Accurate Prediction of Real-Time MPEG-4 Variable Bit Rate Video Traffic

Kang Yong Lee, Moonseong Kim, Hee-Seon Jang, and Kee-Seong Cho

ABSTRACT—In this letter, we propose a novel algorithm to predict MPEG-coded real-time variable bit rate (VBR) video traffic. From the frame size measurement, the algorithm extracts the statistical property of video traffic and utilizes it for the prediction of the next frame for I-, P-, and B- frames. The simulation results conducted with real-world MPEG-4 VBR video traces show that the proposed algorithm is capable of providing more accurate prediction than those in the research literature.

Keywords— Traffic prediction, MPEG video, VBR, cubic spline interpolation.

I. Introduction

Recently, multimedia applications, such as video conferencing, video on demand (VoD), and IP broadcasting have become major parts of network services and the efficient delivery of real-time variable bit rate (VBR) video traffic has been the subject of many studies in both academia and industry. However, because real-time VBR video traffic typically is very bursty and complex depending on the encoding types and the original scenes, it is very difficult to achieve the desired quality of service (QoS) requirements. To meet these challenges, there have been many attempts to adaptively manage network resources, such as network bandwidth allocation by predicting the traffic dynamics in advance [1].

In this letter, we propose a novel traffic prediction algorithm for real-time MPEG-4 VBR video traffic. Based on the past traffic patterns observed, the proposed algorithm captures the traffic characteristics using the cubic spline (CS) interpolation method [2] and predicts the next video frame. Using intensive simulation study, we compare the performance of the proposed algorithm with three well-known prediction algorithms in the research literature: the prediction scheme using adaptive network fuzzy inference system (ANFIS) [3], least mean square (LMS)–based prediction [4], and neural network (NN)–based MPEG-4 video traffic prediction algorithm [5].

Manuscript received June 29, 2007; revised Aug. 23, 2007.

Kang Yong Lee (phone: + 82 42 860 5520, email: kanglee@etri.re.kr) and Kee-Seong Cho (email: chokis@etri.re.kr) are with the Broadband Convergence Network Research Division, ETRI, Daejeon, Rep. of Korea.

Moonseong Kim (email: moonseong@skku.edu) is with the Department of Electrical and Computer Engineering, Sungkyunkwan University, Suwon, Korea.

Hee-Seon Jang (email: hsjang@ptu.ac.kr) is with the Department of Business and Administration, Pyeongtaek University, Pyeongtaek, Rep. of Korea.

II. Prediction Algorithm Using CS Interpolation

The proposed algorithm first extracts the statistical properties from the observed past traffic patterns by calculating the probability density function (PDF) of frame size. Thereafter, it utilizes the PDF for the prediction of next frame. We call this proposed algorithm PuP, a prediction algorithm using PDF.

1. PDF Estimation Using CS Interpolation

For PDF calculation, we use the CS interpolation method which is widely used in piecewise polynomial approximation [2]. The CS interpolation fits a polynomial to a set of data points, based on the idea of connecting every adjacent pair of control points by a cubic. Note that each pair of control points has a different cubic. The individual cubic functions are then joined together at every control point into a smooth curve, resulting in a complete fitting function, $f(x)$, over the entire range.

Suppose we have n control points, (x_i, y_i), $i = 1,\cdots, n$. Then $f(x)$ consists of n-1 cubic segments, $S_i(x)$, of the following form:

$$S_i(x) = \alpha_i (x - x_i)^3 + \beta_i (x - x_i)^2 + \gamma_i (x - x_i) + \delta_i, \qquad (1)$$

where $x_i \le x \le x_{i+1}$ and $i = 1,\cdots, n-1$. The four unknowns, α_i, β_i, γ_i, and δ_i can be derived from

$$S_i(x_i) = y_i, \qquad S_i(x_{i+1}) = S_{i+1}(x_{i+1}),$$
$$S'_i(x_{i+1}) = S'_{i+1}(x_{i+1}), \quad S''_i(x_{i+1}) = S''_{i+1}(x_{i+1}). \tag{2}$$

Note that $S'_i(\cdot)$ and $S''_i(\cdot)$ in (2) are the first and second derivatives of $S_i(\cdot)$, respectively.

Finally, we get the PDF of the random variable x, $f_X(x)$, as

$$f_X(x) = f(x) \Big/ \int_{x_1}^{x_n} f(x)\ dx\ .$$

2. Real-Time MPEG-4 VBR Video Traffic Prediction

In MPEG-coded video sequences, I-, P-, and B-frames are arranged in a deterministic pattern called group of picture (GOP). PuP first separates the video frames into I, P, and B subgroups by GOP pattern, and measures the size of frames within each subgroup. For the prediction of the next frame, it considers the last N frames it measured. In other words, to predict the k-th frame, it calculates the PDF of the frame size using the N frames from the (k-N)th to the (k-1)th, which we define as $rWin_k$. As shown in Fig. 1, to predict the size of the k-th I-frame, our algorithm calculates the PDF of the frame size for the frames from I_{k-N} to I_{k-1}, that is, $rWin_k$. Note that $N \geq 1$ and $k \geq 2$.

First, PuP finds the maximum frame size, P_{max}, and the minimum frame size, P_{min}, among $rWin_k$. Then, we divide the closed interval [P_{min}, P_{max}] into m subintervals equally, as shown in Fig. 2.

Let us define the i-th subinterval SI_i as

$$SI_i = \begin{cases} [P_i,\ P_{i+1}) & \text{for } 1 \leq i \leq m-1, \\ [P_i,\ P_{i+1}] & \text{for } i = m. \end{cases}$$

Based on the frame size, we categorize every frame of $rWin_k$ into each subinterval. For example, if the size of a frame is larger than P_3 and does not exceed P_4, the frame belongs to SI_3. Let us denote the number of frames belonging to a subinterval SI_i as q_i. Then, for PDF calculation, we choose m control points as

$$x_i = \left\lceil \frac{P_{i+1} - P_i}{2} \right\rceil \text{ and } y_i = q_i, \text{ for } i = 1, \cdots, m.$$

In order to calculate the PDF of the frame size for $rWin_k$, PuP uses m control points for CS interpolation. Consequently, PuP obtains the PDF of the frame size for $rWin_k$, that is, $f_X(x)$. Furthermore, from the $f_X(x)$, the mean μ_k and the standard deviation σ_k also can be derived easily. Finally, PuP predicts the

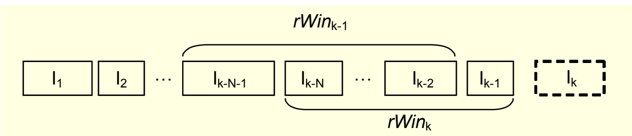

Fig. 1. Prediction step and *rWin*.

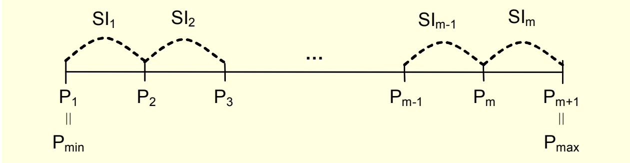

Fig. 2. Subintervals.

frame size of I_k, $\bar{I}_k$, as

$$\bar{I}_k = \mu_k + \frac{E_{k-1}}{\sigma_{k-1}} \times \sigma_k,$$
$$E_{k-1} = I_{k-1} - \bar{I}_{k-1}, \quad k \geq 2. \tag{3}$$

where E_{k-1} is the prediction error for the size of frame I_{k-1}, and σ_{k-1} is the standard deviation derived from PDF calculated by $rWin_{k-1}$ in the previous step, as shown in Fig. 1. This means that PuP learns the prediction error from the previous step and utilizes it for the next prediction. In the same way, the prediction algorithm is applicable without any modification to the prediction of P- and B-frames.

III. Performance Analysis

Using real-world MPEG-4 video traffic traces, we conducted intensive simulations to compare the performance of our algorithm and three well-known prediction algorithms, ANFIS, LMS, and NN. The high quality video traffic traces used in this study are obtained from [6]. More details of traffic characteristics and encoding schemes about the video traces are available in [6].

As a performance metric, we use root mean square error (RMSE). The RMSE is the ratio between the sum of the square of the prediction errors and the sum of the square of the actual values. It can be written as

$$RMSE\ (\%) = \frac{\sum (Size - \overline{Size})^2}{\sum Size^2} \times 100,$$

where $Size$ is the actual value and $\overline{Size}$ is the predicted one.

Table 1 summarizes the performance evaluations in terms of RMSE. Note that N and the number of subintervals are fixed as five and three, respectively, in all the simulations. They were selected as producing the best prediction performance after conducting the simulations with different values. In [3], RMSE was used as a performance metric. We used the same video traces in our evaluations as in [3]; hence, the RMSE for ANFIS, LMS and NN in Table 1 are from [3]. The results reveal that our algorithm outperforms ANFIS, LMS and NN in most cases. PuP shows 14% improvement on average over ANFIS, 40% over LMS, and 49.7% over NN. The best prediction

Table 1. RMSE of I-, P-, B-frames for PuP, ANFIS, LMS, and NN.

Movie	Type	PuP	ANFIS	LMS	NN	%Improv. over ANFIS	%Improv. over LMS	%Improv. over NN
Aladdin	I	**2.03**	2.11	2.35	2.60	3.79	13.62	21.92
	P	**3.97**	10.60	12.48	10.30	62.55	68.19	61.46
	B	**3.57**	3.12	3.88	8.20	-14.42	7.99	56.46
ARD Talk	I	**1.12**	0.77	1.76	0.90	-45.45	36.36	-24.44
	P	**2.59**	5.03	6.10	5.90	48.51	57.54	56.10
	B	**1.41**	1.12	1.42	3.20	-25.89	0.70	55.94
Jurassic Park I	I	**1.01**	0.69	0.80	0.80	-46.38	-26.25	-26.25
	P	**1.60**	2.28	3.88	4.00	51.22	57.76	60.00
	B	**1.52**	2.40	2.28	2.20	36.67	33.33	30.91
Die Hard III	I	**1.98**	2.90	4.41	2.90	31.72	55.10	31.72
	P	**2.52**	7.25	12.30	9.00	65.24	79.51	72.00
	B	**1.59**	3.55	3.44	4.00	55.21	53.78	60.25
Lecture Room	I	**0.17**	0.05	0.12	0.20	-240.00	-41.67	15.00
	P	**1.32**	3.43	4.71	6.90	61.52	71.97	80.87
	B	**1.03**	1.49	2.22	28.30	30.87	53.60	96.36
Silence of the Lambs	I	**1.86**	1.74	2.28	3.60	-6.90	18.42	48.33
	P	**2.71**	13.18	14.29	11.00	79.44	81.04	75.36
	B	**1.36**	2.85	7.30	27.80	52.28	81.37	95.11
Star Wars	I	**1.49**	1.66	2.60	1.50	10.24	42.69	0.67
	P	**6.84**	10.16	14.15	9.20	32.68	51.66	25.65
	B	**1.01**	0.64	1.31	3.50	-57.81	22.90	71.14
Skiing	I	**1.71**	2.70	3.24	2.00	36.67	47.22	14.50
	P	**1.88**	3.60	4.41	6.20	47.78	57.37	69.68
	B	**1.31**	1.39	1.87	14.70	5.76	29.95	91.09

improvement is approximately 96.4%. In rare cases, the predictions of PuP are somewhat inaccurate compared to other prediction schemes; however, for P-frames, our algorithm always makes more accurate predictions than other prediction algorithms. Moreover, it offers good performance for B-frame prediction except for Aladdin and ARD Talk. This is because P- and B-frames have strong correlations with other P- and B-frames, and they are encoded from other I- and P-frames, while I-frames are encoded with no information from any other frames.

The performance results indicate the success of our proposed algorithm in predicting real-time VBR video traffic.

IV. Conclusion

We presented a novel prediction algorithm for real-time MPEG-4 VBR video traffic. To capture the traffic property from a VBR video stream, we calculate the PDF using the CS interpolation method and utilize it for the prediction of the next frame. The simulation results conducted with real-world MPEG-4 VBR video traffic showed that PuP significantly improves prediction accuracy compared to that of the ANFIS, LMS, and NN algorithms.

References

[1] Euee Seon Jang, "Low-Complexity MPEG-4 Shape Encoding towards Realtime Object-Based Applications," *ETRI Journal*, vol. 26, no. 2, 2004, pp. 122-135.

[2] C.F. Gerald, *Applied Numerical Analysis*, Addison-Wesley, 1999.

[3] A. Abdennour, "Short-Term MPEG-4 Video Traffic Prediction Using ANFIS," *Int'l J. of Netw. Mgmt.*, vol. 15, 2005, pp. 377-392.

[4] A. Adas, "Using Adaptive Linear Prediction to Support Real-Time VBR Video Under RCBR Network Service Model," *IEEE/ACM Trans. Netw.*, vol. 6, no. 5, Oct. 1998, pp. 635-644.

[5] A. Bhattacharya, A.G. Parlos, and A.F. Atiya, "Prediction of MPEG-Coded Video Source Traffic Using Recurrent Neural Networks," *IEEE Trans. Signal Process.*, vol. 51, no. 8, Aug. 2003, pp. 2177-2190.

[6] MPEG-4 and H.263 Video Traces for Network Performance Evaluation, Technical University, Berlin, 2001.