Docker-Based Evaluation Framework for Video Streaming QoE in Broadband Networks

Cise Midoglu
Simula Research Laboratory
cise@simula.no

Anatoliy Zabrovskiy
Alpen-Adria-Universität Klagenfurt
anatoliy.zabrovskiy@aau.at

Özgü Alay
Simula Metropolitan Center for Digital Engineering
ozgu@simula.no

Daniel Hölbling-Inzko
Bitmovin Inc
dhi@bitmovin.com

Carsten Griwodz
University of Oslo
griff@ifi.uio.no

Christian Timmerer
Alpen-Adria-Universität Klagenfurt
christian.timmerer@aau.at

ABSTRACT

Video streaming is one of the top traffic contributors in the Internet and a frequent research subject. It is expected that streaming traffic will grow 4-fold for video globally and 9-fold for mobile video between 2017 and 2022. In this paper, we present an automatized measurement framework for evaluating video streaming QoE in operational broadband networks, using headless streaming with a Docker-based client, and a server-side implementation allowing for the use of multiple video players and adaptation algorithms. Our framework allows for integration with the MONROE testbed and Bitmovin Analytics, which bring on the possibility to conduct large-scale measurements in different networks, including mobility scenarios, and monitor different parameters in the application, transport, network, and physical layers in real-time.

CCS CONCEPTS

• **Information systems → Multimedia streaming**; • **Networks → Network performance evaluation**; *Mobile networks*; *Public Internet*.

KEYWORDS

adaptive streaming, network measurements, OTT video analytics, QoE

ACM Reference Format:
Cise Midoglu, Anatoliy Zabrovskiy, Özgü Alay, Daniel Hölbling-Inzko, Carsten Griwodz, and Christian Timmerer. 2019. Docker-Based Evaluation Framework for Video Streaming QoE in Broadband Networks. In *Proceedings of the 27th ACM International Conference on Multimedia (MM'19), Oct. 21–25, 2019, Nice, France.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3343031.3350538

1 INTRODUCTION

Video is the most prominent type of traffic in the Internet today, and is expected to grow 9-fold between 2017 and 2022, taking up to f global mobile data traffic in 2022 [1]. It is also a widely approached research subject, including studies ranging from the design and comparison of video players and adaptation algorithms [4, 7, 11, 12], to the end-to-end evaluation of commercial streaming services [3, 8–10]. However, video streaming Quality of Experience (QoE) is not explored in Mobile Broadband (MBB) as much as in fixed networks and WLANs. There are not many studies involving extensive end-to-end experimentation in operational MBB, with respect to different players and adaptation algorithms. Mobile performance measurement studies often evaluate commercial services such as YouTube, through their native API [8, 9] or through active probing for content quality, availability, and CDN performance [3]. Others, aiming to benchmark different Adaptive Bitrate (ABR) algorithms, often rely on trace-driven link emulations instead of real-life measurements, as well as their own implementation of different algorithms instead of the complete instrumentation of existing video players [7].

Since end users are increasingly expecting video services wherever they are, even while they are on the move, *mobility* becomes a challenging trend. Empirical studies focusing on video streaming measurements need real-life mobility measurements in large scale, in order for the research community to develop more insights. Another challenging aspect is the collection of network-level *metadata*. Most of the existing measurement studies do not include the collection of high-granularity network-level information alongside streaming QoE metrics [11]. However, the association of network and application layer parameters are key to understanding and improving the performance of adaptive video players [4, 12]. The research community can benefit from the collection of metadata-rich and geo-tagged video streaming QoE datasets, which comprise measurements from operational MBB under mobility. These are the motivations to implement a measurement framework which allows for large-scale experimentation in both fixed and mobile operational networks ("in the wild"), with the native player implementations (without emulation), collecting a large set of video streaming QoE metrics alongside network metadata.

In this work, we provide open source software assets developed as part of such a framework, called Video-Bitmovin-ITEC-MONROE (VBIM). VBIM consists of (1) a client container able to run video players in headless mode, (2) a server setup offering a variety of adaptive video players and adaptation algorithms[1], as well as a video asset, (3) possibility to run the client container on the MBB measurement testbed Measuring Mobile Broadband Networks in Europe (MONROE) [5, 6] for gathering extensive metadata, and

MM '19, October 21–25, 2019, Nice, France

ACM ISBN 978-1-4503-6889-6/19/10.
https://doi.org/10.1145/3343031.3350538

[1] Our setup currently hosts 3 video players with 5 adaptation algorithms.

(4) an optional integration with a third party OTT video analytics platform Bitmovin Analytics [2], showing how such products can aid research by providing the academic and developer communities with an extensive set of QoE-related metrics in real-time. We open-source the software with documentation on a GitHub repository[3], where implementation details, and instructions on how to use the framework (as well as how to run experiments on the MONROE testbed) can be found.

2 FRAMEWORK

We design and implement the active measurement framework VBIM, which incorporates headless streaming with different video players in a virtualized Docker environment, while also monitoring parameters in the application, transport, network, and physical layers.

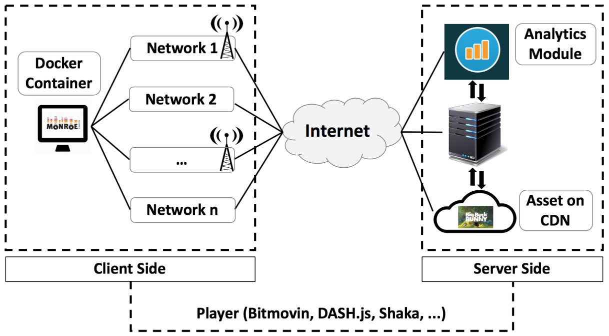

Figure 1: Measurement framework.

Figure 1 presents our measurement framework in a simplified manner. The core comprises two components: (1) VBIM client which can run on any system supporting Docker virtualization, including MONROE measurement nodes, and can be configured to use any available network interface, (2) VBIM server including landing page(s) where different players, test content, and optional analytics services can be incorporated.

2.1 Core Components

2.1.1 Client Side. The VBIM client is designed to emulate a user initiating streaming session(s), by opening a Google Chrome browser and streaming a video from an HTTP server. Characteristics of the client side implementation are as follows.

Docker implementation: To achieve a uniform and reproducible measurement environment, the client is designed as a Docker container. Thus, it is independent from system software settings[4]. The container core is written in Python, and the demo-ready Docker image can be found on Dockerhub[5].

Testbed compatibility: The container can be run without user interaction on simple measurement nodes (e.g., downsized, headless Linux PCs widely used in MBB testbeds such as MONROE). It employs Selenium instrumentation[6] to emulate an end-user who streams multimedia content via the Chrome browser using various video players.

Network monitoring: The container includes optional ping and traceroute measurements towards desired servers (configurable) after the video playback, to collect additional information.

Configuration: The container is configured using an input file or a JSON-formatted string. Just as a single measurement run can be scheduled, VBIM has a multiple-configuration mode which allows for scheduling multiple runs at once, convenient for batch experimentation (e.g., a measurement with `Player A` using `Algorithm X`, followed by another with `Player A` using `Algorithm Y`, and another with `Player B` using `Algorithm Z`, within a "batch").

Metadata: If the container is run on a testbed capable of providing metadata through ZeroMQ (ZMQ) in a publish-subscribe manner[7], this information is collected throughout the experiment.

Outputs: The client container saves the measurement configuration details, ping and traceroute results (if enabled), and complete console output during the measurement. If the experiment is run on the MONROE platform, metadata events from the active interface also appear in the output.

2.1.2 Server Side. The VBIM server hosts HTML files serving as a landing page for the headless browser implementation on the client. The landing page incorporates the (paths to the) selected player and analytics builds, as well as video content.

Players: In the scope of this work, we host 3 pages for: (1) Bitmovin Player v8[8], (2) Shaka Player v2.5[9], and (3) DASH.js Player v2.9[10], all integrated with Bitmovin Analytics v2.3[11], and the same video content described below. The framework is easily extendible to more players and algorithms, by adding a new landing page linking to any JavaScript player build[12]. There is no limit to the number of landing pages that the client can select from.

Video content: We use the `BigBuckBunny`[13] video encoded in 15 different quality levels (from 100 Kbps to 15 Mbps) using the Bitmovin encoding service[14] and uploaded to a CDN. The video content is currently not configurable from the client side. However, it is possible to extend the client and server implementations with minimal effort, to allow for the selection of a video from a list using configuration parameters.

2.2 MONROE Platform Integration

The MONROE platform [5, 6], now operated and maintained by the MONROE Alliance, was developed within the EU H2020 project of the same name, and provides the first European hardware-based

[2] https://bitmovin.com/video-analytics/

[3] https://github.com/cise-midoglu/vbim

[4] The client container is based on the `base:web` image provided by the MONROE project under https://hub.docker.com/r/monroe/base/. It runs on `debian:stretch`.

[5] https://hub.docker.com/r/cmidoglu/vbim-demo

[6] https://www.seleniumhq.org/

[7] ZMQ is a distributed asynchronous messaging library, which provides a publish/subscribe mechanism where consumers can subscribe to selected parts of the information published by a provider (http://zeromq.org/). For details regarding the ZMQ implementation on the MONROE platform, see [2].

[8] https://bitmovin.com/docs/player

[9] https://github.com/google/shaka-player

[10] https://github.com/Dash-Industry-Forum/dash.js

[11] https://bitmovin.com/docs/analytics/quickstarts/integrating-bitmovin-analytics-with-[bitmovin-player|dash-js|shaka]

[12] For a list of players supported by Bitmovin Analytics, see https://bitmovin.com/docs/analytics/quickstarts.

[13] https://peach.blender.org/

[14] https://bitmovin.com/encoding-service/

measurement platform for multi-homed and large scale end-to-end experimentation in commercial MBB networks. The platform comprises a set of 150 nodes, both mobile (e.g., operating in delivery trucks and on board public transport vehicles, such as trains and buses) and stationary (e.g., volunteers hosting nodes in their homes). Each node is multi-homed to 3 different Mobile Network Operators (MNOs) using commercial-grade subscriptions in several countries in Europe. MONROE is currently operational in Italy, Norway, Spain, Sweden.

MONROE nodes integrate 2 small programmable computers and run software based on Debian GNU/Linux. Each node collects metadata from the modems, such as carrier information, technology, signal strength, GPS location and sensor data. Metadata information from classes including `modem`, `GPS`, `sensor`, and `event` is made available to experimenters during execution. All software components used in the platform are open source and available under [2].

Since the VBIM client container is based on a MONROE image, the measurement nodes of this platform can be used as streaming clients for large scale experimentation in multiple operational MBB networks.

2.3 Bitmovin Analytics Integration

In our current setup, the landing pages for all players include an integration with Bitmovin Analytics web collector v2.3, in accordance with the guidelines specified under[11]. The JavaScript is executed on the browser of the client in runtime, and more than 70 parameters, including performance metrics such as startup time (including a breakdown into player and video startup times), video quality, number of switches, and stalls, are recorded for each session.

Configuration parameters pertaining to the Bitmovin Analytics module can be passed from the client container, via the server, to the analytics instance, by means of appending these optional parameters to the requested Uniform Resource Locator (URL).

This optional component allows for the collection of a wide range of QoE-related metrics during playback, which can be observed in real-time from the Bitmovin Dashboard, or exported in raw format using the Bitmovin Analytics Export Service.

2.4 Running an Experiment

Since the VBIM client is designed as a Docker container, it can be run on any machine supporting Docker virtualization, including MONROE measurement nodes.

Running standalone: An experiment can be started locally with the `docker run` command and the public image[5]. If a configuration file is not specified, the experiment uses the default values.

Running on MONROE: In order to run an experiment on deployed MONROE nodes, the web scheduler[15] can to be used. It allows for the container image and configuration parameters to be specified from the Graphical User Interface (GUI), along with the country, number and type of nodes, data and traffic quotas, time and date of execution. Periodic measurements can also be scheduled from the interface using the "recurring" option. Experiment results can be retrieved from the same interface.

Output files: There are 5 types of files that can be retrieved from the client-side for each session. All of them follow a naming convention where the filename stub `MONROE.EXP.VBIM_NODE.<nodeid>_INTERFACE.<interfacename>_TIME.<time>_SESSION.<sessionid>` is appended by a suffix relevant to the file.

The *Configuration Log* is a JSON file, tagged with the filename suffix: `_SUMMARY.json`. It contains a summary of all the experiment settings, for a given session. The *Ping Log* This is a JSON file, tagged with the filename suffix: `_PING.json`. It contains a parsed summary of the ping measurement results, for a given session. The *Traceroute Log* is a JSON file, tagged with the filename suffix: `_TRACEROUTE.json`. It contains a parsed summary of the traceroute results, for a given session. The *Console Output* is a JSON file, tagged with the filename suffix: `_CONSOLEOUTPUT.json`. It contains the complete console output from the Chrome browser, for a given session (logging preferences set to `'browser':'ALL'`). *Metadata* is a compressed folder, tagged with the filename suffix: `_METADATA.tar`. It contains one JSON file per modem event[16], for a session.

Additionally, the optional Bitmovin Analytics metrics can be exported as a CSV file from the Bitmovin Analytics backend using the Bitmovin Analytics Exporter Service, by specifying a valid user license and a time period. This file aggregates the available information related to all the streaming sessions undertaken with the same user license. It contains one entry per each player "event" (in all of the sessions), where the `sessionId` field can be used to group the entries related to each session.

Retrieving results: If the experiment is run locally, all the client-side output files can be found in the directory specified by the `Docker` command. If the client is a MONROE node, the results can be retrieved from the MONROE database with the import script found in our repository. Bitmovin Analytics metrics can either be exported as described above, or viewed in real-time using the Bitmovin Dashboard.

3 USE CASES AND INTENDED AUDIENCE

The VBIM measurement framework, and respective datasets that can be collected with it, are intended for developers and researchers. VBIM allows conducting repeatable large-scale experiments in operational networks and use the results for a wide range of purposes, e.g., (1) comparing video streaming performance in mobile and fixed broadband networks, (2) benchmarking MNOs, video players, ABR algorithms, (3) investigating the influence factors from all layers of the communication stack on video streaming QoE and player behavior, (4) deriving insights about streaming performance under user mobility, (5) investigating and comparing existing QoE models, (6) creating ML/AI models for player behavior prediction in different networks, and/or (7) developing player selection algorithms depending on the location of the client.

Figure 2 demonstrates an example use case, by comparing 2 streaming sessions in terms of the segment qualities they have utilized from the video asset's bitrate ladder. The y axis corresponds to the frequency of requests they have made for segments with a particular

[15] https://monroe-system.eu, requires valid certificate

[16] In the case of MONROE, this is a `MONROE.META.DEVICE.MODEM` event, and includes information fields such as the name of the active interface, network operator, latitude, longitude, MCC-MNC, CID, LAC, and signal strength (RSSI, RSRP, RSRQ) whenever available.

bitrate, where the x axis represents the bitrate ladder of the video asset. We see that the first session achieves a higher *maximum* quality, whereas the second session achieves a higher *average* quality.

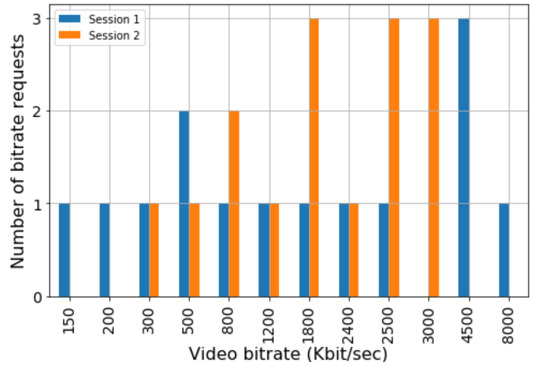

Figure 2: Bitrate distribution for 2 sessions using the same video player.

Figure 3 demonstrates another use case, by benchmarking the startup times of 3 video players using the 4G networks of 7 different MNOs (network operators anonymized). We can observe that the player performances can vary significantly from one network to another, with `player 2` causing the highest startup times overall. It is possible to conduct such joint inquiries, as well as single-dimension benchmarks.

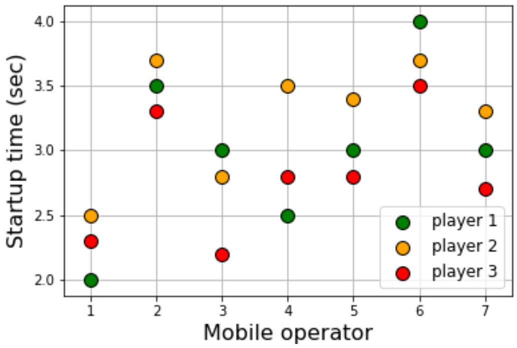

Figure 3: Startup time for multiple sessions using different players in different networks.

4 CONCLUSION

We present an extensive measurement framework (VBIM) for running large-scale end-to-end video streaming experiments in operational networks, using different players and rate adaptation algorithms, and monitoring QoE-related metrics alongside application and network parameters in real-time. This framework can be used to derive insights related to different influence factors on video streaming performance, such as network operator, technology, location, user mobility, video player, and adaptation algorithm. It can be used in conjunction with existing infrastructure, as well as standalone. A live demonstration integrating the MONROE testbed and Bitmovin Analytics will be possible during the presentation of this work, aiming to show the added value of such platforms.

Although our current setup has been established primarily with Video-on-Demand (VoD) services in mind, it is possible to use this framework with live streams as well, provided that the stream is available from a fixed URL in an appropriate format (e.g., DASH manifest or HLS playlist), and live playback is supported by the selected player. This requires a single-line change in our source code corresponding to the landing pages.

As future work, we are planning to integrate a number of QoE models into the VBIM framework, such as the ITU P.1203[17], to automatically derive session scores from the currently available metrics. We are also interested in validating the framework, in terms of the influence of the active monitoring itself on the streaming performance (more specifically, the quantification of this influence, and the consequent demonstration that the effect is negligible), as well as demonstrating the use cases we have addressed here with a large-scale measurement campaign in MBB.

ACKNOWLEDGMENTS

This work was funded by the Austrian Research Promotion Agency (FFG) under the Next Generation Video Streaming project No. 861341 ("PROMETHEUS"), EU H2020 research and innovation programme under grant agreement No. 644399 ("MONROE"), and the Norwegian Research Council project No. 250679 ("MEMBRANE").

REFERENCES

[1] Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update, 2016–2021 White Paper, 2019.

[2] MONROE Project Repository. https://github.com/MONROE-PROJECT/, 2019.

[3] S. Ahsan et al. Impact of Duration on Active Video Testing. In *Proceedings of the 26th International Workshop on Network and Operating Systems Support for Digital Audio and Video (NOSSDAV)*, 2016.

[4] Z. Akhtar et al. Oboe: Auto-tuning Video ABR Algorithms to Network Conditions. In *Proceedings of the 2018 Conference of the ACM Special Interest Group on Data Communication (SIGCOMM)*, 2018.

[5] Ö. Alay et al. Experience: An Open Platform for Experimentation with Commercial Mobile Broadband Networks. In *Proceedings of the 23rd Annual International Conference on Mobile Computing and Networking (MobiCom)*, 2017.

[6] M. P. Quirós et al. Results from Running an Experiment as a Service Platform for Mobile Networks. In *Proceedings of the 11th Workshop on Wireless Network Testbeds, Experimental evaluation & CHaracterization (WiNTECH)*, 2017.

[7] D. Raca et al. DASHbed: a testbed Framework for Large Scale Empirical Evaluation of Real-Time DASH in Wireless Scenarios. In *Proceedings of the 10th ACM Multimedia Systems Conference (MMSys)*, 2019.

[8] A. Schwind et al. Concept and Implementation of Video QoE Measurements in a Mobile Broadband Testbed. In *IEEE/IFIP Workshop on Mobile Network Measurement (MNM)*, 2017.

[9] A. Schwind et al. Dissecting the performance of youtube video streaming in mobile networks. *International Journal of Network Management*, 2019.

[10] S. Xu et al. Dissecting VOD Services for Cellular: Performance, Root Causes and Best Practices. In *Proceedings of the 2017 Internet Measurement Conference (IMC)*, 2017.

[11] A. Zabrovskiy et al. AdViSE: Adaptive Video Streaming Evaluation Framework for the Automated Testing of Media Players. In *Proceedings of the 8th ACM on Multimedia Systems Conference (MMSys)*, 2017.

[12] G. Zhang et al. Ensemble Adaptive Streaming - A New Paradigm to Generate Streaming Algorithms via Specializations. *IEEE Transactions on Mobile Computing*, 2019.

[17] https://www.itu.int/rec/T-REC-P.1203.3-201710-I