# Anticipating Resource Management and QoE for Mobile Video Streaming under Imperfect Prediction

Imen Triki, Rachid El-Azouzi, Majed Haddad

# Anticipating Resource Management and QoE for Mobile Video Streaming under Imperfect Prediction

Imen Triki, Rachid El-Azouzi and Majed Haddad
LIA/CERI, University of Avignon,
Agroparc, BP 1228, 84911, Avignon, France
Email: {imen.triki , rachid.elazouzi , majed.haddad} @univ-avignon.fr,

***Abstract*—By leveraging geolocation and contextual information for mobile users, the prediction of the future throughput becomes more and more feasible. Many approaches on context-aware content delivery have been explored to balance the operators' limited resources with users' requirements. However, the perfect knowledge of the future context cannot be easily performed in real world, which represents a hurdle for most context-aware approaches. In this paper, we address a context-aware delivery algorithm for adaptive video streaming (NEWCAST) that have already been explored in [1] under perfect knowledge of future capacity, to balance the user's perception of the video and the cost of network usage. In order to make NEWCAST more resistant to eventual throughput prediction errors and adapt it to short-term horizons, we propose 4 algorithms that efficiently reduce the number of stalls by at least $75\%$.**

## I. Introduction

The strong emergence of smartphones on daily life as well as the high broadband access supplied by operators have triggered pervasive demand on video streaming mobile services, requiring the exploration of novel approaches on video content delivery. To afford video streaming services at sustainable costs, the idea of adjusting the bit rate of video traffic depending on the (time-varying) available bandwidth (called adaptive streaming technologies) has been actively investigated during the recent years. In industry, many adaptive video streaming solutions exist and are now undergoing a standardization toward DASH (Dynamic adaptive streaming over HTTP), such as Microsoft's smooth streaming, Adobe's HTTP dynamic streaming and Apple's live streaming. One of the key challenges that DASH presents for content providers, operators and device manufacturers, is to accurately assess the users' perceptions in order to enhance service provisioning and to optimize network adaptation. Since the users' perceptions on the video quality directly impacts their engagement in video streaming sessions, many interests are being given to the user's quality of experience (QoE). Actually, most of DASH solutions manage the video quality based on some predefined key QoE factors such as video freezing frequency, data prefetching delay, total number of switches, etc. These solutions mostly rely on the information of the available link/path bandwidth during the latest time interval to predict the available bandwidth during the next interval. Predicting the available bandwidth accurately is important for a good QoE in video streaming since the video quality changing too often or too severely will negatively affect the users' perception [2].

To overcome this hurdle, context-aware content delivery approaches have been initiated in this sense, steering researchers toward new context-predictive techniques on users' mobility and throughput. At the core, lies the idea of exploiting the strong correlation between users' rates and locations [3], [4] in order to design radio maps where historical average signal strength are geographically mapped [5]. Actually, the main idea behind predicting users' future contextual information is to proactively counter possible service fluctuations and to wisely exploit available resources in the future. Studies on users' mobility patterns have also shown that people daily routes exhibit a high degree of spatial and temporal regularity, especially on public transportation [6] or on road ways to/from frequently visited places. Coupled with radio maps, these mobility patterns can give higher accuracy on average throughput predictability along users' trips. [7] addressed other contextual factors such as user speed, time of day, and humidity to predict users' future throughput with more accuracy based on measurement studies.

Many works on video quality adaptation have been performed since the appearance of adaptive video streaming solutions. In the literature, adaptive streaming protocols are classified in two major classes: throughput-based approaches and buffer-based approaches. While the first class estimates the throughput from previously downloaded segments to adapt the quality of the current segments [8], the second class only sees the playback buffer state evolution to decide whether increasing or decreasing the video bitrate [9]. More and more interests are being steered today toward combining the two latter classes [10].

Despite the richness of research and the very large number of articles that can be found in the literature on video delivery techniques, only few works have exploited the knowledge of future throughput variations for QoE. In [11], authors present a threshold-based transmission schedule which optimizes network utilization while guaranteeing users' QoE for simple classical video streaming. This framework was generalized in [1] to strike a balance between network utilization and users' QoE for adaptive video streaming. In particular, [1] proposes a proactive video content delivery algorithm, called NEWCAST, that adjusts the video quality over a long future horizon, assuming a perfect throughput prediction. Further work in [10], assumes the knowledge of the future and copes with long term throughput prediction to periodically optimize the users QoE.

From a practical point of view, perfect throughput prediction is not always possible [6]. This paper aims at showing the impact of inaccurate bandwidth prediction on some QoE key performance indicators (KPIs), such as average video quality, average bitrate switching number and average video freezing (also referred to as video stalls) number. While in [1], we assume a perfect knowledge of the future, in this paper, we assume an inaccurate throughput prediction and propose 2 algorithms to make NEWCAST robust to throughput prediction errors. The first algorithm Adaptive ShorT-tERm Newcast (A-STERN) is a direct application of NEWCAST on successive short-term moving horizons to the future and aim at reducing the number of stalls, whereas the second algorithm Short TeRm Enslaved nEwcasT (STREET) aims at reducing the number of quality switching. The obtained results lead us to believe that these algorithms can be efficient and robust in realistic environment network even if the prediction of the capacity variation is not accurate.

The paper is organized as follows: in the second section we present our optimization framework as developed in [1], we give a short overview on the optimal solution and a short description of NEWCAST. In the third section, we highlight the shortcoming of NEWCAST in real environment under throughput (also called capacity) prediction errors, whereas in the fourth and the fifth sections we present A-STERN and STREET algorithms.

## II. Problem formulation

We consider a video streaming server where each video file is divided into $N$ small segments of the same number of frames $S$ according to the group of pictures (GoP) DASH parameter. Each segment is encoded with $L$ different bitrates representing the video quality-levels . Let $l_j$ be the video quality-levels $j$, and $b_j$ its associated bitrate, where $b_i < b_j$ for $i < j$. Assume that, while streaming a video, the client requests segments from the server such that only one quality-level can be selected at a certain time. Let $b(t)$ be the video bitrate streamed at time $t$. In order to characterize this bitrate with respect to the best quality-level , we define $\gamma(t)$ as $\gamma(t) = \frac{b_L}{b(t)}$. At the client side, we assume that the playback buffer is of sufficiently large size to avoid buffer overflows. We further assume that the playback frame rate $\lambda$ (in fps) is the same for all quality-levels. To avoid video stalls, a prefetching state is introduced at the beginning of the streaming session, i.e., before starting the video, the media player prefetches a number of frames $Q_0$, which we call hereafter the start-up frames.

In our problem, we aim at optimizing system utilization and video quality, while taking into account user's perception on the playback buffer stalls. This optimization problem exploits the prediction of the future network capacity over a finite horizon. Let $\tilde{c}(t)$ be the predicted network capacity and $r(t)$ be the user's bitrate at time $t$, such that $0 \leq r(t) \leq \tilde{c}(t)$. Next, we define the objective cost function, which is the sum of two terms: the average system utilization cost and the weighted average quality of the video. We define the network utilization cost as in [11], namely

$$\sigma = \frac{1}{T} \int_0^T \frac{r(t)}{\tilde{c}(t)} dt \tag{1}$$

In the expression above, $\frac{r(t)}{\tilde{c}(t)}$ stands for the proportion of resources used at time $t$, and $T$ defines the video length in seconds.

During the session, the number of frames streamed with quality-level $j$ is given by

$$\int_0^T \frac{\delta_{\{b(t)=b_j\}} r(t)\lambda}{b(t)} dt = \int_0^T \frac{\gamma_j(t) r(t)\lambda}{b_L} dt \tag{2}$$

where

$$\gamma_j(t) = \begin{cases} \gamma(t) & \text{if } b(t) = b_j, \quad j \in [1 \dots L] \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

Let us denote by $w_j$ the weight associated to quality-level $j$, where $w_i < w_j$ for $i < j$. The value of $w_j$ represents the user's perception on the video quality-level $j$. It actually models the user's preference. Hence, we express the average normalized quality of the video as

$$\rho = \frac{\sum_{j=1}^{j=L} w_j b_j}{b_L} \tag{4}$$

which is equivalent to

$$\rho = \frac{\sum_{j=1}^{j=L} w_j \int_0^T \gamma_j(t) r(t)\lambda dt}{b_L \cdot (N \cdot S)} = \frac{\sum_{j=1}^{j=L} w_j \int_0^T \gamma_j(t) r(t) dt}{S_L} \tag{5}$$

where $S_L$ is the video total size in bits when it is coded with the highest bitrate $b_L$.

It goes without saying that higher QoE comes at higher system utilization. However, it may happen that a user wishes to reduce his system utilization cost in return for a better video quality, or that a content provider wishes to reduce delivery cost at the expand of a minimal video quality that guarantees clients' engagement. Another interesting example is when an operator prefers saving network resources for further usage. All these situations are covered by the model we adopted. To do so, we define a positive constant parameter $\pi$ that allows to balance between system utilization (at the network side) and QoE (at the client side). Our optimization cost function can then be formulated as

$$\mathcal{F} = \sigma - \pi \cdot \rho.$$

In this paper, we consider the case where there are no rebuffering events (or playback stalls) during the streaming session. Thus, when minimizing our cost function $\mathcal{F}$, we ensure that the playback buffer does not fall empty.

Let $u(t)$ and $l(t)$ be the *cumulative* number of arrival frames and the *cumulative* number of frames watched by the user till time $t$ under the no-rebuffering events assumption. Then, the buffer underflow constraint can be defined as $u(t) \geq l(t)\ \forall t \leq T$. Given the network bitrate $r(t)$ and the corresponding streamed video bitrate $b(t)$, we compute the network frame rate as $\frac{\lambda\ r(t)}{b(t)}$.

Let $(r, \gamma)$ denote the video transmission strategy, where $r$ defines the transmission schedule and $\gamma$ characterizes the quality-levels strategy. The overall optimization problem can then be summarized as follows

$$\min_{(r,\gamma)} \mathcal{F}(r,\gamma) = \frac{1}{T}\int_0^T \frac{r(t)}{\tilde{c}(t)}dt - \pi \cdot \frac{\sum_{j=1}^{j=L} w_j \int_0^T \gamma_j(t) r(t) dt}{S_L} \tag{6}$$

$$s.t \begin{cases} \int_0^t \frac{\lambda\ \tilde{c}(t)\gamma_1}{b_L} \geq l(t) & \forall t \leq T \\ \int_0^t \sum_{j=1}^{j=L} \frac{\lambda\ r(t)\gamma_j(t)}{b_L} \geq l(t) & \forall t \leq T \\ \int_0^T \sum_{j=1}^{j=L} \frac{\lambda\ r(t)\gamma_j(t)}{b_L} = l(T) & \end{cases}$$

where the first constraint ensures the existence of at least one solution that defines a mono-coded video using the lowest bitrate level $b_1$, the second one ensures that there are no rebuffering events, and the third one guarantees that all the video is streamed at the end of the streaming session $T$.

*Remark* 1. Another key QoE parameter is the average video quality variation. It tracks the switching of the quality from one segment to another during the streaming session. Although we did not take this parameter into account in the optimization problem (6), we will show by the sequel that the structure of the optimal solution also minimizes the video quality switching.

In [1], we assumed that $\tilde{c}(t)$ is the exact network capacity, i.e., perfect capacity prediction assumption. Under this setting, we showed that the optimal solution of the optimization problem in (6) is as follows:

**Theorem 1.** *Assume that there exists a feasible solution that satisfies the constraints in (6). Then, there exists an optimal strategy $(r_{th}, \gamma_{r_{th}})$ of the optimization problem in (6), where $r_{th}$ is a threshold transmission schedule, namely*

$$r_{th}(t) = \begin{cases} \tilde{c}(t) & \textit{if } \tilde{c}(t) \geq \alpha \\ 0 & \textit{otherwise,} \end{cases} \tag{7}$$

*and $\gamma_{r_{th}}$ is an ascending bitrate level strategy, i.e., the quality-levels of segments increases during the session. This translates mathematically as follows*

$$\forall\ 0 \leq t \leq t' \leq T, \quad b(t) \leq b(t') \textit{ i.e., } \gamma_{r_{th}}(t) \geq \gamma_{r_{th}}(t').$$

A thorough and detailed analysis of the results above is provided in [1]. For succinctness, we leave it out and simply claim that streaming the video at higher capacity regions is cheaper than streaming it at lower capacity regions (in terms of system utilization cost). This can be performed without violating the no-rebuffering constraint. We also notice that, given such a transmission schedule, the no-rebuffering constraint may be relaxed by applying an ascending bitrate strategy. In fact, reordering the qualities in an ascending way may add more flexibility toward the constraints, as it results in higher number of streamed segments at the beginning of the session. To sum it up, we can say that reordering the levels only allows to send more data beforehand, which can be performed without buffer overflows since we assume a sufficiently large playback buffer size.

In [1], we proposed an algorithm, called NEWCAST, that performs close to the optimal solution under perfect capacity prediction over the entire future horizon. NEWCAST can be summarized as follows: We set the threshold transmission schedule $\alpha$ to its lowest value ($\tilde{c}_{min}$), then we keep increasing it progressively till reaching the case where it becomes impossible to stream the entire video using the lowest quality. This case determines the maximum possible scheduling threshold $\alpha$. At each step, we heuristically set the ascending order of segments' quality-levels to maximize the average video quality. By computing and storing at each step the resulting objective function $\mathcal{F}$, we end up choosing the strategy that gives the minimum value of $\mathcal{F}$.

To implement this solution in a real environment, a video delivery optimization framework should be developed at the client side to set the streaming strategy at the beginning of the streaming session. This framework sends both the threshold $\alpha$ to the network scheduler and the quality levels' sequence to the media player. While the scheduler manages the resource allocation, the player requests the qualities one by one to the server as set by the framework.

## III. NEWCAST UNDER IMPERFECT PREDICTION OVER THE ENTIRE HORIZON

As claimed in [12], a perfect prediction of the capacity may not be available over a large horizon window. That said, it is more plausible that the prediction becomes more accurate over a short horizon window. In the literature, we find that prediction accuracy depends on three major factors: (i) the accuracy on the user's mobility model, (ii) the space mapping of the users' average throughput, and (iii) the variation of the real throughput around the user space-mapped average throughput [6]. Accordingly, in this paper, we assume that prediction error increases as long as we move forward in time. The real capacity $c_{real}$ is then modeled as a Gaussian white noise around the predicted capacity $\tilde{c}$, namely

$$\forall\ t\ in\ [0,T],\ c_{real}(t) = \mathcal{N}\left(\tilde{c}(t), \mathcal{SD}_{err}(t)\right)$$

where $\mathcal{SD}_{err}(t) = \mathcal{SD}_{err} \cdot \log(t)$ is the error standard deviation at time $t$.

In our previous work [1], the evaluation tests of NEWCAST algorithm were done through simulations using Matlab. To be as close as possible to real video settings, we set the video parameters according to DASH standard and some Youtube specifications [13], [14]. The network part, however, was simulated through a randomly generated capacity around a given mean throughput.

| | |
|---|---|
| **Window Size** | 3 min 10 s |
| **Mean throughput** | 2 Mbps |
| **Capacity Time Slot** | 1 s |
| **Video Length** | 3 min |
| **Segment Length** | 1s |
| **Video frame rate** | 30 fps |
| **Playback cache** | 4s |
| **Bitrate levels Mbps** | [0.4 0.75 1 2.5 4.5] |
| **Levels weights** | [0.09 0.17 0.22 0.55 1] |

TABLE I
SIMULATION SETTING PARAMETERS.

In this work, we use the same approach to evaluate the performance of NEWCAST under throughput prediction errors. All the parameter settings are listed in Table I. To see how NEWCAST reacts in simulated real environments, we generate many examples of possible real throughput variations $c_{real}$ using one predicted network capacity $\tilde{c}$ and the aforementioned prediction error model with different values of $\mathcal{SD}_{err}$.

NEWCAST is run using the knowledge of the future throughput $\tilde{c}$ to decide on the future streaming strategy that determines the optimal scheduling threshold $\alpha^*$ and the bitrates' strategy $\gamma^*$. After generating a real throughput $c_{real}$, we apply that strategy (same threshold and same quality-levels) to compute the real system utilization cost and the number of stalls during the streaming session. In our simulations, we set $\pi$ to a low value ($\pi = 3$) to prioritize the system cost and make the system more sensible to prediction errors.

In Fig. 1, we evaluate the robustness of NEWCAST by representing the distribution of video stalls during the streaming session as function of $\mathcal{SD}_{err}$. Results show that the probability of having zero stalls is relatively low and decreases from 0.49 to 0.17 as the error on the predicted capacity increases, whereas the average number of stalls increases from 1.53 to 1.89. That said, when the prediction of the capacity over a long future horizon is imperfect, NEWCAST algorithm fails to guarantee a good QoE. This leads us to consider short-term future horizon in order to obtain accurate throughput prediction.

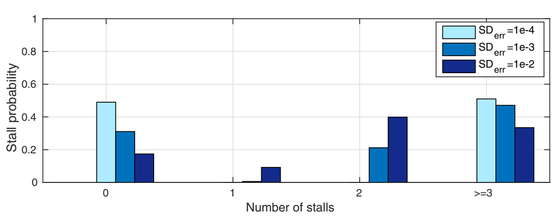


Fig. 1. NEWCAST: Distribution of stalls as function of $\mathcal{SD}_{err}$, $\pi = 3$.

## IV. ADAPTIVE SHORT-TERM NEWCAST (A-STERN) FOR BETTER STALL AVOIDANCE

A-STERN is a modified version of NEWCAST that considers short-term throughput prediction over successive short term future horizons. It takes as input the evolution of the buffer occupancy computed by NEWCAST over long future horizon window with long-term predicted capacity $\tilde{c}$ to adjust the number of segments that need to be streamed at each short future horizon window. Let $\tilde{u}(t)$ be the number of cumulative received frames predicted by NEWCAST at time $t$. At the beginning of each short horizon, A-STERN computes the cumulative number of segments that should be streamed using $\tilde{u}$ in order to force the playback buffer to follow the same evolution as predicted by NEWCAST. To reduce quality-levels' switching rate when moving from one short horizon to another, A-STERN ignores the startup phase mode of NEWCAST where the first segments are streamed with a greedy transmission mode using the lowest quality-level. Let $\mathcal{H}_i$ be the $i^{th}$ short horizon for $i > 0$, $\tilde{c}_{\mathcal{H}_i}$ its associated predicted capacity and $X_{\mathcal{H}_i}$ the number of segments that should be streamed over $\mathcal{H}_i$.

The adaptive side of A-STERN lies in making real-time updates on the real buffer state to see whether the number of streamed frames matches the predicted function $\tilde{u}$ or not. Let $u_{real}$ be the real cumulative number of received frames. We make the following statement: If $|u_{real}(t) - \tilde{u}(t)| \geq \epsilon$ , $t \in \mathcal{H}_i$, $i > 0$ where $\epsilon$ designs a number of frames, then the player stops streaming the video and generates a new strategy over a new short horizon $\mathcal{H}_{i+1}$ (see Algorithm. 1).

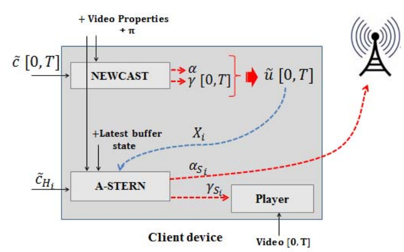


Fig. 2. Illustration of A-STERN interactions in a real environment.

In our simulations, we set $\pi_{\textbf{A-STERN}}$ to be one time equal to $\pi_{\textbf{NEWCAST}}$ and one time greater than $\pi_{\textbf{NEWCAST}}$ (3 and 6).

Fig. 3 depicts the probability of having stalls with A-STERN compared to NEWCAST as function of $\pi_{\textbf{A-STERN}}$ and $\mathcal{SD}_{err}$. It is clear that A-STERN makes the system more robust to video stalls. Indeed, for a $\pi_{\textbf{A-STERN}} = 3$, the probability of having zero stalls increases from 0.17 to 0.64, and for $\pi_{\textbf{A-STERN}} = 6$ it increases to 0.97 at the highest error, which led to reduce the average number of stalls from 1.90 to 0.38 and 0.03.

In Fig. 4, we plot the average number of switches for A-STERN and NEWCAST during the streaming session. The number of switches is notably high for both values of $\pi_{\textbf{A-STERN}}$: around 16 and 25 at the highest error, versus 3 switches with NEWCAST. For the sake of illustration, we show in Fig. 5 snapshots on video quality-levels' variation for a given real throughput $c_{real}$.

As for the overall system performance, it is noticed from Fig. 6 that A-STERN increases the system cost from 32% to 43% ( $\pi_{\textbf{A-STERN}} = 3$) and 71% ($\pi_{\textbf{A-STERN}} = 6$) at the highest prediction error. The average video quality has evenly increased.

**Algorithm 1:** Adaptive Short-TERm Newcast (A-STERN)

**Input:** $\mathcal{H}_1 = [t_0, t_1], X_{\mathcal{H}_1} = \frac{\tilde{u}(t_1)}{SegmentSize}, i = 1;$

**while** *Still segments to stream* **do**

  $t = t_{i-1};$

  *Predict* $\tilde{c}_{\mathcal{H}_i}$;

  *Check if it is possible to stream* $X_{\mathcal{H}_i}$ *segments over* $\tilde{c}_{\mathcal{H}_i}$, *otherwise, reduce* $X_{\mathcal{H}_i}$;

  $[\alpha_i, \gamma_i]$=**NEWCAST**$(\tilde{c}_{\mathcal{H}_i}, X_{\mathcal{H}_i})$;

  *Extract QualityVect$_i$ from* $\gamma_i$ ;

  *[BufferState, $t_{fin}$] = StreamVideo (* $c_{real}[t_i$*:end]*, $\alpha_i$, *QualityVect$_i$*);

  ***Do simultaneously to streaming***

  **while** $t \in [t_{i-1}, t_i[$ **do**

    **if** $\frac{|u_{real}(t) - \tilde{u}(t)|}{SegmentSize} \geq \epsilon$ **then**

      *Change strategy* ;

      $t_i = t + 1;$

      $t_{i+1} = t_i + \mathcal{H}_i;$

      $\mathcal{H}_{i+1} = [t_i, t_{i+1}];$

      $X_{\mathcal{H}_{i+1}} = \frac{\tilde{u}(t_{i+1})}{SegmentSize}$ - *BufferState(t)* ;

      $i = i + 1;$

      $t = 0$ ;

    **else**

      $t = t + 1$ ;

    **end**

  **end**

**end**

**return** $(\{\alpha_i\}, \{\gamma_i\})$

These figures clearly show significant amelioration on the overall quality, mainly by reducing the number of stalls, but, this unluckily leads to an increase on the number of switches and on the system utilization cost, which was our motivation to develop the next algorithm.

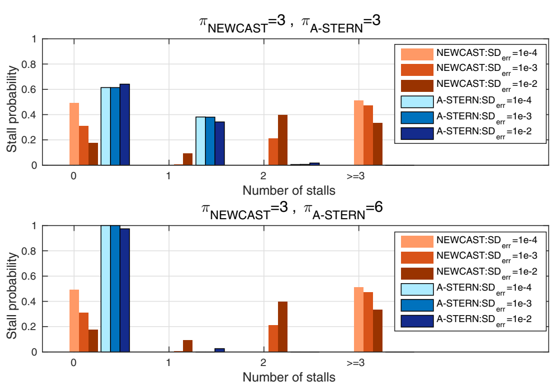


Fig. 3. A-STERN vs. NEWCAST: Probability of having stalls as function of $\mathcal{SD}_{err}$ and $\pi_{\text{A-STERN}}$ for a short-term horizon of 10s, $\epsilon$=3 segments.

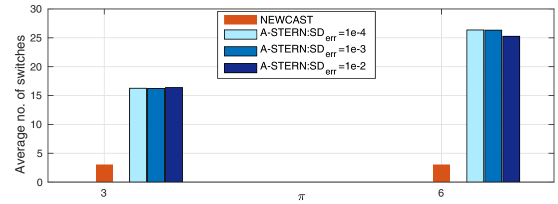


Fig. 4. A-STERN vs. NEWCAST: Average number of switches as function of $\mathcal{SD}_{err}$ and $\pi$ for a short-term horizon of 10s.

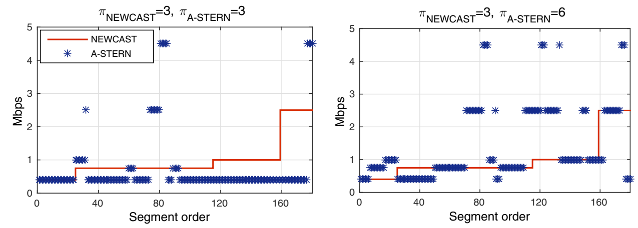


Fig. 5. A-STERN vs. NEWCAST: Snapshots on video quality-levels' variation as function of $\pi$ for $\mathcal{SD}_{err} = 10^{-2}$ Mbits and a short-term horizon of 10s.

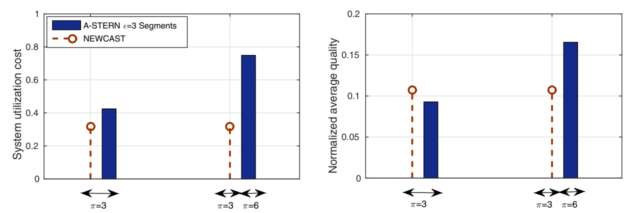


Fig. 6. A-STERN vs. NEWCAST: System utilization cost and normalized average video quality as function of $\pi$ for $\mathcal{SD}_{err} = 10^{-2}$ Mbits and a short-term horizon of 10s .

## V. Short TeRm Enslaved nEwcasT (STREET) for smoother quality variation

Unlike the previous algorithm that sets its bitrate strategy independently of NEWCAST's predicted quality using the capacity $\tilde{c}$, STREET algorithm attempts to follow the same predicted quality as NEWCAST to give similar performance with a reduced number of stalls. It performs as follows: At the beginning of each short-term horizon $\mathcal{H}_i$, it computes the number of segments $X_{\mathcal{H}_i}$ to stream by seeing $\tilde{u}$. If it is possible to perform the streaming with the same quality as NEWCAST, it sets up the highest short-term threshold $\alpha_i$ to minimize the system utilization cost over $\mathcal{H}_i$, otherwise it reduces the video quality-levels of the segments by running NEWCAST. In the latter case, STREET ignores the startup phase mode of NEWCAST to avoid additional switches.

Figures. 7 and 8 show the performance of STREET in terms of quality-levels' switching compared to NEWCAST and the A-STERN. It is noticed that STREET succeeds to follow a near bitrate trend as NEWCAST with a near number of switches (around 3.03 for both values of $\pi_{\textbf{STREET}}$). The system utilization cost and the average video quality are evenly near to NEWCAST performance (around 39% and 0.48 Mbps for both values of $\pi_{\textbf{STREET}}$). As for the distribution of stalls, it is depicted from Fig .10 that the probability of having zero stalls decreases compared to A-STERN to 0.32 for both values of $\pi_{\textbf{STREET}}$ , but is still

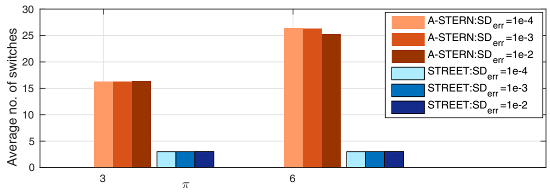


Fig. 7. STREET vs. A-STERN: Average number of switches as function of $\mathcal{SD}_{err}$ and $\pi$ for a short-term horizon of 10s, $\epsilon = 3$ segments.

relatively high compared to NEWCAT. As a result, the average number of stalls increases compared to A-STERN to 0.71 but is still relatively low compared to NEWCAST. Overall, we consider STREET as the best algorithm that mostly achieves the same performance as NEWCAST while reducing the risk of video stalls by around 75%.

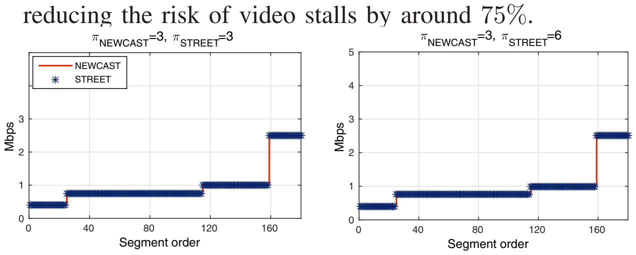


Fig. 8. STREET vs. NEWCAST: Snapshots on quality-levels variation as function of $\pi$ for $\mathcal{SD}_{err} = 10^{-2}$ Mbits and a short-term horizon of 10s.

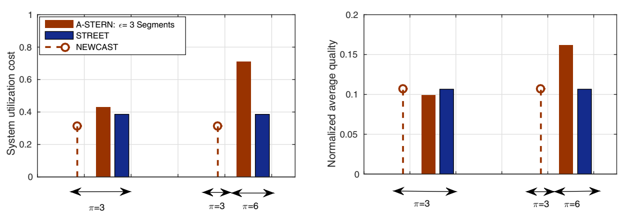


Fig. 9. STREET vs. a-STERN and NEWCAST: System utilization cost and normalized average quality as function of $\pi$ for $\mathcal{SD}_{err} = 10^{-3}$ Mbits and a short-term horizon of 10s.

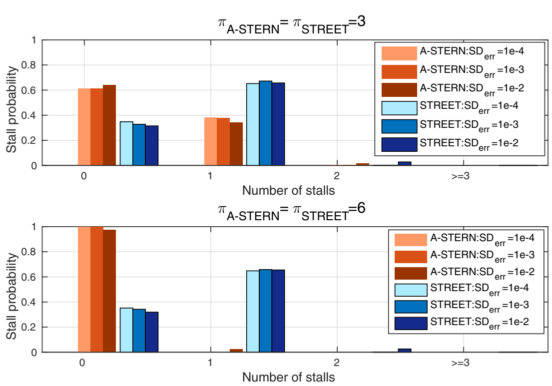


Fig. 10. STREET vs. A-STERN: Probability of stalls as function of $\mathcal{SD}_{err}$ and $\pi$ for a short-term horizon of 10s.

## VI. CONCLUSION

In this paper, we proposed 2 novel approaches to anticipate resource management and QoE for video streaming under imperfect prediction. More specifically, we adapted NEWCAST algorithm to imperfect prediction of the future capacity. We showed that the proposed A-STERN algorithm can significantly reduce the number of stalls by leveraging short-term horizon throughput prediction, which usually results in accurate throughput prediction compared to long-term horizon throughput prediction. However, we observed that short-term horizon prediction combined to buffer occupancy results in increasing the number of quality-levels' switching. This led us to propose STREET algorithm to smooth the video quality variation. Our numerical results showed the efficiency of our proposed solutions, whereas feasible communication protocols between the operator and our proposed frameworks are still arguable.

## REFERENCES

[1] I. Triki, R. El-Azouzi, and M. Haddad, "NEWCAST: Anticipating Resource Management and QoE Provisioning for Mobile Video Streaming," in *17th International Symposium on a World of Wireless, Mobile and Multimedia Networks (IEEE WoWMoM)*, June 2016.

[2] S. Egger, B. Gardlo, M. Seufert, and R. Schatz, "The Impact of Adaptation Strategies on Perceived Quality of HTTP Adaptive Streaming," in *Proceedings of the 2014 Workshop on Design, Quality and Deployment of Adaptive Video Streaming*, 2014.

[3] R. Margolies, A. Sridharan, V. Aggarwal, R. Jana, N. K. Shankaranarayanan, V. A. Vaishampayan, and G. Zussman, "Exploiting mobility in proportional fair cellular scheduling: Measurements and algorithms," in *IEEE Conference on Computer Communications, INFOCOM*, 2014.

[4] J. Yao, S. S. Kanhere, and M. Hassan, "An empirical study of bandwidth predictability in mobile computing," in *Proceedings of the Third ACM International Workshop on Wireless Network Testbeds, Experimental Evaluation and Characterization*, WiNTECH, 2008.

[5] "Opensignal project." http://opensignal.com/, Link verified on May, 20th, 2016.

[6] H. Abou-zeid, H. S. Hassanein, Z. Tanveer, and N. AbuAli, "Evaluating mobile signal and location predictability along public transportation routes," in *IEEE Wireless Communications and Networking Conference, WCNC*, 2015.

[7] D. Han, J. Han, Y. Im, M. Kwak, T. T. Kwon, and Y. Choi, "MASERATI: Mobile Adaptive Streaming Based on Environmental and Contextual Information," in *Proceedings of the 8th ACM International Workshop on Wireless Network Testbeds, Experimental Evaluation and Characterization*, 2013.

[8] G. Tian and Y. Liu, "Towards Agile and Smooth Video Adaptation in Dynamic HTTP Streaming," in *Proceedings of the 8th International Conference on Emerging Networking Experiments and Technologies*, 2012.

[9] T. Y. Huang, R. Johari, N. McKeown, M. Trunnell, and M. Watson, "A buffer-based approach to rate adaptation: Evidence from a large video streaming service," in *ACM Conference on Special Interest Group on Data Communication (SIGCOMM)*, 2014.

[10] X. Yin, A. Jindal, V. Sekar, and B. Sinopoli, "A Control-Theoretic Approach for Dynamic Adaptive Video Streaming over HTTP," *SIGCOMM Comput. Commun. Rev.*, pp. 325–338, 2015.

[11] Z. Lu and G. de Veciana, "Optimizing stored video delivery for mobile networks: The value of knowing the future," in *INFOCOM, 2013 Proceedings IEEE*, pp. 2706–2714, April 2013.

[12] S. Mekki and S. Valentin, "Anticipatory quality adaptation for mobile streaming: Fluent video by channel prediction," in *16th IEEE International Symposium on A World of Wireless, Mobile and Multimedia Networks, WoWMoM 2015, Boston, MA, USA*, 2015.

[13] "Live encoder settings, bitrates and resolutions." https://support.google.com/youtube/answer/2853702?hl=en.

[14] S. Lederer, C. Müller, and C. Timmerer, "Dynamic Adaptive Streaming over HTTP Dataset," in *Proceedings of the 3rd Multimedia Systems Conference*, 2012.