Layered Congestion Control for Scalable Video Coding based on the Efficient Bandwidth Inference

Sheng-Shuen Wang , Hsu-Feng Hsiao, and Suh-Yin Lee
Department of Computer Science
National Chiao Tung University
Hsinchu, Taiwan
{sswang, hillhsiao, sylee }@cs.nctu.edu.tw

***Abstract*—Scalable video coding allows greater granularity adaptation at bit level to the time-vary heterogeneous networks. In this paper, we propose a congestion control algorithm based on the bandwidth estimation techniques. In order to cooperate with H.264/MPEG-4 AVC SVC extension which can achieve fine granularity of scalability, we design an adaptation strategy to smartly switch to an efficient bandwidth inference mode during the channel probing period according to the one way delay and jitter profile so that the subscription decision of SVC layers can be made quickly to better utilize the network resource. In case of the unavoidable network congestion, we unsubscribe scalable video layers according to the recently received throughput instead of only dropping one layer at a time to rapidly accommodate the streaming service to the channels. The simulations show that the proposed algorithm converges fast to the available bandwidth and can adapt to the fluctuated network efficiently.**

Keywords—bandwidth estimation, congestion control, end-to-end, packet train.

I. INTRODUCTION

Many applications of multimedia communication over IP network, such as VoIP, multimedia on demand, IPTV, and video blog, have been integrated with our daily life rapidly. Congestion control plays an important role in the growing demand of network resource by those multimedia services. The knowledge of available bandwidth of the bottleneck link is crucial in terms of making effective use of network resource to improve Quality of Service (QoS) in many distributed applications, such as the overlay construction of peer to peer system, optimization of dynamic server selection, and also congestion control for the streaming applications.

In order to dynamically adapt to the end-to-end channel status and the device capability without transcoding, Scalable Video Coding (SVC), as an amendment to the H.264/MPEG-4 AVC standard created by Joint Video Team (JVT), intends to encode a video sequence once and the encoded bit stream is able to allow a diversity of different receivers to acquire and decode a subset of the encoded bit stream. Scalable video coding enables not only efficient distribution of real-time multimedia streaming over heterogeneous networks but also a most promising solution for one-to-many congestion control over multicast networks.

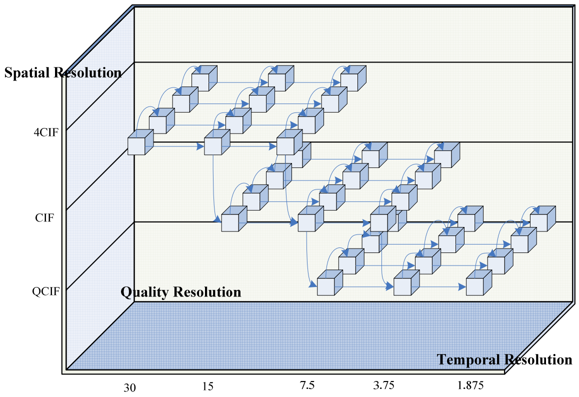

Fig 1: Spatial temporal quality cube for combined scalability.

A scalable video bit stream contains a non-scalable base layer which is in compliance with H.264/MPEG-4 AVC and one or more enhancement layers which may result from spatial, temporal or quality scalability of the scalable tools. For transformed residual signal in each temporal level, progressive refinement slices have been introduced so that Network Abstraction Layer (NAL) units can be truncated at any arbitrary point to support Fine Granular quality Scalability (FGS) or flexible bit-rate adaptation.

The third octet in an NAL unit header records three layer identifications, including temporal_level (3bits), dependency_id (3bits), and quality_level (2bits). Dependency_id is used to indicate the dependency hierarchy of inter-layer coding between different spatial resolutions. Temporal_level and quality_level specify the temporal scalability and FGS respectively. Thus, there are at most 2^3 x 2^3x2^2 layers. The combination can be complicated such as the 3D cube shown in Fig. 1. Due to the large number of possible scalable video layers, how to quickly converge to the time-varying available bandwidth becomes a critical factor for real-time video streaming.

One-way delay trend detection is utilized in Pathload [7] to measure the end-to-end available bandwidth by sending periodic packet trains. Since each packet train is used to determine only one decision that if the probed rate is greater or smaller than the available bandwidth, usually binary search is adopted to adapt the probing rate to the available bandwidth. In contrast to acquiring initial available bandwidth, layered congestion control algorithm proposed in BIC [5], similar to

0-7695-3084-2/07 $25.00 © 2007 IEEE
DOI 10.1109/ISM.Workshops.2007.76

Pathload, generates periodic burst packet trains from the upper layer over multicast network so that the probing periods of each receiver can be synchronized. Each receiver uses one-way delay trend detection to make decision of joining one additional layer at a time and leaves a scalable video layer when packet loss rate exceeds a specified threshold. As a result, it is not suitable for receivers that might require joining or leaving several scalable video layers in a short time, due to dramatically fluctuant channels. In [2], a hierarchical sub-layer probing which adopts coarse to fine layer partitioning to improve the efficiency of the probing interval was proposed. It might help to reduce the number of probing periods when compared to BIC, but on the other hand, the probing packets might overshoot easily and congest the networks.

In this paper, we focus on congestion control of end-to-end video streaming by adopting top-down and bottom-up schemes adaptively to infer available bandwidth and the corresponding scalable video layers. The remainder of this paper is organized as follows. In Section II, the bandwidth estimation and layered congestion control algorithm are presented. In Section III, we evaluate the performance of our proposed algorithm and the conclusion of this paper is given in Section IV.

II. Bandwidth Estimation and Layered Congestion Control

Several studies have been devoted to the research of available bandwidth estimation in recent years. Probe-based methods by means of packet train [1] analysis are widely adopted to infer network utilization. Packet train is a sequence of probing packets of equal packet size and the probing packets are arranged either back-to-back or with some specified inter-packet dispersion.

A. Bandwidth Estimation

There are two major types of packet-train based algorithms: one-way-delay (OWD) based analysis model and dispersion based analysis model [3].

Given that a sender *S* transmits *K* packets of packet size *L* through H hops with respective capacity C_i to its receiver *R*, the OWD D^k of the *k*-th packet can be modeled as the summation the transmission delay (L/C_i), processing delay (σ_i), and queuing delay (d_i^k) of each and every link *(i= 1 ...H)* along the path.

$$D^k = \sum_{i=1}^{H}(\frac{L}{C_i} + \sigma_i + d_i^k). \quad (1)$$

The OWD difference between adjacent packets can be expressed as the contribution from queuing delay as shown in (2).

$$\Delta D^k = D^{k+1} - D^k = \sum_{i=1}^{H}(\Delta d_i^k). \quad (2)$$

If the probing rate is faster than the available bandwidth, the network queues will build up and the probing packets will be delayed ($\Delta D^k > 0$). Full search algorithm [5] is one of the schemes to detect the delay trend by (3).

$$S = \frac{\sum_{k=2}^{N}\sum_{l=1}^{k-1} I(D^k > D^l)}{\frac{M(M-1)}{2}} \quad (3)$$

$$\begin{cases} \mathrm{I}(\mathrm{D}^k > \mathrm{D}^l) = 1 \text{ if } \mathrm{D}^k > \mathrm{D}^l, \\ \mathrm{I}(\mathrm{D}^k > \mathrm{D}^l) = 0, \text{otherwise.} \end{cases}$$

On the other hand, dispersion based model exploits the information of the inter-arrival time between two successive probing packets at the receiver. Given that the network capacity of the tight link *C*, the available bandwidth *A* ($=C-\lambda$) can be estimated by solving the following equation for the traffic load λ [1] if probing packets are in JQR[9], otherwise $\delta_{in} = \delta_{out}$.

$$\delta_{out} = \frac{L}{C} + \frac{\lambda}{C}\delta_{in} . \quad (4)$$

In other words, for the probing packets passing through hop *i* with the arrival rate R_{i-1} to hop *i* and departure rate $R_i = L/\delta_i$ from the same hop, R_{i-1} and R_i will have the following relationship,

$$R_i = R_{i-1}\frac{C_i}{\lambda_i + \max\{R_{i-1}, A_i\}} . \quad (5)$$

where λ_i is the traffic load of hop *i*. Obviously, the departure rate will be less than or equal to the arrival rate ($R_{i-1} \geq R_i$). However available bandwidth A is the minimum of all A_i; thus we can induce that

$$R_{in} \geq R_{out} \geq A. \quad (6)$$

Based on (3) and (6), the top-down bandwidth estimation algorithm is proposed in [3].

B. Layered Congestion Control

Our proposed congestion control algorithm consists of two phases: initial phase and transmission phase.

1) Initial Phase: During the start phase, we have no idea about the information of bandwidth, so we may use the capacity of bottleneck, or the bit rate of the largest layer as the probing rate. In [3] shows that top-down approach is more efficient than the binary search to determine the initial available bandwidth, especially in the low utilization networks. Packet train is sent to test whether the probing rate is greater than available bandwidth and the received rate is used as the next probing rate iteratively until no delay trend is detected by (3).

2)Transmission Phase: This phase can be divided into normal period and probing period. We use packet loss rate as a metric to detect congestion occurrence during normal periods. If packet loss rate is greater than the threshold in normal period, we drop layers immediately. According to (6), we can use the corresponding layer of received rate at client as the sending rate in stead of dropping only one layer as in BIC. Thus the sending rate can be quickly adjusted to avoid more packet losses. On the other hand, if the packet loss rate in normal period is less than the threshold, we observe the

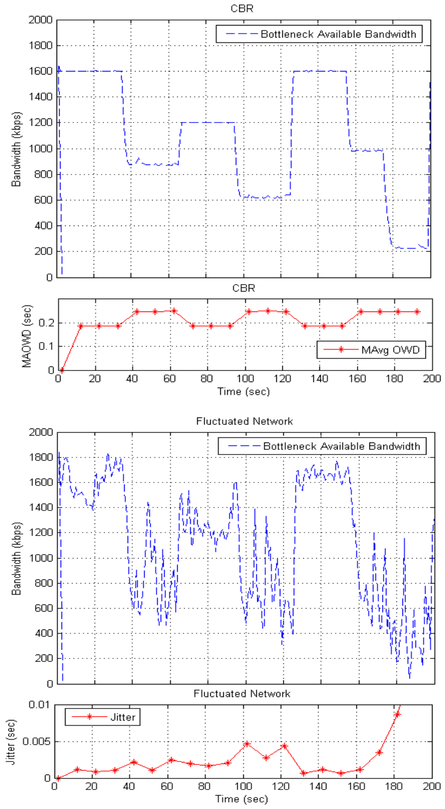

Figure 2: MAOWD and jitter during CBR and fluctuated network status respectively.

Moving Average of OWD (MAOWD) in (7) and the interarrival jitter *J* [8] in (8) for each packet *i* in normal period to decide whether to use top-down approach or bottom-up approach in the probing periods to estimate the number of appropriate scalable video layers. In (7), MA(OWD_i) is the moving average of OWD of packet *i*, and α is the smoothness parameter. The interarrival jitter J_i of packet *i* is calculated by the OWD of packet *i* and previous packet *i-1* as shown in (8).

$$MA(OWD_i) = (1-\alpha)MA(OWD_{i-1}) + \alpha OWD_i \,. \qquad (7)$$

$$J_i = J_{i-1} + (|(OWD_i - OWD_{i-1})| - J_{i-1})/16 \,. \qquad (8)$$

Bottom-up approach is to probe layers in a one by one manner from current layer. It is more efficient when available bandwidth doesn't have significant increment because it may probe only one layer to detect the delay trend. As to the dramatic bandwidth variation, top-down approach is faster to converge than the bottom-up approach. However, it is unfavorable due to sending the packet train at the speed equal to the speed of the highest scalable video layer when no more residual bandwidth is available. So our goal is to find out when the available bandwidth has large increment to take top-down approach. The simulation results in Fig. 2 are performed in the network topology as shown in Fig.3 under constant bit rate (CBR) and fluctuated background traffic to show the corresponding moving average of OWD and jitter, respectively. In fluctuated network, jitter drops when residual bandwidth increases or network congestion causes packet lost. As to packet loss, we have corresponding strategy mentioned above to deal with. However in the CBR network situation, the jitter trend is not distinct, so we use Moving Average OWD. From the above observation, our congestion control algorithm takes top down scheme in probing period when MAOWD or jitter drop than a threshold without packet loss during normal period. Otherwise, that means the network status is stable, so we just have to take bottom-up strategy to probe one more layer to determine whether there is enough residual bandwidth to increase one more layer. In other words, we drop layers by packet lost rate and increase layer by predicting network condition from the mean and variance of OWD.

III. SIMULATION AND DISCUSSION

We use ns2 network simulator to conduct simulations with the topology shown in Fig. 3 and the length of a packet train is 30 packets with packet size 550 bytes. The capacities along the path are 10, 7.5, 5.5, 2, 6, and 8 Mbps, respectively. We assume that the link with capacity 2 Mbps is the tight and narrow link and the queue length is 30 packets. The threshold of full search algorithm is 0.63. As to various scalability of SVC, we design 24 layers with bit rates 16, 32, 48, 64, 80, 96, 112, 128, 160, 192, 224, 256, 320, 384, 438, 512, 640, 768, 896, 1024, 1280, 1536, 1792, and 2048 Kbps, respectively.

Because BIC is designed for multicast and only one layer is added or dropped at each probing period, we improve the BIC to fit the end-to-end congestion control better. During the probe period, layers are probed sequentially until delay trend is detected. When packet loss rate exceeds a threshold in normal period, scalable video layers are dropped continuously. Since top-down approach has been shown to have better performance than the binary search in [3], the modified BIC also takes top-down strategy in initial path.

A. Background Traffic of Constant Bit Rate Flows

CBR cross traffic with packet size 550 byte is generated for each link so that the available bandwidth can exhibit large changes over a period of time. The packet loss rate threshold is 0.05.

From Fig. 4 and Fig. 5, the modified BIC has similar estimation accuracy with our proposed algorithm. The mean absolute difference (MAD) of estimation is 66.4 and 54.4 kbps, respectively. For the convergence time to the available bandwidth, our proposed algorithm is better than the modified BIC, especially when there are severe bandwidth fluctuations. The mean probing times of the modified BIC and our proposed method are 0.97 and 0.62 sec, respectively. The standard deviations of each are 1.0 and 0.2 sec, respectively. That is because we take adaptive strategy by bottom-up

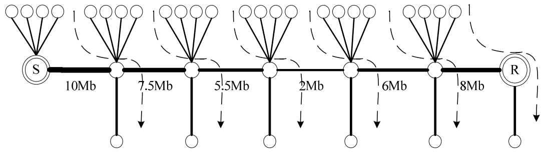

Figure 3: Network topology used in the NS2 simulations

approach when the network is stable and by top-down approach when the network suffers from severe bandwidth fluctuations. Our proposed algorithm can drop scalable video layers fast when congestion occurs, and increase the transmission of scalable video layers when spare bandwidth can be utilized.

B. *Background Traffic with Pareto Distribution*

To mimic the real Internet, there are four random sources at each link with Pareto distribution and the cross traffic is simulated with different packet sizes as follows: 40% of the background traffic are 40 bytes, 50% are 550 bytes, and 10% are 1500 bytes. Comparing Fig. 7 with Fig.8, the modified BIC needs to take longer time to converge to the available bandwidth, especially at time 130 sec. The mean probing times of modified BIC and our proposed method are 1.13 and 0.9 sec. The standard deviations of each are 1.46 and 0.66 sec, respectively.

In summary, the modified BIC and our proposed algorithm have similar performance in terms of the accuracy of estimate available bandwidth. However, the proposed algorithm converges much faster and shows smaller convergence time variation.

IV. CONCLUSION

Congestion control based on the bandwidth inference for H.264/MPEG-4 AVC SVC is proposed. Top-down and bottom-up approaches are used to estimate the available bandwidth according to the distribution of MAOWD and jitter. The dropped layers are decided by the decreasing of receiving rate whenever packet loss rate is greater than a threshold. The simulation results show that our proposed algorithm can adapt to the available bandwidth faster, and is applicable over the fluctuated networks.

REFERENCES

[1] C. Dovrolis, P. Ramanathan, and D. Moore, “Packet-dispersion techniques and a capacity- estimation methodology,” IEEE/ACM Transaction on Networking, Vol. 12, NO. 6, pp963-977, Dec. 2004.

[2] J. L. Lin, S. C. Pei and J. N. Hwang, “Fine-grain layered multicast based on hierarchical bandwidth inference congestion control,” International Symposium on Circuits and Systems (ISCAS), May 2005.

[3] S. S. Wang and H. F. Hsiao, “Fast end-to-end available bandwidth estimation for real-time multimedia networking,” International Workshop on Multimedia Signal Process (MMSP), Oct. 2006.

[4] T. Wiegand, G. Sullivan, and J. Reichel, etc. …, “Scalable video coding standard joint draft 6,” Doc. JVT-S201, April 2006.

[5] Q. Liu and J. N. Hwang, “A new congestion control algorithm for layered multicast in heterogeneous multimedia dissemination,” International Conference on Multimedia and Expo (ICME), Jul. 2003.

[6] J.-R. Ohm, M.van der Schaar, and J. W. Woods, “Interframe wavelet coding-motion picture representation for universal scalability,” Signal Processing: Image Communication, Vol. 19, Issue9, pp877-908 Oct. 2004

[7] M. Jain and C. Dovrolis, “Pathload: a measurement tool for end-to-end available bandwidth,” in Proc. Passive Active measurements, Fort Collins, CO, Mar. 2002

[8] H. Schulzrinne, S. Casner, R. Frederick, V. Jacobson “A Transport Protocol for Real-Time Applications,” RFC 1889, Jan. 1996

[9] N. Hu, and P. Steenkiste, “Evaluation and characterization of available bandwidth probing techniques,” IEEE Journal on Selected Areas in Communications, Vol. 21, No.6, Aug 2003.

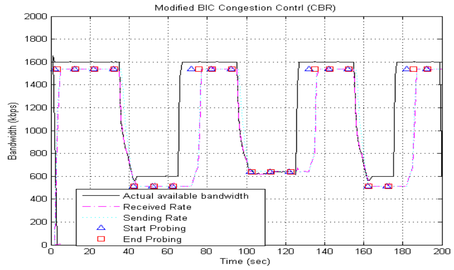

Figure 4: The modified BIC congestion control over CBR cross traffic.

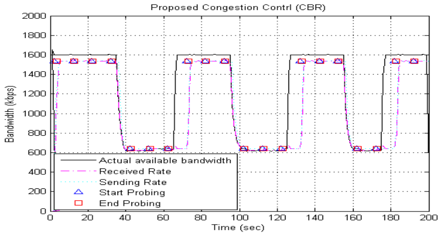

Figure 5: The proposed congestion control over CBR cross traffic.

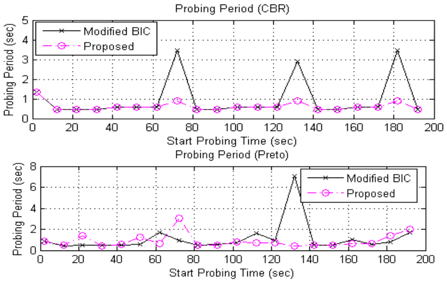

Figure 6: Time period for each probing over CBR and Pareto traffic.

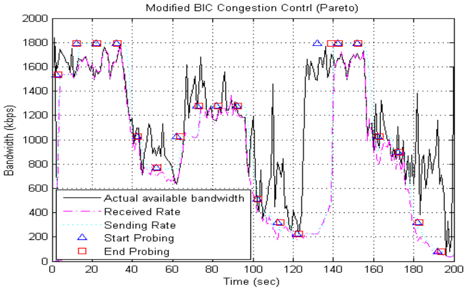

Figure 7: The modified BIC congestion control over fluctuated bandwidth.

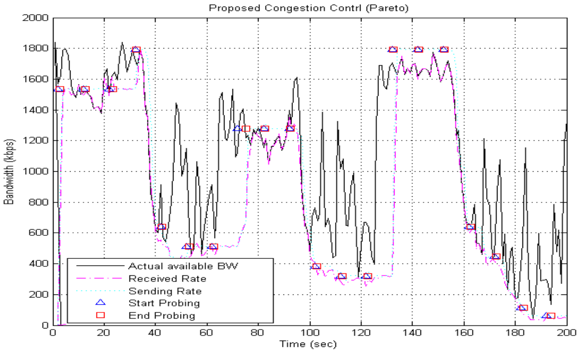

Figure 8: The proposed congestion control over fluctuated bandwidth.