A Network-Adapative SVC Streaming Strategy with SVM-Based Bandwidth Prediction

Xu Yanling, Wu Baolin, and Yang Liushan

Abstract—**This paper presents an online streaming strategy for SVC (Scalable Video Coding).The crucial technique adopted is the combination of network bandwidth prediction and rating mechanism for different streaming options. An SVM (Support Vector Machine) algorithm is used for prediction of network bandwidth variation, with which the system then evaluates every strategy that can be made at that time. The experiment shows this strategy not only cut down video playing back interruption times, but also improves overall video quality users perceive.**

Index Terms—**Bandwidth prediction, support vector machine, SVC streaming.**

I. Introduction

The techniques for video content delivery have evolved from UDP (User Datagram Protocol) to HTTP (Hypertext Transfer Protocol) based methods. HTTP based solutions have many advantages including easy deployment and better utilization of web cache infrastructure. Furthermore, they are Firewall and NAT (Network Address Translate) friendly. Thus many mainstream video providers, such as YouTube, deliver contents via HTTP. The providers also allow users to choose from multiple versions with different quality. To do this, videos will be encoded into different and independent files with much redundant information. So space waste is one shortcoming in mainstream approaches. Another defect is although the mechanism for choosing different video versions is provided; users are not usually making the right choice and suffer from playback interruption or low quality of downloaded videos.

Better space utilization can be achieved by adopting H.264/SVC [1] standard, which also improves network efficiency. If user wants to upgrade cached video to higher quality, only enhancement layers need to be requested. While in traditional approaches, the contents that are cached locally will be abandoned. By encoding videos into different dependent layers, less space is occupied. However, to overcome the other defect of traditional video streaming, i.e. user may not be aware of current bandwidth and trends to select wrong video version, further investigation should be conducted. Although SVC standard provides a mechanism for choosing video quality on the fly, the strategy is still a challenging and open problem [2]. How to get full use of network bandwidth and cut down playback interruption times, as well as ensuring playback smoothness are not trivial. Some research has been done in SVC streaming strategy, among which Xiang's method [2] (we call it OS, i.e. Optimization Strategy) is one of the state-of-art methods. Although Xiang proves the performance is optimal, it is based on an unrealistic assumption of the bandwidth. Xiang assumes that bandwidth variation can be modeled as random process, but no evidence is provided and it doesn't apply to different applications. In this paper, we bring up with a new strategy (We call it BPB, i.e. Bandwidth Prediction Based algorithm) based on network bandwidth prediction, which is much more flexible and easy to implement.

Our contributions are: an SVM method is presented to predict bandwidth. Then a rating strategy for different streaming options is introduced based on the prediction. After simulating some scenarios in experiments, we show that the strategy for streaming improves overall video quality that is subject to available bandwidth as well as ensuring playing smoothness.

The rest of the paper is organized as follows: Section II is the related works and Section III shows the design of an SVM based bandwidth predictor. Then Section IV is the rating algorithm for evaluating different streaming options. Simulations and experiments are analyzed in Section V, followed by concluding remarks and further research issues in Section VI.

II. Related Works

A. Network Bandwidth Prediction

The techniques for predicting available bandwidth can be classified as linear based and non-linear based approaches. Linear approach is easy to understand and implement. So many projects like NWS (Network weather service), adopt linear predictor. In other projects, like RPS [3] (Resource Prediction System), the prediction mechanism is based on time-series model, such as AR (Autoregressive) model, MA (moving average) model. In [4], Yao makes a comparison about these linear methods. But linear model is inadequate to adapt to complex network conditions. Non-linear model can be applied in wider context. From ANN (Artificial Neural Network) to SVM, the models are more and more robust. In [5], [6], it shows ANN is a powerful tool to predict network bandwidth. Although ANN outperforms linear methods, ANN has some disadvantages [7] because it's based on ERM (Empirical Risk Minimization) principle, so it's prone to over-fitting and under-fitting. To overcome this disadvantage, SVM is the new method that attracts many researchers' attention [8], [9]. When using SVM, the feature selected for training plays a big role in prediction accuracy. L.Hu [10] makes a conclusion about current SVM methods and brings up with new feature selection approach, adopting Nu-SVR (Nu-Support Vector Regression) to do the

Manuscript received August 30, 2013; revised November 1, 2013.

The authors are with the School of Software Engineering, Tongji University, Shanghai, P. R. China (e-mail: xuyanling@tongji.edu.cn, wubaolin2009@126.com, ylx0327@gmail.com).

DOI: 10.7763/IJFCC.2014.V3.297

prediction.

B. SVC Video Download Strategy

As traditional HTTP based methods are getting more and more popular for video content provider, many commercial products like Microsoft Smooth Streaming, implement adaptive video streaming. Besides, C. Liu [11] comes up with some adaptive download strategy for video downloading. In [12], many strategies are compared and new method is introduced. But all the methods mentioned above are not applied for SVC streaming.

In [13] a simple SVC streaming strategy is present, which oversimplifies prediction by adopting a naïve linear filter. One of the state-of-art algorithms for SVC streaming is Xiang's method [2]. Although the result is optimal when underlying bandwidth conforms to MDP (Markov Decision Process), the modeling itself doesn't scale well. Another shortcoming of Xiang's algorithm is that the method is not online.

To present more robust SVC streaming algorithm, we bring in bandwidth prediction to facilitate the process of choosing best strategy. The major tasks in this paper are: First, we adopt a machine learning approach to predict network bandwidth. Then we bring up an efficient algorithm called BPB for streaming SVC videos. The result shows the advantages over current algorithms in this field.

III. Bandwidth Prediction

In this section, we bring up with an SVM-based approach to do the prediction. The features used for SVM are previous measurements of bandwidth. We consider two different features, one is absolute value of band and the other is differential value. We also compare this algorithm with ANN (artificial neural network) approach.

In [8], the author comes up with an SVM-based approach to predict TCP throughput. He observes that the prediction error is conforming to Gauss distribution; our experiment reconfirms it. Meanwhile, in order to prove that SVM-based approach is a better choice, we come up with two ANN-based methods for comparison.

After receiving a video segment, the rough bandwidth is measured for this period. Then we add it to ANN or SVM to train. In order to improve responsiveness, two AI(Artificial Intelligence) modules are employed, one is for retraining and the other is for predicting. The retraining process occurs when every 5 video segments are downloaded. Thus the time between feeding features to AI module and getting predicted result is ignorable, since AI module for predicting is not involved in training process. After retraining finishes, the parameters of two AI modules are synchronized. The architecture of the AI module is shown in Fig. 1.

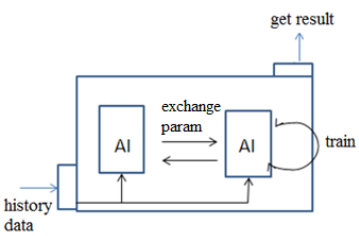

Fig. 1. Predictor architecture.

For ANN, we use two methods, one is differential value based (DIFF-ANN) and the other is absolute value based (ABS-ANN). For DIFF-ANN, we use (T_{n-1}-T_{n-2}, T_{n-2}-T_{n-3}… T_{n-5}-T_{n-6}) as the feature vector. While for ABS-ANN, the feature is absolute bandwidth. In our experiment, ANN consists of 3 layers, each of which contains 5, 10, 14 units.

For SVM, we use SVM regression to do the prediction. For absolute value based approach (ABS-SVM), the feature vector of T_n is (T_{n-1}, T_{n-2}…T_{n-5}).We also implement differential bandwidth based method (DIFF-SVM),that is, the feature for(T_n-T_{n-1}) is (T_{n-1}-T_{n-2},T_{n-2}-T_{n-3},…, T_{n-5}-T_{n-6}).

The results are presented in Section V.

IV. SVC Download Strategy

In this section, the bandwidth predictor is adopted to facilitate SVC download strategy.

Good SVC download strategy should present users with high quality video, as well as ensuring playback smoothness. Besides, it should avoid video play interruption due to buffer underflow. We can achieve these goals by combining bandwidth and download strategy together. (1) is adopted to give ratings for each action when previous request is finished.

$$R(a) = \big(1 - P(a)\big) * (P1 * AVQ + P2 * PS)) \quad (1)$$

$R(a)$ is the ratings for a specific action. P (probability), AVQ(*average quality*) and *PS(playback smoothness)* consider interrupt ratio, video quality and playback smoothness respectively.

Supposing *next* indicates the segment to be downloaded next time slot. The action set, which is denoted as ACTION_SET, is {-5,-4,-3,-2,-1,1,2,3,4}, and its meaning is in Table I:

TABLE I: Action Set

Action	Meaning
$a \in [-5, -1]$	upgradethe segment with id *next* - a to a higher layer(if possible)
$a \in [1, 4]$	download the *next* segment with layer id a

For one action $a \in$ ACTION, AVQ indicates the average quality. We only consider the local range of [*next-5, next*]. AVQ is just arithmetic average layer id of local segments.

$$AVQ = \frac{\sum_{i=next-5}^{next} segment\_max\_layer(i)}{6} \quad (2)$$

segment_max_layer(i) is the highest layer id in segment i.

PS indicates playback smoothness. Continuous video play with the same layer is called one run r and its length is denoted asNr.

$$PS = \frac{\sqrt{\sum_r Nr^2}}{6} \quad (3)$$

The parameter balancing video quality and playback smoothness is also important.P_1 and P_2 are used for this purpose. Here we chooseP_1 = 1000.0 and P_2=1.5 since after many experiments, this configuration makes a good balance.

The innovative part of this algorithm is the calculation of interruption probability. This step is involved in (4).

$$P(a) = \frac{1}{1+e^{f(a)}} \quad (4)$$

$$f(a) = -\frac{dtime(a) + decodetime(layer) - rtime(a)}{200}$$

*a*is one action in ACTION_SET.

dtime is the predicted download time from band predictor. *rtime* is the time before the chosen segment is to be played. This *P*(*a*) function is inspired from the activation function commonly used in ANN, which gets a real number as its input and outputs a number between 0 and 1. Decoding time should not be ignored. The estimation of decoding time is rough; it only depends on the layer id to be downloaded, as shown in Table II. But it's very rough and aren't flexible among different videos. How to give more rigid model for estimating the decoding time for different layer is left for future research.

TABLE II: DECODING TIME ADJUST PARAM

Layer ID	Decode time (in ms)
0	500
1	800
2	1200
3	1300

Intuitively *P* (*a*) function indicates the longer download time, the higher risk of interruption. The higher layer to be downloaded, the higher risk of interruption it will be due to decoding time.

After getting all ratings, we then run the same method recursively. Since ACTION_SET contains 9 different actions, a 3-step calculation involves 9^3different possible paths. The algorithm calculates all possible paths then select the action that leads to maximum of (5).

$$R(a) \times R(next(a)) \times R(next(next(a))) \quad (5)$$

*next (a)*indicates one of the action taken after the *a* action. In our experiment, we find 3-step is enough to get good results and 4 or more step is not necessary since the performance doesn't improve much. If video play buffer size is more than *BUFFER_MAX_SIZE* * *THRESHOLD_SLEEP* (250KB and 1.2 respectively), no action is taken until some segments in buffer are consumed.

V. PERFORMANCE

In this section, we introduce some metrics for evaluating bandwidth algorithm and show the results. Then we'll turn to the SVC download algorithm, talking about metric and performance about it.

A. Bandwidth Prediction

The bandwidth in time *t* is denoted as *b*(*t*) and the predicted value is *p*(*t*). Two metrics are adopted to verify the effectiveness of ANN-based and SVM-based methods.

1) Mean of absolutedifference (MAD)

$$\text{MAD} = \frac{|\, p(t) - b(t)|}{num\_predictions}$$

2) Mean of square error (MSE):

$$\text{MSE} = \frac{(p(t) - b(t))^2}{num\_predictions}$$

Besides, we also consider the relative error distribution for each method.

For ANN, we find that ABS-ANN is very ineffective, the reason is mainly to lack of training data set. So we omit the results for it. We compare the DIFF-ANN and ABS-SVM method, as shown in Table IV.

3) One sample from the prediction.

Here we show one sample result of ABS-SVM and DIFF-ANN.

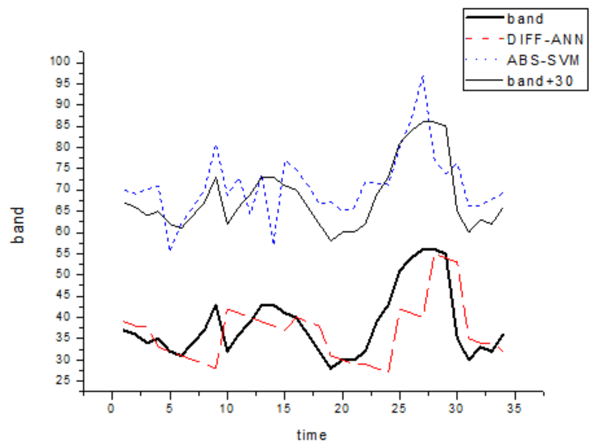

Fig. 2. Prediction result.

In Fig. 2, we can see the ABS-SVM can fit the bandwidth approximately. While the ANN seems faillinng to fit it well. ANN has some delayed behavior, that is, it predicts bandwidth variation until the variaion emerges. The main reason is that ANN can't identify underlying pattern when the traning set is small. Later results aslo show SVM outperforms ANN using other metrics.

4) MAD and mse

TABLE III: MAD AND MSE

Table Head	MAD	MSE
DIFF-ANN	6.92	79.3
ABS-SVM	4.91	44.9
DIFF-SVM	6.45	131.27

In this table, ABS-SVM outperforms other methods. Although DIFF-SVM has better MAD than DIFF-ANN, its MSE is much more than others and as shown in Table III, DIFF-SVM are more likely to generate extreme value which lies far away from the underlying bandwidth.

5) Error distribution

TABLE IV: ERROR DISTRIBUTION

method	DIFF-ANN	ABS-SVM	DIFF-SVM
<-35	0	0	9
[-35,-25]	1	0	6
[-25,-15]	8	5	6
[-15,-5]	26	44	16
[-5,5]	92	119	127
[5,15]	55	23	35
[15,25]	19	2	2
>25	0	1	0

The performance of DIFF-SVM is poor; it tends to predict unrealistic bandwidth. As for DIFF-ANN and ABS-SVM, ABS-SVM is more accurate so the ABS-SVM method is adopted as the tool for prediction bandwidth. We also find the error distribution is approximately conforming to Gauss distribution.

B. Performance of SVC download

The QoE (Quality of Experience) Metrics for evaluating SVC download are IR (Interruption Ratio), APQ(average playback quality) and PS(playback smoothness).

1) IR(Interruption ratio)

$$IR = \frac{interrupted\ time}{200},$$

200 indicate the total number of segments for this experiment.

2) APQ(Average playback quality)

This metric indicates the average video quality the user perceived. There are totally N runs and each run r has length *Nr* with layer index i. APQ (average playback quality) is defined as

$$\text{APQ}= \frac{\sum_r Nr * i}{\sum_r Nr}$$

3) PS(Playback smoothness):

Longer playing of the same layer means that user can merely realize the existence of underlying switch between different layers. Frequent switching between different layers leads to unacceptable user experience. The PSis defined as

$$PS = \sqrt{\frac{\sum_{r=1}^{N} Nr^2}{N}}$$

To do the experiment, we encode the “Big Buck Bunny” video into 200 segments with 4 different layers and display 24 frames per second. The way we conduct the experiments is:

1) Convert .mov file .yuv file using ffmpeg.
2) Get the first 3400 frames of the yuv file and then split it into 200 segments with 17 frames per segment.
3) For each segment, use JSVM 9.12 to encode the video into 4 different layers, the configurations of each layer is in Table V.
4) For each segment encoded as .svc file, extract different layers into different files.
5) Put all the files into a HTTP server running light tpd.

TABLE V: LAYER CONFIGURATION

Resolution	Average bit-rate(KB)	Y-PSNR(Y-Peak Signal to Noise Ratio)	Layer Index
320x180	14.1	35.47	0
320x180	29.86	39.44	1
640x360	45.47	35.90	2
640x360	78.27	39.31	3

In aspect of bandwidth, we try to limit the maximum bandwidth of the client using NetLimiter for this purpose. 4 different experiments are conducted, with different download bandwidth limited to 20KB/s, 40KB/s and 60KB/s and varying bands in the fourth. The result for experiment1, 2, 3 are presented in Table VI.

4) Experiment 1 (download rate 20KB/s)

TABLE VI: EXPERIMENT PERFORMANCE

Experiment	Strategy	IR	AVQ	PS
1	OS	0.055	0.69	2.5
	BPB	0	0.87	4.1
2	OS	0	2.07	2.6
	BPB	0	2.31	4.7
3	OS	0	2.65	4.3
	BPB	0	2.87	7.2

In this experiment, although the underlying bandwidth is less than layer 1’s average bit-rate, the average quality is close to layer 1. The PS is inacceptable range. The OS algorithm, under this circumstance, behaves badly. The main reason for that is that network bandwidth is totally different from Xiang’s assumption. The underlying network’s difference is a major disadvantage of Xiang’s algorithm. In Xiang’s method, the bandwidth is modeled as MDP and experimental settings conform to the assumption. While in experiment 1, when the underlying bandwidth differs from Xiang’s, unrealistic assumption leads to poor performance. BPB tries to mine underlying pattern in different network environment, it’s more applicable to various settings.

5) Experiment 2 (download rate 40KB/s)

In Table VII, we can see that our algorithm want to make full use of the underlying network. The AVQ is good while the PS is acceptable. BPB outperforms OS in this case. In Xiang’s method, the average bandwidth lies in this interval, but the experiment shows the MDP assumption doesn’t fit for every environment settings well. When the underlying network can’t be simply modeled as MDP, the weakness of OS is exposed.

6) Experiment 3 (download rate 60KB/s)

Here BPB algorithm performs better both in terms of AVQ and PS. Furthermore, in BPB, the Interruption ratio is really well (no interruption at all). When the bandwidth is nearly enough for providing best quality, BPB can have better knowledge of bandwidth, thus leading to better strategy.

7) Experiment 4 (manually change restricted bandwidth on the fly)

We restricted the bandwidth on time 0,50,100,150 with 50KB/s, 40KB/s, 30KB/s, 20KB/s. We only show partial results.

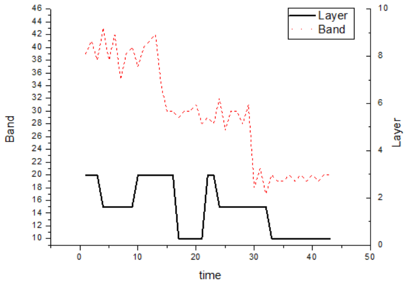

Fig. 4. Experiment 4 results.

It can be inferred that BPB fits the bandwidth well. When bandwidth changes, BPB tends to choose lower quality layer due to potential under flow. Then after buffer gets filled, BPB prefers to higher quality, which shows the adaptability. AVQ and PS are 2.79 and 3.9 respectively, no interruption occurs.

VI. Conclusion

In conclusion, we can see that BPB has many advantages over current algorithm in real network with restricted bandwidth. Furthermore, OS algorithm performs badly due to unrealistic assumption of network bandwidth variations. BPB overcomes it by adopting more robust AI-based methods. It’s an online algorithm with great flexibility, making full use of bandwidth. The combination of bandwidth predictor and streaming strategy is an innovative idea, which can be deployed quickly in SVC-based multimedia provider and player.

After conducting these experiments, there are also some considerations and new innovative ideas we are thinking about. First, get the parameters automatically using some methods involving Artificial Intelligence. Next, give a more rigid model for estimating different layer’s decoding time, which will make the probability function more accurate.

Acknowledgment

Without Xiang’s support for providing the test bed for our project, it’ll be much harder to get fast prototype of SVC player for testing the algorithm. We want to shows our thanks to him very much.

References

[1] H. Schwarz, D. Marpe, and T. Wiegand, “Overview of the scalable video codingextension of the H. 264/AVC standard,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 17, no. 9, pp. 1103–1120, 2007.

[2] S. Y. Xiang, L. Cai, and J. P. Pan. “Adaptive scalable video streaming in wireless networks,” in *Proc. the 3rd Multimedia Systems Conference MMSys’12*, 2012, pp. 167-172.

[3] P. A. Dinda, “Design, implementation, and performance of an extensible toolkit for resource prediction in distributed systems,” *IEEE Transactions on Parallel and Distributed Systems*, vol. 17, no. 2, pp. 160-73, Feb. 2006.

[4] J. Yao, S. S. Kanhere, and M. Hassan, “An empirical study of bandwidth predictability in mobile computing,” in *Proc. the Annual International Conference on Mobile Computing and Networking, MOBICOM*, 2008, pp. 11-18, and *Proc. MobiCom'08 the 3rd ACM International Workshop on Wireless Network Testbeds, Experimental Evaluation and Characterization, WiNTECH'08*.

[5] A. Esswaradass, X. H. Sun, and W. Ming, “A neural network based predictive mechanism for available bandwidth,” in *Proc. 19th IEEE International Parallel and Distributed Processing Symposium*, 2005, pp. 10.

[6] Z. X. Liu, X. P. Guan, and H. H. Wu, “Bandwidth prediction and congestion control for ABR traffic based on neural networks,” *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 3973 LNCS, pp. 202-207, 2006.

[7] V. N. Vapnik, “The nature of statistical learning theory,” 2nd ed, Springer-Verlag, New York, 1999.

[8] M. Mariyam, S. Joel, B. Paul, and X. J. Zhu, “A machine learning approach to TCP throughput prediction,” in *Proc. 2007 ACM SIGMETRICS International Conference on Measurement and Modeling of Computer Systems,* San Diego, California, USA, June 12. 16, SIGMETRICS ’07, pp. 97–108.

[9] Q. He, C. Dovrolis, and M. Ammar, “On the predictability of large transfer TCP throughput,” *roc. SIGCOMM’05*, Philadelphia, Pennsylvania, USA, vol. 35, no. 4, pp. 145–156,August 21-26, 2005.

[10] L. Hu, X. Che, and X. Cheng, “Bandwidth Prediction based on Nu-Support vector regression and parallel hybrid particle swarm optimization,” *International Journal of Computational Intelligence Systems*, vol. 3, no. 1, pp. 70-83, 2010.

[11] C. Liu, I. Bouazizi, M. Gabbouj *et al.*, “Rate adaptation for adaptive HTTP streaming,” in *Proc. the 2011 ACM Multimedia Systems Conference*, MMSys’11, pp. 169-174, 2011.

[12] J. Jiang, V. Sekar, and H. Zhang, “Improving fairness, efficiency, and stability in HTTP-Based adaptive video treaming with FESTIVE,” *Association for Computing Machinery*, pp. 97-108, 2012.

[13] K. RobertKofler *et al.,* “An evaluation of TCP-Based rate-control algorithms for adaptive internet streaming of H.264/SVC,” in *Proc. the 2010 ACM SIGMM Conference on Multimedia Systems*, 2010, pp. 157-167.

Xu Yanling was born in China in 1977. She received the B.E., M.E. degrees from Shandong University, Jinan, China in 2000 and 2003. She received Ph.D. degree from Shanghai Jiaotong University, Shanghai, China in 2007. Since 2007, she has been with the college of software engineering, Tongji University, where she is currently an Associate Professor. Her main areas of research interests are video encoding, error concealment and wireless Network. Dr. Xu has published many papers on IEEE Transactions on Consumer Electronics about error concealment and received many projects sponsored by government and universities. She has also been given outstanding youth award and other awards by Tongji University.

Wu Baolin was born in China in 1989. He received B.E. degree from Tongji Univerisy, Shanghai, China in 2011. Since 2011, he was a graduate student in College of Software Engineering, Tongji University, Shanghai, China. His main areas of research are video encoding and embedded systems. Mr. Wu has published a paper about Linux scheduler on PACRIM 2011. Besides, he has been awarded plenty of scholarship by Tongji University.

Yang Liushan was born in China in 1992. She was enrolled as an undergraduate student in College of Software Engineering, Tongji University, Shanghai, China. Her main areas of research are Artificial Intelligence, video transmitting over wireless networks and software engineering. Ms. Yang has published many papers about software engineering. She got many awards from computer skills contest and plenty of scholarship by Tongji University and other organizations.