Bandwidth Prediction Schemes for Defining Bitrate Levels in SDN-enabled Adaptive Streaming

Ali Edan Al-Issa[§], Abdelhak Bentaleb[*], Alcardo Alex Barakabitze[§], Thomas Zinner[†], Bogdan Ghita[§]
[§]Centre for Security, Communications and Network Research, [*]School of Computing, [†]INET: Internet Network Architectures,
[§]Plymouth University, [*]National University of Singapore, [†]TU Berlin,
E-mail: ali.alissa, alcardoalex.barakabitze, bogdan.ghita@plymouth.ac.uk, bentaleb@comp.nus.edu.sg, zinner@inet.tu-berlin.de

***Abstract*—The majority of Internet video traffic today is delivered via HTTP Adaptive Streaming (HAS). Recent studies concluded that pure client-driven HAS adaptation is likely to be sub-optimal, given clients adjust quality based on local feedback. In [1], we introduced a network-assisted streaming architecture (BBGDASH) that provides bounded bitrate guidance for a video client while preserving quality control and adaptation at the client. Although BBGDASH is an efficient approach for video delivery, deploying it in a wireless network environment could result in sub-optimal decisions due to the high fluctuations. To this end, we propose in this paper an intelligent streaming architecture (denoted BBGDASH$^+$), which leverages the power of time series forecasting to allow for an accurate and scalable network-based guidance. Further, we conduct an initial investigation of parameter settings for the forecasting algorithms in a wireless testbed. Overall, the experimental results indicate the potential of the proposed approach to improve video delivery in wireless network conditions.**

***Index Terms*—SDN, QoE; DASH; OpenFlow; Stability; Adaptive Video Streaming; Streaming architecture; Bounded Bitrate Guidance; Network-Assistance**

I. INTRODUCTION

Multimedia data currently represents the majority of Internet traffic. The last report of Cisco [2] indicates that video traffic will represent 82% of all consumer Internet traffic by 2022. To cope with this increase, most service providers have adopted the HTTP adaptive streaming (HAS) technology [3], which provides a flexible, client-based adjustment of video quality and streaming bitrate. HAS covers different implementations that have similar adaptation principles, with Dynamic Adaptive Streaming over HTTP (DASH) [4] being one of the most popular ones. In principle, video content in HAS is encoded by the streaming source with multiple bitrate representations and chunked into small segments of video duration (i.e. 2 to 15 seconds). The client includes an adaptation algorithm that employs different approaches dynamically adapt the requested quality level to the network conditions. Despite the great benefits they bring, purely client-based HAS solutions also face challenges relating to quality fluctuations when competing for a shared link [5].

The defining metric for benchmarking the success of video applications is Quality of Experience (QoE). In an adaptive context, QoE is optimised by estimated throughput or the occupancy of the player's playback buffer and selecting an appropriate bitrate. This aligns video delivery to the fluctuating in network conditions, but may also lead to incorrect or out-of-sync adaptation when network conditions fluctuate too rapidly and, subsequently, to a high number of bitrate changes and buffer starvation, particularly in a wireless environment.

Research studies [6], [7] demonstrated that reliable bandwidth prediction could boost the performance of the adaptation algorithms, especially when it is combined with rate stabilization functions at the client side. In addition software-defined networking (SDN) provides a centralised network control which can be used for enhancing the network-based video delivery.

This paper presents two time-series-based forecasting approaches for identifying clients bitrate levels and therefore providing enhanced client assistance for delivering optimal QoE. Second, it extends our previous architecture [1] to an intelligent streaming architecture (denoted BBGDASH$^+$), also investigated in this paper in terms of performance and potential benefits.

The rest of this paper is organised as follows. In section II we present the related work. Section III presents the description of the proposed QoE-driven network-assisted architecture for HTTP adaptive video streaming BBGDASH$^+$. Section IV provides the system model and the algorithm description. Section V presents an experimental setup while section VI provides the performance analysis and results. Finally, we conclude this paper in Section VII.

II. RELATED WORK

The use of HAS for video delivery led to new approaches that optimise the perceived QoE by adapting the requested video quality according to network conditions. Most of the studies either locally optimise the adaptation decision or delegate the decision logic to an additional central entity (e.g. as part of SDN). Jiang *et.al* [8] propose a client-based approach, named FESTIVE, which consists of three modules (bandwidth estimation, bitrate selection, and randomised chunk scheduling) to avoid video segments overlapping among the different players. Further, Spiteri *et.al* introduce in [9] a Buffer Occupancy based Lyapunov Algorithm (BOLA), in which the ABR decision is formulated at the client side as a utility maximisation problem. On the other hand, the centralised alternatives aim to optimise the received QoE and provide a fair allocation among users. Bentaleb *et.al* presents in [10] an

978-3-903176-24-9 © 2019 IFIP

SDN-enabled DASH architecture (SDNDASH) to maximise per-client QoE by dynamically allocating network resources and guide each client for the optimal bitrate level for the next chunks to be downloaded. Authors in [11] design a QoE-based SDN architecture that considers the cooperation between network operators and content providers. Further, Cofano et al., [12] investigated several network-assisted streaming strategies that allow active cooperation between the video streaming applications and the network.

We note that, with purely client-based HAS adaptations, the distributed per-client based optimization might lead to an unfair QoE allocation among the involved clients. On the other hand, moving the adaptation logic to an external element affects the endpoint-based adaptation principle at the core of HAS and may lead to scalability issues.

Motivated by the above challenges of HAS and the scalability issues of moving the whole decision logic to an additional entity, we presented our previous architecture [1] as a proof-of-concept implementation for the hybrid bitrate guidance. In our previous work, the network component guides clients with a set of bitrate levels to let them adapt the quality locally. However, identifying the number of bitrate levels for each client in a wireless network condition is not a trivial issue, and it could result in a significant QoE degradation when the client be bounded with a wrong set of bitrate levels. Therefore, in order to cope with this issue, we present two time-series-based forecasting approaches that identify the optimal set of bitrate levels for each client base on the network conditions. Additionally, we extend the implementation of our previous architecture to an intelligent streaming architecture (denoted BBGDASH$^+$). Further, we conduct a set of experimental evaluations with different configuration parameters to investigate the behaviour of the proposed approaches in real network conditions.

III. System Architecture

In this section, we present the proposed system architecture for bounded bitrate guidance that consists of three planes, as shown in Fig1. We describe each plane as follows:

A. Data Plane

The data plane consists of a set of SDN-based forwarding devices. These devices (i.e. Open vSwitch(OVS)) are responsible for forwarding network flows and allocating the available resources based on the received policies from the control plane via the southbound interface.

B. QoE Management Plane

The QoE management plane consists of six modules, namely BBGDASH$^+$ manager, Bandwidth Estimator, QoE Monitor, Database Archive, Policy Enforcer, and Bandwidth Forecasting . We provide the details of each component in the proposed architecture as follows:

1) BBGDASH$^+$ Manager: The BBGDASH$^+$ manager is responsible for computing the optimal set of bitrate levels per DASH player. It communicates with the other components of the QoE management plane in order to get the required information (i.e. number of active DASH player, available resources, etc.) for allocating network resources and guiding DASH player.

2) Bandwidth Estimator: The bandwidth estimator requires previous knowledge of the link capacity of the network path to measure the available bandwidth. In the proposed architecture, the control plane frequently queries the forwarding devices to get the network statistics. These collected network statistics are then smoothed and archived to be used as an input for an ARIMA-based time-series forecasting.

3) QoE Monitor: The QoE monitor module performs real-time measurement for the perceived QoE. This module considers the devices, video, and other QoE related metrics for calculating the end-user QoE. Furthermore, this module is sufficiently flexible to allow the deployment of any other QoE-metrics based model for measuring the perceived QoE at the end-user.

4) Database Archive: The database module stores information related to the number of DASH clients, configuration parameters, estimated bandwidth and the QoE metrics of video streaming sessions, which is used as input by the BBGDASH$^+$ manager module and the forecasting module to compute the suitable set of bitrate levels. As shown in Fig. 1, the database is used to store network stats and DASH flow rules that are currently active in the streaming sessions.

5) DASH Server and Client Entities: The DASH server hosts DASH content, encoded into multiple versions and segmented into small chunks, accompanied by manifest files describing the content. At the client-side, a modified dash.js player embeds the proposed guidance schema. In addition, the MPEG Server and Network Assisted DASH (SAND) architecture [13] sends the QoE-related metrics and receives the optimal guidance to/from the proposed BBGDASH$^+$ agent.

6) Policy Enforcer: This module applies the BBGDASH$^+$ recommendation at application and the network levels. The application's levels action is implemented by sending the bitrate levels range for each DASH player while the network level action is applied through allocating the network resource dynamically. This module also informs the video server about each DASH client levels which, in turn, prevents the greedy clients from requesting higher bitrate levels than the ones allocated by BBGDASH$^+$ manager.

C. SDN Control Plane

The control plane aims to support the delivery of video services and to provide the QoE-based resource allocation per DASH client; in order to provide this functionality, it interfaces between the data plane and the QoE management plane. In the proposed architecture, we implemented the network agent that carries instructions or defined policies from the QoE management plane and translate those policies into a set of rules or actions on the data forwarding plane.

IV. System Model and Problem Formulation

A. System Model

We model the system as an undirected graph $\mathcal{G} = (\mathcal{X}, \mathcal{Y})$, where $\mathcal{X} = \{P, F, A, S\}$ represents the set of nodes and $\mathcal{Y}$ is

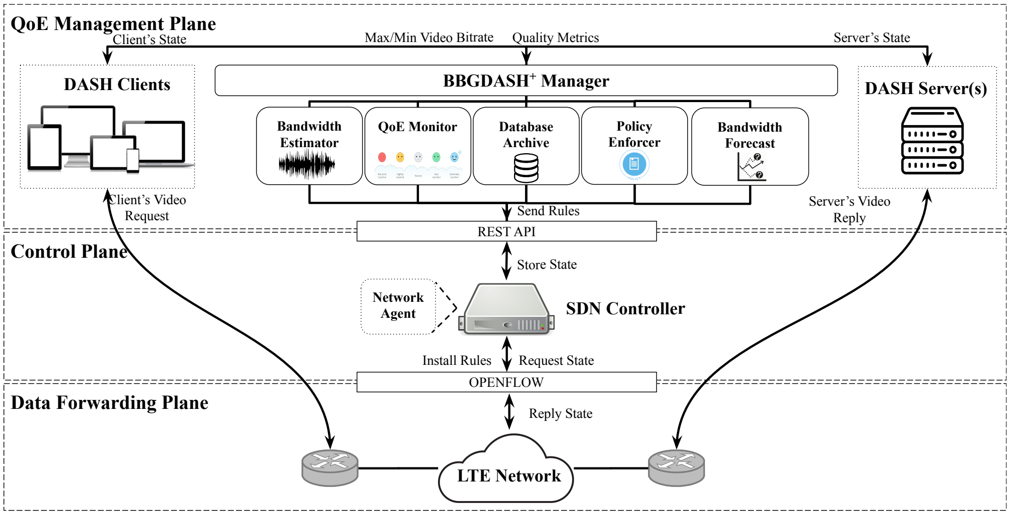


Fig. 1: Proposed QoE-Driven Network-Assisted Architecture for HTTP Adaptive Video Streaming

the set of links between the nodes. The set of nodes in the proposed architecture includes a number of subsets encompassing a number of DASH players $p_i \in P$, one DASH server S, forwarding devices F, and the SDN controller A. Each player p_i has at least one link $y \in Y$ connecting it to the access node $f \in F$. We consider that the only network bottleneck is in the access network with total estimated capacity of BW_e.We note that each player $p_i \in P$, $i = [1, \ldots, N]$, has a specific set of requirements for screen resolution r_{pi} and plan subscription ST_{p_i}, and may request a video $v_{p_i} \in V$ that has a distinct set of bitrate representations $L^{v_{pi}}$.

Given the context, enforcing the same bitrate allocation for all players may result in an unfair distribution and inefficient of both QoE and network resources allocation. In order to meet the end-user QoE requirements and achieve a high level of fairness among the players that request video with different characteristics, then we an intelligent a network-assisted that base on the concept of *max-min fairness* for allocating the available resources. The main objective is to maximise the minimum quality across the p_i players. The objective function f is described as follows:

$$f = \begin{cases} \max\left(min_{l^j_{p_i} \in L^{v_{p_i}}}\right) \\ s.t \ \sum_{i=1}^{N} BW_{p_i} \leq BW_e, \\ \qquad BWp_i \geq 0, l^j_{p_i} \in L^{v_{p_i}} \\ \qquad \forall p_i \in P, i = [1, \ldots, N], j = [1, \ldots, M_{p_i}], \end{cases} \tag{1}$$

where BW_{p_i} is the bandwidth allocated for p_i, $l^j_{p_i}$ is the selected bitrate by p_i, and M_{p_i} is the total number of bitrate levels for the video v_{p_i}.

B. ARIMA-based Bandwidth Forecasting

Future bandwidth prediction is a time series forecasting problem [14] where, for a given timeslot, the objective of the prediction model (μ) is to accurately find the predicted bandwidth given a set of previous bandwidth measurement samples. Formally, given a set of n time series observations at time elapsed t, denoted y_t, a future predicted value of time series for next h steps (horizon), denoted $\hat{y}_{t+h}$, is defined as:

$$\hat{y}_{t+h} = \mu(y_t, y_{t-1}, y_{t-2}, \ldots, y_{t-n}). \tag{2}$$

In this context, the main challenge is choosing the model that provides the highest forecasting accuracy. Several studies [6], [7] acknowledged that statistical approaches tend to have a higher forecasting accuracy than machine learning models [15]. Following from this conclusion, this study applied the autoregressive integrated moving average (ARIMA), a classical and universal statistical modelling tool, for predicting $\hat{y}_{t+h}$, given its ability to handle non-stationary time series data through applying the differencing method. In order to model ARIMA processes, we consider the variable y_t as a set of the network bandwidth measurement samples. Therefore, the first part of the ARIMA model is an autoregressive (AR) process of order p for the number of time lags y_t, as follow:

$$(1 - \phi_1 B - \cdots - \phi_p B^p) \ \hat{y}_{t+h} = c + \varepsilon_t. \tag{3}$$

where c is a constant, ϕ_j are coefficients or the parameters of the moving average part and ε_t is the error term.

Another process of the ARIMA model is the moving average (MA) of order q, which is a linear combination of the current white noise term and the q of the most recent past white noise terms, and can be defined as follow:

$$\hat{y}_{t+h} = c + (1 - \theta_1 B - \cdots - \theta_q B^q) \ \varepsilon_t, \tag{4}$$

where c is constant and ε_t are the error terms.

The AR and MA models can be integrated along with differencing the time series to give a wide variety of effects.

This combination leads to an *ARIMA(p,d,q)* model, which can be written in backshift notation as:

$$(1-\phi_1 B-\cdots-\phi_p B^P)\ (1-B)^d\ \hat{y}_{t+h} = c+(1-\theta_1 B-\cdots-\theta_1 B^q)\ \varepsilon_t, \tag{5}$$

where ϕ_i, θ_i are parameters to be determined.
The first 95% confidence interval for the predicted data can be given as:

$$\alpha = \hat{y}_{t+h} \pm 1.96\hat{\sigma}, \tag{6}$$

where $\hat{\sigma}$ is the standard deviation of of the residual errors.
We apply root-mean-square-error (RMSE) to evaluate the accuracy of the forecasting model and the goodness of its parameters, as it indicates the squared error between the predicted $\hat{y}_{t+h}$ and the observed value(s) for the same throughput levels.

C. Throughput prediction

The main two elements in the proposed forecasting methodology are to depict the best fit of the ARIMA model that achieves the highest accuracy. The $auto.arima()$ Python library function was used to determine the optimal online fit of the ARIMA model. The function is based on a variation of the Hyndman-Khandakar algorithm [16], which integrates unit root tests, maximum likelihood estimation, and cross-validation techniques. The algorithm starts with identifying the number of differences d required to generate a stationary data. After differencing the data d times, the algorithm chooses values of p and q by minimizing the Akaikes Information Criterion (AIC), which utilises the stepwise search to navigate the model space rather than considering every possible combination of p and q.

An accurate bandwidth prediction is essential for providing proper guidance for DASH players. The presented methodology starts with splitting the traces into two datasets. The training set comprises 60% of the trace and is used for validating the parameters of the prediction model, while the validation set is used for estimating the prediction error. The prediction methodology retrains the prediction model with the new observed values every h seconds. Further, we fixed h in this study and to be proportional to the duration of the requested video chunk.

D. Video Bitrate Boundary Identification

The main objective of the proposed approach is to identify the optimum set of bitrate levels that DASH players should be using to adopt the quality locally. To achieve that, we introduce two algorithms, namely Confidence-Based Bounding (CBB) (Algorithm 1) and Error-Based Bounding (EBB) (Algorithm 2) that exploit the power of time series forecasting to identify the optimal boundaries of the requested bitrate.
For each step i, CBB starts with forecasting the throughput $\hat{y}_{t+h}$ for the horizon h, $h>0$, then it computes the confidence interval value (α), which identifies the initial boundaries of the requested bitrate levels. In the next step, the algorithm maps the continuous values of the computed boundaries into discrete bitrate values based on Algorithm 3 and it ends by sending the computed boundaries to each DASH player.

Algorithm 1 Confidence Based Bounding (CBB)

```
for each $step\ i$ do
    Forecast the throughput $\hat{y}_{t+h}$ using (5).
    Find the initial boundaries using (6).
    Quantise the initial boundaries to the discrete video bitrate boundaries: Call algorithm 3
```

In the Algorithm 2, we replaced step 3, which defines the initial boundaries based on the prediction error rather than the confidence interval.

Algorithm 2 Error Based Bounding (EBB)

```
for each $step\ i$ do
    Forecast the throughput $\hat{y}_{t+h}$ using (5).
    Find the initial boundaries as:
        $\alpha = \hat{y}_{t+h} \pm RMSE$    (7)
    Quantise the initial boundaries to the discrete video bitrate boundaries: Call algorithm 3
```

E. Perceptual Quality and Cluster Identification

Video perceptual quality measurement has a non-linear relationship with bitrate [10], [17], [18]. To embed this in our approach, we adopt the bitrate to perceptual quality mapping function from [19], which takes three features of a video streaming session including device resolution, content type, and service plan type and then map all of them in one common SSIMplus-based space (the Structural SIMilarity plus index) [20]. With this model, the existing BBGDASH$^+$ players can be clustered into five non-overlapping clusters denoted $Cl = \{Cl_1, \ldots, Cl_5\}$. Hence, clustering players into a set of clusters helps BBGDASH$^+$ to send per-cluster bitrate recommendation rather than per-client.

F. QoE-driven quality optimization

To achieve a fair QoE allocation among DASH players and an efficient resource utilization, we present a dynamic programming based algorithm that provides each player with the optimal bitrate guidance. The proposed Algorithm 3 provides fair QoE distribution and efficient resource allocation among different players with different requirements that may request videos with heterogeneous representation levels.

V. EXPERIMENTAL SETUP

The proposed architecture in Section III has been implemented in a testbed environment in order to investigate the performance of the proposed approaches. In this section, we explain the testbed setup and the experimental methodology.

TABLE I: Notation and Symbols Description.

Symbol	Descriptions
N	Total Number of DASH players
Cl	Set of clusters $cl_k, k = [1, \dots, 5]$ of specific screen resolution r_{cl_k}
N_{cl_k}	Number of DASH players in cluster cl_k
$l_{cl_k}^{ini,p_i}$	Initial bitrate selection for player p_i in cluster cl_k
$l_{cl_k}^{opt}$	Optimal bitrate for cluster cl_k
S_f	Scaling factor
BW_e	The total available bandwidth for HAS players at time t
BW_{p_i}	The bandwidth consumed by player p_i
BW^{left}	The remaining bandwidth after the first allocation
$L^{v_{p_i}}$	The complete set of bitrate levels $[l_{p_i}^1 : l_{p_i}^{M_{p_i}}]$ of video v_{p_i} requested by player p_i
$l_{p_i}^{diff}$	The difference between the current bitrate level and the next level of the video v_{p_i} requested by player p_i
h	Prediction horizon related to the duration of the requested video chunk
V_1	Maximum allowed bitrate set for the players
V_2	Minimum allowed bitrate set for the players
α	The set of the max. and min. bandwidth computed by the prediction alg.

A. Evaluation Testbed

A set of experiments was conducted on an SDN-based testbed. The data plane was implemented on Mininet using a DASH player connected to an Apache server serving video content via two OpenFlow switches OVS1 and OVS2. The control plane was implemented using Ryu SDN [21]. On the video server side, we used the 600s Big Buck Bunny video, encoded using FFmpeg at 4 different resolutions (360p, 480p, 720p, and 1080p) and 20 bitrate levels using two passes. Then each video is segmented into set of 4 seconds chunk using GPAC MP4Box [22]. The client side used the dash.js JavaScript-based DASH player [23]. Additionally, in order to allow DASH clients to send the QoE-related metrics and receive the optimal guidance to/from the proposed BBGDASH$^+$ agent (see Fig. 1), a websocket channel has been initiated between DASH players and the BBGDASH$^+$ agent. For the archiving purposes, MySQL V5.7 is deployed in management plane for archiving the QoE and the network related parameters.

B. Experiment Design

The implemented testbed was used to investigate the performance of the presented approaches under wireless network conditions, where the throughput is subject to considerable fluctuations. In order to replicate the same network conditions, we shaped the link between OVS1-OVS2 based on a real set of traces gathered from IEEE 802.11 wireless local area networks (WLANs) [24].

***First Experiment*:** The first experiment aims to investigate the impact of the forecasting horizon and the sampling rate of the network bandwidth measurement on the accuracy of the prediction algorithm and their implications on the proposed schemes (i.e. EBB and CBB). To find the optimal configurations, we investigate the prediction algorithm under four different horizons (i.e. 2, 4, 8, and 12 seconds). Further, for each horizon, different sampling rates of the network bandwidth measurement have been also examined.

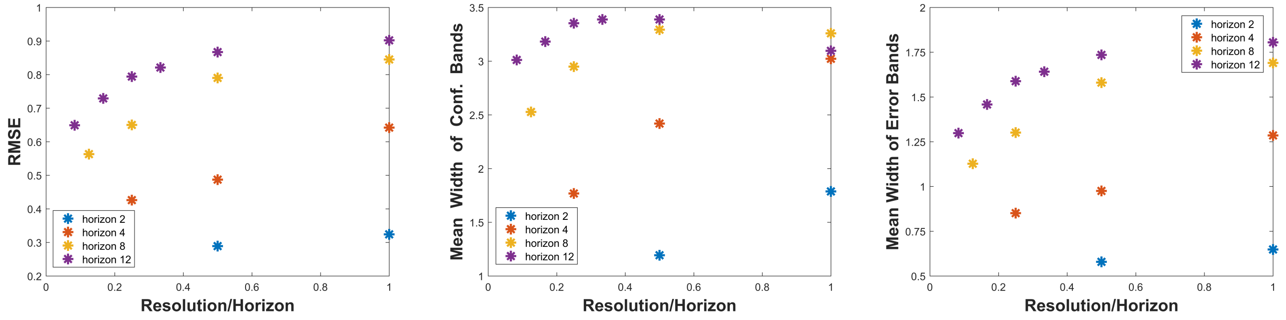
Algorithm 3 Bitrate Selection Algorithm

```
Input: $N, L^{v_{p_i}}, Cl, N_{cl_k}, l_{cl_k}^{opt}, \alpha$
for $j \leftarrow 1$ to 2 do
    CALCULATE $S_f \leftarrow \frac{\alpha[j]}{\sum_{k=1}^{5} N_{cl_k} * l_{cl_k}^{opt}}$
    for each $cluster\ cl_k \in Cl$ do
        $l_{Cl_k}^{ini,p_i} \leftarrow l_{cl_k}^{opt} * S_f$
    for each $cluster\ cl_k \in Cl$ do
        for each $player\ p_i \in P$ do
            FINDMAX$(l_{p_i}^{\sigma} \in L^{v_{p_i}}) \mid l_{p_i}^{\sigma} \leq l_{Cl_k}^{ini,p_i}$
            INSERT$(l_{p_i}^{\sigma})\ into\ V_1[]$
        for each $player\ p_i \in P$ do
            if $l_{p_i}^{\sigma} \neq l_{p_i}^{M_{p_i}}$ then
                FIND$(l_{p_i}^{\sigma+1})$
                $l_{p_i}^{diff} = (l_{p_i}^{\sigma+1}) - (l_{p_i}^{\sigma})$
            else
                $l_{p_i}^{\sigma+1} = \infty$
            INSERT$(l_{p_i}^{\sigma+1})\ into\ V_2[]$
            INSERT$(l_{p_i}^{diff})\ into\ V_3[]$
        CALCULATE$(BW^{left}) = \sum_{i=1}^{N} l_{Cl_k}^{ini,p_i} - l_{p_i}^{\sigma}$
        SORT$(V1)$
        while $BW^{left} \geq$ MIN$(V_3)$ do
            for each $l \in V_1$ do
                if $V_3[l] \leq BW^{left}$ and $V_2[l] \neq \infty$ then
                    $V_1[l] = V_2[l]$
                    $BW^{left} = BW^{left} - V_3[l]$
                else CONTINUE()
        RETURN$(V_j)$
```

***Second Experiment*:** The goal of the second experiment is to evaluate the stability of the received video when a DASH player stream a video under wireless network conditions without any application/network-assistance. In this experiment, DASH player relies only on the local ABR algorithm (i.e. throughput-based) to estimate the network conditions and to adapt the quality of the requested segment.

***Third Experiment*:** In the third experiment, the first network-based bitrate guidance approach (i.e. CBB) is investigated under a different set of configurational parameters (i.e. horizon and sampling rate) to see the efficiency of the proposed approach for providing an efficient video delivery. The potential of CBB has been evaluated when it runs under four different horizons (i.e. 4, 6, 8, and 12 sec.). Further, for each of the examined horizon h, two different sampling rates of the network bandwidth measurement (i.e. 1 and h seconds) are provided as an input to the prediction algorithm.

***Fourth Experiment*:** Another network-assistance bitrate guidance approach (i.e. EBB) is evaluated in the fourth experiment to be compared with the other approaches. For a fair comparison, the same configurational parameters of the third



(a) Prediction Accuracy.

(b) Mean Width of Conf. Bands.

(c) Mean Width of Error Bands

Fig. 2: Impact of measurements' resolution and forecasting horizon on the prediction schemes

experiment have been deployed with this experiment.

VI. EXPERIMENTAL RESULTS

This section presents the results of the experiments described above to evaluates the proposed error-based and confidence-based bounding guidance. The overall aim is to gradually determine the effectiveness of the two methods, from accurately predicting the bandwidth, through enforcing it in a timely fashion, and ultimately achieving better QoE.

1) Prediction Accuracy: Fig. 2a shows the results of horizon/sampling rates(resolution) against the forecasting accuracy that are measured with the RMSE. It can be observed from Fig. 2a that, the lowest horizon has the highest accuracy of prediction under the highest sampling rates. This implies that, at 0.5, the RMSE for horizon 2 is 0.3 giving a good prediction accuracy compared to horizon 8 and 12, which lead to a large RMSE of 0.8 and 0.9 respectively.

2) Mean Width of Confidence Bands: Fig. 2b shows the results of horizon/sampling rates (resolution) against the mean width of confidence bands. The results indicate that as the horizon increases the mean width of the confidence bands also increase. For example, at 0.5 the mean width of confidence bands for horizon 8 and 12 is 3.3 Mbps and 3.4 Mbps respectively. Similarly, a lower horizon provides a lower mean width of confidence bands; at 0.5, a horizon value of 2 results in a mean width of the confidence bands of 1.2 Mbps.

3) Mean Width of Error Bands: Fig. 2c indicates the results of the horizon/sampling rates(resolution) against the mean width of the error bands. It can be observed that the mean width of the error bands has a lower value than the mean width's value of the confidence bands, as the later bounds the true values with probability of 0.95. For example, at 0.5, horizon 8 and 12 have a mean width error bands of 1.58 Mbps and 1.73 Mbps respectively. However, the lowest horizon provides a lower mean width of the error bands. Horizon 2, for example, leads to a mean width error bands of 0.56 Mbps at 0.5 while that of horizon 4 is 0.96 Mbps.

4) The Performance of Normalized QoE: In Fig. 3 the different approaches are compared in term of the normalized QoE when they have been deployed under different combinations of the forecasting horizons and measurement sampling rates (resolutions). The results show that the end user's QoE using EBB/R=1 under a forecasting horizon of 4 outperforms the other approaches, as the last provides the highest average bitrate levels and the lowest switching amplitude. It is worth mentioning here that EBB is more sensitive for the prediction accuracy than other approaches. Further, It is obvious that the non-assisted approach (Dash.js) performs poorly in every aspect compared to other approaches.

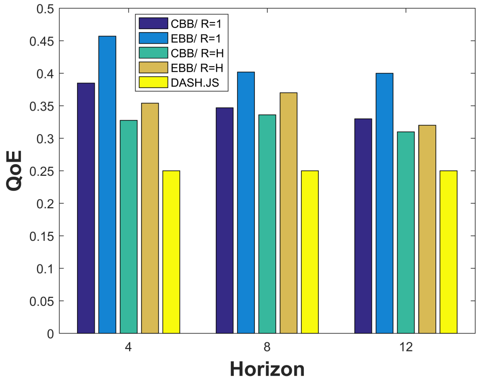


Fig. 3: Normalised QoE.

VII. Conclusion

This paper present a novel QoE-driven network-assisted architecture for HTTP adaptive video streaming called BBGDASH$^+$ that provides a real-time bandwidth prediction using SDN. We propose an Error-Based Bounding (EBB) and Confidence-Based Bounding (CBB) that exploit the power of time series forecasting to identify the optimal bitrate boundaries of the requested bitrate in a wireless environment. We conduct an initial investigation of the proposed system to evaluate the impact of the configurational parameters (i.e. prediction horizon and measurement sampling rate) on the accuracy of the predicted boundaries. The results show that, the proposed algorithms (EBB and CBB) can significantly improve the end-users QoE compared to the purely client based HAS applications.

References

[1] A. Alissa, A. Bentaleb, T. Zinner, B. Ghita, and I. Mkwawa., "A Max-Min Bounded Bitrate Guidance for SDN Enabled Adaptive Video Streaming," *22nd Conference on Innovation in Clouds, Internet and Networks (ICIN*, February 2019.

[2] "Cisco Visual Networking Index: Forecast and Trends, 2017 - 2022 White Paper." https://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/white-paper-c11-741490.html.

[3] T. Stockhammer, "Dynamic adaptive streaming over http–: standards and design principles," in *Proceedings of the second annual ACM conference on Multimedia systems*, pp. 133–144, ACM, 2011.

[4] I. Sodagar, "The MPEG-DASH Standard for Multimedia Streaming Over the Internet," *IEEE MultiMedia*, vol. 18, pp. 62–67, April 2011.

[5] B. Abdelhak, C. B. Ali, Z. Roger, and S. Harous, "SDNHAS: An SDN-Enabled Architecture to Optimize QoE in HTTP Adaptive Streaming," *IEEE Transactions on Multimedia*, vol. 19, pp. 2136–2151, Oct 2017.

[6] T. Mangla, N. Theera-Ampornpunt, M. Ammar, E. Zegura, and S. Bagchi, "Video through a crystal ball: Effect of bandwidth prediction quality on adaptive streaming in mobile environments," in *Proceedings of the 8th International Workshop on Mobile Video*, p. 1, ACM, 2016.

[7] X. K. Zou, J. Erman, V. Gopalakrishnan, E. Halepovic, R. Jana, X. Jin, J. Rexford, and R. K. Sinha, "Can accurate predictions improve video streaming in cellular networks?," in *Proceedings of the 16th International Workshop on Mobile Computing Systems and Applications*, pp. 57–62, ACM, 2015.

[8] J. Jiang, V. Sekar, and H. Zhang, "Improving fairness, efficiency, and stability in HTTP-based adaptive video streaming with festive," *IEEE/ACM Transactions on Networking*, vol. 22, no. 1, pp. 326–340, 2014.

[9] K. Spiteri, R. Urgaonkar, and R. K. Sitaraman, "BOLA: Near-optimal bitrate adaptation for online videos," *Proceedings - IEEE INFOCOM*, vol. 2016-July, 2016.

[10] A. Bentaleb, A. C. Begen, and R. Zimmermann, "SDNDASH: Improving QoE of HTTP Adaptive Streaming Using Software Defined Networking," *Proceedings of the 2016 ACM on Multimedia Conference - MM '16*, pp. 1296–1305, 2016.

[11] E. Liotou, K. Samdanis, E. Pateromichelakis, N. Passas, and L. Merakos, "Qoe-sdn app: A rate-guided qoe-aware sdn-app for http adaptive video streaming," *IEEE Journal on Selected Areas in Communications*, vol. 36, no. 3, pp. 598–615, 2018.

[12] G. Cofano, L. D. Cicco, T. Zinner, A. Nguyen-Ngoc, P. Tran-Gia, and S. Mascolo, "Design and performance evaluation of network-assisted control strategies for http adaptive streaming," *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, vol. 13, no. 3s, p. 42, 2017.

[13] E. Thomas, M. van Deventer, T. Stockhammer, A. C. Begen, and J. Famaey, "Enhancing MPEG DASH Performance via Server and Network Assistance," *SMPTE Motion Imaging Journal*, vol. 126, no. 1, pp. 22–27, 2017.

[14] C. Chatfield, *The analysis of time series: an introduction*. Florida, US: CRC Press, 6th ed., 2004.

[15] S. Makridakis, E. Spiliotis, and V. Assimakopoulos, "Statistical and machine learning forecasting methods: Concerns and ways forward," *PloS one*, vol. 13, no. 3, p. e0194889, 2018.

[16] R. J. Hyndman, Y. Khandakar, *et al.*, *Automatic time series for forecasting: the forecast package for R*. No. 6/07, Monash University, Department of Econometrics and Business Statistics , 2007.

[17] M. Seufert, S. Egger, M. Slanina, T. Zinner, T. Hossfeld, and P. Trangia, "A Survey on Quality of Experience of HTTP Adaptive Streaming," *Ieee Communication Surveys & Tutorials*, vol. 17, no. 1, pp. 469–492, 2015.

[18] G. Cermak, M. Pinson, and S. Wolf, "The relationship among video quality, screen resolution, and bit rate," *IEEE Transactions on Broadcasting*, vol. 57, no. 2, pp. 258–262, 2011.

[19] A. Bentaleb, A. C. Begen, R. Zimmermann, and S. Harous, "SDNHAS: An SDN-enabled architecture to optimize QoE in HTTP adaptive streaming," *IEEE Transactions on Multimedia*, vol. 19, no. 10, pp. 2136–2151, 2017.

[20] A. Rehman, K. Zeng, and Z. Wang, "Display device-adapted video quality-of-experience assessment," in *Human Vision and Electronic Imaging XX*, vol. 9394, p. 939406, International Society for Optics and Photonics, 2015.

[21] "RYU SDN Framework. http://osrg.github.io/ryu/."

[22] "GPAC:Multimedia Open Source Project," *What's New*, pp. 0–60.

[23] "DASH Industry Forum. http://www.dashif.org/.."

[24] K. Miller, A.-K. Al-Tamimi, and A. Wolisz, "Qoe-based low-delay live streaming using throughput predictions," *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, vol. 13, no. 1, p. 4, 2017.