9, we introduce a counter to help us monitor the termination condition. In each iteration, we first calculate the ratio between the cost in the previous iteration and the cost in the current iteration, and increase the counter if the ratio is less than some threshold. Once the counter reaches some number, the whole algorithm terminates. Finally, we delete the relay nodes that have no throughput and output solution..

**Algorithm 1** EM Heuristic Algorithm

**Input:** $N$ terminals, the source node, the constraint $M$ on the number of relay servers

**Output:** a solution to the Constrained SIF problem Relay positions, flow assignments

1: Randomly generate $M$ relays in the smallest box containing the source and all the terminals;
2: (**Flow Assignment**) Fix the relay positions, and solve the LP in (1)-(7) to get the flow rates;
3: (**Relay Position Tuning**) Fix the flow rates assigned in Step 2, and solve the convex optimization problem for the relay positions, e.g., using the Equilibrium method in [1].

(**Random Seed Generation**)
4: **for** $i$ = 1 to $M$, **do**
5: **if** the total flow on Relay $i < \varepsilon$ **then**
6: Randomly generate a new relay $i$ in the smallest box containing the source and all the terminals;
7: **end if**
8: **end for**
9: **if** the *termination condition* does not hold, **go to** Step 2;
10: Delete the relay server nodes that have no flow on them.

### 3. Simulation and Performance

We have simulated the EM algorithm in a 2-D Euclidean space. We choose 11 nodes uniformly at random in the unit box. 10 of them are set as the terminals, while the remaining one is set as the source. We set 4 as the constraint on the number of relays. Since the multicast cost will be proportional to the receiving rate, we assume the receiving rate is 1 at each terminal and assign a large capacity 10 to each link. For the termination condition, we will increment the counter if the former cost is no greater than 1.05 times the current cost. We stop the algorithm when the counter reaches 1000. Fig. 1 shows the resulted relay server positions and topology. The final total cost in terms of the sum of bandwidth-delay products is 6.72.

### 4. Conclusions

In this letter, we present a solution for the min-cost video multicast problem by considering the problem in a delay space based on network coordinates. We solve the resulted Constrained Space Information Flow problem using an EM algorithm, that can take into account the constraint on the number of relay servers and can scale to a large pool of candidate servers. Preliminary simulations show that our algorithm can yield fast convergence to the local optimal solutions to the non-convex combinatorial optimization problem.

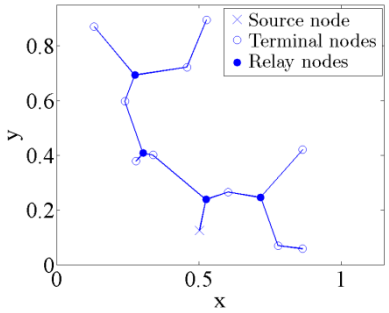



Fig. 1 The simulation result for randomly chosen 10 terminals and 1 source. The delay node number constraint is 4.

### References

[1] Jiaqing Huang, Xunrui Yin, Xiaoxi Zhang, Xu Du and Zongpeng Li, "On Space Information Flow: Single Multicast," in the Proc. of Network Coding (NetCod), 2013 International Symposium on 7-9 June 2013.

[2] F. Dabek, R. Cox, F. Kaashoek, and R. Morris, "Vivaldi: A decentralized network coordinate system," in Proc. of ACM SIGCOMM, 2004.

[3] R. Ahlswede, N. Cai, S.T.R Li, and R. W. Yeung, Network Information Flow, IEEE Trans. On Information Theory, 46(4):1204-1216, 2000.

[4] S. Boyd and L. Vandenberghe, Convex Optimization (7e), Cambridge University Press, 2007.

# Compressive Sensing for Video Coding: A Brief Overview

*Quan Zhou and Liang Zhou*
*College of Telecom & Inf Eng, Nanjing University of Posts & Telecom, Nanjing, P.R. China*
*{quan.zhou, liang.zhou}@njupt.edu.cn*

## 1. Introduction

Conventional approaches for video coding follow the major standard, such as MPEG [1] and H.26X [2], which are well advanced and widely employed in last few decades. These developed standards underlie nearly all the video compressive protocols used in consumer devices and visual electronics, such as VCDs and DVDs. Since the video needs to be compressed only once in the encoder, while the decompression in decoder is required to be performed many times, it is expected that the decoding should be implemented as simple and quickly as possible. To this end, the existing video coding paradigms, such as MPEG as well as H.26X, essentially design a complex encoder and a simple decoder. In order to investigate the spatial-temporal redundancies for video data, the encoder has to be restricted with specially designed hardware, leading to the increasing cost of the cameras.

In the past ten years, with the emergence of smart-phones and continued growth of laptops, netbooks, tablets, and sensors, there is a huge increase in a number of mobile devices able to support new video signal applications. For example, the video surveillance and sports broadcasting, these mobile devices are in fact video cameras. However, delivering video data and conveying visual information from mobile devices over cellular or mobile broadband networks confronts many challenges, e.g., limited channel bandwidth; high required quality and reliability; and most importantly, the insecure, time varying, and unplanned operating environments. This casts an emergent problem that needs to be well addressed for the video coding scheme: *due to the limited computational and energy resources, the traditional video coding standard involved complex encoders may not be suitable for recent environment of video delivery and transmission.*

Recent years have witnessed substantial research and developments of a new theory called Compressive Sampling, also known as Compressive Sensing (CS) [3, 4, 5], which may show great promise to solve above problem. The most striking characteristic of CS theory is that one can recover certain signals only using very fewer samples than traditional methods use, such as Shannon's sampling theory [6]. A lower sampling rate implies less computation and energy requirement for data processing, resulting in lower power requirements for the special designed hardware. For video coding, due to the substantial amounts of redundancy in the spatial-temporal domain of video data, the CS theory seems to be potentially applied using only few samples with good fidelity. Our intent of this letter is to briefly overview the basic CS theory in current literatures, present the underlying mathematical insights of this theory, survey a couple of important results of applying CS theory for video coding, and specify some possible research directions in the future.

## 2. Basic Theory of CS

The famous Shannon's sampling theorem, the so-called Nyquist sampling rate, asserts: If one needs to recover a certain signal without error, the sampling rate must be at least twice the maximum frequency present in the signal [6]. On the contrary, the CS theory tells us: If a certain signal is projected to a feature space spanned by some predefined orthogonal basis, for example, Wavelet basis and Fourier basis, then a large fraction of projected coefficients can be "thrown away", while the perceptual loss is hardly noticeable from the reconstructive signal. To make this possible, CS theory depends on two major principles: *sparsity*, which involves the signals of interest, and *incoherence*, which involves the sensing modality.

- *Sparsity*. For the continuous time signal, *sparsity* expresses the idea that the significant signal content, so-called "information rate", may be much smaller than what is suggested by signal bandwidth. On the other hand, for the discrete time signal, *sparsity* implies that the number of freedom degrees for the original signal, is comparably much smaller than its finite signal length. More precisely, the CS theory investigates the common fact that due to the sparse structure, most nature signals are redundant and thus compressible. In other words, the nature signals always have concise representations with proper selected basis.

Mathematically, suppose we are given a discrete signal $f \in \mathrm{R}^n$ sampled from a certain continuous signal, which can be linearly represented by a series of orthonormal basis $\boldsymbol{\Psi} = [\varphi_1, \varphi_2, ..., \varphi_n]$ as follows:

$$f(t) = \sum_{i=1}^{n} x_i \varphi_i(t) \qquad (1)$$

where $\boldsymbol{x} = [x_1, x_2, ..., x_n]$ is the coefficient vector of $f$ and defined as $x_i = <f, \varphi_i>$. Then the sparsity implies that when a signal has the form of sparse expansion,

most coefficients with small magnitude is able to be discarded. Formally, one can only keep top $S$ terms with largest $x_i$ in Eqn. (1) and set the rests as zero, thus defines the sparse vector $\boldsymbol{x}_s$ with at most $S$ nonzero entries. If $S << n$, we will call $f$ as *S-sparse*, and $\boldsymbol{x}_s$ is sparse in a strict sense.

Of course, the above principle underlies the essence of current lossy compression, such as JPEG-2000 [7] as well as many other coding standards: one could encode the indexes $i$ and the corresponding magnitude of the $S$ significant coefficients. The main advantage of such process lies in that it requires no prior knowledge of all the coefficients in $\boldsymbol{x}$, as well as the indexes that imply the significant pieces of information in advance.

- *Incoherence*. Suppose we have a pair of orthogonal basis ($\boldsymbol{\Psi}$, $\boldsymbol{\Phi}$) of $R^n$, where the first basis $\boldsymbol{\Psi}$ is used for sampling or sensing and the second basis $\boldsymbol{\Phi}$ is used to represent signal $f$. According to [8], then the coherence between these two bases can be defined as:

$$\mu(\boldsymbol{\Psi}, \boldsymbol{\Phi}) = \sqrt{n} \cdot \max_{1 \le k, j \le n} |< \varphi_k, \phi_j >| \qquad (2)$$

The implication of *incoherence* is now clear: It measures the largest correlation between any two elements of sensing basis $\boldsymbol{\Psi}$ and representation basis $\boldsymbol{\Phi}$ [9]. If there exists correlated elements in $\boldsymbol{\Psi}$ and $\boldsymbol{\Phi}$, the coherence thus tend to be large, otherwise, it tend to be small. Regarding to how large and how small, it follows the linear algebra that:

$$\mu(\boldsymbol{\Psi}, \boldsymbol{\Phi}) \in [1, \sqrt{n}] \qquad (3)$$

For CS theory, the pair of two bases is always designed to achieve low coherence.

In summary, sparsity and incoherence together quantify the compressibility of a certain signal. The higher sparsity and incoherence a nature signal has, the more compressible this signal is.

**3. CS based Video Coding**

Research on the application of CS theory for video system has been started from last decade. We start by briefly reviewing the open literature in recent reports.

- *Distributed video compression/coding*. The pioneer work using CS in distribution video coding (DVC) is proposed in [11]. Unlike conventional approach that exploits source statistics in encoder, DVC methods investigate video redundancy at the decoder. Two or more independent encoders are cooperative to encode statistically dependent source. The statistical dependent bit-streams from each encoder are sent to a common decoder for joint decompression. Some other DVC literatures can be referred to [12, 13, 14].

- *Mobile broadcasting*. Another implementation of CS theory is mobile broadcasting [15], where the video data is divided into self-contained video cubes with some proper selected random matrices as sensing basis. The proposed method is scalable with channel capacity, particularly in wireless broadcast situations.

- *Rate-energy-distortion in video streaming*. Unlike the previous coding standard, such as MPEG [1], the concept of rate-energy-distortion [16] is proposed to trade-off coding efficiency, energy resource and recovery quality. In this application, an empirical model is designed when limited energy is available for video compression and transmission.

**4. Possible Applications in Video Coding**

The applications of CS theory suggest that a large amount of mathematical and computational methods could have an enormous impact in areas where the conventional multimedia information processing has significant limitations, especially in the field of video acquisition/coding. This section presents some possible research directions that could significantly expand the ability of traditional coding scheme using CS theory.

- *Video compression/coding*. In some cases, the sparse basis $\boldsymbol{\Psi}$ may be unknown in the initialization of video compression. Furthermore, the $\boldsymbol{\Psi}$ may have no sparse structure, leading to the impractical implement at the decoder. To address this problem, the random design of basis $\boldsymbol{\Phi}$, also known as "Random Sensing" [17], can be seen as a universal coding paradigm, as it can be designed without considering the structure of basis $\boldsymbol{\Psi}$. In multi-signal setting such as sensor networks, this universality may be particularly suitable for distributed source coding [12].

- *Channel coding*. In order to protect from random interference during video transmission, the important principles of CS theory, such as sparsity, randomness and convex optimization, can be applied to design fast correcting codes to rectify the transmission errors, as explained in [18].

- *Video data acquisition and sensing*. The traditional video coding schemes depend on the fact that the video data have been acquired using physical cameras. In some important situations, however, it may be very hard to obtain discrete-time video sequence from an analog signals, resulting in the difficulty in the subsequently compression. According to the CS theory, instead to perform analog-digital conversion, the most feasible approach is to design the sampling hardware that directly record discrete and low-rate measurements of the incident analog signals. We refer the reader to [8] by Candes and Wakin elsewhere in this issue for related discussions.

**5. Conclusion**

In summary, the characteristic of sparse structure for the multimedia data makes CS a very useful and powerful mathematical tool for a broad scope of natural science and engineering applications. CS theory, however, is still a fresh field and it is even more recent to apply it into video systems. As a result, there are many avenues for the future research and the thorough quantitative/qualitative analyses are still lacking. For example, the relationship between the underlying features and the sparse characteristics of videos, as well as the involved visual-based quantitative assessment principles are still far from well understood. Even so, CS methodology will undoubtedly play a more important role in the society of advanced multimedia signal processing. All the related CS-based theories, techniques, and practical applications are encouraged to be in-depth investigated in the near future.

**References**


[1] P. Symes, "Digital Video Compression," *McGraw-Hill*, 2004.

[2] T.Wiegand, G. Sullivan, G. Bjontegaard, and A. Luthra, "Overview of the H.264/AVC video coding standard," in *IEEE Transactions on Circuits and System for Video Technology*, vol. 13, no. 7, pp. 560-576, 2003.

[3] E. Candès, J. Romberg, and T. Tao, "Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information," in *IEEE Transactions on Information Theory*, vol. 52, no. 2, pp. 48-509, 2006.

[4] E. Candès and T. Tao, "Near optimal signal recovery from random projections: Universal encoding strategies?" in *IEEE Transactions on Information Theory*, vol. 52, no. 12, pp. 5406-5425, 2006.

[5] D. Donoho, "Compressed sensing," in *IEEE Transactions on Information Theory*, vol. 52, no. 4, pp. 1289-1306, 2006.

[6] C. Shannon, "Classic paper: Communication in the presence of noise," in *Proceedings of the IEEE*, vol. 86, no. 2, pp. 447–457, 1998.

[7] D.S. Taubman and M.W. Marcellin, "JPEG 2000: Image Compression Fundamentals, Standards and Practice," *Norwell, MA: Kluwer*, 2001.

[8] E. Candes and M. Wakin, "An introduction to compressive sampling," in *IEEE Signal Processing Magazine*, pp. 21-30, 2008.

[9] M. Y. Baig, E. M. K. Lai and A. Punchihewa, "Compressive Video Coding: A Review of the State-Of-The-Art," *Video Compression*, 2013

[10] R. Coifman, F. Geshwind, and Y. Meyer, "Noiselets," in *Applied and Computational Harmonic Analysis*, vol. 10, no. 1, pp. 27–44, 2001.

[11] J. Slepian and J. Wolf, "Noiseless coding of correlated information sources," in *IEEE Transactions on Information Theory*, vol. 19, no. 4, pp. 471–480, 1973.

[12] D. Baron, M.B. Wakin, M.F. Duarte, S. Sarvotham, and R.G. Baraniuk, "Distributed compressed sensing," 2005, Preprint.

[13] L. Zhou, X. B. Wang, T. Wei, G.M. Muntean, and B. Geller, "Distributed Scheduling Scheme for Video streaming over Multi-Channel Multi-Radio Multi-Hop Wireless Networks", in *IEEE Journal on Selected Areas in Communications*, vol. 28, no. 3, pp. 409-419, 2010.

[14] T. T. Do, C. Yi, D. T. Nguyen, N. Nguyen, G. Lu, and T. D. Tran, "Distributed Compressed Video Sensing," in *Information Sciences and Systems*, pp. 1-2, 2009.

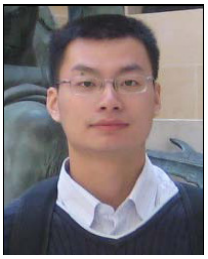
[15] L. Chengbo, H. Jiang, P. Wilford, Y. Zhang and M. Scheutzow, "A new compressive video sensing framework for mobile broadcast", in *IEEE Transactions on Broadcast*, vol. 59, no. 1, pp. 197-205, 2013.

[16] S. Pudlewski and T. Melodia, "Compressive Video Streaming: Design and Rate-Energy-Distortion Analysis", in *IEEE Transactions on Multimedia*, vol. 15, no. 8, pp. 2072-2086, 2013.

[17] E. Candes and J. Romberg, "Sparsity and incoherence in compressive sampling," in *Inverse Problems*, vol. 23, no. 3, pp. 969–985, 2007.

[18] E. Candès and T. Tao, "Decoding by linear programming," in *IEEE Transactions on Information Theory,* vol. 51, no. 12, pp. 4203-4215, 2005.


**Quan Zhou** received the B.S. degree in Elect and Inf Eng from China University of Geosciences, Wuhan, China, in 2002, and the M.S. and Ph.D. degree in Elect. and Inf. Eng from Huazhong University of Sci. and Tech., Wuhan, China in 2006 and 2013, respectively. He is now the assistant professor in college of Telecom and Inf. Eng, Nanjing University of Posts and Telecom, Nanjing China. His major research interests include image processing, computer vision, pattern recognition, and multimedia communication.



**Liang Zhou** received his B.S. degree and M.S. degree (with honors) both major at Elect Eng from Nanjing University of Posts and Telecom, Nanjing, China in 2003 and 2006, respectively. In March 2009, he received his Ph.D. degree major at Elect Eng both from Ecole Normale Superieure (E.N.S.), Cachan, France and Shanghai Jiao Tong University, Shanghai, China. From Mar. 2009. His research interests are in the area of multimedia communications and services, in particular, resource allocation, scheduling, and cross-layer design. He is the editor board of IEEE Transactions on Circuits and System for Video Technology, IEEE Communications Surveys & Tutorial, and Guest Editor for other journals.

# Offloading Policy of Video Transcoding for Green Mobile Cloud

*Weiwen Zhang and Yonggang Wen*
*School of Computer Engineering, Nanyang Technological University*
*{wzhang9, ygwen}@ntu.edu.sg*

## 1. Introduction

Video transcoding [1] is an enabling technology to support the growing demand of video consumption on mobile devices. According to Cisco VNI report [2], mobile video consumption will increase 16-fold between 2012 and 2017. This trend poses challenges on the design of mobile application platform, because mobile devices can support limited video formats and resolutions. Transcoding can allow videos to be played on diverse mobile devices, by adapting videos into a particular format (e.g., mp4) along with a resolution reduction. This transcoding process, however, is computation-intensive, which can drain the battery lifetime of mobile devices. Therefore, a new computing paradigm is increasingly demanded for mobile devices to consume video content.

Cloud computing [3] offers a natural way to accomplish transcoding tasks. Mobile devices can upload videos to the cloud for transcoding and thus, computation is shifted from the mobile device to the cloud, which is referred to computation offloading [4]. By computation offloading, significant energy consumption can be saved on mobile devices, enabling more media applications [5].

In this paper, we first present a generic green mobile cloud system to provide Transcoding as a Service (TaaS) to mobile devices. We then propose an optimization framework of offloading policy to reduce the energy consumption on both the mobile device and the cloud. For the mobile device, we formulate offloading policy as a constrained optimization problem, in order to minimize the energy consumption on the mobile device while satisfying a delay deadline. We find the operational region on which execution mode, i.e., mobile execution or cloud execution, is more energy efficient. For the cloud, using the framework of Lyapunov optimization, we propose an online algorithm to dispatch transcoding tasks to service engines. This algorithm can reduce energy consumption while achieving the queue stability in the cloud. The optimization framework of offloading policy can jointly reduce the energy consumption on both the mobile device and the cloud, which provides guidelines for the design of green mobile cloud.

## 2. System architecture for green mobile cloud

In this section, we present a generic green mobile cloud system to provide TaaS to mobile devices.

In Figure 1, the system of green mobile cloud consists of mobile devices, a dispatcher and a set of service engines. The dispatcher receives offloading requests from mobile devices and dispatches these requests to service engines for computation. The service engine can be a physical server or a virtual machine. If the video requested by users is cached in service engines or stored at the back-end storage, the video can be rendered immediately to users without transcoding; otherwise, video transcoding is performed either on the mobile device or one of the service engines in the cloud. Particularly, if the service engine has an identical image of the mobile device, and transcoding can be conducted on that service engine without the mobile user sending the input file.

In this architecture, the dispatcher makes the offloading decision for both the mobile device and the cloud. The dispatcher can have the information of mobile devices (i.e., profile of transcoding tasks) and service engines in the cloud (i.e., queue length). Based on this information, we can design a policy for the dispatcher. In this paper, leveraging optimization theory, we adopt a model-based approach to the offloading policy. We can also adopt a rule-based approach using machine learning; that approach remains our future work.

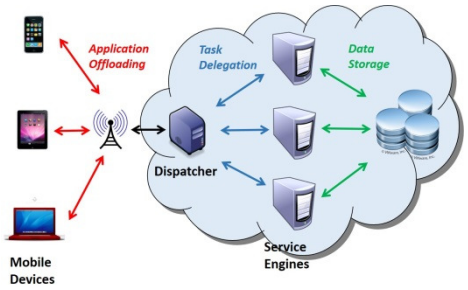



Figure 1. Architecture of green mobile cloud.

## 3. Optimization framework.

In this section, we propose an optimization framework of offloading policy for green mobile cloud.

As illustrated in Figure 2, a transcoding task can be executed in two alternative modes:

- Mobile execution: a transcoding task is executed locally on the mobile device;
- Cloud execution: a transcoding task is offloaded and scheduled by the dispatcher to one of the service engines for execution.

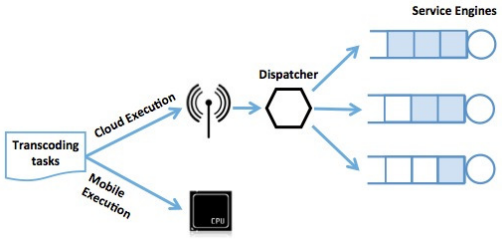



Figure 2. Execution modes for transcoding tasks.

To save energy consumption on the mobile device, we formulate delay-constrained optimization problems for the mobile execution and the cloud execution, respectively. Given the profile of the transcoding task, we determine which execution is more energy-efficient. First, for the mobile execution, its energy consumption $\xi_m$ is minimized by optimally configuring the clock frequency via dynamic voltage scaling (DVS), i.e., $\xi_m^* = \min_{\psi \in \Psi}\{\xi_m(L, T_d, \psi)\}$, where $\psi$ is the set of all feasible clock-frequency vectors $\psi$, $L$ is the input data size and $T_d$ is the time deadline of video transcoding. Second, for the cloud execution, its energy consumption $\xi_c$ is minimized by optimally setting the transmission rate on the mobile device, i.e., $\xi_c^* = \min_{\phi \in \Phi}\{\xi_{tran}(L, T_{tran}, \phi)\} + \xi_{recv}(L')$, where $\Phi$ is the set of all feasible data scheduling vectors, $T_{tran}$ is the transmission time, $L'$ is the output data size, $\xi_{tran}$ and $\xi_{recv}$ are the energy consumed by transmitting and receiving data, respectively. Particularly, $T_{tran} = T_d - T_{recv} - T_0$, where $T_{recv}$ is time of receiving output data and $T_0$ is queueing delay determined by queue length of the service engine that executes the transcoding task depending on the dispatching decision. Then, the offloading policy for mobile devices is obtained by comparing the optimal energy consumption of the mobile execution and the cloud execution.

To save energy consumption on service engines, we formulate the dispatching decision as a stability-constrained optimization problem. We aim to minimize the time average energy consumption while satisfying the queue stability. Particularly, we consider a discrete time model and assume that transcoding time can be estimated and queue length of each service engine can be observed for each time slot (i.e., $\mathbf{A}(t)=\{A(t)\}$ and $\mathbf{Q}(t)=\{Q(t)\}$). We define the time average energy consumption $\overline{E}$ and the time average queue length $\overline{Q}$ as the average of summation of energy consumption and remaining transcoding time by all the service engines over a long period of time, respectively. The energy consumption for each service engine is a product of its power $P$ and the transcoding time $A(t)$. In addition, the queue stability is defined as $\overline{Q} < \infty$.

**4. Energy-efficient offloading policy**

In this section, we present the energy-efficient offloading policy for the mobile device and the cloud, under the optimization framework.

**Offloading policy for mobile devices.**

We can adapt the results in [6] to obtain the minimum energy consumption on the mobile device. Specifically, the minimum energy consumption of the mobile device by the mobile execution is $\xi_m^* = ML^3/T_d^2$, where $M$ is a constant depending on the chip architecture on the mobile device. The minimum energy consumption on the mobile device by the cloud execution is $\xi_c^* = C(n) L^n/(T_d - L'/r' - T_0)^{n-1} + P'L'/r'$, where the first term refers to the energy consumption of transmitting the input data and the second term refers to the energy consumption of receiving the output data. In addition, $C(n)$ is a function of $n$ for the cloud execution, and $P'$ and $r'$ are the power and rate of receiving the output data, respectively.

We consider the application of transcoding FLV files with 1920x1080 resolution size into mp4 files with 480x360 resolution size. We can determine which execution is more energy-efficient for the mobile device, by comparing $\xi_m^*$ and $\xi_c^*$. We model $L'$ as a linear function of $L$, using data fitting over the output data size, i.e., $L' = aL + b$, where $a$ = 0.02658 and $b$ = 0.3316. We also set $r'$=500KB/s, $n$=2 and $T_0$=0.5s, where $T_0$ is the queueing delay based on the dispatching decision in the cloud. Figure 3 shows there is an operational region that separates the mobile execution and the cloud execution for input data size $L$ and specified time delay $T_d$. Given ($T_d$, $L$), if it is above the curve, then the cloud execution is optimal; otherwise, the mobile execution is optimal.

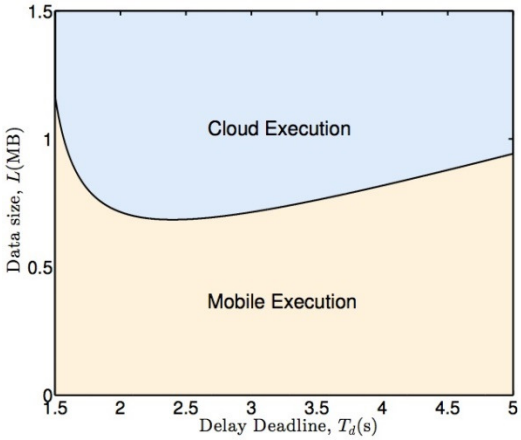



Figure 3. Operational region of the mobile execution and the cloud execution for mobile devices.

**Offloading policy for cloud.**

Using the framework of Lyapunov optimization [7], we propose an online algorithm to dispatch transcoding

tasks to the cloud. Upon receiving the transcoding task at time slot $t$, the dispatcher estimates its transcoding time for each service engine, $\mathbf{A}(t)$, and observes the queue length on each service engine, $\mathbf{Q}(t)$. Then, the transcoding task is dispatched to the service engine with the minimum value of $A(t)(Q(t)+VP)$, where $V$ is a control variable. The queue length of the chosen service engine is the queueing delay of offloading for mobile devices. In addition, the queue length is updated at every time slot for each service engine.

The online algorithm can reduce energy consumption while achieving the queue stability. We do the simulation, assuming there are 10 physical servers in the cloud. Figure 4 shows the time average energy consumption and the time average queue length that are normalized and calculated over 50000 time slots under different $V$. With the increase of $V$, the time average energy consumption decreases and converges to the optimal value. However, with the increase of $V$, the time average queue length grows linearly. Hence, the cloud operator can dynamically tune the variable $V$ for the energy-delay tradeoff.

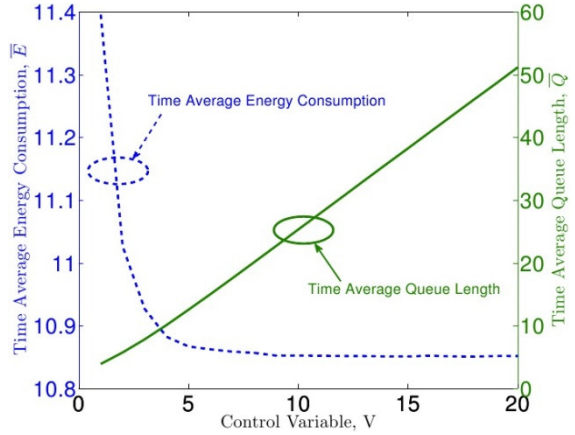



Figure 4. Tradeoff between time average energy consumption and time average queue length.

## 5. Conclusion

In this paper we presented a generic mobile cloud system. We proposed energy-efficient offloading policy for TaaS to minimize the energy consumption of transcoding on both the mobile device and the cloud while achieving low delay. For mobile devices, we obtained the operational region of the optimal execution for mobile devices. For the cloud, we proposed an online algorithm to dispatch transcoding tasks to service engines, which can reduce energy consumption while achieving the queue stability. In the future, we will have VM resource management such that CPU and memory resource can be dynamically allocated. We will also use machine learning method to design the offloading policy.

## References

[1] A. Vetro, C. Christopoulos, and H. Sun, "Video transcoding architectures and techniques: an overview," Signal Processing Magazine, IEEE, vol. 20, no. 2, pp. 18–29, 2003.

[2] Cisco Visual Networking Index: Forecast and Methodology, 2012 - 2017, Cisco, 2013.

[3] M. Armbrust, A. Fox, R. Griffith, A. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, I. Stoica, and M. Zaharia, "A view of cloud computing," Communication of the ACM, vol. 53, no. 4, pp. 50–58, 2010.

[4] K. Kumar and Y. H. Lu, "Cloud computing for mobile users: Can offloading computation save energy?" IEEE Computer, vol. 43, no. 4, pp. 51–56, 2010.

[5] Y. Xu and S. Mao, "A survey of mobile cloud computing for rich media applications," IEEE Wireless Communications, vol. 20, no. 3, pp. 46–53, 2013.

[6] W. Zhang, Y. Wen, K. Guan, D. Kilper, H. Luo, and D. Wu, "Energy-efficient mobile cloud computing under stochastic wireless channel," in IEEE Transactions on Wireless Communications, 2013, pp. 4569–4581.

[7] M. J. Neely, "Stochastic network optimization with application to communication and queueing systems," Synthesis Lectures on Communication Networks, vol. 3, no. 1, pp. 1–211, 2010.

**Weiwen Zhang** received a Bachelor's Degree in Software Engineering and Master's Degree in Computer Science from South China University of Technology, in 2008 and 2011, respectively. He is a PhD student in Nanyang Technological University in Singapore. His research interests include mobile computing, cloud computing and green media.

**Yonggang Wen** received his PhD degree in Electrical Engineering and Computer Science (minor in Western Literature) from Massachusetts Institute of Technology, Cambridge, USA. He is an assistant professor with school of computer engineering at Nanyang Technological University, Singapore. He has published over 80 papers in top journals and prestigious conferences. His latest work in multi-screen cloud social TV has been featured by global media (more than 1600 news articles from over 29 countries) and recognized with ASEAN ICT Award 2013 (Gold Medal) and IEEE Globecom 2013 Best Paper Award. His research interests include cloud computing, green data center, big data analytics, multimedia network and mobile computing.

# SPECIAL ISSUE ON NETWORK CALCULUS

## Applications of Network Calculus to Multimedia Communications

*Guest Editor: Florin Ciucu (University of Warwick)*

{florin@dcs.warwick.ac.uk}

Network calculus emerged in the early 1990s as an alternative to the classical queueing theory. Motivated by the need to analyze complex network scenarios with non-Poisson arrivals, network calculus has evolved as an elegant analytical framework attractive to computer scientists and engineers. In this sense, a key principle lies in the abstraction of cumbersome mathematical technicalities related to arrival processes, scheduling, and multi-node analysis through the innovative concepts of envelopes and service processes. This ability to analyze systems in a simple and yet intuitive manner has made network calculus applicable in diverse areas such as switched Ethernets, systems-on-chip, avionic network, or the smart grid.

The goal of this issue of the E-Letter is to illustrate the applicability of network calculus in the area of multimedia communications. The issue consists of five papers, co-authored by network calculus experts, covering both applications and theory. In particular, the issue concerns topics such as the evaluation of the perceived quality of video streaming, the analytical computation of latencies in multimedia sensor networks, the design of System-on-Chips in multimedia systems, improving the accuracy of network calculus performance metrics, and the modelling and analytical understanding of transcoders in end-to-end multimedia streaming.

The first paper, titled "A Network Calculus Extension to EvalVid", is co-authored by Heidinger and Lim from BMW Group Research and Technology, Germany. The paper overviews the integration of network calculus into EvalVid -- a framework to assess the perceived quality of video streaming. In particular, by relying on network calculus delay bounds, the paper illustrates the variability of the perceived quality (i.e., as the Mean Opinion Score) as a function of the play-out buffer. The results are shown for various scenarios such as automotive in-car and cabin networks.

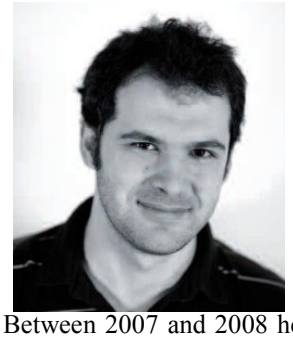
The second paper, titled "Performance Analysis for Wireless Multimedia Sensor Networks Based on Stochastic Network Calculus", is co-authored by Wang, Wang, and Peng from Guangxi University, China. The authors address the derivation of probabilistic end-to-end delay bounds in a multi-hop multimedia sensor network, whereby flows are assigned certain weights. The paper illustrates the elegance of the stochastic network approach to derive end-to-end results for bursty data carried over a wireless environment. The underlying analytical challenges are addressed using the convenient representations of arrivals and service by envelopes and service processes.

The third paper, titled "A General Stochastic Framework for Low-Cost Design of Multimedia SoCs", is co-authored by Raman et al. The authors consider the problem of better designing resource-constrained System-on-Chips (SoCs) in multimedia systems through analytically understanding the required resources. Concretely, the authors apply the stochastic network calculus to dimension play-out buffers in video decoders. A key message is that, in contrast to its deterministic counterpart, the stochastic network calculus can lend itself to better dimensioning of resources in the presence of randomness in both the streaming data and processing.

The fourth paper, titled "Copula Analysis for Stochastic Network Calculus", is co-authored by Wu, Dong, and Srinivasan from the University of Victoria, Canada. This paper addresses a key theoretical challenge in the stochastic network calculus, i.e., the tightness of the produced probabilistic bounds. In particular, the authors integrate the copula theory into the framework of the stochastic network calculus to improve performance bounds when arrival processes are not necessarily independent. The obtained results have the potential to better design scheduling systems in multimedia systems.

The fifth paper, titled "A Delay Calculus of Streaming Media with Video Transcoding", is co-authored by Wang and Schmitt from University of Kaiserslautern, Germany. The paper addresses the problem of end-to-end delay bounds of video streaming in a network with transcoders. The authors demonstrate in particular the suitability of the stochastic network calculus to model the lossy behavior of transcoders, and show that end-to-end latencies grow in the number of transcoders.

We hope that the five contributions convincingly demonstrate the potential and suitability of network calculus to better understand and design multimedia communications systems.



**Florin Ciucu** was educated at the Faculty of Mathematics, University of Bucharest (B.Sc. in Informatics, 1998), George Mason University (M.Sc. in Computer Science, 2001), and University of Virginia (Ph.D. in Computer Science, 2007). Between 2007 and 2008 he was a Postdoctoral Fellow in the Electrical and Computer Engineering Department at the University of Toronto. Between 2008 and 2013 he was a Senior Research Scientist at Telekom Innovation Laboratories (T-Labs) and TU Berlin. Currently he is an Assistant Professor in the Computer Science Department at the University of Warwick. His research interests are in the stochastic analysis of communication networks and the smart grid, resource allocation, and randomized algorithms for large systems. He is a recipient of the ACM Sigmetrics 2005 Best Student Paper Award.

# A Network Calculus Extension to EvalVid

*Emanuel Heidinger[1], Hyung-Taek Lim[2]*
*[2]BMW Group Research and Technology*
*[1]heidinge@mytum.de, [2]Hyung-taek.lim@bmw.de*

## 1. Introduction

Network Calculus has shown good value for guaranteeing worst case performance bounds in the field of industrial Ethernet networks. E.g., those networks are found in the automotive industry [19][22], aeronautics industry [11], and automation industry[13]. In those networks, the considered requirements usually range from 1ms for synchronization, over 10ms for worst case audio latency to 100ms for worst case signaling delay (cf. [23]). The performance guarantees are required to provide robustness to safety-relevant functions, such as audio announcements in the aircraft cabin, or acoustic warning signals due to distance alerts in the automotive scenario.

In addition to the briefly explained benefit of Network Calculus in safety-relevant use cases, we now provide insights on how the Network Caluculus results can be applied for video streaming use cases. Since video streaming is frequently used for entertainment, it is believed that video streaming does not have that stringent requirements in terms of reliability. However, video streaming may also play a role in scenarios, that are closer to safety-relevance than this is the case for in-car-entertainment or in-flight-entertainment, e.g.,

- video surveillance in aircraft cabins, or
- rear view camera for parking assistance.

For those scenarios, a simulative approach is certainly helpful. However, when it comes to terms, completely omitting analytically methods is not always possible.

In this letter we provide the following insights: Section 2 gives an overview on the video evaluation framework EvalVid [14], which is used to determine quality of video streaming using Network Calculus delays. Section 3 recalls on Network Calculus results in the field of automotive in-car networks as well as aeronautic cabin networks. Section 4 gives insights in how Network Calculus results can be employed on EvalVid quality metrics. Section 5 draws the conclusions and gives an outlook on ongoing work.

## 2. Video Evaluation Framework EvalVid

EvalVid [14] is a video evaluation framework by Klaue et al. that determines quality metrics such as PSNR and MOS. Figure 1 gives a simplified overview on EvalVid. The EvalVid-API allows to add network delay to video packets traversing the network. These delays usually come from real measurements via TCPDump[15] or network simulation with OPNET network simulator [16].

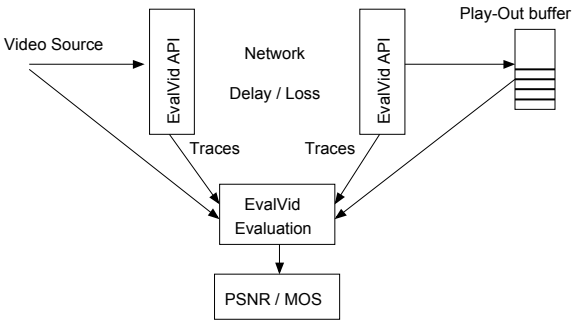



Figure 1: Schematic of EvalVid [14]

The approach given in this letter is as follows: We derive deterministic Network Calculus bounds for video streams over sample onboard networks. These Network Calculus bounds are then provided to EvalVid's API in order to dimension the play-out buffers on the receiver's side.

The given approach cannot only be used for standard switched Ethernet, but also for AVB networks [12]. However, shaping algorithms as inherent parts of AVB may result in higher worst case delays which in turn may require higher play-out buffer reserves.

## 3. Network Calculus Results

Network Calculus is a framework for determining worst case bounds in queuing networks that has emerged since the early nineties. The basic Network Calculus is primarily based on the work of Cruz [9][10]. One decade later, Le Boudec [18] added nice algebraic operations to this theory by shifting towards the (min,+)-algebra. The mentioned frameworks target at deterministic performance bounds, i.e., the determined bounds hold under any circumstance. Today, the Network Calculus community has moved towards stochastic extensions of the Network Calculus. The work of Ciucu et al. [8] summarizes the basic movements and stochastic extensions to the deterministic calculus.

This work addresses wired video transmission such that it shall suffice to determine deterministic performance bounds. We address the following scenarios and topologies: (a) In-Car-Entertainment as discussed in

[21] by Lim et al., and (b) In-Flight-Entertainment as discussed in [11] by Heidinger et al. The in-car deterministic Network Calculus results were presented in [20] and are summarized in Table 1 together with the cabin results given in [11].

| **Topology** | **Video Flow** Maximum Delay in *ms* |
|---|---|
| In-Flight Cabin Scenario | 7.5560 |
| In-Car Star-based | 1.3550 |
| In-Car Daisy chain-based | 1.7588 |
| In-Car Tree-based | 1.7581 |

Table 1: Network Calculus Deterministic Bounds

## 4. Network Calculus and Video Evaluation

As a matter of fact, there is no single frame loss in the deterministic world of Network Calculus as long as enough buffer space is available. As a result, if the receiver provides a large play-out buffer, the video transmission itself shall have no negative side effect on the video quality. However we saw scenarios in the introduction, e.g., video surveillance or rear view camera, where a smart play-out level in the lower range is desired.

The question is, how can we apply the Network Calculus results to give predictions about the perceived video quality. A very first idea that immediately comes up is delaying each frame by the maximum worst case. But in the field of video streaming, other worst case scenarios play an even greater role. We now consider high variability in the transmission time of each frame: One frame might be delivered as fast as possible and no interfering flow has negative impact on this frame. Imagine now that the succeeding frame is interfered by all other imaginable flows, i.e., it experiences the worst case. As a consequence, we have to dimension a play-out buffer that it is able to handle that variability.

In order to do so, we do not only require the worst case transmission time as provided by a standard Network Calculus analysis, we also calculate the minimal transmission time.

We propose the following approach: Values ranging from minimum delay to maximum delay are feed to the EvalVid API. These values follow a distribution that does not necessarily have to correspond to the expected distribution. In fact, our intention is to choose a distribution with a worst, negative impact in terms of variability.

As a first approach, we assume values being uniformly distributed, and employ the given approach to the Akiyo video in CIF format [17]. In our setup, the Akiyo video has the MOS value of 3.17 before transmission. Figure 2 shows the impact of shifting the length of play-out buffer towards the analytical delay bound. We see that not only the network delay has to be addressed while dimensioning the play-out buffer, but also the packetization of the video stream.

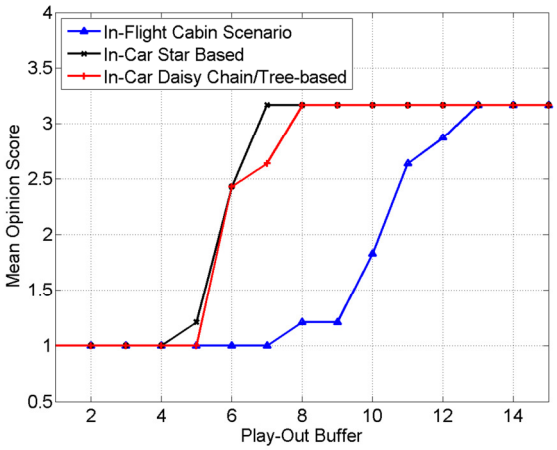



Figure 2: Mean Opinion Score

## 5. Conclusion

There is a definite trend that today's onboard networks move towards switched Ethernet not least due to the availability of low-cost switching ASICs. However, there exists no built-in mechanism in standard switched Ethernet to predict exact arrival of transmitted frames. Due to the non-preemptiveness, frames may even be delayed by frames with lower priority.

Network Calculus provides an analytical framework to capture worst case arrivals, and shows good acceptance in certification. Ongoing work targets at an improvement of the engineering process, i.e., how can we use the results from the Network Calculus analysis to dimension the onboard network and its components.

In this letter we proposed to combine standard video evaluation with analytical Network Calculus methods. The results can be directly used to give close upper bounds for the play-out buffers without degrading user's perceived quality yet achieving low latency play-outs.

**Acknowledgments:** Portions of this research were done while Emanuel Heidinger was a PhD student at the Department of Computer Science, Technische Universität München.

## References


[8] F. Ciucu and J. Schmitt. Perspectives on Network Calculus: No Free Lunch, but Still Good Value. In Proceedings of the Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications, SIGCOMM 2012, pages 311–322. ACM, 2012.

[9] R.L. Cruz. A Calculus for Network Delay, Part II, Network Analysis. IEEE Transactions on Information Theory, 37(1):132–141, 1991.

[10] R.L. Cruz. A Calculus for Network Delay, Part I, Network Elements in Isolation. IEEE Transactions on Information Theory, 37(1):114–131, 1991.

[11] E. Heidinger, N. Kammenhuber, A. Klein, and G. Carle. Network Calculus and Mixed-Integer LP Applied to a Switched Aircraft Cabin Network. In Proceedings of the 20th International Workshop on Quality of Service, IWQoS 2012, pages 1–4, 2012.

[12] IEEE 802.1 AV Bridging Task Group. Website, Accessed 2011-01-25. http://www.ieee802.org/1/pages/avbridges.html.

[13] J. Imtiaz, J. Jasperneite, and L. Han. A performance study of Ethernet Audio Video Bridging (AVB) for Industrial real-time communication. In Proceedings of the 14th Conference on Emerging Technologies & Factory Automation, ETFA 2009, pages 1–8. IEEE, 2009.

[14] Jirka Klaue, Berthold Rathke, and Adam Wolisz. Evalvid–a framework for video transmission and quality evaluation. In Computer Performance Evaluation. Modelling Techniques and Tools, pages 255–272. Springer, 2003.

[15] TCPDUMP/LIBPCAP public repository. Website, Accessed 2014-02-13. http://www.tcpdump.org/.

[16] OPNET. Application and Network Performance with OPNET. Website, Accessed 2011-07-04. http://www.opnet.com/

[17] Jirka Klaue. EvalVid - A Video Quality Evaluation Tool-set. Website, Accessed 2014-01-30. http://www2.tkn.tu-berlin.de/research/evalvid/.

[18] J.Y. Le Boudec. Network Calculus: A Theory of Deterministic Queuing Systems for the Internet. Springer-Verlag, Berlin, 2004.

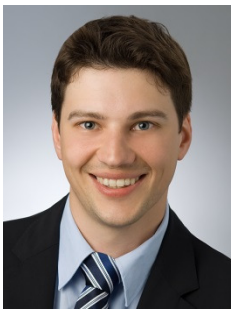
[19] H.T. Lim, D. Herrscher, L. Volker, and M.J. Waltl. IEEE 802.1 AS time synchronization in a switched Ethernet based in-car network. In Proceedings of the Conference on Vehicular Networking Conference, VNC 2011, pages 147–154. IEEE, 2011.

[20] Hyung-Taek Lim and Emanuel Heidinger. Performance bounds in in-car and aeronautic networks.

[21] Hyung-Taek Lim, Lars Volker, and Daniel Herrscher. Challenges in a future IP/ethernet-based in-car network for real-time applications. In Design Automation Conference (DAC), 2011 48th ACM/EDAC/IEEE, pages 7–12. IEEE, 2011.

[22] Martin Manderscheid and Falk Langer. Network calculus for the validation of automotive ethernet in-vehicle network configurations. In Cyber-Enabled Distributed Computing and Knowledge Discovery (CyberC), 2011 International Conference on, pages 206–211. IEEE, 2011.

[23] Radio Technical Commission for Aeronautics. DO-214/SC-164: Audio Systems Characteristics and Minimum Operational Performance Standards for Aircraft Audio Systems and Equipment Systems and Equipment. 1993.

**Emanuel Heidinger** received his diploma (equiv. to MSc) and doctoral degree from TU München, all in computer science, in 2006, and 2013 respectively. During his studies, he worked in the PhD program at EADS Innovation works. His research interests lie in the field of Network Calculus, Discrete Event Simulation and Switched Queuing Networks for Aeronautic Cabin Networks.

**Hyung-Taek Lim** received his diploma (equiv. to MSc) from TU Berlin in computer engineering in 2009. During his study, he worked in the PhD program at BMW Group Research and Technology for the topic Ethernet AVB. His research interests lie in the field of discrete event simulation and Quality-of-Service in wired and wireless communication in an in-car network.

# Performance Analysis for Wireless Multimedia Sensor Networks Based on Stochastic Network Calculus

*Nao Wang, Gaocai Wang and Ying Peng*
*School of Computer and Electronics and Information, Guangxi University, 530004, China*
*gcwang@gxu.edu.cn*

## 1. Introduction

In recent years, wireless multimedia sensor networks (WMSNs) are able to ubiquitously obtain multimedia content such as video, audio streams, images, and scalar sensor data from the environment because of the availability of inexpensive hardware. In a WMSN system, wireless channels have time-varying location dependent characteristics and different wireless sensors experience different channel conditions at a given time. These stochastic wireless channels are affected inherently by sensors' shadowing, interference and path losses due to changing environments and possible mobility. In a WMSN, on one hand, the multimedia content should be delivered with predefined levels of Quality of Service (QoS) under resource and performance constraints such as bandwidth, energy and delay. On the other hand, the traffic generated by sensors is highly burstiness, and various kinds of data input and output sensors by occupying stochastic wireless channel to satisfy different performance demands and guarantee. Some uncertain factors existing in stochastic wireless channel become challenges for performance guarantee in a WMSN. Traditional deterministically performance guarantee methods based on queuing theory and effective bandwidth are relatively weak in making performance analysis of stochastic networks and lead to low utilization of network resource.

Deterministic network calculus (DNC) has already become an elegant and effective theory for worst case performance analysis in network dimensioning by two tools: arrival curve and service curve, since it was proposed by Cruz and Chang. On the basis of DNC, some related works have been proposed to derive deterministically performance guarantee (such as deterministic delay and backlog bounds) in some traditional and fashionable networks. As an improved version of DNC, a novel theory called stochastic network calculus (SNC) in which extends DNC two curves with probability meanings, has been proposed systematically. One can provide stochastic performance guarantee in network dimensioning and get higher network utilization at the expense of minor performance degradation under SNC framework.

We mainly focus on performance based on popular SNC in wireless multimedia and data networks. More especially, sensed data may originate from various types of events that have different levels of importance. Consequently, the content and nature of the sensed data also varies. As an example that highlights the need for network level QoS, consider the task of bandwidth assignment for multimedia mobile medical calls, which include patients' sensing data, voice, pictures and video data. Unlike the typical source-to-sink multi-hop communication used by classical sensor networks, the WMSN architecture maybe use 3G/4G cellular system in which individual nodes forward the sensed data to a cellular phone or a specialized information collecting entity. Different priorities are assigned to video data originating from sensors on ambulances, audio traffic from elderly people, and images returned by sensors placed on the body. In order to achieve this, parameters like hand-off dropping rate, latency tolerance and desired amount of wireless effective bandwidth are taken into consideration.

In this paper, we mainly focus on some performance metrics of WMSNs. We firstly limit traffic arrival process by using exponentially bounded burstiness (EBB) model of SNC and establish stochastic arrival curve. Then, we give a new suitable stochastic service curve which can provide better QoS performance and obtain stochastic service curve. We give delay bounds of single node and the end-to-end delay bounds.

## 2. Network model

In Figure 1, we introduce reference architecture for WMSNs, where three sensor networks with different characteristics are shown, possibly deployed in different physical locations.

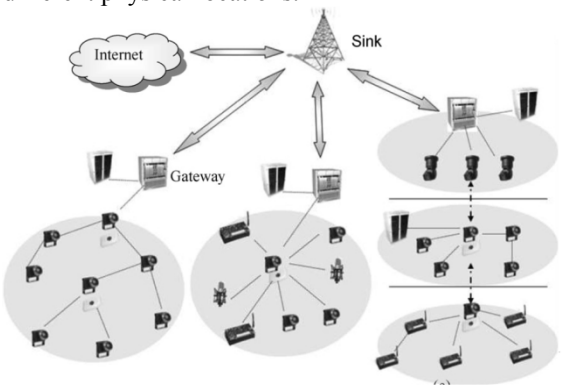



Figure 1 A network architecture for WMSN

We define edge nodes (EN) which sense information and data from environment. Sense nodes (SN) can sense information and relay data for edge nodes. Sink node deliver data to control center.

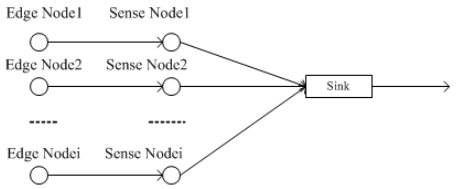



Figure 2 wireless sensor network traffic model

We give some traffic process and scheduling rules for Figure 1 and 2 models.

(1) Edge nodes send sensing data to sense nodes by Posisson process with parameter $r$.

(2) There exist arrival multi-flows in a sense node. $R(t)$ denotes arrival data and $x$ is instantaneous burstiness.

(3) Sense nodes buffer arrival data.

(4) Sense nodes provide FCFS service for arrival flows. The service capability is $C$, and satisfies service curve $\beta(t)$. Departure curve is $R^*(t)$.

(5) $R^*(t)$ can be regarded as arrival flows for next sense node.

**Stochastic arrival curve**

We give a suitable stochastic arrival curve for sensed data in a WMSN. In SNC, EBB traffic model is a typical stochastic arrival traffic model, it has already been proved that many types of input traffic processes such as Poisson, Bernoulli and exponential ON/OFF have EBB. So, given the bursty nature of voice and video data, queueing disciplines are needed that can accommodate sudden peaks, we abstract arrival traffic process of all flows and adopt EBB model bound. We have the following Theorem 1.

**Theorem 1:** Stochastic Arrival Curve: Suppose all sensed data arrive at sensor node according to M/M/1 stochastic process and arrival curve satisfies $R(t)$, $R(t)$ and $\alpha(t)$ denote arrival data and arrival curve, respectively. $\alpha(t)=rt + x$, $r$ is arrival rate and $x$ is instantaneous burstiness. $f(x)=Me^{-xt}$ is violation probability. For $0\le s\le t$, if $x\ge 0$, we have

$$\Pr\{R(s,t) - rt > x\} \le f(x) \tag{1}$$

**Stochastic service curve**

The performance of WMSNs depends on different stochastic service curve.

**Theorem 2:** Stochastic Service Curve: Suppose flows have arrival process $A(t)$ and departure process $R^*(t)$, a sensor node said to provide a stochastic service curve $\beta(t)=C(t-T)$ with violation probability $g(x)=1-Me^{-xt}$ for flows if for all $t\ge 0$ and $x \ge 0$.

$$\Pr\{R^*(t) - R(t) \otimes (rt - x) \ge 0\} \le g(x) \tag{2}$$

Where, $T$ denotes the delay of traffic in a sensor node. For a single sense node, the sense node will schedules arrival flow according FCFS if there are multiple arrival flows. Suppose $N$ ($N$=1,2,…) arrival flows, for any arrival flow $i$, if the $i$ flow has priority weight $u_i$, the flow gets service $S_i(s, t)$ in time interval $(s, t)$

$$\frac{S_i(s,t)}{S_j(s,t)} \ge \frac{u_i}{u_j}, i, j = 1,2,3 \cdots N \tag{3}$$

And the flow gets minimal service rate $R_i$ is

$$R_i = (u_i / \sum_{i \in E} u_i) C \tag{4}$$

Where, $E$ is the set of flow for waiting service in time interval $(s, t)$.

## 3. Performance evaluation

In this section, we give some performance metrics for WMSNs according to above results.

### 3.1 A delay bound for a single node

In buffer of a sense node, data will be forwarded to the next sense node according first come first serve mechanism. On the other hand, we suppose the sense node serves data by using token bucket model $(r, x)$. So, we can compute a delay bound of single node.

$$\begin{aligned} f \otimes g(x) &\le \Pr\{D(t) > h(\alpha + x, \beta)\} \\ &= \Pr\{\sup_{t\ge 0}\{\inf\{D(t) \ge 0 : \alpha(t) + x \le \beta(t)\}\}\} \\ &= \Pr\{\sup_{t\ge 0}\{\inf\{D(t) \ge 0 : rt + x \le C(t - \frac{x}{C})\}\}\} \\ &= \Pr\{\sup_{t\ge 0}\{\inf\{D(t) \ge 0 : (C - r)t \ge x + x\}\}\} \\ &= \Pr\{\sup_{t\ge 0}\{\inf\{D(t) \ge 0 : t \ge \frac{2x}{C - r}\}\}\} \\ &= \Pr\{\sup_{t\ge 0}\{D(t) \ge 0 : t \ge \frac{2x}{C - r}\}\} \end{aligned} \tag{5}$$

According to the delay of single node, we can deduce the end-to-end delay of a path in a WMSN. A weight $u_i$ is set to every arrival flow because of a limited service capability C, when the flows arrive at the sense node.

$$\Pr\{D(t) > h(\alpha + x, \beta)\} = \Pr\{\sup_{t\ge 0}\{D(t) \ge 0 : t \ge \frac{2x}{R_i - r}\}\} \tag{6}$$

### 3.2 The end-to-end delay bound

Next, we will compute the end-to-end delay bound by using convolution formula.

**Theorem 3:** Suppose a path of a WMSN has $n$ ($n$=1,2,…,$N$) sensor nodes. If every sensor node provides service curve $\beta^1(t)$, $\beta^2(t)$,…, $\beta^n(t)$, the end-to-end service curve $\beta_{net}(t)$ can be expressed by

$$\beta_{net}(t) = \beta^1 \otimes \beta^2 \otimes \cdots \otimes \beta^n(t) \tag{7}$$

An illustration for the end-to-end delay is given in Figure 3.

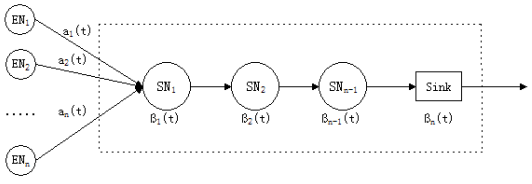



Figure 3 An illustration for the end-to-end delay

**Theorem 4**: Suppose a path of a WMSN has $n$ ($n$=1,2,…,$N$) sensor nodes, and the flows transmit to sense node from $h$ edge nodes with rate $r$. The service capability is $C$, and $hr \le C$. Then, the end-to-end delay $D_{E2E}$ satisfies the following

$$D_{E2E} \le \frac{n(n+3)}{2x(R_i - hr)} \log(\frac{M_n^{\frac{n(n+3)}{2}}}{\varepsilon}) \qquad (8)$$

Here, ε is violation probability of the path.

### 3.3 Simulation results

We give some numerical results based on above theoretical results. Figure 4 shows that the delay of single node changes with the increase of the priority weight $u_i$. Figure 5 shows the end-to-end delay bound under different node sizes.

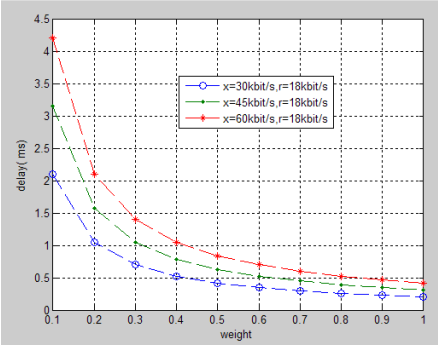



Figure 4 Different delay bounds for single node

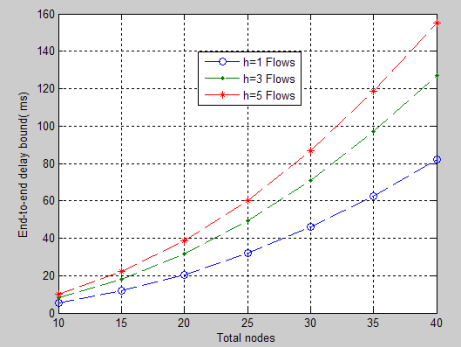



Figure 5 The end-to-end delay bounds

### 4. Conclusion

This paper mainly focuses on some performance metrics of WMSNs. We firstly limit traffic arrival process by using exponentially bounded burstiness (EBB) model of SNC and establish stochastic arrival curve. Then, we give a new suitable stochastic service curve which can provide better QoS performance and obtain stochastic service curve. We give delay bounds of single node and the end-to-end delay bounds.

**Acknowledgment:** This research is supported in part by the National Natural Science Foundation of China under Grant Nos. 61262003, 61063045 and 61103245, in part by the Natural Science Foundation of Guangxi Province under Grant No.2010GXNSFC013013

### References

[24] I. Akyildiz, T. Melodia, K. Chowdhury, A survey on wireless multimedia sensor networks, Computer Networks 51 (4) (2007) 921–960.

[25] Y. Liu, Ch. K. Tham, Y.M. Jiang, A calculus for stochastic QoS analysis. Performance Evaluation, 64(6), pp.547-572, 2008.

[26] J. Debardelaben, Multimedia sensor networks for ISR applications, in: 37th Conference on Signals, Systems and Computers, vol. 2, 2003, pp. 2009–2012.

[27] I. Akyildiz, T. Melodia, K. Chowdhury, Wireless multimedia sensor networks: applications and testbeds, Proceedings of the IEEE 96 (10) (2008) 1588–1605.

[28] J. Boudec and P. Thiran. Network calculus: A theory of deterministic queuing systems for the internet. Springer, LNCS, 2008.

[29] R. L. Cruz. A Calculus for Network Delay, Part I: Network Elements in Isolation. IEEE Transactions on Information Theory, 36(2): 114-131. 1991

[30] R.L.Cruz.A Calculus for Network Delay, Part II: Network Analysis. IEEE Transactions on Information Theory, 37(1):132-141. 1991.

[31] Florin Ciucu, Almut Burchard, Jorg Liebeherr. A Network Service Curve Approach for the Stochastic Analysis of Networks. In: Proceedings of the ACM International Conference on Measurement and Modeling of Computer Systems(SIGMETRICS'2005), Banff, Alberta, Canada, 2005. 279-290.

[32] Florin Ciucu, Almut Burchard, Jorg Liebeherr. Scaling Properties of Statistical End-to-End Bounds in the Network Calculus. IEEE Transactions on Information Theory, 2006, 52(6): 2300-2312.

**Nao Wang** received her B.E., M.E., Degree from Central South University, China. Currently, she is a lecture in Guangxi University. Her research interests include networks optimization.

**Gaocai Wang** received his B.E., M.E., and Ph.D. degrees in computer science from Central South University, China, in 2004. Afterward, he went to Tsinghua University, University of Toledo and Texas Southern University as a Postdoctor. Currently, he is a professor in the Department of Computer Science, Guangxi University. His research interests include computer networks, routing algorithms, performance evaluation. He has published more than 60 journal and conference papers in these areas.

**Ying Peng** received her B.E., M.E., Degree from Huazhong Normal University, China. Currently, she is a Ph.D candidate and lecture in Guangxi University. Her research interests include energy consumption optimization.

# A General Stochastic Framework for Low-Cost Design of Multimedia SoCs

*Balaji Raman[1], Ayoub Nouri[1], Deepak Gangadharan[2], Marius Bozga[1], Ananda Basu[1], Mayur Maheshwari[1], Axel Legay[3], Saddek Bensalem[1], and Samarjit Chakraborty[4]*
*VERIMAG (France) [1], Technical University of Denmark[2], INRIA Rennes (France) [3], Technical University of Munich (Germany) [4]*
*Balaji_Raman@mentor.com*

## 1. Introduction

An apt choice of a modeling framework is essential to design resource-constrained System-on-Chips (SoCs) in multimedia systems (such as video/audio players, etc.). Such a modeling framework must exploit the inherent stochastic nature of the multimedia applications to design low-cost systems. The uncertainty in such systems is due to high variability present in the input multimedia stream, in terms of number and complexity of items that arrive per unit time to the system. Consequently, the variability is exhibited both in arrival and in processing time.

Many of the existing analytical frameworks proposed so far are either incompatible or inflexible to capture the key characteristics of the system being modeled:

- Worst-case execution time modeling and analysis framework cannot capture behavior of soft real-time systems, leading to pessimistic designs with exorbitant use of hardware resources (such as buffer size).
- Average-case execution time analysis framework cannot provide QoS guarantees and are thus hardly trustworthy.

To address the above limitations, we sought a framework for analyzing multimedia systems that account for the stochastic nature of the streaming application. We need a model characterizing input stream and execution of the multimedia stream as stochastic; instead of capturing event arrivals and executions with worst or average cases.

Recently, stochastic network calculus based approaches have been proposed for performance analysis of multimedia systems. These approaches, however, used the probabilistic calculus only partially: the input stream objects of a multimedia stream (e.g., frames) and their execution time are assumed to be deterministic. Another study used probabilistic real-time calculus to analyze hard real-time systems. This later research, however, did not focus on any specific application domain. Nonetheless, if stochastic network calculus is fully adopted for system-level design, then design of multimedia embedded platforms can benefit, too. The analysis can provide probabilistic bounds on the output quality of the system. The worst-case and average case analysis of the system is special-case scenarios of the analytical framework.

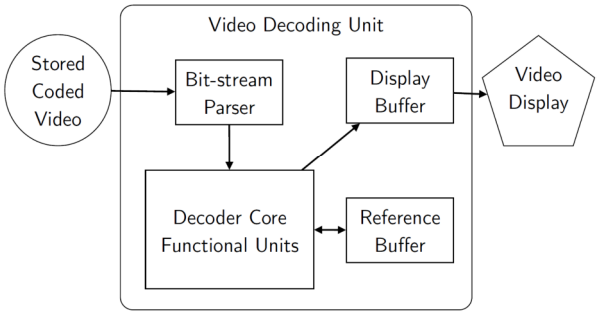



**Figure 6: Display buffer in a generic video decoder of a SoC.**

## 2. Case study: Video Decoder

We analyze an abstraction of a video decoder SoC (shown in Figure 1) with the analytical approach. In the abstract SoC model, the input video stored is fed to the input buffer in terms of stream objects such as macroblocks, frames, etc. A pipeline of functional units process the input stream. Processed items are temporarily stored in the output buffer before their display.

Now, we will state the problem addressed in this paper. We assume that the multimedia SoC contains a single processing unit and two buffers.

**Problem Statement:** To estimate the probability that the buffer underflow (U(t)) is less than two consecutive frames in 30 frames. We are given the following:

- a set of video clips of certain bit-rate (r) and resolution,
- maximum frequency of the processing unit of the multimedia SoC (f),
- consumption rate of the output device (c),
- start-up values for the initial delay (d), input-buffer size (b), and playout buffer size (B).

### 2.1. Analytical Model of the Multimedia SoC

In this subsection, first, we give an overview of our analytical approach for this case study. Second, we present formulation for evaluating the QoS constraints.

We can estimate the maximum size of the output buffer using our stochastic analytical framework. Previous works that estimate the output buffer size using deterministic real-time calculus proceed as follows. For all given video clips, Maxiaguine et al. construct upper bounds on the number of items that arrive to the input buffer and that execute in the processor. These two bounds together yield another upper bound on the number of items that arrive to the display buffer. Thus, given a rate at which items are consumed from the output buffer, Maxiaguine et al. estimate the maximum buffer size required.

We, too, compute deterministic upper bounds on arrival, execution, and output, however, only for a sub-set of given video clips; the remaining video clips can violate the deterministic bounds. For example, the number of items arriving to the output buffer over a time interval, for a certain clip, can be larger than the deterministic output bound. Now, we explain how to quantify this deviation from the deterministic bound in a stochastic setting (as we are using probabilistic network calculus).

Assume that the stream objects arrival and execution are stochastic—an apt characterization of multimedia streams. So, what is the probability that the number of stream objects that arrive to the output buffer over a time interval is larger than the deterministic output bound?

First, we estimate the maximum probability of violating the deterministic bounds for the arrival and execution; then, using these two bounds, we estimate it for the output. Using this stochastic bound on the output, given the constant consumption rate, we can compute the probabilistic distribution of the buffer size.

We specify tolerable loss in video quality as: less than two consecutive frame loss within 30 frames, less than 17 aggregate frame loss within 100 frames, etc. Previous work models the loss of stream objects as buffer underflows, which occurs whenever the display device finds insufficient items to read from the output buffer. Raman et al. show that the amount of buffer underflow can be controlled using an application parameter, namely, the initial playout delay, the delay after which the video starts to display.

We, too, tune the initial delay parameter to restrict the maximum buffer underflow, albeit, in a stochastic setting. We obtain probability values for a QoS property to hold for a range of initial delays. The buffer size corresponding to each such delay can be estimated. Therefore, the system designer can choose an initial delay based on his resource and quality constraints.
In what follows, we formulate the probability distribution of the output buffer size using the stochastic real-time calculus model.

In practice, the designer is typically given a set of input clips to design the SoC with given display constraints. So,in our problem setting, the designer can construct an upper and lower bound using synthetic traces (which are obtained from actual traces) and use the actual traces to construct the bounding functions. Now, we explain the construction of these synthetic traces.

Let the given set of input video clips be partitioned into two sets, $S_A$ and $S_B$, based on the designer's requirement that all clips in $S_B$ must be processed with no loss in video quality. The deterministic upper bound on the arrival and the output, introduced in the previous section, are constructed using clips in $S_B$. Now, we discuss how to synthetically generate clips in case we do not have a set $S_B$.

The information we have about the clips in set SA are the number of bits and number of cycles corresponding to each macroblock. For certain macroblocks, we modify the number of bits and cycles, assuming we are given lower bounds on these parameters. Any number of bits lower than the actual bound in the input traces are replaced with a value of the lower bound. Thus we obtain synthetic traces forming clips in set SB. Now we explain how we estimate stochastic bounds using the actual clips (i.e. clips from set SA) and synthetic traces (or if available clips from set SB).

The upper and lower bounds on the arrival given from the formula in the previous subsection are calculated for the synthetic traces. That is, from the modified inverse $\phi$ function, we compute $\alpha_l(\Delta)$ and $\alpha_u(\Delta)$. This leads to the definition of the stochastic arrival curve.

Assume the output arrival curve is an arrival process to the playout buffer. The probability distribution at the playout buffer can be computed using the bounding function h.

$$P\,(U(t) > a) \leq h(a - (\alpha_* \oslash ct))(0),\ t \geq 0$$

In the above equation $\alpha_*$ is the output arrival function given by $\alpha_u \oslash \beta_u\,(t)$.

### 3. Results

This section sketches QoS probabilities estimated from the stochastic network calculus approach presented in the previous section.

We implemented the analytical framework described in MATLAB. The experiments were conducted for a low-

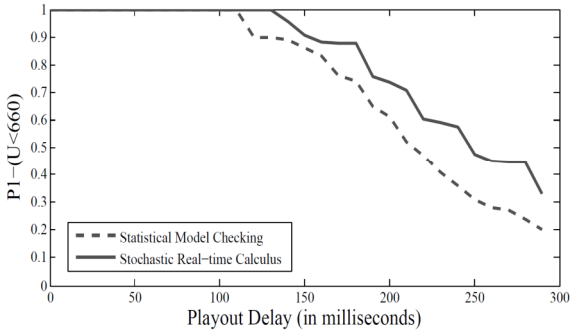



**Figure 7: Probabilistic bounds for mobile.m2v.**

bit rate and low resolution clips (352 x 240) obtained from an open source. The bit-rate of the input video is 1:5 Mbits per second and the frame output rate is 30fps. We used an MPEG2 implementation optimized for speed. The MPEG2 source was annotated to get the number of bits corresponding to each compressed macroblock. The execution cycles for each macroblock is obtained from the software simulator SimpleScalar. Recapitulate that the number of bits and execution cycles per macroblock are inputs to the analytical framework. We chose the video files *cact.m2v*, *mobile.m2v*, and *tennis.m2v* for our experiments.

To construct the synthetic clips we set the lower bound for bits (*e.g* to 60) and the lower bound for execution cycles (*e.g* to 9000). Then from the actual trace containing the number of bits and execution cycles per macroblocks, synthetic traces are obtained; any value below the lower bound is modified to the lower bound.

Figures show the probability that the buffer underflow is greater than two consecutive frames over any time interval. Figure 2 also show the probability estimates from the statistical model checking framework. Following are the main observations:

- Increase in playout delay decreases the amount of buffer underflow, so, probability that the buffer underflow is more than two consecutive frames decreases.
- The estimates from stochastic real-time calculus upper bound the statistical model checking results as the analytical framework captures the worst-case behavior.

## 4. Conclusion

This paper proposed an application of stochastic network calculus for the design of multimedia system-on-chips. There are two main advantages of choosing this framework (i.e. stochastic network calculus) for modeling multimedia systems: (1) for characterizing multimedia streams as stochastic and thus leading to efficient SoCs (in terms of hardware resources, etc.,), and (2) for providing QoS guarantees. We illustrated these advantages with applying the network calculus framework for designing video decoders. We also proved the accuracy of our methodology using a simulation technique whose designs can be expressed in formal semantics.

## References

[1] B. Raman, A. Nouri, D. Gangadharan, M. Bozga, A. Basu, M. Maheshwari, J. Milan, A. Legay, S. Bensalem, and S. Chakraborty. A General Stochastic Framework for Low-Cost Design of Multimedia SoCs. Technical Report TR-2012-7, Verimag Research Report, 2012 (also published in IEEE SAMOS XIII).

[2] N. H. Zamora, X. Hu, and R. Marculescu. System-level performance/ power analysis for platform-based design of multimedia applications, ACM Transactions on Design Automation of Electronic Systems, (TODAES), 12(1):1–29, January 2007.

[3] Y. Jiang and Y. Liu. Stochastic Network Calculus. Springer, 2008.

[4] L. Santinelli and L. Cucu-Grosjean. Toward probabilistic real-time calculus. ACM SIGBED Review, 8(1):54–61, March 2011.

[5] B. Raman, G. Quintin, W. T. Ooi, D. Gangadharan, J. Milan, and S. Chakraborty. On buffering with stochastic guarantees in resource constrained media players. In Proc. of the IEEE/ACM/IFIP International Conference on Hardware/Software Codesign and System Synthesis (CODES+ISSS), pages 169–178, September 2011.

[6] A. Maxiaguine, S. Kunzli, S. Chakraborty, and L. Thiele. Rate analysis for streaming applications with on-chip buffer constraints. In Proc. Of the Asia and South Pacific Design Automation Conference (ASP-DAC), pages 131–136, January 2004.

[7] D. Wijesekera and J. Srivastava. Quality of Service (QoS) Metrics for Continuous Media. Multimedia Tools and Applications, 3(2):127–166, July 1996.

# Copula Analysis for Stochastic Network Calculus

*Kui Wu, Fang Dong, Venkatesh Srinivasan*
*Computer Science Department*
*University of Victoria*
*BC, Canada V8W 3P6*
*{wkui, fdong, venkat}@uvic.ca*

## 1. Introduction

Since its introduction in early 1990s [1], network calculus has been widely adopted to analyze complex queueing systems, such as multimedia networks, where the Markovian property of arrivals generally does not hold and thus traditional queueing theory is hard to apply. Network calculus has evolved along two tracks – deterministic [2, 3] and stochastic [4]. In the domain of stochastic network calculus (SNC), due to some difficulties specific to stochastic networks, it is only in recent years that critical network calculus properties, such as concatenation property [4, 5] have been proved.

The practical use of SNC has been questioned due to the lingering problem in deriving tight stochastic performance bounds [6, 7]. In [7], the problem has been *implicitly* raised, and caution has been advised if model transform is used to derive performance bounds. In [6], the problem has been analyzed in more detail and has been given an *explicit* term, called the quasi-deterministic problem.

While substantial efforts have been devoted to improving the bounds [6, 8], the problem is tackled only for special types of traffic and service models, using probability inequalities, *e.g.*, Chernoff bounds and martingale inequalities. In general the independence assumption is required, *e.g.*, the independence of the traffic arrivals and the independence between the arrivals and the service.

This paper points out the potential of copula theory in SNC. With copula analysis and a simple case study, we clearly show the region where copulas can be helpful and the best possible bound that SNC can achieve.

## 2. Background

### A. Stochastic network calculus

We introduce the notation and key concepts of stochastic network calculus [4, 9, 10]. We assume that all arrival curves and service curves are non-negative and wide-sense increasing functions. Conventionally, $A(t)$ and $A^*(t)$ are used to denote the *cumulative* traffic that arrives and departs in time interval $(0,t]$, respectively, and $S(t)$ is used to denote the cumulative amount of service provided by the system in time interval $(0,t]$. For any $0 \le s \le t$, let

$A(s,t) \equiv A(t) - A(s)$, $A^*(s,t) \equiv A^*(t) - A^*(s)$ and $S(s,t) \equiv S(t) - S(s)$. By default, $A(0) = A^*(0) = S(0) = 0$.

We denote by $\mathrm{F}$ the set of non-negative wide-sense increasing functions, *i.e.*,

$$\mathrm{F} = \{f(\cdot) : \forall 0 \le x \le y, 0 \le f(x) \le f(y)\},$$

and by $\overline{\mathrm{F}}$ the set of non-negative wide-sense decreasing functions, *i.e.*,

$$\overline{\mathrm{F}} = \{f(\cdot) : \forall 0 \le x \le y, 0 \le f(y) \le f(x)\}.$$

For any random variable $X$, its distribution function, denoted by

$$F_X(x) \equiv Prob\{X \le x\},$$

belongs to $\mathrm{F}$, and its complementary distribution function, denoted by

$$\overline{F}_X(x) \equiv Prob\{X > x\},$$

belongs to $\overline{\mathrm{F}}$.

The $(\min,+)$ *convolution* of functions $f$ and $g$ is useful for SNC, and is defined under the $(\min,+)$ algebra [1-3]:

$$(f \otimes g)(t) \equiv \inf_{0 \le s \le t} \{f(s) + g(t-s)\}. \qquad (1)$$

Stochastic traffic arrival curve and stochastic service curve are core concepts in stochastic network calculus, with the former used for traffic modeling and the latter used for service modeling. In the literature, there are different definitions of stochastic arrival curve and stochastic service curve [4, 9]. We use simple models in this paper for illustration purpose.

*Definition 1*: **The** *t.a.c* **model** [4]: A flow $A(t)$ is said to have a *traffic-amount-centric* (*t.a.c.*) stochastic arrival curve $\alpha \in \mathrm{F}$ with bounding function $f \in \overline{\mathrm{F}}$, denoted by

$$A \sim_{ta} < f, a >,$$

if for all $t \ge s \ge 0$ and all $x \ge 0$, it holds

$$Prob\{A(s,t) - \alpha(t-s) > x\} \le f(x). \qquad (2)$$

Note that the above model is actually quite general and covers some broadly-used models. For example, the stochastically bounded burstiness (BSS) model [11] is a special case of the *t.a.c* model by setting $\alpha(t-s)=\rho\cdot(t-s)$. Following the same notation, when the traffic arrival $A(t)$ follows the BSS model with upper rate of $\rho$ and bounding function of $f$, we denote it as $A \sim_{SBB} < f, \rho >$.

For service models, the following model is normally adopted:

*Definition 2*: **The** *w.s.* **model** [4]: A server is said to provide a flow $A(t)$ with a *weak stochastic* (*w.s.*) service curve $\beta \in \mathrm{F}$ with bounding function $g \in \bar{\mathrm{F}}$, denoted by

$$S \sim_{ws} < g, \beta >,$$

if for all $t \geq 0$ and all $x \geq 0$, it holds

$$Prob\{A \otimes \beta(t) - A^*(t) > x\} \leq g(x). \qquad (3)$$

**B. Copulas**

A copula is a function that links univariate marginals to their multivariate distribution. In practice, it is normally easy to find the univariate marginals, but it is hard to obtain the multivariate distribution. With a copula, we can capture the joint distribution of random variables once their marginal distributions are obtained.

*Definition 3*: An *N*-dimensional copula is a function $C$ having the following properties [12]:

1) Dom $C = I^N = [0,1]^N$;
2) If any argument of $C$ is zero, then $C = 0$. In addition, $C$ is nondecreasing in each argument[1];
3) $C$ has margins $C_n$ which satisfy $C_n(u) = C(1,\ldots,1,u,1,\ldots,1) = u$ for all $u$ in $I$.

Sklar's theorem [12] is the core of the copula theory, because it provides a way to analyze the dependence structure of multivariate distributions. It also builds a link between the joint distribution and the marginal distributions.

*Theorem 1*: **(Sklar's theorem)** Let $F$ be an *N*-dimensional distribution function with continuous margins $F_1, F_2 \ldots, F_N$. Then $F$ has a unique copula representation:

$$F(x_1, x_2, \ldots, x_N) = C(F_1(x_1), F_2(x_2), \ldots, F_N(x_N)) \quad (4)$$

Sklar theorem implies that we can construct a copula if the joint distribution and the marginal distributions are given. Similarly, we can calculate the joint distribution if we know the copula and the marginal distributions. This seems to suggest that the joint distribution and copula are equivalent. The following theorem, however, shows a special feature of copulas that the joint distribution function does not possess.

*Theorem 2*: **The invariant property of copulas** [12]**:** Let $X$ and $Y$ be continuous random variables with copula $C_{XY}$. If $\alpha$ and $\beta$ are strictly increasing on the range of $X$ and the range of $Y$, respectively, then $C_{\alpha(X)\beta(Y)} = C_{XY}$. In other words, $C_{XY}$ is invariant under strictly increasing transformations of $X$ and $Y$.

**3. Stochastic network calculus with copulas**

**A. Basic Lemmas**

For simplicity, we denote $[x]_1 \equiv \min\{x,1\}$ and $[x]^+ \equiv \max\{x,0\}$ in the following.

In stochastic network calculus, we are often interested in the complementary distribution function of $Z = X + Y$, *i.e.*, $Prob\{Z > z\}$. The following two lemmas have been widely used in the derivation of stochastic bounds.

*Lemma 1*: **General case** [4]: For the sum of two random variables $X$ and $Y$, $Z = X + Y$, no matter whether $X$ and $Y$ are independent or not, $\bar{F}_Z(z) \leq \bar{F}_X \otimes \bar{F}_Y(z)$.

*Lemma 2*: **Independent case**: Assume that non-negative random variables $X$ and $Y$ are independent and $\bar{F}_X(x) \leq f(x)$ and $\bar{F}_X(x) \leq g(x)$, where $f, g \in \bar{\mathrm{F}}$. Then, for all $x \geq 0$, $Prob\{X+Y > x\} \leq 1-(\bar{f} * \bar{g})(x)$, where $\bar{f}(x) = 1-[f(x)]_1, \bar{g}(x) = 1-[g(x)]_1$, and $*$ is the Stieltjes convolution operation.

The following lemma, from copula analysis, is useful for SNC:

*Lemma 3*: **Copula case** [12]: Let $Z$ be the sum of two random variables $X$ and $Y$. Then

$$\hat{\bar{F}}_Z(z) \geq \bar{F}_Z(z) \geq \check{\bar{F}}_Z(z), \qquad (5)$$

where

$$\hat{\bar{F}}_Z(z) = 1 - \sup_{x+y=z} \{W(F_X(x), F_Y(y))\}, \qquad (6)$$

$$\check{\bar{F}}_Z(z) = 1 - \inf_{x+y=z} \{\bar{W}(F_X(x), F_Y(y))\}, \qquad (7)$$

$$W(u,v) = [u+v-1]^+, \qquad (8)$$

[1] We slightly modify the definition for ease of understanding. Refer to [12] for a more mathematically rigorous definition.

$$\bar{W}(u,v) = [u+v]_1. \quad (9)$$

**B. Copula analysis**

Markov modulated processes have been extensively used for representing multimedia traffic [13]. It has been shown that Markov modulated traffic could be captured with the stochastically bounded burstiness (SBB) model. Assume that we are given two Markov modulated processes $A_1 \sim_{SBB} < f_1, \rho_1 >$ and $A_2 \sim_{SBB} < f_2, \rho_2 >$ As a concrete example, we assume that both bounding functions, $f_1$ and $f_2$, have the exponential form [11] with mean values of $\alpha$ and $\beta$, respectively. We are interested in modeling the superposition of $A_1$ and $A_2$, $A = A_1 + A_2$.

Let $X$ and $Y$ be exponentially distributed random variables with mean values of $\alpha$ and $\beta$, respectively, and let $Z = X + Y$. With Lemma 1, we have the following bound, denoted as the *general bound* since it holds for any $X$ and $Y$:

$$\bar{F}_Z(z) = \begin{cases} 1, & z < \theta \\ e^{-\frac{z-\theta}{\alpha+\beta}}, & z \geq \theta \end{cases} \quad (10)$$

where $\theta = (\alpha+\beta)\ln(\alpha+\beta) - \alpha\ln(\alpha) - \beta\ln(\beta)$.

If we know that $X$ and $Y$ are independent, we have the following bound, denoted as the *independent bound*:

$$\bar{F}_Z(z) = \begin{cases} [(1+\frac{z}{\gamma})e^{-\frac{z}{\gamma}}]_1, & \alpha = \beta = \gamma \\ [\frac{\alpha e^{-\frac{z}{\alpha}} - \beta e^{-\frac{z}{\beta}}}{\alpha - \beta}]_1, & \alpha \neq \beta \end{cases} \quad (11)$$

Based on Lemma 3, we have the following copula upper and lower bounds, whose proof is omitted to save space.

*Theorem 3*: Let $X$ and $Y$ be exponentially distributed random variables with means $\alpha$ and $\beta$, respectively. Let $\hat{\bar{F}}_z$ and $\check{\bar{F}}_z$ be as in Lemma 3. Then

$$\hat{\bar{F}}_z = \begin{cases} 1, & z < \theta \\ e^{-\frac{z-\theta}{\alpha+\beta}}, & z \geq \theta \end{cases} \quad (12)$$

and

$$\check{\bar{F}}_z = \begin{cases} 1, & z < 0 \\ e^{-\frac{z}{\max(\alpha,\beta)}}, & z \geq 0 \end{cases} \quad (13)$$

where $\theta = (\alpha+\beta)\ln(\alpha+\beta) - \alpha\ln(\alpha) - \beta\ln(\beta)$.

It is easy to show that the superposition of $A_1 \sim_{SBB} < f_1, \rho_1 >$ and $A_2 \sim_{SBB} < f_2, \rho_2 >$ follows $A \sim_{SBB} < g, \rho_1 + \rho_2 >$, where $g$ can be calculated with Equations (10) (general bound), (11) (independent bound), (12) (copula upper bound), and (13) (copula lower bound), respectively. The copula lower bound indicates the tightest possible bound that we can obtain with SNC when the upper rate is $\rho_1 + \rho_2$.

Figs. 1 and 2 show two numerical examples. We have two interesting observations from the figures:

- The general bound is the same as the upper copula bound, indicating that the general bound is actually the worst bound that we can get with SNC.
- There is a clear gap between the independent bound and the lower copula bound. This gap implies that if the dependence of random variables is unknown, independent case analysis does not always lead to the best bound. There is much room for us to explore how to improve stochastic bounds with copulas.

**4. Conclusion and future work**

With a simple case study, we illustrated the benefit of applying copula theory in SNC for tighter performance bounds. Our preliminary analysis, while by no means comprehensive, sheds light on several important issues in SNC, including the region where we can take advantage of dependency of random processes and the tightest bound that SNC can possibly achieve. This knowledge is fundamental for designing better scheduling and multiplexing strategies for multimedia systems.

Our future work includes constructing and validating copula models for network traffic with real-world experiments, and investigating the achievability of the lower copula bound.

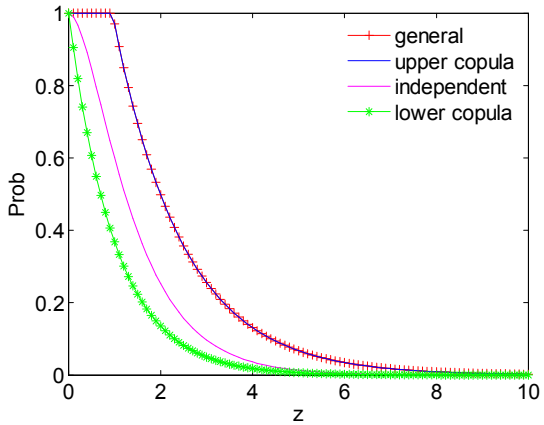



Figure 1. Different Bounds with $\alpha = 0.5, \beta = 1$.

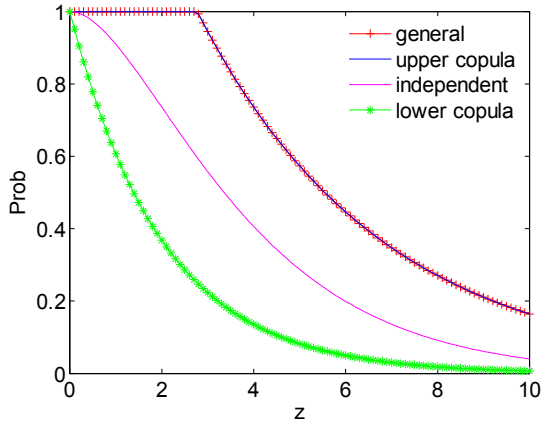



Figure 2. Different Bounds with $\alpha = 2, \beta = 2$.

**References**

[1] R. L. Cruz, "A calculus for network delay. I. Network elements in isolation," in *IEEE Trans. Information. Theory*, vol. 37, no. 1, pp. 114–131, 1991.

[2] C.-S. Chang, *Performance Guarantees in Communication Networks*. Springer-Verlag, 2000.

[3] J. Le Boudec, and P.Thiran, *Network calculus: a theory of deterministic queuing systems for the internet*. Springer-Verlag, 2001.

[4] Y. Jiang, *Stochastic network calculus*. Springer, 2008.

[5] F. Ciucu, A. Burchard, and J. Liebeherr, "A network service curve approach for the stochastic analysis of networks," in *IEEE Trans. Information Theory*, vol. 52, no.6, pp. 2300-2312, 2006.

[6] F. Ciucu and J. Schmitt, "Perspectives on Network Calculus - No Free Lunch but Still Good Value," in *ACM Sigcomm*, pp. 311–322, 2012.

[7]K. Wu, Y. Jiang, and J. Li, "On the model transform in stochastic network calculus," in *Quality of Service (IWQoS), 2010 18th International Workshop on*, pp. 1–9, 2010.

[8] Y. Jiang, "Network Calculus and Queueing Theory: Two Sides of One Coin," in *Proceedings of VALUETOOLS 2009*, 2009.

[9] Y. Jiang, "A Basic Stochastic Network Calculus," in *Proceedings of ACM Sigcomm 06*, pp. 123–134, 2006.

[10] C. Li, A. Burchard and J. Liebeherr, "A Network Calculus With Effective Bandwidth," in *IEEE/ACM Trans. Netw.*, vol. 15, no. 6, pp. 1442–1453, 2007.

[11] D. Starobinski and M. Sidi, "Stochastically bounded burstiness for communication networks," in *IEEE Trans. Information Theory,* vol. 46, no.1, pp. 206-212, 2000.

[12] R. Nelsen, *An Introduction to Copulas*. Springer, 2006.

[13] M. Schwartz, *Broadband Integrated Networks*. Prentice Hall PTR New Jersey, 1996.

**Kui Wu** received the Ph.D. degree in Computing Science from the University of Alberta, Canada, in 2002. He joined the Department of Computer Science at the University of Victoria, Canada in 2002 and is currently a Professor there. His current research interests include network performance evaluation, network security, and cloud computing. He is an IEEE senior member.

**Fang Dong** received Master degree in Electronics and Communication Engineering and Bachelor degree in Physics at Wuhan University, China. She is now working towards her Master degree in Computer Science at the University of Victoria, Canada. Her current research interest is stochastic network calculus.

**Venkatesh Srinivasan** is an Associate Professor in the Computer Science Department at the University of Victoria. Prior to that, he was a postdoctoral fellow at the Max-Planck Institute for Informatics, Center for Discrete Mathematics and Theoretical Computer Science at Rutgers University and the Institute for Advanced Study. He received his PhD degree in Computer Science from the Tata Institute of Fundamental Research. His research interests are in Algorithms and Complexity of Computing.

# A Delay Calculus for Streaming Media Subject to Video Transcoding

*Hao Wang and Jens Schmitt*
*DISCO Lab, University of Kaiserslautern, Germany*
*{wang, jschmitt}@informatik.uni-kl.de*

## 1. Introduction

Streaming video to a diverse range of target devices across a network is enabled by video transcoding. A transcoder converts the video stream to be delivered from one quality level to another, either using the same standard, or even switching to another one. Both of these options, often called homogeneous and heterogeneous video transcoders [1], can be deployed on one or multiple intermediate nodes along the path from the video provider to the consumers. One simple scenario is to deliver, e.g., HD videos, to a certain set of desktop PCs, laptops, tablets, and smart phones (see Figure 1). The transcoding techniques can be very diverse: some packets may be randomly discarded, or information in the stream is used to get the transcoded flow. In general, transcoding is a lossy process.

To achieve a certain Quality-of-Experience (QoE) we usually need to control the end-to-end delay performance of a video stream. Network calculus can provide much insight in the network system performance: for example, an optimal smoother based on network calculus is introduced in [2], yet it cannot capture the lossy transcoding behavior. In previous work [3] of ours, we developed a scaling element for the network calculus to model loss processes (though not limited to these). Therein the model applies at the granularity of an equal sized data unit (e.g., one bit). Now, we investigate the scaling element at the granularity of variable data units as they typically arise in streaming media, where packets with variable lengths are often encountered. We call this new scaling element *packet scaling*. The packet scaling decides whether a whole packet can be forwarded or must be discarded. This happens after the processed bits have been re-packetized with the (same) packetizer. In this paper, we assume that the packetizers after each fluid service remain the same. This is suitable to the case where streaming media is homogeneously transcoded. The heterogeneous case could be based on an extension of [3]. See also Figure 2. For such a network, when analyzing the end-to-end delay the mathematical challenge is to preserve the convolution-form network [4] taking into account the influence of packetization.

## 2. Model of transcoding with network calculus

The time model is discrete. Before defining the packet scaling element we first give the definitions of packetizer [4, 5], packet delay, and packetized server.

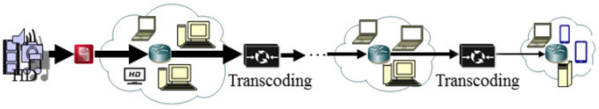



Fig. 1. A network scenario with video transcoding.

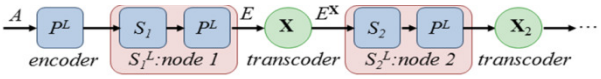



Fig. 2. Model of transcoding with network calculus.

For the *arrival* (*departure*) *process* $A(t)$ ($D(t)$), which is the amount of traffic in $[0, t]$ ($A(s,t) := A(t) - A(s)$ for $0 \le s \le t$), we define:

**Definition 1** (Packetizer). Consider a packet process $L$, which is a sequence of cumulative packet lengths $L(n) = l_0 + \ldots + l_n$, $n = 0, 1, 2, \ldots$ and $l_0 = 0$, an $L$-packetizer $P^L$ is a network element satisfying for all $A(t)$, $t \ge 0$

$$P^L\big(A(t)\big) = \sum_{i=1}^{N_t} l_i\,, \qquad (1)$$

where $N_t = \max\{m: \sum_{i=1}^{m} l_i \le A(t)\}$. We say that a flow $A(t)$ is $L$-packetized if $A(t) = P^L(A(t))$ for all $t \ge 0$.

**Definition 2** (Packet Delay). A process $W(t)$ is called packet delay (process), if for all $t \ge 0$

$$W(t) = \inf\{d \ge 0: P^L\big(A(t)\big) \le P^L\big(D(t+d)\big)\}\,.$$

Next we define a *dynamic server*, which provides a lower bound on the service of a (sub-)system. It is a random process $S(s,t)$ such that the convolution inequality $D(t) \ge A \otimes S(t) := \inf_{0 \le s \le t}\{A(s) + S(s,t)\}$ holds for all $t \ge 0$. When the inequality holds with equality, the dynamic server is said to be *exact*. Note that the convolution of two concatenated dynamic servers $S_1 \otimes S_2$ is still a dynamic server (concept of convolution-form network). We define a *packetized server* consisting of a dynamic server $S$ and a packetizer $L$ and providing a dynamic server $S^L$. When $l_{max}$ exists, from [6] we know a possible $S^L$, $S^L(s,t) = [S(s,t) - l_{max}]_+$. Next, we define the packet scaling element.

**Definition 3** (Packet Scaling Element). A packet scaling element consists of an $L$-packetized arrival process $A(t) = \sum_{i=1}^{N_t} l_i$, a packet scaling process $\mathbf{X}$ taking non-negative integer values and a scaled packetized flow defined for all $t \ge 0$ as $A^{\mathbf{X}}(t) = \sum_{i=1}^{N_t} l_i X_i$.

When we analyze concatenated packet scaling elements, we cannot easily obtain $(A^{X_1})^{X_2}(t) =$

$\sum_{i=1}^{\sum_{j=1}^{A(t)} X_{1,j}} X_{2,i}$. For this case, we use different subscripts for those packet sequences after each round of scaling. Assume an $L$-packetized $A(t) = l_1 + \cdots + l_{N_t}$, $N_t$ is given in Eq. (1). We denote the packets resp. number of packets in the arrivals after each round of scaling as $l_{k,i}$ resp. $m^{(k)}$, clearly $m^{(0)} = N_t$, $m^{(k)} = \sum_{i=1}^{m^{(k-1)}} 1_{\{X_{k,i}>0\}}$, $k > 0$. Further, we denote the concatenated scaled process for all $t \geq 0$ as

$$A^{(k)}(t) := \left(\cdots (A^{X_1})^{X_2}\right)^{X_k}(t), \text{where}$$
$$\text{right} = l_{k-1,1}X_{k,1} + \cdots + l_{k-1,m^{(k-1)}}X_{k,m^{(k-1)}}$$
$$= l_{k,1} + \cdots + l_{k,m^{(k)}}. \quad (2)$$

### 3. End-to-end delay of a network with transcoders

In this section, we compute the end-to-end packet delay for the case with multiple transcoders. According to our previous work, there are two methods to compute the end-to-end delay: one is to commute the service and scaling elements, the other is to get the leftover service for the flow of interest if the server has FIFO scheduling [7]. In this paper, we use the first.

**Lemma 1** (Commutation). *Consider system (a) and (b) in Figure 3, we define* $T^L(s,t) := \sum_{i=N_s+1}^{N_t} l_i X_i$ *as the exact service curve in (b), where* $A(s) = \sum_{i=1}^{N_s} l_i$, $A(s) + S^L(s,t) = \sum_{i=1}^{N_t} l_i$. *If A, S,* $\mathbf{X}$, *and L are independent, then for all* $t \geq 0$, $F(t) \leq E^{\mathbf{X}}(t)$.

Before we present the end-to-end delay bound, we need to show two useful lemmas. We omit the proofs of Lemma 1, 2 due to space restrictions.

**Lemma 2** (Stationarity Bound). *Assume that the packets* $l_i$ *of a packet process L are i.i.d., the* $\mathbf{X}_i$*'s of a scaling element* $\mathbf{X}$ *are also i.i.d., A and B are two L-packetized arrival processes, then for all s, t , x > 0,*

$$\Pr(A^{\mathbf{X}}(t) - B^{\mathbf{X}}(s) \geq x) \leq Pr\left(\left(A(t) - B(s)\right)^{\mathbf{X}} \geq x\right).$$

**Lemma 3** (Recursive MGF Bound of Scaled Process). *Assume that A is an arrival process,* $S_i^L$ *is a packetized server,* $l_i$*'s are i.i.d. with maximal length* $l_{max}$*, and the* $\mathbf{X}_i$*'s are Bernoulli processes. If we denote* $V_{n-1}(\theta_n)$ *as*

$$E\left[e^{\theta\left(\cdots\left(A(t-s)-S_1^L(s,u_1)\right)^{\mathbf{X}_1}-\cdots-S_{n-1}^L(u_{n-2},u_{n-1})\right)^{\mathbf{X}_{n-1}}}\right],$$

*then for all* $0 \leq s \leq u_1 \leq \cdots \leq u_{n-1} \leq t$, *and* $n > 1$

$$V_{n-1}(\theta_n) \leq e^{-\theta_{n-1}S_{n-1}^L(u_{n-2},u_{n-1})}V_{n-2}(\theta_{n-1}),$$

*where* $\theta_n > 0$ *is given.*

Proof. Because $l_i$'s are *i.i.d.* and $\mathbf{X}_{n-1}$ is Markov process, we can view $V_{n-1}(\theta_n)$ as the MGF of $\left(\sum_{i=1}^{P} l_i - \sum_{i=1}^{Q} l_i\right)^{\mathbf{X}_{n-1}}$, where $P$, $Q$ are *r.v.*'s. Thi is a Markov-modulated process. It has $\theta$-envelope $R_{n-1}(\theta)$ ([5]). Thus we get $V_{n-1}(\theta_n) \leq E\left[e^{\theta_n R_{n-1}(\theta_n)(P-Q)}\right]$. On one hand we know that

$$M_{\sum_{Q+1}^{P} l_i}(\theta_{n-1}) = E\left[e^{\log M_l(\theta_{n-1})(P-Q)}\right].$$

On the other hand, we have

$$M_{\sum_{Q+1}^{P} l_i}(\theta_{n-1}) \leq e^{-\theta_{n-1}S_{n-1}^L(u_{n-2},u_{n-1})}V_{n-2}(\theta_{n-1}).$$

Let $\theta_n R_{n-1}(\theta_n) = \log M_l(\theta_{n-1})$, this completes the proof. ■

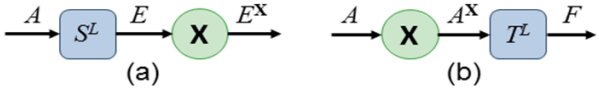



Fig. 3. Commuting packetized server and packet scaling.


Fig. 4. A network with multiple streaming transcodings.

Next, we consider Figure 4 and provide an end-to-end delay bound and the corresponding order of growth.

**Theorem 1** (End-to-end Delays in a Network with Transcoding). *Consider the network scenario from Figure* 4 *where an L-packetized arrival process A*(*t*) = $P^L(A(t))$ *traverses a series of alternate stationary and (mutually) independent bit level service elements together with an L-packetizer and scaling elements denoted by* $S_1$, $S_2$, ..., $S_n$ *and i.i.d. loss processes* $\mathbf{X}_1$, $\mathbf{X}_2$, ..., $\mathbf{X}_{n-1}$, *respectively. Assume the packets of L* $l_i$ *are i.i.d. Assume the MGF bounds* $M_{A(s,t)}(\theta) \leq e^{\theta r_A(\theta)(t-s)}$ *and* $M_{S_k(t)}(-\theta) \leq e^{-\theta C_k t}$, *for k* = 1, ..., *n, and some* $\theta > 0$. *We also assume that the maximal packet length of L is* $l_{max}$. *Under a stability condition, given in the proof, for* $\theta_i > 0, i = 1, \ldots, n$, *we have the following end-to-end steady state delay bounds for all* $d \geq 0$

$$\Pr(W > d) \leq e^{\left(\sum_{i=1}^{n}\theta_i+\theta_1\right)l_{max}K^n b^d}, \quad (3)$$

*where the constants K and b are given in the proof as well. Moreover, the ε-quantiles scale as* $\mathcal{O}(n)$, *for some* $\varepsilon > 0$.

*Proof.* Due to space restrictions we only sketch the proof. The proof follows Theorem 1 in [3]. We use Lemma 1, 2, 3 and Eq. (2) here, and letting $b = \sup\{e^{-\theta_k C_k} : 1 \leq k \leq n\}$ we compute the end-to-end delay bound on $W_n(t)$ as

$$\Pr(W_n(t) > d)$$
$$\leq \sum_{0\leq s\leq t} e^{\mathrm{d}\cdot\log b}e^{\left(\sum_{i=1}^{n}\theta_i+\theta_1\right)l_{max}}e^{\left(\log b+\theta_1 r_A(\theta_1)\right)(t-s)} \cdot \sum_{s\leq u_1\leq\cdots\leq u_{n-1}\leq t+d} 1$$
$$\leq b^d e^{\left(\sum_{i=1}^{n}\theta_i+\theta_1\right)l_{max}K^n},$$

Here we let $K = \left(1+\frac{d}{n}\right)^{1+\frac{d}{n}} \Big/ \left(\frac{d}{n}\right)^{\frac{d}{n}}$ and used $\log b + \theta_1 r_A(\theta_1) < 0$ as the stability condition. Taking $t \to \infty$ proves the result. Finally, the order of growth of the $\varepsilon$-quantiles for some $0 < \varepsilon < 1$ follows directly as $\mathcal{O}(n)$. ■

## 4. Evaluation

To evaluate the results, we use the following example settings. First, we let the packet sizes be discrete uniformly distributed *i.i.d. r.v.*'s, $l \sim$ U[$a$,$b$]. Then we know $M_l(\theta) = \frac{e^{a\theta_1} - e^{(b+1)\theta_1}}{(b-a+1)(1-e^{\theta_1})}$. Let $a$ = 1, $b$ = 16 for illustration. Clearly, $l_{max}$ = 16. Next, we use a Bernoulli scaling process $\mathbf{X} \sim$ B($p$) for all transcoders, so that we know $R(\theta) = \frac{1}{\theta}\log(1-p+pM_l(\theta))$. Further we assume the original arrival traffic of homogeneous data units (e.g., bits) to be a Poisson process Poi($\lambda$), while the service for this traffic at the servers is work-conserving with constant rate $C_i$. Further, let $\lambda$ = 1; the number of transcoders varies from 1 to 9; the capacities at the rest of the nodes are statically set as $C_1..C_{10}$ = [1.25, 1.15, 1.05, 0.95, 0.85, 0.80, 0.75, 0.70, 0.65, 0.60]; $\varepsilon$ is $10^{-3}$. We use Omnet++ 4.3 to do simulations. We measure $10^6$ packet delays at the end receiver node.

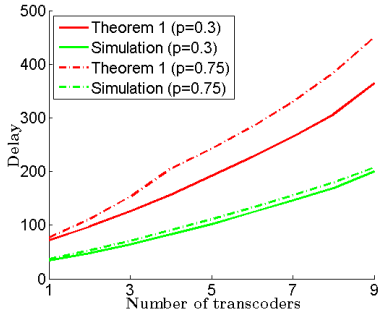



Fig. 5. Delay bounds with Theorem 1 and simulation.

Figure 5 compares the stochastic end-to-end delay bounds obtained with Theorem 1 and simulation. Both curves exhibit the $\mathcal{O}(n)$ order of growth. It is also illustrated that the delay bounds are increasing in probability $p$. That means if the streaming data are longer kept at a node due to transcoding, the burstier the load of the next node becomes. Clearly, the bounds from Eq. (3) of Theorem 1 are not tight when compared to simulations. That is because we consider the increased latency for each packet after being served by the packetized server as $\frac{l_{max}}{C}$, while actually most packets have smaller increased latency. To avoid using $l_{max}$ we would have to consider the inherent correlation among arrivals, service and packet scaling elements, which makes it very difficult to analyze the delay bounds even for a single node without scaling element (see [8]).

## 5. Conclusion

In this paper, we applied network calculus to the end-to-end delay analysis of streaming video across networks subject to in-network transcoding. We have shown that the delays can be bounded and grow in the order of number of transcoders. The analysis can be applied to a simple scenario where a data unit from the streaming video is randomly selected, or, can be extended to more complicated scenarios of transcoding if we specialize the scaling for specific transcoding.

## References

[1] I. Ahmad, X. Wei, Y. Sun, and Y. Zhang, "Video transcoding: an overview of various techniques and research issues," in *IEEE Trans. Multimedia*, vol. 7, no. 5, pp. 793-804, 2005.

[2] P. Thiran, J.-Y. Le Boudec, and F. Worm, "Network calculus applied to optimal multimedia smoothing," in Proc. of INFOCOM, pp. 1474-1483, 2001.

[3] F. Ciucu, J. Schmitt, and H. Wang, "On expressing networks with flow transformations in convolution-form," in *Proc. of INFOCOM*, pp. 1979-1987, 2011.

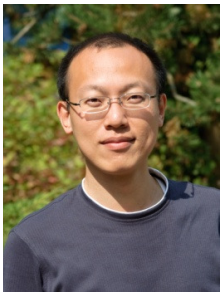
[4] J.-Y. Le Boudec and P. Thiran, *Network Calculus*, Springer Verlag, Lecture Notes in Computer Science, LNCS 2050, 2001.

[5] C.-S. Chang, *Performance Guarantees in Communication Networks*, Springer Verlag, 2000.

[6] A. Burchard, J. Liebeherr, and F. Ciucu, "On superlinear scaling of network delays," in *IEEE/ACM Trans. Networking*, vol. 19, no. 4, pp. 1043-1056, 2011.

[7] H. Wang, F. Ciucu and J. Schmitt, "A leftover service curve approach to analyze demultiplexing in queueing networks," in *Proc. of VALUETOOLS*, pp. 168-177, 2012.

[8] Y. Jiang, "Stochastic service curve and delay bound analysis: a single node case," in *Proc. of ITC 25*, pp. 1-9, 2013.



**Hao Wang** received his B.Sc. degree in Computer Science and Technology from Northeastern University, China and his M.Sc. degree in Computer Science from University of Kaiserslautern, Germany. His research interests include performance modeling of distributed computer systems, especially using stochastic network calculus.

**Jens B. Schmitt** is professor for Computer Science at the TU Kaiserslautern. Since 2003 he has been the head of the Distributed Computer Systems Lab (disco). His research interests are broadly in performance and security aspects of networked and distributed systems. He received his Ph.D. from TU Darmstadt in 2000.

# MMTC OFFICERS

**CHAIR**

Jianwei Huang
The Chinese University of Hong Kong
China

**STEERING COMMITTEE CHAIR**

Pascal Frossard
EPFL, Switzerland

**VICE CHAIRS**

Kai Yang
Bell Labs, Alcatel-Lucent
USA

Chonggang Wang
InterDigital Communications
USA

Yonggang Wen
Nanyang Technological University
Singapore

Luigi Atzori
University of Cagliari
Italy

**SECRETARY**

Liang Zhou
Nanjing University of Posts and Telecommunications
China

## E-LETTER BOARD MEMBERS


| | | | |
|---|---|---|---|
| Shiwen Mao | Director | Aburn University | USA |
| Guosen Yue | Co-Director | NEC labs | USA |
| Periklis Chatzimisios | Co-Director | Alexander Technological Educational Institute of Thessaloniki | Greece |
| Florin Ciucu | Editor | TU Berlin | Germany |
| Markus Fiedler | Editor | Blekinge Institute of Technology | Sweden |
| Michelle X. Gong | Editor | Intel Labs | USA |
| Cheng-Hsin Hsu | Editor | National Tsing Hua University | Taiwan |
| Zhu Liu | Editor | AT&T | USA |
| Konstantinos Samdanis | Editor | NEC Labs | Germany |
| Joerg Widmer | Editor | Institute IMDEA Networks | Spain |
| Yik Chung Wu | Editor | The University of Hong Kong | Hong Kong |
| Weiyi Zhang | Editor | AT&T Labs Research | USA |
| Yan Zhang | Editor | Simula Research Laboratory | Norway |