Network Performance Assessment with Quality of Experience Benchmarks

Decebal Constantin Mocanu*, Giuliano Santandrea†, Walter Cerroni†, Franco Callegati† and Antonio Liotta*
*Department of Electrical Engineering
Eindhoven University of Technology, The Netherlands
Email:{d.c.mocanu,a.liotta}@tue.nl
†Department of Electrical and Information Engineering
University of Bologna, Italy
Email: {giuliano.santandrea2, walter.cerroni, franco.callegati}@unibo.it

***Abstract*—Today the performance of network services and devices is mainly assessed using Quality of Services (QoS) factors. These provide statistics about the quality of the network behavior but cannot accurately reflect how the unpredictable impairments which might occur in the network end up affecting the perception of the final beneficiary of these services, i.e. the user. This situation arises because QoS-based performance analysis does not capture the combined end-to-end properties of networks and applications. In this paper, we introduce a new network performance methodology based on Quality of Experience benchmarks, whereby we estimate the quality of the service as it is perceived by the user. We illustrate this approach in the context of video streaming services, showing how to evaluate quality degradation in Software Defined Networks. Our approach is better suited to the evaluation of dynamic networks and helps better pinpointing the critical factors that affect the applications the most.**

I. Introduction

Presently, assessing the performance of networks services and devices is mainly done using Quality of Services (QoS) factors, such us jitter, latency and packet loss. These factors show statistics about the quality of the network behavior, and in many cases they are reflecting well the specificity of the networks, but cannot reflect accurately how the unpredictable impairments which might occur in the network could affect the perception of the final beneficiary of these services, i.e. the user. In practice, it will be hard to meet in full the users quality requirements by operating merely on QoS benchmarks, as it is shown in [1]. This situation arises because QoS-based performance analysis does not capture the combined end-to-end properties of networks and applications. According to [2], those services and applications which will take into account the user expectations will gain more success and will be adopted widely into the future. Hence, the Quality of Experience (QoE) assessment techniques [3], [4], [5] can be used to assess the network services performance, by focusing on the quality perceived by the user [6]. In the scope of these arguments, in this paper we introduce a new methodology to assess the performance of network services and devices, using end-to-end QoE benchmarks to take the user's experience into account. Thus, we look at the synergy between video streaming services and objective Quality of Experience algorithms [7]. To achieve the aforementioned goal, additionally, this article is proposing a novel video synchronization algorithm to detect completely lost frames in the streamed video received on the client side. In comparison with Evalvid [8], a framework with similar aims, our proposed method is lighter, and does not involve the manipulation of large YUV files, the assessor needing only just mp4 videos. Besides that, our proposed video synchronization algorithm doesn't need the QoS throughput to find the location of the lost frames in the impaired video, as Evalvid does, or further manipulations of the video files as in the case of frame tagging technique. In its simplicity, the algorithm needs just the original and the impaired videos to synchronize the frames and estimate the network performance by approximating the human viewpoint, as much as possible. The proposed procedure was evaluated in the context of Software Defined Networks (SDN) and Network Function Virtualization (NFV) concepts. We find that our approach is better suited to evaluate the interplay between time-constrained applications (video streaming) and virtual networks (SDN with service migration).

II. Network performance methodology

A. General Architecture

The key modules of the Quality of Experience benchmarking framework for networks performance are sketched in Fig. 1. It can be distinguished a data-flow initiated in a server (or server farm or, more generally, in a service cloud), going through the tested network and onto the client machine. In practice, this data-flow is a video service. Hence, the server is starting to stream a video and on the other hand the streamed video is recorded into the receiving terminal. Afterwards, the received video is resynchronized in such a way that it can be compared with the original unimpaired video using state-of-the-art objective QoE algorithms, to measure quality degradation, as it would be perceived by a typical user. In other words, this means to match all the frames from the impaired video with the corresponding frames from the original videos such that the average score over all matched frames, given by Image Quality Assessment (IQA) algorithms [9], is maximized.

B. Streaming Videos

FFmpeg[1] is used to stream a MP4 video[2] from the server to the client with the Real-time Transport Protocol (RTP) over UDP. We prefer to use RTP in the detriment of other actual technologies, because RTP ensures us that each frame is transmitted uniformly distributed in time over the network.

[1]http://www.ffmpeg.org/, September 21th 2014
[2]Please note that other types of video files can be used.

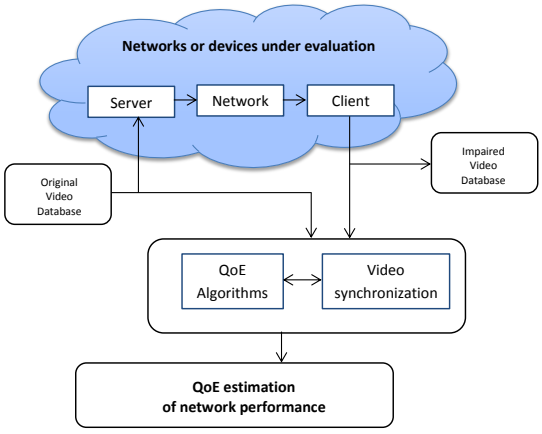

Fig. 1: General architecture and information flow for the proposed network performance methodology.

This means that we can assess the network performance almost continuously and not as in the case of, e.g. HTTP streaming where the video is download in small chunks of at least 1 second. The first step of the proposed methodology is to stream the video through the network under scrutiny. On the client side, FFmpeg records the stream as a MP4 file. At this point of the work-flow, there are two sets of videos: (1) one set with the original unimpaired videos and (2) one with the impaired videos. These two sets can be firstly synchronized using the algorithm introduced in Subsection II-C, and secondly compared using the algorithms from Subsection II-D.

C. Video Synchronization Algorithm

In order to quantify the quality of an impaired video, a frame-by-frame comparison has to be done between the original and the streamed video using full reference objective QoE algorithms for IQA. Nevertheless, one of the impairments which might appear in the captured video is that some frames are missing from it due to packet losses caused by transmission or congestion problems in the tested network. This misalignment given by even a single missing frame will propagate all the way to the end of the video file, yielding erroneous results. To avoid this issue, in this paper we are proposing a greedy algorithm to handle frames losses, and to synchronize the frames from the original and the impaired video, independently on the QoS measured on the client side. This synchronization practically means to detect the missing frames in the streamed video and to align the other frames from it with their corresponding frames in the original one for a fair comparison using any IQA algorithm. To do this one can use the dynamic programming approach to calculate the IQA values for all possible combination between the frames from the original video and the ones from the distorted video to pick the best matches, as it is done in the case of Dynamic Time Warping algorithm [10]. However, this approach has a computational complexity of $\mathcal{O}(n_{ov}^2 \times \mathcal{O}_{IQA})$, where n_{ov} is the total number of frames in the original videos, and $\mathcal{O}_{IQA}$ represents the computational complexity of the IQA algorithm. To reduce the computational complexity, we are making the following observation: if k frames are missing from the distorted video, for sure, for any frame with index i from the impaired video, the only possible places where its matched frame from the original video is located are one of the frames which have the indexes from i to $i+k$. Furthermore, if a greedy selection procedure to find the matching pair frames is applied, the computational complexity of the proposed algorithm is reduced to a maximum of $\mathcal{O}(n_{ov}k \times \mathcal{O}_{IQA})$. The pseudo-code for the aforementioned algorithm is given in Alg. 1, where $threshold_{IQA}$ can be set to the upper bound of the possible values given by the IQA algorithm, or it can be set to a lower value for a faster computational time, while trading off the performance of the algorithm. After the algorithm is run, the frames which are missing from the impaired video are representing by a 0 IQA value in the QoE array. Also, by averaging all the values from the QoE array a quality estimation for the impaired video is obtained. Furthermore, the QoE array might be compared with the QoS factors of the tested network, to discover hidden relations between them.

%Inputs;
F_{ov} - array with the frames of the original video;
F_{iv} - array with the frames of the impaired video;
$threshold_{IQA}$ - threshold for the IQA algorithm;
%Initialization;
n_{ov} = the number of frames from the original video;
n_{iv} = the number of frames from the impaired video;
$k = n_{ov} - n_{iv}$ %the number of missing frames;
$pos_{actual} = 1$;
$QoE = zeros(n_{ov})$ %array to save the IQA values of the matched frames;
%parse each frame of the impaired video;
for $i = 1 : n_{iv}$ **do**
  $max_{IQA} = -\infty$;
  $pos_{match} = 0$;
  %search the corresponding frame in original video;
  for $j = pos_{actual} : (i + k)$ **do**
    $current_{IQA} = IQA(F_{iv}[i], F_{ov}[j])$;
    if $max_{IQA} < current_{IQA}$ **then**
      $max_{IQA} = current_{IQA}$;
      $pos_{match} = j$;
    end
    if $max_{IQA} > threshold_{IQA}$ **then**
      break;
    end
  end
  %save the IQA value for the matched frames;
  $pos_{actual} = pos_{match}$;
  $QoE[pos_{actual}] = max_{IQA}$
end
%calculate the mean for all matched frames
$QoE_{average} = mean(QoE)$;

Algorithm 1: Video Synchronization Algorithm

D. Objective QoE algorithms

In the previous discussed video synchronization algorithm, any QoE algorithms for IQA can be used. The best way to assess the quality for each frame of the video would be to use subjective studies [11]. Still, this is time consuming and unpractical for a network specialist, who would like to fine tune different parameters of the network, to offer the best services for the users in a short time. Thus, objective IQA

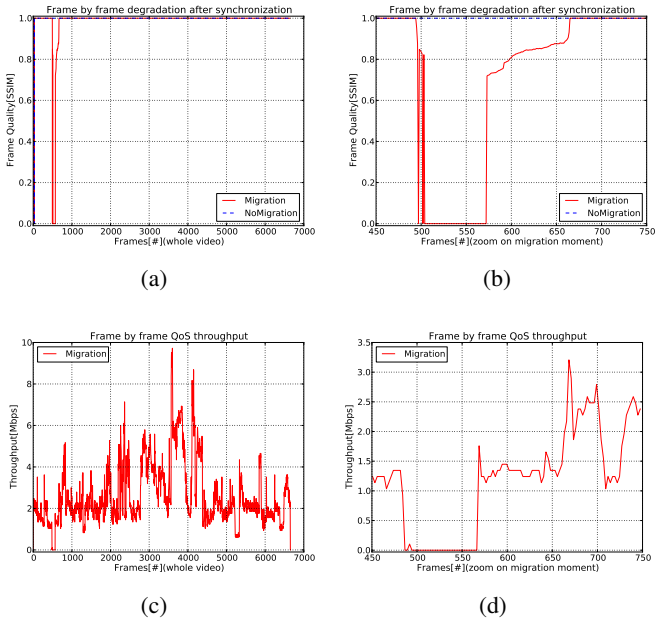

Fig. 2: Streaming through a local area network characterized by low latency and low jitter. (a) The video quality achieved for all the frames, after synchronization. (b) The video quality achieved by zooming on the migration moment, after synchronization. (c) The throughput measured at the receiver for all frames. (d) The throughput measured at the receiver with zoom on the migration moment.

algorithms are better suitable for this purpose because they can be highly automatized. However, among these, we choose to use a full reference IQA metric, widely used in the literature as Structural Similarity (SSIM) [12]. It provides a good trade-off between computational complexity and accuracy, and it is considered to be well correlated with the quality perception of the Human Visual System. We would like to highlight that SSIM can be replaced easily with any other IQA algorithm.

III. Experiments and Results

The proposed methodology to assess the performance of network services from the user point of view was tested in a SDN context, involving the emerging NFV paradigm. The main motivation behind this choice is to assess the impact on QoE of flexible infrastructure resource sharing and virtual network function management procedures, which are specific of SDN/NFV scenarios and may introduce additional service degradation with respect to traditional network paradigms. In particular, the experimental setup consists of two virtual machines (VMs) implementing a video-on-demand server and an access router, respectively. The virtual access router, connecting a client device to the virtual video server, is specialized to provide specific customers with their required network profile (e.g., global connectivity, network address translation, firewall, guaranteed bandwidth, etc.). While a given end-user is watching a video stream coming from the virtual server, the network provider moves both VMs from a physical host to another one, e.g. to perform data center resource maintenance and server consolidation, to follow a mobile user to a different

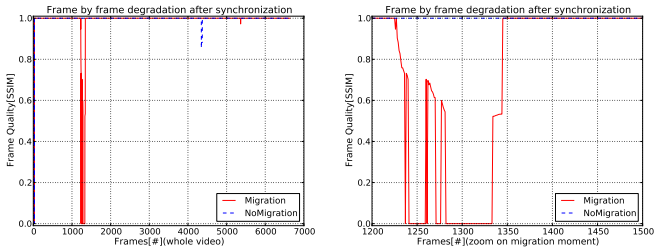

Fig. 3: Streaming through Internet. (a) The video quality achieved for all the frames, after synchronization. (b) The video quality achieved by zooming on the migration moment, after synchronization.

network region where more bandwidth resource is available, or to apply any of the flexible network resource management techniques made available by SDN/NFV. A more detailed explanation of this setup is given in [13]. Under these conditions, we have used our proposed methodology to evaluate how the sequential migration process of the two VMs affects the quality of the video perceived by the end-user on the client device. To be able to perform the evaluation, the video synchronization algorithm has been implemented from scratch in Python. The operations needed to process the video files were done using a python wrapper for OpenCV [14]. For streaming, we used a high quality video with a high level of motion, encoded in MP4 format, at 25 frames per seconds, and with the length of 6651 frames. The experiments were split in two scenarios, based on the physical location of the client computer.

A. Scenario 1. Streaming in the Local Network

In this first scenario, both the cloud server implementing the video streaming virtual machine and the client computer were located in Bologna, Italy. The client and the server were connected through a low-latency local area network. Two test cases are described further. (1) *Low jitter.* The video has been streamed through a local network, characterized by low jitter conditions given by the random behavior of the network. First, the virtual machines were migrated during the streaming process. Then, the virtual machines were not migrated during the streaming. These situations are shown comparatively for the whole length and with a zoom on the migration moment in Fig. 2a, and Fig. 2b, respectively. The overall quality degradation measured with SSIM averaged over all frames was 0.9865, which shows that the streamed video is very similar to the original one. Still, when the migration was performed, in total there were 78 missing frames in the received video. Hence, the user did not receive the stream for approximately 3 seconds. In addition, for further 3 seconds around the migration phase, the received quality is also degraded. In turns, this shows the limitations of a mere QoS approach, which would not be able to give a sufficiently accurate insight of this situation, as it is depicted in Fig. 2c and Fig. 2d. Per contra, our QoE method places quality degradation in frame-by-frame timeline and directly in relation to the network (migration) dynamics. (2) *No jitter.* The previous case has been repeated and the uncontrolled behavior of the network has resulted in

TABLE I: Network performance evaluation. QoE vs. QoS.

	QoE			QoS
	Downtime (s)	Decreased Quality Period (s)	Total (s)	Downtime (s)
Scenario 1.1	≈ 3.12	≈ 3.96	≈ 7.08	≈ 3.19
Scenario 1.2	≈ 3.40	≈ 3.72	≈ 7.12	≈ 3.38
Scenario 2	≈ 3.52	≈ 1.60	≈ 5.12	≈ 3.42

no jitter. The results were similar, this letting us to speculate that jitter has a low influence on the tested setup.

B. Scenario 2. Streaming across Internet domains

In the second scenario, the cloud server with the video streaming virtual machine was located in Bologna, Italy, while the client computer was located in Eindhoven, The Netherlands. The client and the server were connected through the ordinary Internet. Like for the previous case, the streaming process was carried out with and without migrating the virtual machines. In comparison with the local area network experiments, the migration phase is characterized by downtime moments which are alternating with low quality moments of the streamed video, as may be observed in Fig. 3a and Fig. 3b. In other words, these can be interpreted as oscillations in the user's quality perception. In this scenario, 88 missing frames were registered in the impaired video, and the overall quality level measured by SSIM, and averaged over all frames, was 0.9859. It is worth mentioning, that this value becomes closer to 1 if, considering the same single service downtime event, the video length is increased. This is correct in terms of QoE if we assume that the user will subjectively consider of good quality a longer video with only a few seconds of disruption.

In all the experiments performed, the downtime of the streaming services introduced by the migration phase affected approximately 80 frames, which accounts to about 3 seconds. It is worth to specify, that such a small downtime value is the result of the sequential VM migration procedure which has been executed on a dedicated 1 Gbps connection between the two physical hosts of the VMs. This downtime has been detected by both, our QoE based methodology and a standard QoS technique. Besides that, in each scenario, there were further 80 frames circa, for which the quality of the received video was lower than usual, while the streaming service was still functional. The QoS technique was not able to detect these lower quality frames, while our QoE technique revealed them. Thus, using our methodology we have been able to accurately detect what happens when a virtual machine which acts as a streaming server is migrating from a physical computer to another. In this way, we find that the user's experience (i.e. the quality of the video received by the user) is affected negatively for approximately 6-7 seconds, as video degradation propagates beyond the 3 seconds of QoS degradation incurred during service migration. A summery of the service degradation period, measured respectively with QoS and QoE metrics, is given in Table I.

IV. Conclusion

In this paper we introduce a novel network performance methodology based on video streaming services and Quality of Experience benchmarks, whereby we estimate the quality of the service as it is perceived by the user. Additionally, we propose a novel algorithm for video synchronization, which is required to automate the overall evaluation process. The methodology was assessed in the context of SDN and NFV concepts. The results show that our proposed procedure is better suited than conventional QoS based approaches, because it is able to detect precisely the effects of the virtual machine migration on the video service. For instance, it detects not just the time when the video streaming server is down, but also the extended quality degradation due to video error propagation effects. Besides that, the QoE analysis is useful for the network's designers to pinpoint the sensitivity of service quality to specific network parameter. As further research directions, we are intending to use the proposed methodology to evaluate other types of network services and devices, in cases where QoS assessment fails to capture the intricacies between network impairments and service or application dynamics.

Acknowledgment

This work was funded by EIT ICT labs, under the project "Smart Networks at the Edge".

References

[1] R. M. Noor and S. Khorsandroo, "Quality of experience key metrics framework for network mobility user," *International Journal of the Physical Sciences*, vol. 6, no. 28, pp. 6521–6528, 2011.

[2] K. D. Moor, W. Joseph, I. Ketyk, E. Tanghe, T. Deryckere, L. Martens, and L. D. Marez, "Linking users' subjective qoe evaluation to signal strength in an ieee 802.11b/g wireless lan environment." *EURASIP J. Wireless Comm. and Networking*, vol. 2010, 2010.

[3] F. Agboma, M. Smy, and A. Liotta, "Qoe analysis of a peer-to-peer television system," *Proceedings of IADISInt. Conf. on Telecommunications, Networks and Systems*, pp. 365–382, 2008.

[4] F. Agboma and A. Liotta, "Quality of experience management in mobile content delivery systems." *Telecommunication Systems*, vol. 49, no. 1, pp. 85–98, 2012.

[5] V. Menkovski, G. Exarchakos, A. Liotta, and A. C. Sanchez, "Quality of experience models for multimedia streaming." *IJMCMC*, vol. 2, no. 4, pp. 1–20, 2010.

[6] A. Liotta, "The cognitive NET is coming," *IEEE Spectrum*, vol. 50, no. 8, pp. 26–31, Aug. 2013.

[7] A. Liotta, D. C. Mocanu, V. Menkovski, L. Cagnetta, and G. Exarchakos, "Instantaneous video quality assessment for lightweight devices," in *Proceedings of International Conference on Advances in Mobile Computing & Multimedia*, ser. MoMM '13. New York, NY, USA: ACM, 2013, pp. 525:525–525:531.

[8] J. Klaue, B. Rathke, and A. Wolisz, "Evalvid - a framework for video transmission and quality evaluation," in *In Proc. of the 13th International Conference on Modelling Techniques and Tools for Computer Performance Evaluation*, 2003, pp. 255–272.

[9] A. K. Moorthy and A. C. Bovik, "Visual quality assessment algorithms: What does the future hold?" *Multimedia Tools Appl.*, vol. 51, no. 2, pp. 675–696, Jan. 2011.

[10] D. J. Berndt and J. Clifford, "Using dynamic time warping to find patterns in time series." in *KDD Workshop*, U. M. Fayyad and R. Uthurusamy, Eds. AAAI Press, 1994, pp. 359–370.

[11] K. Seshadrinathan, R. Soundararajan, A. C. Bovik, and L. K. Cormack, "Study of subjective and objective quality assessment of video," *Trans. Img. Proc.*, vol. 19, no. 6, pp. 1427–1441, Jun. 2010.

[12] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image Quality Assessment: From Error Visibility to Structural Similarity," *IEEE Transactions on Image Processing*, vol. 13, no. 4, pp. 600–612, Apr. 2004.

[13] A. Manzalini, R. Minerva, F. Callegati, W. Cerroni, and A. Campi, "Clouds of virtual machines in edge networks," *IEEE Communications Magazine*, vol. 51, no. 7, pp. 63–70, 2013.

[14] G. Bradski, *Dr. Dobb's Journal of Software Tools*.