2010 2nd International Conference on Education Technology and Computer (ICETC)

Evalvid-RASV: Shaped VBR Rate Adaptation Stored Video System

A. Suki M. Arif, Suhaidi Hassan, Osman Ghazali, Shahrudin Awang Nor
InterNetWorks Research Group, UUM College of Arts and Sciences
Universiti Utara Malaysia, 06010 UUM Sintok, Kedah, Malaysia
{suki1207, suhaidi, osman, shah}@uum.edu.my

***Abstract*—Video traffic is a variable bit rates (VBR) data source in nature and it generates highly bursty traffic. Recent implementations mostly buffer the media source in order to regenerate it in the form of constant bit rates (CBR). Consequently, it will add more delays to the system and thus unable to support the original nature of the video data. Inspired by the works of Hamdi et al. [1] and Lie and Klaue [2], we developed Evalvid-RASV. This system is working on the VBR concept (open-loop video coding), but it is "shaped" so that it will not produce uncompromised bursty traffic without additional delay. With the knowledge of video characteristics in advance, Evalvid-RASV was developed to utilize the information resulting a better algorithm. In addition, we implemented the system in Evalvid-RA environment. It is an environment which is able to perform rate adaptation to the media data source and has an integrated video performance evaluation tools, especially user-perceived video quality. Our experiments have shown that Evalvid-RASV outperforms open-loop VBR in term Peak Signal Noise Ratio (PSNR) value and acceptable delay time.**

Keywords- Stored Video; Evalvid; Leaky-bucket algorithm; Shaped VBR; bursty traffic;

I. INTRODUCTION

Recent years have witnessed explosive growth of the video application via the Internet. The surge of video media applications over the Internet are attributed to the increasing capacity of the Internet and its cost-effectiveness [3]. The Video data source is highly bursty in nature. It is a variable bit rates (VBR) data. Such type of data will cause fluctuating data to the network. Thus it will probably congest the network and in the worst case scenario it may lead to congestion-collapsed network.

Therefore, many applications nowadays are applying constant bit rates (CBR) video traffic. In this arrangement, the video data source will be buffered and then regulated so that in the network interface it will be constant. The weaknesses of these implementations are the additional delay due to buffering. Furthermore, it is unable to support the original nature of the video data, the visual quality of CBR video is not constant and tends to vary according to the video content [4].

Hence, there is a clear need for unconventional solution by taking advantages of both worlds. By that, we mean that we should try to encode the video with an open loop VBR as much as possible, but in the same time we need to control traffic admission into the network without causing extra delays. Thus, we can avoid unpredictable large bursty rate variations without the rigidity and systematic coding delay of CBR coders.

The other issue is that most of the studies are relying on network performance metrics only and they may not produce adequate realistic evaluation to the result of the video application research. There are growing concerns in the video application research to use better video evaluation framework for better perceived quality video measurement, such as the use of Mean Opinion Score (MOS), *psycho-visual* or PSNR [2].

Although there are still many arguments on how to best evaluate the performance of video quality delivered in a simulated environment, many believe that user perceived quality are good enough [2].The growing concern on this issue has lead to the introduction of Evalvid [5]. It is a complete tool or framework to evaluate the quality of video transmission either in real or simulated network.

Motivated by the lack of studies on the controlled-VBR for stored video in a user-perceived quality video performance evaluation framework, we developed Evalvid-RASV. Specifically, we are inspired by the works of Hamdi et al. [1] and Lie and Klaue [2].

In [1], Hamdi et al. introduced the concept of shaped VBR (SVBR). SVBR is a preventive traffic control which allows VBR coding video traffic direct into the network but will regulate unpredictable large bursty traffic by utilizing leaky bucket algorithm. Whereas, Lie and Klaue [2] have implemented Evalvid-RA, a tool-set for rate adaptive VBR video performance evaluation in ns-2, based on Evalvid version 1.2 and Evalvid-NS2 [6].

Basically, Evalvid-RASV implements SVBR in Evalvid-RA environment for a stored video transmission system. In stored video transmission, we can obtain the needed information before calculating SVBR algorithm, thus the SVBR can be optimized further. Moreover, the result can be tested with Evalvid tool-set in order to gain better accuracy of the video performance evaluation.

The remainder of this paper is organized as follows. In the next section, we provide a brief background on the related studies on this issue. In Section III we will elaborate

Evalvid-RASV algorithm which is based on shaped VBR and implemented in Evalvid-RA framework. Then in Section IV, we will explain on how the experiments are done. We will also discuss the results of the experiments in this section. Finally, we conclude the paper in Section V.

II. RELATED WORKS

As stated in the previous section, the need for the using VBR media traffic is clear. However, producing an efficient VBR encoded video delivery is still a challenging tasks. Traditionally, VBR posed problems to networks due to the bursty nature of VBR video traffic. As a result, it contributes significant variation of network traffic, jitters and delays [7].

To address these problems, many studies are controlling video rates. Among the early works, the researchers are adjusting the source coding rate based on the quantization parameters. The application will increase quantization parameter (Q) of MPEG encoder to decrease the source output rate [8]. In fact, in term of users satisfaction, they prefer continuous lower quality display than good quality display with intermittent pause [9]. Other studies maximized a joint control of encoder rate and channel rate to maintain the end-to-end delay of transmitted videos to the appropriate level which is suitable for reliable display [10, 11].

Although most of the previous works are controlling the video rates, using buffering or pre-fetch methods, they are not really constraining the open-loop coding. We believe that unconstraint open-loop VBR coding may not be able to produce satisfactory performance guarantees. Furthermore, a good number of them are using some kind of buffering that will add more delay to the solutions.

Hamdi et al. [1] proposed a novel algorithm to limit the open-loop burst while at the same time allowing open-loop VBR coding provided they are within permitted constraint. They proposed SVBR based on leaky bucket algorithm to perform admission control. The leaky bucket used by them can be considered as an imaginary buffer, thus no extra delay is introduced. Moreover, they assumed that for a fast moving scene with complex image structure, we can reduce slightly the scene quality since human eyes do not have enough time to notice the image details. In addition to that, they have suggested of applying the algorithm at Group of Picture (GoP) granularity, consequently this will yield to a less complex algorithm and lower delay.

However, the algorithms are not specific to the proposed stored video application and it is still open to other researchers to optimize all the parameters used. Furthermore, it is not integrated with the user perceived video quality performance evaluation tool-set. Later on, Lie and Klaue [2] have included the user perceived quality by integrating Evalvid and SVBR resulting in what they call as Evalvid-RA. The tool-set is able to generate real rate adaptive video application traffic.

A. User-Perceived Quality Video Performance Evaluation

Most of the previous research on the video transmission applications are using network performance metrics, for instance less delays/jitters, less packet loss, higher bandwidth utilization/throughput, etc. to evaluate the result [6]. Although those metrics certainly influence the video quality, they do not always reflect the level of quality of video transmission. According to Klaue et al. [5] the transformation could be different for every coding scheme, loss concealment scheme and delay/jitter handling. Consequently, it not sufficient to adequately rate user perceived quality [2, 5, 6] and the perceived quality impression of a human observation is nevertheless the most important factor.

The growing concern of sufficient or adequate user perceived quality evaluation has lead to the introduction of Evalvid [5]. The Evalvid is a complete tool or framework to evaluate the quality of video transmission either in real or simulated network. Therefore, the researchers will be able to evaluate their network designs or setups in terms of perceived video quality by the end user. By using EvalVid-NS2 [6], it enables researchers and practitioners in general to simulate and analyze the performance of real video streams with consideration for video semantics under a vast range of network scenarios.

III. EVALVID-RASV ALGORITHM

Evalvid-RASV uses leaky-bucket algorithm to regulate the media traffic into the network. Leaky-bucket algorithm is implemented by manipulating two main parameters; leak rate r and bucket size b, noted as LB(r,b). Excessive bursty media data will be restricted by this algorithm. Thus, the traffic passing through LB(r,b) should satisfy bursty restriction, written as follows;

$$R_i(t_j - t_k) \le r_i(t_j - t_k) + b_i \tag{1}$$

Where $R_i(t_j\text{-}t_k)$ is the number of bytes emitted into the network by node i from time t_j to t_k.

It is important here to control/maintain the leaky bucket counter within the following range;

$$0 > \text{LB}(r,b) > b \tag{2}$$

For that purpose, Hamdi et al. [1] suggest that we control the size of leaky-bucket "credit" after operation of any k^{th} GoP by utilizing the equation follows;

$$X(k+1) = \min\{b,(\max\{0,X(k)-r\}+R(k))\} \tag{3}$$

Here X(k+1) is a counter of the number of leaky-bucket credits spent at the start of the $(k+1)^{th}$ GoP. Whereas, R(k) is the number of bytes generated in GoP-k. They proposed the initial value for X(0) as b/2, and for every X(k), the value should be in the middle range, that is around b/2.

In order to maintain the value of X(k) so that it is around b/2, we need to control the size of the R(k). The size of the R(k) is very much related to the quantization parameter Q used. We will use the notation Q(k) to represent the quantization value used in a k^{th} GoP. We can use the following expression to determine the quantization parameter value to obtain the desired bit allocation R(k+1);

$$Q(k+1) = Q(k)R(k)/R(k+1) \tag{4}$$

A. Determining Q for GoP (k+1)

For the purpose of determining the correct Q value for the next GoP, we need to determine the size of the next GoP bytes, that is R(k+1). We can use (3) as a basis and the value of X(k+1) should be near to $b/2$. Therefore, R(k+1) should be around;

$$\begin{aligned} &\text{If } X(k+1)-r <= 0 \text{ then} \\ &\quad R(k+1) \approx b/2; \\ &\quad \text{else} \\ &\quad R(k+1) \approx b/2 - (X(k+1)-r); \end{aligned} \tag{5}$$

The next thing to do is to get the corresponding Q(k+1) which will satisfy (5). This is easy to accomplish in stored video system. We can generate a table which consist all the possible Q values (2 to 31 in our case) and the equivalent GoP size for each GoP and Q. We should bear in mind that GoP size for this purpose is not only the sum of all frame sizes in the GoP. We have to consider the overhead as a result of fragmenting frames into packets for the transmission. Accordingly, with the help of simple programming, we can determine which Q value will produce R that can satisfy (5) for the GoP of interest.

B. Burstiness Control

The above-mentioned algorithm did not fully address the bursty or fluctuation which will occur in VBR-based video application. For that, Hamdi et al. [1] proposed their formula. For stored video system we can obtain next GoP information ahead of the k^{th} GoP processing. Thus we want to propose the modified equation for the stored video system;

$$\begin{aligned} &\text{If } R_{real}(k) > r \text{ then} \\ &Q(k+1) = \frac{Q(k)R_{real}(k+1)}{(1-\varepsilon_1(x))R_{real}(k+1)+\varepsilon_1(x)r} \\ &\text{else} \\ &Q(k+1) = \frac{Q(k)R_{real}(k+1)}{(\varepsilon_2(x))R_{real}(k+1)+(1-\varepsilon_2(x))r} \end{aligned} \tag{2}$$

where $R_{real}(k+1)$ is equal to R(k+1) with the overhead for GoP-k. Whereas, x=X(k)/b and ε_i should be in the following range: $0 <= \varepsilon_i(x) <= 1$. The actual value is subjected to tuning.

IV. EXPERIMENTS AND RESULTS

We have tested the algorithm in both simple and complex network using small and large number of frames video clip (for the reason of limited space, we just layout the result for longer video clip. Nevertheless, the results are more or less similar). As a comparison, we run as well "non-adaptive" VBR (open-loop) with the smallest Q (the highest quality) and the largest Q (the lowest quality). All of the experiments are conducted in ns2 simulation. The results are evaluated base on PSNR values; the higher the PSNR value, the better the video quality. Another metric is on the acceptable delay time. We assume that the acceptable delay time is not more than 15 seconds (most literatures indicate that acceptable delay time is several seconds).

A. Simulation Setting

In setting the simulation experiments, we attempt to closely match the real Internet environment wherever possible. Most of the topologies, setting and parameters used in these studies have been based on various works of others, in particular video transmission research.

For simple network experiments, a well known dumb-bell topology is extended into the Evalvid-RASV as depicted in Fig. 1. We believe that it is important to keep our simulation topology relatively simple, yet sufficiently representative of the real world. As recommended in [12], with simple models or experiments, we will get some results or insights. Next, we be able to introduce complexities. In order to simulate a more complex topology or to introduce cross-traffic, a parking-lot topology has been used as illustrated in Fig. 2.

For propagation delays at both end links we used 2 ms and at the bottleneck link used 50 ms one-way delay. Thus, end-to-end round trip propagation delay was 108 ms. This value is closely representative of typical WAN delays on the Internet, which is 105 ms [13].

For bandwidth speed at the receiver link, we used 340 Kbps to represent the lowest broadband home Internet access speed in Malaysia and 100 Mbps to represent typical LAN speed. In the bottleneck link, we used 1.5 Mbps as the bandwidth speed. This bandwidth is sufficiently provisioned so that congestion only occurs on the bottleneck link [14]. All these parameters setting are listed in table I.

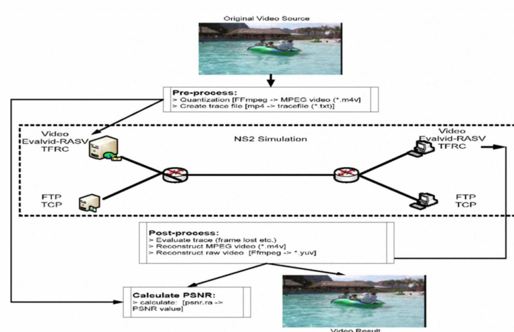

Figure 1. Evalvid-RASV with Dumb-bell Topology

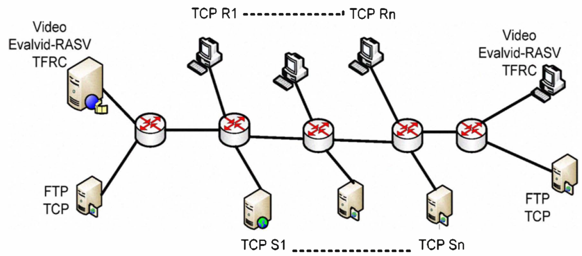

Figure 2. Evalvid-RASV with Parking-lot Topology

TABLE II. SIMULATION PARAMETERS

Simulation parameters	Dumb-bell Topology	Parking-lot Topology
End-user delay	2ms	2ms
Bottleneck delay	50ms	20ms
End-user bandwidth	340Kbps	1Mbps
Number of competing traffics	1	4 (including cross-traffic)
Competing transport protocol	TCP	TCP
VBR rates	340Kbps	1Mbps
Number of video frames	7099	7099
GoP size and number of GoP	12 & 547	12 & 547

In setting-up the parking-lot topology experiment, the bottleneck delay was set as 40 ms. This is done due to additional delay potentially resulted from many cross-traffic transmissions. For the bandwidth speed at receiver link, we used 1 Mbps to represent another rate of broadband home Internet access speed in most places in Malaysia.

B. *Results and Discussions*

Fig. 4a depicts PSNR values from first frame to the last available frame in the dumb-bell topology. Whereas, Fig 4b shows the PSNR values in the parking-lot topology. For each topology, we compared Evalvid-RASV with VBR of quantization value 2 and quantization value 31. The VBR used above is of open-loop video coding, without rate adaptation.

For play-back delay time comparison, we charted them in Fig. 5a and 5b. Fig. 5a illustrates delay time for VBR with quantization value 31 and Evalvid-RASV in both dumb-bell and parking-lot topologies. We separated delay time for VBR with quantization value 2 in Fig. 5b because the delay time goes up to more than 2000 seconds, thus it can improves readability.

We would like to highlight the fact that the packet numbers in Fig. 5b have been regenerated using a simpleawk program. This is due to the fact that VBR Q=2 generated more than 100 000 packets. Regrettably, our tool is not able

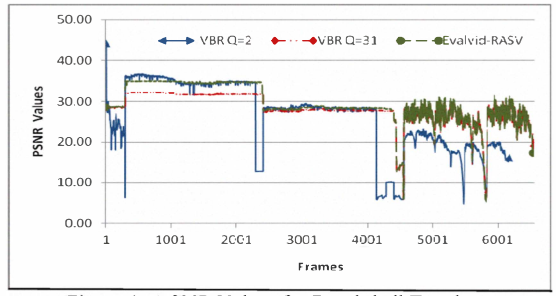

Figure 4a. PSNR Values for Dumb-bell Topology

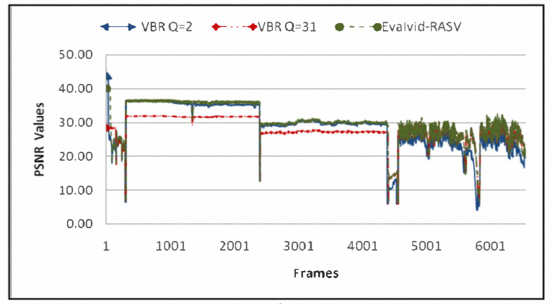

Figure 4b. PSNR Values for Parking-Lot Topology

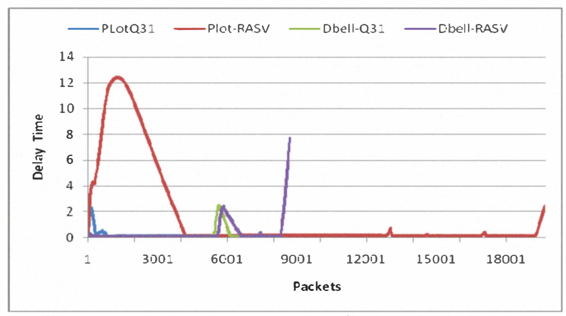

Figure 5a. Delay Time for VBR Q=31 and Evalvid-RASV

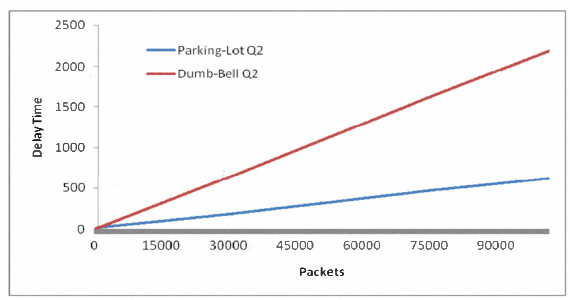

Figure 5b. Delay Time for VBR Q=2

to plot the chart with that size of data. Thus, our simple *awk* program grouped the packets into 30-packets group and choose the highest delay in each group as the delay time.

Table 2 tabulated the experiment results. The main parameters that we measured are number of frames lost during the many times of experiment (column two), the average PSNR of each test for all correctly-received frames (column three), number of packets which have been received after 15 seconds delay (fourth column), maximum and average delay time in seconds (fifth and sixth column).

From the table, we can say that Evalvid-RASV has better algorithm because it has the highest PSNR value compared to open-loop VBR in both dumb-bell and parking-lot topology experiments. In dumb-bell topology experiment, Evalvid-RASV has 29.17 PSNR average value compared to 26.06 and 28.15 for VBR with Q=2 and Q=31 respectively. Whereas, in parking-lot topology, Evalvid-RASV has PSNR average value of 30.27 compared to 28.80 and 28.07 for VBR Q=2 and Q=31 accordingly. Although the differences are small, it is significant because it involves more than 7000 frames.

Though VBR Q=2 in dumb-bell topology (Fig. 4a) started with a very high value (around 45), later on the value degraded intermittently below 10. It may suffer packet lost due to high traffic generated from high quality clips. The receiver link was not able to support such high volume of the video traffic. Due to high number of packet lost, the video quality degraded significantly. It is reflected by the low PSNR values gained.

Furthermore, only around 6700 frames have been received correctly after the experiment. The average PSNR value was calculated based on correctly-received frames. It means that should we compare higher number of correctly-received frames, the PSNR average value should be lower.

In term of packets/frames delay, none of the packets/frames in both Evalvid-RASV experiments has exceeded 15 seconds constraint. Thus all of the packets/frames received are playable. Both of the packets/frames delay for VBR Q=2 experiments suffered from long delay. For the experiment in parking-lot topology, the delay goes up to 10 minutes (600 seconds) and the dumb-bell topology experiment, the average frames have arrived after 18 minutes (1080 seconds).

TABLE III. EXPERIMENT RESULTS

	# Frames Lost	Ave. PSNR	# pkts delay > 15 s	Max. Delay (s)	Ave. delay (s)
VBR Q=2 in Dumb-bell	381	26.06	101016	2187.3	1089.6
VBR Q=31 in Dumb-bell	10	28.15	0	2.496	0.219
E-RASV in Dumb-bell	23	29.17	0	7.69	0.409
VBR Q=2 in Parking-lot	33	28.80	101055	620.77	315.24
VBR Q=31 in Parking-lot	7	28.07	0	2.36	0.209
E-RASV in Parking-lot	7	30.27	0	12.47	1.7

Although the VBR Q=31 experiment produced very short delay (less than three seconds), the video quality is the lowest. Furthermore, when we inspect the result of Evalvid-RASV experiments closer, we believe that the algorithm can be fine tuned further.

V. CONCLUSION

The outcome from the experiment revealed that Evalvid-RASV have produced a better result both in terms video of quality gained and acceptable delay time. We have shown that even in low bandwidth link and in network with many cross-traffics, Evalvid-RASV has performed satisfactorily.

We believed that the encouraging outcomes are gained as a result of "shaping" policy of the algorithm and the exploitation of the availability of the video characteristics beforehand. We also believe that the algorithm can be improved further by analyzing in-depth on why at certain period the PSNR values dropped drastically and how to increase further PSNR values when the network is not congested.

REFERENCES

[1] H. Hamdi, J. W. Roberts, and P. Rolin, "Rate control for VBR video coders in broad-band networks," *Selected Areas in Communications, IEEE Journal on,* vol. 15, pp. 1040-1051, 1997.

[2] A. Lie and J. Klaue, "Evalvid-RA: Trace Driven Simulation of Rate Adaptive MPEG-4 VBR Video," *Multimedia Systems,* vol. 12, 2008.

[3] S. Liang and D. Cheriton, "TCP-RTM: Using TCP for Real Time Applications," *Submitted to IEEE ICNP '02,* 2002.

[4] K. Chun-Wai, J. Y. B. Lee, M. Hamdi, and V. O. K. Li, "Turbo-slice-and-patch: an algorithm for metropolitan scale VBR video streaming," *Circuits and Systems for Video Technology, IEEE Transactions on,* vol. 16, pp. 338-353, 2006.

[5] J. Klaue, B. Rathke, and A. Wolisz, "EvalVid - A Framework for Video Transmission and Quality Evaluation," in *the 13th International Conference on Modelling Techniques and Tools for Computer Performance Evaluation*, 2003, pp. 255--272.

[6] C.-H. Ke, C.-K. Shieh, W.-S. Hwang, and A. Ziviani, "An Evaluation Framework for More Realistic Simulations of MPEG Video Transmission," *Journal of information science and engineering,* vol. 24, pp. 425-440, 2008.

[7] L. Guanfeng and L. Ben, "Balancing Interruption Frequency and Buffering Penalties in VBR Video Streaming," in *INFOCOM 2007. 26th IEEE International Conference on Computer Communications. IEEE*, 2007, pp. 1406-1414.

[8] D. J. Reininger, R. Dipankar, and J. Y. Hui, "Bandwidth renegotiation for VBR video over ATM networks," *Selected Areas in Communications, IEEE Journal on,* vol. 14, pp. 1076-1086, 1996.

[9] C. Krasic, "A Framework for Quality-Adaptive Media Streaming: Encode Once - Stream Anywhere," PHD Dissertation, School of Science & Engineering, Oregon Health & Science University, 2004.

[10] H. Chi-Yuan, A. Ortega, and A. R. Reibman, "Joint selection of source and channel rate for VBR video transmission under ATM policing constraints," *Selected Areas in Communications, IEEE Journal on,* vol. 15, pp. 1016-1028, 1997.

[11] A. R. Reibman and B. G. Haskell, "Constraints on variable bit-rate video for ATM networks," *Circuits and Systems for Video Technology, IEEE Transactions on,* vol. 2, pp. 361-372, 1992.

[12] R. Jain, *The Art of Computer System Performance Analysis*: John Wiley, 1991.

[13] S. Boyden, A. Mahanti, and C. Williamson, "Characterizing the Behaviour of RealVideo Streams," in *SCS Symposium on Performance Evaluation of Computer and Telecommunication Systems (SPECTS)*, Philadelphia, PA, USA, 2005.

[14] Z. Wang, S. Banerjee, and S. Jamin, "Media-Aware Rate Control," Tech. Rep., CSE-TR-501-04, University of Michigan, 2004.