

Buffer management for scalable video streaming

Amparito A. Morales Figueroa, Lorenzo Favalli

Department of Electrical, Computer and Biomedical Engineering, University of Pavia
Via Ferrata 5, 27100 Pavia (PV), Italy

amparito.morales01@universitadipavia.it, lorenzo.favalli@unipv.it

ABSTRACT

In a video streaming scenario, to cope with bandwidth variations, buffer management and intelligent drop packets policies play a critical role in the final quality of the video received at the user side. In this context, we present a buffer management strategy implemented at the source of a video communication system. This scheme uses priority information from the H.264/SVC encoder, network information from a Bandwidth Estimation approach (BE), based on Hidden Markov Model (HMM) and monitors buffer fullness: when it exceeds a defined threshold, the selective discard strategy takes place. To get more flexibility, we employed SNR quality scalability (Medium Grain Scalability), to get more than one rate point for each enhancement layer. Low priority packets correspond to higher quality layers and are discarded first, with the aim to preserve as much as possible more relevant lower layers packets. Dependencies created by the encoding process are kept into account. We show that the strategy presented ensures that the video transmitted has the highest possible quality under the given network conditions and buffer resources.

Keywords

buffer management; drop packets policy; Hidden Markov Model; bandwidth estimation; H.264/SVC; scalable video coding

1. INTRODUCTION

Delivering variable bit-rate video streaming over the Internet is a challenging task. In order to guarantee the presentation quality of multimedia applications, QoS requirements such as packet loss rate, bandwidth, delay and jitter must be satisfied. Normally, transmission of video requires high bandwidth and low delay, while a certain amount of packet loss can be accepted [18].

Buffer management and scheduling algorithms represent the main class of techniques traditionally applied to maximize performance in communication networks, either wired or wireless. Video transmission makes no exception. Several works in the literature, present buffer management strategies applied to video communications that consider the specific features of video.

As an example, a study in [11] found that, for the case of low bit rate coded sequences, in most cases higher quality low frame rate videos are preferable to lower quality full frame rate ones.

Following this consideration and the result from network analysis that outlined the fact that most of the loss occurs at the transmission point and not at the nodes inside the network, [1] proposes a simple buffer management scheme implemented at the transmission source which drops low priority packets in response to congestion, where priority is related to frame rate. In this work, the video sequences are encoded into multiple priority layers using H.26L video encoder and a robust scalable sub-band/wavelet video coder. The buffer management algorithm performs a greedy strategy on the packets being transmitted, trying to maximize the quality of the video which can be reconstructed from the transmitted packets. Basically, the algorithm constantly controls the buffer size. When the current size exceeds a defined threshold, the packet is dropped considering its transmission probability. On the other hand, if the size exceeds the maximal allowed buffer size, the arriving packet is dropped without considering its priority.

In [2] the buffer management and congestion control presented in [1] are applied in video streaming. Also in this case the buffer is implemented at the source and it queues packets from the encoder and dequeued packets are transmitted using a randomized binomial scheme. The buffer algorithm is the same as the one presented in [1]. However, different from the previous work, the Round Trip Times (RTT) are used to estimate the drop probability of a packet.

An integrated video communication scheme for stored variable bit-rate (VBR) video streaming in a congested network is presented in [18]. This scheme regulates the transmission rate through a refined rate control algorithm based on the Program Clock References (PCR) value embedded in the video streams. Furthermore, multiple buffers for different importance levels, along with an intelligent selective frame discard algorithm are applied at the source. An architecture which includes an input buffer at the server coupled with the congestion control scheme of TCP at the transport layer is presented in [7]. This work assumes that the available bandwidth is sufficient to deliver the high priority frames and the goal is to maximize the number of transported low priority frames subject to the constraint that the loss rate for the high priority frames would be minimal.

In [12], different reactive and proactive queue management schemes are discussed and investigated with respect to enhancing the objective video quality of a streaming application in a HSDPA network. A proactive buffer management scheme with data differentiation is proposed, which significantly increases the video quality by taking into account MPEG frame dependencies.

A Weighted Multi-Playback Buffer Management (Weighted MPBuff) is proposed in [13]. The Weighted MPBuff and scheduling algorithm provides more protection to the lower layers compared to the higher layers. To archive weighted protection, the sender schedules the video data considering the transmission

sequence of the video and defining different amounts of GOPs for each layer. The playout buffer is composed by as many buffers as coded layers, and the buffer size decreases from the base layer to the highest enhancement layer. Each time period, the receiver calculates the video buffer time using the highest timestamp and the timestamp of the current playback time. This information is used to decide whether the video data unit in layer $i+1$ will be scheduled or not. If the current available bandwidth is not enough to transmit the layer $i+1$, also the layers above will not be sent.

Considering a multi-stream video transmission scenario, [5] proposes a perceptual quality-aware active queue management, which is designed for scalable video traffic. In order to reduce the queuing delay and queue length, the proposed scheme selectively drops packets from layers that have little influence on video quality, minimal perceptual quality reduction (PQR) in a stream. The dropping strategy regards the fact that losses in base quality layers cause considerably higher quality reduction than losses in quality enhancement layers.

Finally, several works have exploited scalability in wireless environments mixing with adaptive modulation and coding (AMC) and resource allocation. As an example, [9] proposes a dynamic sharing of the resources by combining SVC with appropriate radio link buffer management for multiuser streaming services. Since the pictures belonging to the lowest temporal layer are most important for decoding, these own the highest priority. Then the respective progressive refinement (PR) fragments have next lower importance, and so on. Hence, by assuming that the temporal resolution has highest priority, this work recommends a dropping order, based on the temporal level of the fragments, in case that a rate adaptation on the fly is necessary. First, the PR fragments of the highest temporal level present are dropped. If there are only packets containing base layer fragments, the base layer of the highest temporal level present is discarded. Furthermore, this work develops a priority labeling technique, which is applied in the radio link buffer, this follows the same aforementioned drop concept.

In this paper we present a buffer management strategy with selective packet discard for a Medium Grain Scalable (MGS) video encoded H.264 video sequence, based on two main blocks:

- 1) a Bandwidth Estimation (BE) approach based on Hidden Markov Model (HMM), where the states sequence simulates the bandwidth of the network and are matched with the rates of the MGS encoder.
- 2) a selective discard of packets in the buffer based on packet priority and its dependencies given the constraints imposed by buffer fullness and BE.

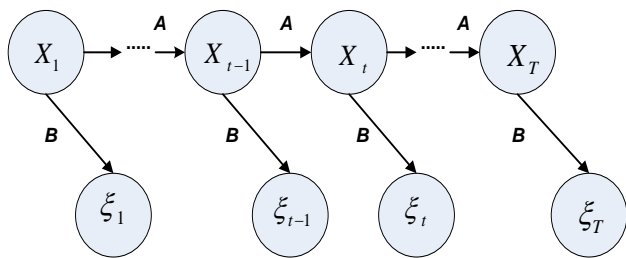


Figure 1. Hidden Markov Model.

When the source transmission buffer length exceeds a given threshold, the buffer management strategy takes place. This

consists in dropping low priority packets and sending the most important packets in the available bandwidth. Since MGS-SVC defines a hierarchical dependency between layers, the packets are discarded taking into account their quality and temporal level. Base layer packets clearly have the highest priority and the packets belonging to the highest enhancement layer have the lowest priority. When more than one packet has the same quality, the packet with the highest temporal level is considered to be discarded first. The algorithm developed guarantees, that the transmitted video has the highest possible quality.

The remainder of the paper is organized as follows. Section II explains the Bandwidth Estimation Model based on Hidden Markov Model and the application of this model in SVC video sequences. Section III describes the buffer management to SVC video sequences and presents the discard packets strategy proposed in this paper. Section IV presents and discusses the results obtained by the simulations. Section V gives the final conclusions and future works.

2. BANDWIDTH ESTIMATION MODEL

The estimation of the available bandwidth (AB) of an end-to-end path has received noticeable attention due to its relevance in several network applications. Transport layer protocols might also use AB information to change the transmission rate according to the amount of available bandwidth in the path, using the network resources efficiently while avoiding congestion [6].

Two main available bandwidth estimation approaches have been reported. The first approach is called the Probe Gap Model (PGM), which bases the estimation on the gap dispersion between two consecutive probing packets at the receiver [16]. The second approach called Probe rate model (PRM) is based on the idea of induced congestion, in which the available bandwidth is determined by the variation in the probing packet rate from sender to receiver.

This paper uses an accurate, non-intrusive and fast end-to-end approach based on PGM. Hence, using the delay variation of probe packets and the Hidden Markov Model it is possible to estimate the AB.

2.1 Hidden Markov Model

The available bandwidth can be modeled by N states, each one representing a certain level of availability. We consider a one-step transition Markov chain to estimate the probability of being in a particular state or AB range. As the available bandwidth cannot be directly observed, the probing packets approach is employed to get the delay time between packets.

Figure 1 represents the Hidden Markov Model (HMM) with discrete hidden states X representing the available bandwidth levels and discrete observation variables ξ representing probing packet pair dispersions. A particular observation has associated a probability B to be generated by a particular hidden state. Transitions between states are governed by probabilities specified in the transition probability matrix A .

As defined in [14], the Hidden Markov Model is defined by a tuple of five elements:

1. number of states (N)
2. number of distinct observation symbols per state (M)
3. state transition probability matrix (A)
4. observation probabilities (B)
5. initial state probabilities (Π)

Since with MGS the video sequences are encoded with a finite number of possible rates, a finer granularity in bandwidth is not useful for our purposes. Therefore, the number of states in the model (N) is related with the number of the target bit-rates used to encode the video sequence. The set of states is defined by $S = S_1, S_2, \dots, S_N$ where the available bandwidth grows from S_1 (low) to S_N (high). The state at time t is denoted by X_t . The number of symbols (M) is the set of symbols corresponding to observed dispersions from the probing sampling method. These symbols are decimal number corresponding to the delay between consecutive packets; these delay values are grouped in M intervals of values to convert every single observation to a discrete symbol.

The matrix A contains the transition probability between the states. Since only one-step transitions between states are considered possible, the number of unknown elements in the matrix is reduced to the three main diagonals:

$$A = \begin{bmatrix} a_{1,1} & a_{1,2} & 0 & \dots & 0 \\ a_{2,1} & a_{2,2} & a_{2,3} & 0 & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & 0 & a_{N-1,N-2} & a_{N-1,N-1} & a_{N-1,N} \\ 0 & \dots & 0 & a_{N,N-1} & a_{N,N} \end{bmatrix}$$

The matrix B has the set of probabilities that indicates how likely is that at time t a specific observation symbol ξ_t is generated by each state from the set S .

$$\begin{aligned} b_{S_1} &= \left[P\left(\frac{\xi_1}{S_1}\right), \dots, P\left(\frac{\xi_M}{S_1}\right) \right] \\ b_{S_2} &= \left[P\left(\frac{\xi_1}{S_2}\right), \dots, P\left(\frac{\xi_M}{S_2}\right) \right] \\ &\vdots \\ b_{S_N} &= \left[P\left(\frac{\xi_1}{S_N}\right), \dots, P\left(\frac{\xi_M}{S_N}\right) \right] \end{aligned}$$

It is expected that the small values of ξ are the result of a highly available bandwidth and therefore more likely generated by a high index state and conversely for high values.

Matrix A and matrix B must be determined based on some experimental data sequence via some estimation algorithm, the most commonly used of which is the Baum Welch (BW) one. Once A and B are given, the HMM can be used to generate an estimated symbol and state sequence with the same statistical properties of the original sequence.

The bandwidth estimation model used in this paper is based on the estimation model presented in [6]. With the purpose to analyze the implemented model and its accuracy, two experiments changing the values of the transition matrix A and the observation probabilities B have been carried on.

In the first experiment, the initial values of the A matrix are constructed using random or equally likely values. In both cases, the obtained results after the BW are quite similar. Therefore, it is possible to construct the initial matrix A (one-step transition) randomly. In the second experiment, we repeat the same procedure for the B matrix. This matrix can be constructed using random values, equally likely values or a defined pattern. This pattern considers the fact that small values of ξ (delay time) are the result of a non loaded network and therefore more likely generated by a high order state (corresponding a high available bandwidth) and vice versa.

From the simulations, the last case presents higher accuracy and coherence with the expected output. The first case, random values, does not represent exactly the model but is pretty close and could be used. Furthermore, in the second case, using equally likely values, the symbol sequence constructed by the model is similar to the initial one, but the state sequence drastically varies and it is not consistent. The reason is due to the fact that matrix B does not have any specific pattern and all the states have the same occurrence probability, that it is not true.

Therefore, to generate the symbol sequence and the state sequence, in this paper, we will use a matrix A composed by random values and a matrix B made by values which represent the pattern defined in the model. Examples of the A and B matrices employed in this work are presented below, where we have 7 states (available bandwidths) and 7 symbols (delay time ranges).

$$A = \begin{bmatrix} 0.4735 & 0.5265 & 0 & 0 & 0 & 0 & 0 \\ 0.0759 & 0.5460 & 0.3780 & 0 & 0 & 0 & 0 \\ 0 & 0.1057 & 0.3018 & 0.5926 & 0 & 0 & 0 \\ 0 & 0 & 0.4603 & 0.4639 & 0.0758 & 0 & 0 \\ 0 & 0 & 0 & 0.4022 & 0.3966 & 0.2011 & 0 \\ 0 & 0 & 0 & 0 & 0.5867 & 0.1040 & 0.3092 \\ 0 & 0 & 0 & 0 & 0 & 0.5362 & 0.4638 \end{bmatrix}$$

$$B = \begin{bmatrix} 0.0916 & 0.1033 & 0.1345 & 0.1542 & 0.1674 & 0.1676 & 0.1814 \\ 0.0347 & 0.0663 & 0.0739 & 0.1631 & 0.2053 & 0.2365 & 0.2202 \\ 0.0300 & 0.1209 & 0.1577 & 0.1926 & 0.1955 & 0.1708 & 0.1326 \\ 0.0414 & 0.0996 & 0.2271 & 0.4306 & 0.1131 & 0.0881 & 0.0001 \\ 0.0975 & 0.2007 & 0.4875 & 0.1608 & 0.0328 & 0.0084 & 0.0124 \\ 0.1858 & 0.2125 & 0.1452 & 0.1301 & 0.1382 & 0.0929 & 0.0954 \\ 0.1866 & 0.1824 & 0.1732 & 0.1639 & 0.1183 & 0.1058 & 0.0698 \end{bmatrix}$$

2.2 Bandwidth Estimation Model applied in SVC video sequences

In SVC a signal is encoded once at the highest quality (resolution, frame rate, quality) with appropriate packetization, and then it can be decoded from partial streams for a specific rate, quality or complexity requirements [17]. There are three types of scalability: SNR scalability (fidelity), spatial scalability (resolution), and temporal scalability (frame rate). Moreover, the quality scalability offers two options: CGS (coarse-grain quality scalable coding) and MGS (medium-grain quality scalability). In the first one, the number of available bit rates is restricted since number of layers that can be defined is not too high because it implies worse coding efficiency. MGS allows partitioning a CGS layer into several MGS layers with the aim of increasing the flexibility of bit stream adaptation and in this way improving the coding efficiency.

In this work, some test *cif* resolution video sequences¹ are encoded using the JSVM (Joint Scalable Video Model), version 9.19.15 using MGS with configuration:

$QP = \{36, 32(6,10), 28(6,10), 24(6,10)\}$, i.e. one base layer and three enhancement layers. Each of these enhancement layers defines two *MGSVector*. The video sequences are encoded using pre-established suitable bit-rates values obtained using the Kush Gauge formula, based on several parameters as video width and height, frame rate and amount of motion present in the video.

$$\begin{aligned} \text{Bitrate}(kbps) &= \text{Width} \times \text{Height} \times \text{Frame rate} \times \\ &\text{motion factor} \times 0.07 \div 1000 \end{aligned}$$

¹ The video sequences are available in: <https://media.xiph.org/video/derf/>

Where the motion factor could take three different values: (a) Low motion =1; (b) Medium motion =2 and (c) High motion =4.

3. BUFFER MANAGEMENT FOR MGS VIDEO SEQUENCES

In a server-client adaptive streaming environment, a client makes request for multimedia objects to a centralized video server. The server sends the packets sequentially and usually selects the content from a set of files where the content is encoded at different rate to match the available bandwidth. Switching from one chunk to the other is made based on the estimates of the bandwidth obtained by some feedback mechanism based on either proprietary or standard protocols such as DASH [15]. This part is transparent to our simulation as we only assume that some information about RTT is received so that the HMM-BE model can adapt the estimated bandwidth periodically. In the present work we assume this happens every second. All encoded packets are sent to the transmission buffer. Buffer fullness obviously depends on the difference between encoding rate and available bandwidth. Losses can be reduced with larger buffers at the expenses of higher delays. We show that using scalable coded sequences and the selective discard procedure proposed, allows reducing losses while maintaining minimum delays.

When a video sequence is encoded with Scalable Video Coding, the encoded video defines one base layer and several enhancement layers, which improve the quality. In SVC, the upper layers depend upon the lower layers. In such way, if a lower layer is missing, it will not be possible to decode the above layers.

In this paper, we propose a *Quality Discard Packets* policy based on the above consideration regarding the priority of the packets. Hence, the base layer packets have the highest priority and the packets belonging to the highest enhancement layer have the lowest priority.

The *Quality Discard Packets* policy is developed considering the scenario shown in Figure 2. When a H.264/SVC video is encoded, the encoder also generates a trace file which specifies various parameters for each single "packet" inside the bit-stream. The parameters include the start position (in units of bytes) of the packets, the length of the packet (in units of bytes), the values of dependency_id (Lid), temporal level (Tid), and quality level (Qid) of the packet, the type of the packet, and two flags which indicate whether the packet is discardable or truncatable [16].

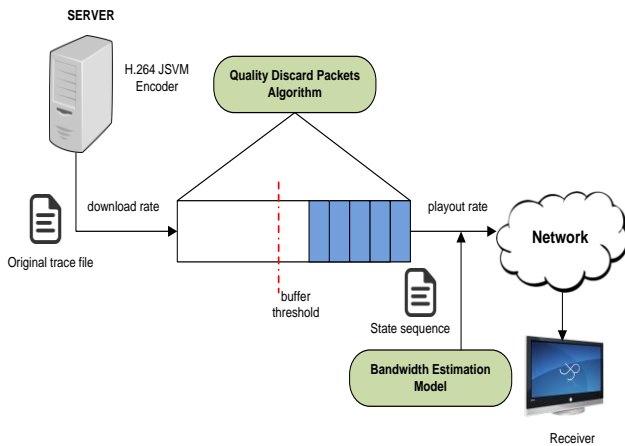


Figure 2. Quality Discard Packets scenario.

We set up the buffer size and define a threshold which is less than the total buffer size. In this work, the threshold is equal to 50% of the maximum buffer size. When this threshold is attained, we assume the buffer is close to the risk of overflowing and the *Quality Discard Packets* policy must be applied.

The implemented algorithm, at the moment to discard a packet, is governed by the following rules and considerations:

- The packets belonging to the highest layers are discarded first than the other ones.
- The video encoded with SVC presents different temporal levels (Tid). These temporal levels, also present a hierarchical dependency. So, the packets with the highest Tid are discarded first.
- At the end, the video received is decoded using the decoder tool included in JSVM [8] and FFMPEG [3] which conceals whole frame losses using temporal frame interpolation. For FFMPEG decoder, a lost P frame is concealed by copying the pixels from the previous reference frame, and a lost B frame is concealed by temporal interpolation between the frame pixels of the previous and the future frames [10].

4. SIMULATIONS RESULTS

To evaluate the performance of the *Quality Discard Packets* algorithm, several *cif* video sequences were encoded in H.264/SVC with MGS. Table 1 presents the main encoding parameters used in the *Football* video sequence.

Table 1. Encoder parameters

Football	
Parameter	Value
FrameRate	30
FramesToBeEncoded	1820
CgsSnrRefinement	1
MGSControl	2
GOPSize	8
BaseLayerMode	2
NumLayers	4

The video sequences are encoded using the *FixedQPEncoderStatic* tool included in the JSVM software. This tool provides two options:

- encode the video fixing the Quantization Parameter value (QP) to each layer.
- set the target bit rate that will be reached by each layer.

Here the second option is used as it allows a better match with the variations of the bandwidth conditions over the network. The following bit rates are used for a CIF sequence:

- Base Layer - 200 kbps
- Enhancement Layer 1 - 400 kbps
- Enhancement Layer 2 - 600 kbps

- Enhancement Layer 3 - 800 kbps

Moreover, each enhancement layer defines two MGSVector, achieving in this way a video encoded with seven target points.

Using the packet size information provided by the trace file, the next step is to calculate the times when the packets arrive to the buffer from the server, as well as the times in which the packets will leave the buffer. The first one, is obtained dividing the size by the maximum bit rate to which the video was encoded. The departure time depends on the *playout rate*, which varies each second of time. The values of the *playout rate* are obtained using the Bandwidth Estimation Model presented in section II. Figure 3 shows an example of the bandwidth fluctuation, obtained with the HMM-BE model, during 60 seconds that is the duration of the employed video sequence.

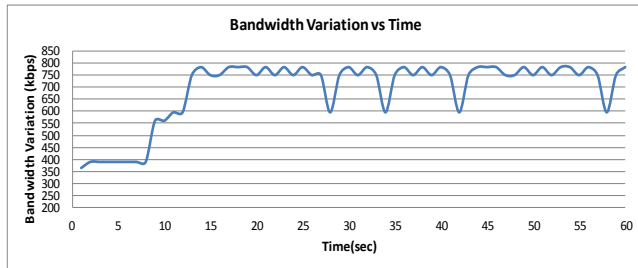


Figure 3. Bandwidth obtained with the HMM-BE model.

The performance of the *Quality Discard Packets* algorithm is compared with a case with NoBufferManagement and pure buffer management strategy unaware of the specific content, the Random Early Detection (RED). The RED calculates the average queue size and then, this value is compared to two thresholds: a minimum and a maximum threshold. When the average queue size is less than the minimum threshold, no packets are discarded. When the average queue is greater than the maximum threshold, every arriving packet is discarded. When the average queue size is between the minimum and maximum thresholds, each arriving packet is discarded with a probability p_a , where p_a is a function of the average queue size [4].

Table 2, 3 and 4 present the results obtained, applying the three approaches aforementioned in the football video sequence. It is important to highlight that the total number of frames in the original sequence is equal to 1820 frames, corresponding around one minute of duration, and the original trace file is composed by 16389 packets. In that way, it is expected that the decoded video at the receiver side owns the same number of frames than the original video.

It is important to notice that when No Buffer Management or RED are used, the buffer size is limited to a size greater or equal to 3MB in order to get a good PSNR value recovering all the frames. However, when the Quality Discard Packets is used, it is possible to get a good PSNR value and recover all the frames, when a buffer size is greater or equal than 100KB is used. This is due to the priority discard packets.

Considering the fact that the current version JSVM cannot decode video streams affected by out order, corrupted or missing NALUs, we decided in these cases to use another decoder, FFmpeg, which uses error concealment techniques in order to reconstruct the original video. In Tables 2 to 4 the blue line indicates the decoder used. The results which are on and upper the blue line were

decoded using the JSVM decoder and the results which are below to the delimiter line were decoded with FFmpeg decoder.

As we can see, in Table 4, if FFmpeg is used for the decoding process and the *Quality Discard Packets* strategy is applied, it is possible to get an acceptable PSNR value recovering all the frames, even with a buffer size constrained. In this case, we are able to have a buffer size equivalent to 15KB presenting a PSNR of 27.2025 and decoding all the frames. This is because since we limit the number of lost packets, the concealment and recovery abilities of the FFmpeg decoder can achieve much better results.

Figure 4 shows in a graphical way a summary of the results of the tables presented above. This represents the percentage of the frames that are decoded at the receiver side.

Table 2. No Buffer Management results

No Buffer Management				
Buffer Size (MB)	PSNR(dB)	discarded packets		Decoded frames
3	31.4235	*Buffer threshold is never attained		1820
1	21.3088		1683	1602
0.5 (500KB)	26.7126		13570	297
0.25(250KB)	26.7677		14777	167
0.1(100KB)	22.2908		15805	57

Table 3. Random Early Detection results

Random Early Detection (RED)				
Buffer Size (MB)	PSNR(dB)	Number of discarded packets		Decoded frames
		large size packet	probability	
3	31.4235	*Buffer threshold is never attained		1820
1	20.8915	0	23	1816
0.5 (500KB)	24.0732	13399	73	325
0.25(250KB)	21.5896	14998	44	149
0.1(100KB)	20.8912	15949	37	43

Table 4. Quality Discard Packets results

Quality Discard Packets				
Buffer Size (MB)	PSNR(dB)	Number of discarded packets		Decoded frames
		large size packet	Algorithm	
3	31.4235	*Buffer threshold is never attained		1820
1	31.0638	0	1674	1820
0.5 (500KB)	30.7514	0	2406	1820
0.25(250KB)	30.643	0	2568	1820
0.1(100KB)	30.6266	0	2391	1820
0.05(50KB)	27.2025	0	2230	1820
0.025(25KB)	27.2025	0	2032	1820
0.020(20KB)	27.2025	1	1969	1820
0.015(15KB)	27.2025	10	1889	1820
0.01(10KB)	20.4387	65	1896	1819
0.005(5KB)	15.5952	219	1885	1717

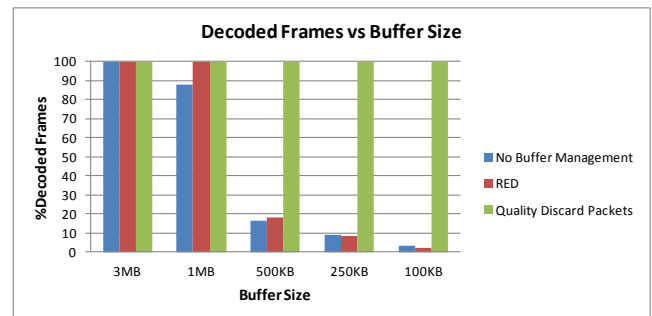


Figure 4. Decoded Frames vs. Buffer Size.

We can see that even when the buffer size decreases, our *Quality Discard Packets* approach overcomes the other ones, recovering all the frames as the original video sequence and getting a good PSNR value. A PSNR value is considered good when it is greater than 30 dB. Moreover, in all the cases when the one hundred percent of the frames are recovered, the attained PSNR is equal to 31.4235 dB.

It is important to mention that when a packet which is signaled as a non-discardable packet in the original trace file, is discarded some distortion is inserted in the belonging frame.

It is evident that the discard packets algorithm proposed in this paper presents better results and due to its "intelligent" discard packets is possible to reconstruct the original video sequence in the highest quality available.

Work is in progress to move the analysis to a scenario where we statistically multiplex several streams and the bandwidth fluctuations affect the multiplexed stream.

A final consideration is related to the applicability of the proposed approach to a large set of independent connections thus comparing it to some adaptive streaming algorithms. It is worth noting that the exploitation of scalable coding avoids the need to pre-encode a large set of copies of the same content. Furthermore, as seen from the tables above, the requirements on buffer size to compensate for the bandwidth fluctuations are clearly smaller than for the other two approaches therefore reducing the latency of the transmission chain.

5. CONCLUSIONS

In this paper, we presented a discard packets strategy which considers the priority of the packets before discard them. This priority is assigned considering the quality (Qid) and temporal (Tid) level of a packet. The proposed discard algorithm takes into account the current available network bandwidth, given by a Bandwidth Estimation Model, to apply an intelligent discard of the packets stored in the buffer. The packets are dropped from low to high priority. In the simulations, we compare our discard strategy with two other approaches and the simulations results have shown that the proposed algorithm outperforms the other two approaches and shows significant reduction in required buffer size and therefore delay. Work still needs to be performed to correctly determine the thresholds, based on the delay constraints and the type of sequence.

6. REFERENCE

- [1] Bajic, I.V., Tickoo, O., Balan, A., Kalyanaraman, S. and Woods, J.W. 2003. Integrated end-to-end buffer management and congestion control for scalable video communications. *Image Processing, 2003. ICIP 2003. Proceedings. 2003 International Conference on* (2003), III–257–60 vol.2.
- [2] Balan, A., Tickoo, O., Bajic, I., Kalyanaraman, S. and Woods, J. Integrated Buffer Management and Congestion Control for Video Streaming.
- [3] FFMPEG : <http://ffmpeg.org/>, The official website.
- [4] Floyd, S. and Jacobson, V. 1993. Random early detection gateways for congestion avoidance. *Networking, IEEE/ACM Transactions on*. 1, 4 (Aug. 1993), 397–413.
- [5] Ghoreishi, S.E., Aghvami, A.H. and Martini, M.G. 2015. Perceptual quality-aware active queue management for video transmission. *Personal, Indoor, and Mobile Radio Communications (PIMRC), 2015 IEEE 26th Annual International Symposium on* (Aug. 2015), 1267–1271.
- [6] Guerrero, C.D. and Labrador, M.A. 2008. A Hidden Markov Model approach to available bandwidth estimation and monitoring. *Internet Network Management Workshop, 2008. INM 2008. IEEE* (Oct. 2008), 1–6.
- [7] Gürses, E., Akar, G.B. and Akar, N. 2003. Selective frame discarding for video streaming in TCP/IP networks. *Packet Video Workshop, Nantes, France (April 2003)* (2003).
- [8] JSVM Software Manual, version 9. 19. 14 (2011).
- [9] Liebl, G., Schierl, T., Wiegand, T. and Stockhammer, T. 2006. Advancedwireless Multiuser Video Streaming using the Scalable Video Coding Extensions of H.264/MPEG4-AVC. *2006 IEEE International Conference on Multimedia and Expo* (Jul. 2006), 625–628.
- [10] Lin, T.-L., Chang, Y.-L. and Cosman, P.C. 2010. Subjective experiment and modeling of whole frame packet loss visibility for H.264. *Packet Video Workshop (PV), 2010 18th International* (Dec. 2010), 186–192.
- [11] Masry, M. and Hemami, S.S. 2001. An analysis of subjective quality in low bit rate video. *Image Processing, 2001. Proceedings. 2001 International Conference on* (2001), 465–468 vol.1.
- [12] Orlov, Z. and Necker, M.C. 2007. Enhancement of video streaming QoS with active buffer management in wireless environments. *European Wireless Conference* (2007).
- [13] Palawan, A., Woods, J. and Ghanbari, M. 2014. Weighted multi-playback buffer management for scalable video streaming. *Computer Science and Electronic Engineering Conference (CEECE), 2014 6th* (2014), 47–51.
- [14] Rabiner, L. 1989. A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE*. 77, 2 (Feb. 1989), 257–286.
- [15] Sodagar, I. 2011. The MPEG-DASH Standard for Multimedia Streaming Over the Internet. *IEEE MultiMedia*. 18, 4 (Apr. 2011), 62–67.
- [16] Xu, D. and Qian, D. 2008. A bandwidth adaptive method for estimating end-to-end available bandwidth. *Communication Systems, 2008. ICCS 2008. 11th IEEE Singapore International Conference on* (Nov. 2008), 543–548.
- [17] Yang, S.-H. and Tang, W.-L. 2011. What are good CGS/MGS configurations for H.264 quality scalable coding? *Signal Processing and Multimedia Applications (SIGMAP), 2011 Proceedings of the International Conference on* (Jul. 2011), 1–6.
- [18] Zhang, Y., Huangfu, W., Xu, J., Li, K. and Xu, C. 2007. Integrated Rate Control and Buffer Management for Scalable Video Streaming. *Multimedia and Expo, 2007 IEEE International Conference on* (Jul. 2007), 248–251.