MULTI-BUFFER BASED CONGESTION CONTROL FOR MULTICAST STREAMING OF SCALABLE VIDEO

Chenghao Liu[1], Imed Bouazizi[2] and Moncef Gabbouj[1]

[1]Department of Signal Processing, Tampere University of Technology, Finland
[2]Nokia Research Center, Tampere, Finland
Email: chenghao.liu@tut.fi, imed.bouazizi@nokia.com and moncef.gabbouj@tut.fi

ABSTRACT

Receiver driven layered multicast streaming provides an attractive solution for transmitting the same video data to multiple receivers while accounting for the heterogeneity in the network resources and device capabilities. Traditionally, congestion control methods in receiver-driven layered multicast have used packet loss as a measure to detect congestion. However, given that wireless networks are characterized by higher packet loss and varying throughputs, packet-loss based congestion control results in inappropriate behavior of those algorithms, and ultimately in sub-optimal usage of the available network resources.

This paper presents a novel multicast congestion control algorithm named Layered Virtual Client Buffer (LVCB)-based receiver-driven multicast for multicasting of scalable video. The proposed LVCB technique tracks the media time for each layer currently present in the receiver buffer. The proposed multicast congestion control method reacts to variations in the media time for each LVCB by dynamically joining/leaving multicast groups, in order to adapt the subscription level to the varying network resources. Furthermore, the presented algorithm solves the problem of mutual affection between receivers without exchange of information about subscription levels. The simulation results show the suitability of the proposed method in wireless as well as wired scenarios.

Keywords— Streaming, Multicast, Layered Media, Congestion Control, SVC

1. INTRODUCTION

Multicast streaming is an attractive solution to transmit video data to multiple receivers simultaneously. However, when the network resources are characterized by heterogeneity, a media stream encoded at a fixed bit rate cannot serve the needs of every receiver. The concept of receiver driven layer multicast (RLM) proposed by McCanne *et al.* [1] provides a solution to this problem.

In receiver driven layered multicast, the server uses scalable video coding techniques to produce a set of layered bit streams and transmit each layer on a different multicast group. In RLM, each receiver periodically joins multicast groups in so-called join experiments until congestion occurs. When multiple receivers observe packet loss, they determine that a problematic receiver is conducting a join experiment which causes the congestion. The RLM protocol expects that receiver to solve the congestion problem by canceling the last join experiment, which has caused the congestion. However, current Internet mainly offers a best effort service and is characterized by time-varying bandwidth. Moreover, the method of observing the packet loss to determine congestion occurrence usually degrades the video quality in the receivers.

In order to solve the problems of packet loss-based congestion detection, Legout *et al.* introduced the packet pair receiver-driven cumulative Layered Multicast (PLM) [2], which carries out the join experiments based on the estimated bandwidth using packet pairs sent from the client to the server. But the PLM estimates the bandwidth under the assumption of a fair queuing network. Later, Liu *et al.* proposed a layered bandwidth inference congestion (BIC) control [3], which sends a probing packet at a fixed rate and detects the delay trend to infer the spare network capacity for first-in, first-out (FIFO) queuing networks. In [4], Lin *et al.* proposed a hierarchical BIC method to take advantage of the coarse/fine scalability provided in the scalable extension of H.264/AVC video coding standard [5]. In the papers [3] and [4], the authors argue that detecting the spare network capacity is critical in the multicast control. For detecting congestion, BIC and hierarchical BIC observe the packet loss. However, the effectiveness of the congestion control protocol is decided not only by the efficiency of detecting the spare network resources but also by detecting network congestion. This is especially true in the mobile streaming scenario, where the mobile receiver needs to adapt the subscription layer to the varying network resources. Otherwise, the loss of packets significantly degrades the reconstructed video quality at the receiver since packet loss in hybrid motion compensation-based video coding technology results in error propagation [6]. Moreover, the

This work was supported by the Academy of Finland, (application number 129657, Finnish Programme for Centres of Excellence in Research 2006-2011).

congestion control methods presented in [3] and [4] require the sender to periodically send the probing packets to estimate the bandwidth. Without such probing packets from the server, the bandwidth cannot be estimated to control the congestion.

In contrast to the previous RLM works, this paper uses the media time stored in the client buffer to detect congestion. The proposed method uses Layered Virtual Client Buffer (LVCB) to maintain the media time for each layer of the bit stream received from the corresponding multicast group and compares the media time variation in the LVCB against an adaptive per-layer threshold to predict congestion well before packet loss occurs. Our congestion control protocol provides a solution to adapt the subscription level (called operation level) to the varying network bandwidth. In this paper, a method to cope with the cross-receiver interference, which does not require the shared learning (sharing the operation level with each other) process, is presented.

The proposed method has the following advantages compared with previous works. First, our method quickly and accurately adapts the subscription level to varying network resources such as present in internet. Second, it is robust against link losses; hence, it can be applied in the wireless networks scenarios. Third, it does not require additional support from the sender as is the case in BIC.

The rest of the paper is organized as follows. Section 2 briefly describes the transport of SVC. The proposed method is presented in Section 3. Simulation results and conclusions are presented in Sections 4 and 5, respectively.

2. TRANSPORT OF SVC

Scalable video coding has been widely investigated in during the past two decades. However, before the standardization of the scalable video coding (SVC) extension of H.264/AVC [5], scalable video coding was rarely used in commercial applications due to the increased complexity and the significant drop in coding efficiency when compared to non-scalable video coding. Alternatively, simulcast, which transmits two or more independent single layer streams simultaneously, was preferred. However, simulcast causes a significant increase in the resulting total bit rate. The SVC extension of H.264/AVC introduces single loop decoding and drift control to address earlier shortcomings of scalable video coding.

SVC enables to produce a single scalable bit stream which contains a base layer and one or several enhancement layers. SVC supports three different types of scalability: spatial scalability, temporal scalability, and quality scalability. To enable the transmission of SVC, Wang *et al.* specified a transport format for SVC [7].

Each layer of the sub-stream can be extracted by extracting the corresponding Network Abstraction Layer (NAL) units, which are identified by the combination of the spatial layer, quality layer and temporal layer identifiers. Therefore, SVC is developing into the media codec of choice in video multicast and broadcast services.

3. PROPOSED METHOD

We propose a novel multicast congestion control method called receiver-driven Layered Virtual Client Buffer (LVCB)-based congestion control.

3.1. LVCB- based Receiver-driven Multicast Congestion Control

A block diagram of a receiver driven LVCB-based multicast congestion control system is shown in Fig. 1. At the multicast server side, the server uses the SVC encoder such as H.264/SVC to encode a scalable bit-stream. As shown in Fig. 1, the *i*th layer of the sub-stream is extracted and transmitted to a multicast group provided with the multicast group ID denoted as Gi(S, D$_i$). Herein, the S and D denote the source and destination respectively. The frame rate of transmission is equal to the frame rate the sub-stream. And the transmission start time of each layer can be set as one of the following two ways. First, the transmission start time of each layer is the same. Second, the base layer is transmitted first. An enhancement layer i is transmitted after a predefined amount of time, denoted as τ, has elapsed since transmitting layer i-1. Therefore, the media time filled in the LVCB at the receiver decreases along with each additional layer of the VCB, as show in Fig. 1.

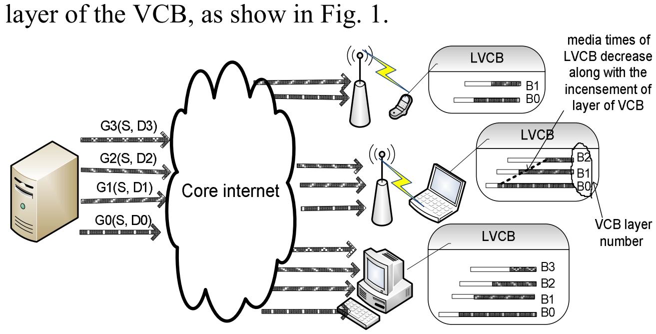

Fig. 1. LVCB receiver driven multicast congestion control

At the multicast receiver side, each receiver firstly request the multicast session information such as available number of multicast groups and multicast group ID. Based on the multicast session information, the receiver allocates multiple Layered Virtual Client Buffer (LVCB) each of which separately track media time of a sub-stream, received from a multicast group as shown in Fig.1. The media time used in our paper denotes the playing out duration of media data stored in the client buffer, which can be calculated as the difference between the timestamp of latest packet and the timestamp of the earliest packet of a sub-stream. We use the verb virtual as, in reality; media data itself from multicast groups are stored in the same physical client buffer.

Fig. 2 shows the flowchart of the LVCB based receiver-driven multicast congestion control consisting of two phases, i.e., the initialization phase and the operation level adaptation loop phase. In the initialization phase, the receiver joins G_0 which carries the base layer, performs initial buffering for D_{ib} seconds before starting playback, schedules the event to join G_1 with joining interval as described in section 3.3 and sets subscription level as 1. In the operation level adaptation loop, the receiver waits for an event. There are three event types named as detect congestion, join group (join G), and end session.

1. Congestion detected event:

The multicast congestion control system detects congestion and leaves a multicast group as described in section 3.2. The congestion detected event is scheduled at a predefined constant interval ψ.

2. Join Group G event:

The subscription level (L) is increased by 1 and group G_{L-1} is joined to explore whether the spare end-to-end bandwidth is available. The event of joining the next enhancement layer in group G_L is scheduled after joining a group G_{L-1}(S, D_{L-1}). Joining G_{L-1} event is also scheduled whenever leaving a group G_{L-1}(S, D_{L-1}). The joining interval is set dynamically as described in section 3.3.

3. End of session event:

The congestion control protocol is terminated.

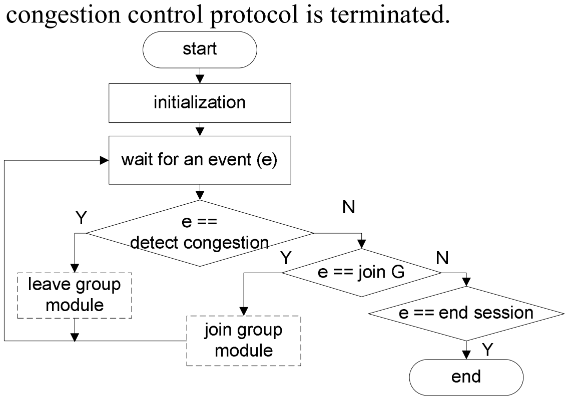

Fig. 2. Flowchart of the LVCB receiver driven multicast congestion control

3.2. Leaving Multicast Group

Fig. 3 shows the leave group module. At an instance t media time denoted as m_t is calculated as formula (1)

$$m_t = \min\{T_i^{max}(t) f_i^p\} - \max\{T_i^{min}(t) f_i^p\}, \quad 0 \le i \le L-1 \qquad (1)$$

where $T_i^{max}(t)$ and $T_i^{min}(t)$ denotes the maximum and minimum timestamp of media units of layer (i) at the current time t, f_i^p denotes the playing phase flag and L denotes the subscription level. When a sub-stream layer is received but hasn't been played back yet, the layer is considered on initial buffering phase, which is specified by setting f_i^p to 0. As soon as playback of a layer starts, a layer is defined to be in the playing phase, specified by setting f_i^p as 1. Upon joining a new multicast group of layer i, the receiver typically buffers a certain amount of time before the playback starts, which is denoted as initial buffering duration D_i^{ib} . In formula (1), the difference between minimum of latest timestamp in all playing phase layers and maximum earliest timestamp in all playing phase layers represents the media time of decodable frames in the client buffer.

To detect congestion, the media time of current time (m_{tc}) is compared with the reference media time at the previous measurement time (m_{tp}) to obtain the variation in media time (v_m) as equation (2).

$$v_m = (m_{tc} - m_{tp}) \qquad (2)$$

The previous measurement of the media time denoted as m_{tp} is updated to reflect the media time of decodable frames in the LVCB whenever a receiver joins or leaves a multicast group.

The variation of the media time is used as the metric to detect congestion. If the v_m is larger than the leaving group threshold ε, congestion is detected as shown in Fig.3 and the receiver leaves the group G_{L-1}.

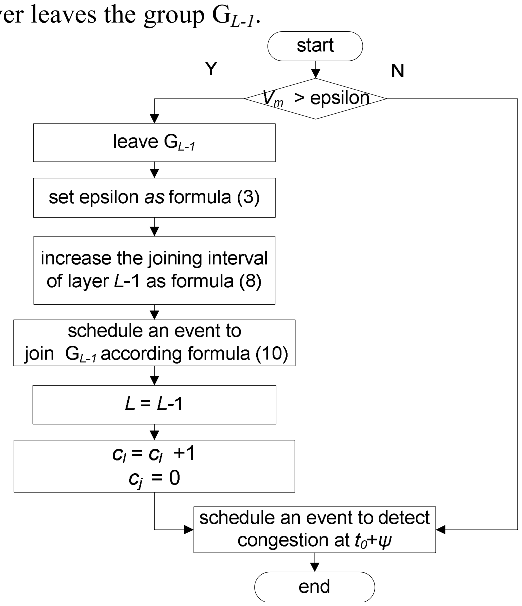

Fig. 3. Detect congestion and leave a group module

It is important to note that the threshold ε is updated every time a leave or join operation is performed.

Case 1: after leaving a group (dropping a layer of sub-stream)

$$\varepsilon = \varepsilon_c (k - c_l)^{1/2}, \quad k > c_l \qquad (3)$$

where ε denotes the threshold for leaving group, ε_c denotes the threshold to leave current layer, i.e., highest subscription layer, k denotes a tolerance factor and c_l denotes the count of consecutive group leaving operations. The c_l is set as 0 when the receiver joins a group and is increased by 1 each time the receiver leaves a group. Multiplying ε_c by k avoids leaving groups because of variation in the bitrate of the encoded video stream; hence, increasing the stability of the subscription level. k is dependent on the bitrate variation of the encoded bit stream and should be set to a large value compared to possible c_l values.

Case 2: after joining a group (adding a layer of sub-stream)

$$\varepsilon = \varepsilon_c(1 + c_j)^{1/2} \tag{4}$$

where ε denotes the media time variation threshold for leaving group, ε_c threshold to leave current layer, c_j denotes the count of consecutive join group operations. c_j is reset to 0 once the receiver leaves a group and is increased by 1 each time the receiver joins a group. The minimum ε is equal to ε_c in the case of re-joining a group after it was left. The low (sensitive) threshold enables the receiver to quickly leave the problematic multicast group to prevent any possible congestion. Several successful joining group operations increase the confidence on the availability of spare network resources; hence, ε is increased so that the receiver successfully reaches the optimum operation level.

In the following, we will describe the equation to calculate the threshold to leave current layer ε_c which is decided with geometric series formula (5).

$$\varepsilon_c = \varepsilon_b \cdot q^{L-1}, \quad 0 < q < 1 \tag{5}$$

where ε_b denotes media time variation threshold of the base layer, q is a proportion and L denotes current subscription level. The desired q can be derived using pre-defined first term and last term threshold values.

The first term ε_b is calculated according to the following equation

$$\varepsilon_b = D_{ib} - D_{mb} \tag{6}$$

where D_{ib} denotes the initial buffering duration and D_{mb} denotes the predefined minimum buffering duration to prevent buffer underflow.

The reasons for the usage of equation (5) are twofold. First, the geometric proportions $(0 < q < 1)$ assigns larger ε_c to the base layer compared to the enhancement layer. It ensures a conservative lower layer dropping decision. After dropping an enhancement layer, the receiver will be more cautious with dropping the next lower layer, thus providing room for absorbing the media time decrease due to still in-transit media data from the dropped higher layer. Second, geometric proportions ($0 < q < 1$) characterized by ε_c induce an earlier reaction to congestion by receivers of high subscription levels than those of lower subscription levels. Consequently, this diminishes the problem of cross-receiver interference to a large extent. Herein the cross-receiver interference denotes the congestion affection between different receivers sharing the same bottleneck bandwidth limitations.

The parameters controlling joining a group are discussed in the section 3.3. Finally, the leave group module schedules an event to detect congestion at $t_0 + \psi$ wherein t_0 denotes the current time and ψ denotes a congestion detection interval.

3.3. Joining a Multicast Group

Each multicast group G$_i$ has a joining interval noted as I_i^j to control the joining operations. The I_i^j is initialized according to equation (7)

$$I_i^j = I_u(1 + (b_i/b_u)^{0.5}) \tag{7}$$

where I_u is the predefined join interval unit, b_i is the sub-stream bitrate of layer i and b_u is the bitrate unit.

I_i^j reveals the opportunity to join a specific layer i and its value is increased after leaving a group G$_i$ according to the following equation.

$$I_i^j = \widehat{I_i^j} + \lambda(b_i/b_u)^{0.5} \tag{8}$$

where I_i^j and $\widehat{I_i^j}$ denote the new and original interval for joining G$_i$, λ is a predefined join interval increasing step.

In addition to each layer's joining interval, a common joining interval called I_j is used to specify the interval between two consecutive group join operations. I_j is dynamically updated according to c_j as follows.

$$I_j = I_b - c_j \cdot \sigma \tag{9}$$

where I_b denotes a predefined base interval, and σ denotes the decreasing offset. The time to evaluate whether joining group causes congestion is biggest in the first time to join a group after having left a group and decreases with the several consecutive and successful join operations.

Finally, joining a group G$_i$ is scheduled at time t_i^j using following formula.

$$t_i^j = \begin{cases} t_0 + I_i^j, & I_i^j > I_j \\ t_0 + I_j, & I_i^j < I_j \end{cases} \tag{10}$$

where t_0 denotes the current time. The time to schedule the next joining group event is described in section 3.1. Upon reaching the scheduled join time, the joining group module is started as shown in Fig. 4. This module increases the subscription level by 1, joins G$_{L-1}$, schedules an event to join G$_{L-1}$, and updates ε according to formula (4).

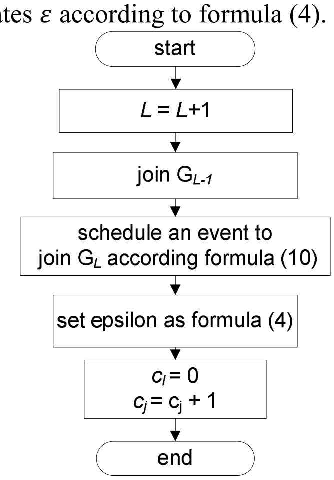

Fig. 4. Join group module

4. SIMULATION RESULTS

In this section, we present the simulation results of the proposed LVCB method which were performed on network simulator NS2 [8]. We compare the results with the traditional receiver-driven layered multicast (RLM) [1]. Fig. 5 shows the network topology used in the simulations. The node 0 is the traffic source, the nodes 4, 5, 6 and 7 are the receivers, where BW denotes the bandwidth and CBR denotes constant bit rate traffic flow. To evaluate the performance of the proposed method in wireless scenarios, the links between node 2 and 4, node 2 and 5, node 3 and 6, node 3 and 7 are defined as lossy links with packet loss rates (PLR) of 0%, 1%, 3%, 5% and 10% respectively. In addition, the CBR traffic flows, from node 1 to node 2 and node 3 with bitrates of 300 Kbps and 500Kbps respectively, starts at time 80s and ends at 130s to analyze the effectiveness of the proposed method in the dynamic cases. The queue size is set as 300 packets. In our simulation, the bandwidths are set larger than the minimum bitrates of 4CIF and CIF to the node 4, 5 and node 6, 7 respectively.

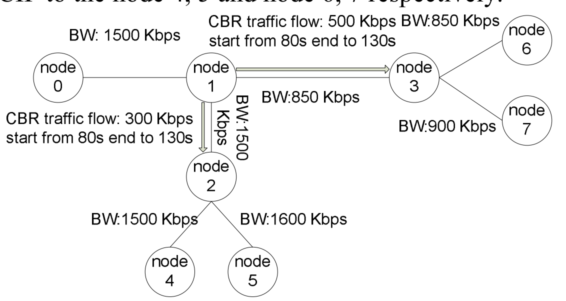

Fig. 5. Network topology

The H.264/SVC reference software JSVM version 9.6 [9] is used to create the scalable video bit stream. Video sequence *Crew* is encoded with combined scalability, i.e., combination of spatial, temporal and quality scalability. Table 1 shows the layer number (L) and the corresponding spatial (S), temporal (T), quality (Q) scalability and bitrate (B) Kbps. The constant quantization parameter (QP) is used to encode the video sequence since no rate control algorithm for enhancement layers is implemented in JSVM version 9.6, therefore the instant bitrate of an encoded stream is varying.

Table 2 shows one of the sample parameters used in our simulation. The proposed method uses each the variation in media time to detect congestion, which smoothes out short-term delay jitter and packet loss effects. Furthermore, the threshold to detect congestion, i.e., ε is dynamically adapted to the changes of the c_l and c_j. Therefore, we believe that the proposed method is robust and the performance of the method does not rely much on the parameter setting.

Fig. 6 shows the average reception throughput for the receivers using the proposed LVCB and RLM congestion control methods. On average, the proposed method outperforms RLM by over 200 Kbps for the link layer loss rates of 1%, 3%, 5% and 10%. In node 4 and node 5, the received throughput with the proposed method ranges from around 950 (layer 10) to 1100 Kbps (layer 12). The received throughput with the RLM, however, is less than 800 Kbps (layer 7) for the link layer loss rates from 1% to 10%. This is attributable to the fact that RLM uses packet loss to detect the congestion and detects false congestion events after observing packet losses that are due to the wireless link. In contrast to RLM, the proposed method uses the media time variation to detect congestion, which explains the good performance in the wireless scenario as shown in Fig. 6.

Fig. 7 shows the overall packet loss results of the proposed LVCB and the traditional RLM. Simulation results of LVCB show that the PLR of node 4 and node 5 are slightly higher than for node 6 and node 7. In the case of 10% link layer loss rate, the PLR caused by congestion amounts to around 4% at the nodes 4, 5 and 1.8% at nodes 6, 7. The reason for the higher reliability in the case of LVCB is due to early congestion detection based on media time variation and the subsequent preventive layer dropping. LVCB can accurately monitor the media time variation as the media time is monitored at each layer separately.

Table 1. Each layer of bitrate with the spatial, temporal and quality scalability

L	S	T	Q	B	L	S	T	Q	B
0	CIF	15	0	227	7	4CIF	30	0	782
1	CIF	30	0	288	8	4CIF	15	1	834
2	CIF	15	1	391	9	4CIF	15	2	890
3	CIF	15	2	538	10	4CIF	30	1	960
4	CIF	30	1	475	11	4CIF	30	2	1026
5	CIF	30	2	634	12	4CIF	15	3	1031
6	4CIF	15	0	672	13	4CIF	30	3	1179

Table 2. Sample parameter

τ	D_i^{ib}	k	ε_b	q	I_u	λ	b_u	I_b	σ
0.2	7	4	0.22	0.86	3	6	100	6	1

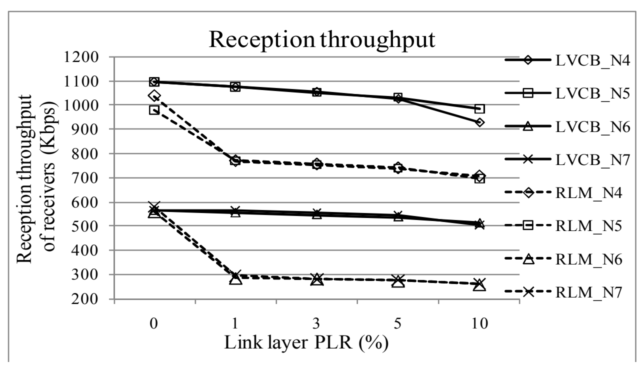

Fig. 6. Average reception throughput of the receivers with proposed LVCB method and RLM method

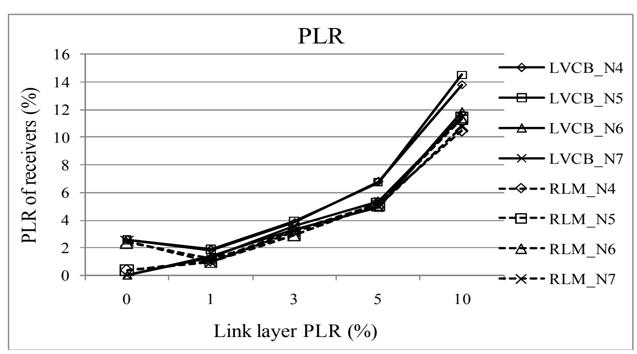

Fig. 7. Average PLR of the receivers with proposed LVCB method and RLM method

Fig. 8, 9 and 10 show the instantaneous subscription level, reception throughput and packet loss results (respectively) for node 5 and at link layer loss rate of 3%. Fig. 8 to Fig. 10 show the following advantages of the proposed method compared to RLM. First, the received average reception throughput matches the corresponding bottleneck bandwidth for the receiver in node 5. The variation in reception throughput at the receiver is due to the variation of encoded bitstream, as we applied constant QP encoding as explained earlier. Second, the proposed method quickly adapts to the bandwidth changes. When the bottleneck bandwidth decreases due to the introduction of CBR background traffic, the proposed method adapts the subscription level to the new bottleneck bandwidth within 7 seconds. When the available bandwidth increases again, the proposed method reaches the optimum subscription level after around 20 seconds. The packet losses during 80 to 130 in Fig. 10 are caused by join experiments. The interval to join a multicast group increases after a previous failure to join as shown in Fig. 8.

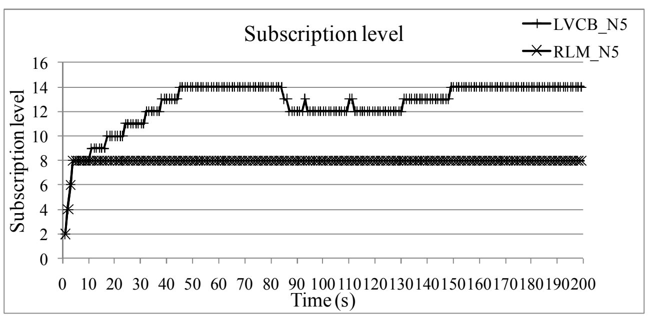

Fig. 8. Subscription level of node 5 with 3% of link layer PLR

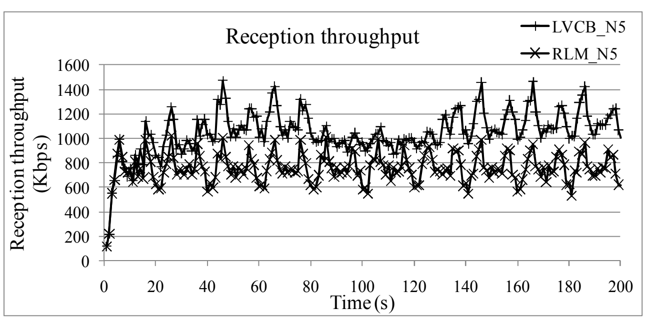

Fig. 9. Reception throughput of node 5 with 3% of link layer PLR

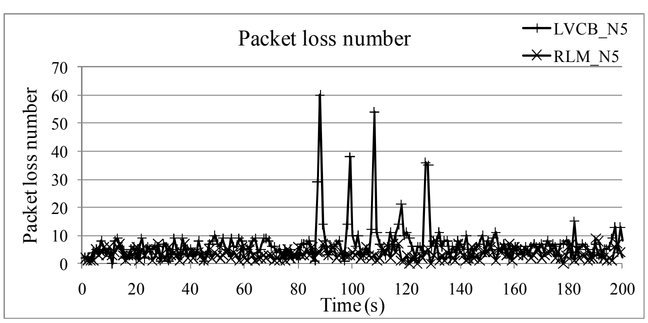

Fig. 10. Packet loss number of node 5 with 3% of link layer PLR

5. CONCLUSION

In this paper, we presented a Layered Virtual Client Buffer (LVCB)-based receiver-driven multicast congestion control method. LVCB stores information about the media time for each media layer received from a multicast group. The proposed congestion control algorithm compares the variation in media time for each layer against an adaptive threshold to detect congestion before packet loss occurs at the receiver side. Furthermore, an algorithm to dynamically set the threshold values is proposed so that the congestion is detected quickly while maintaining a stable subscription level by avoiding leaving multicast groups due to transient media time variations. Simulation results show that the advanced congestion control method provides good performance with dynamic network resources and link-introduced packet losses. Hence, the proposed multicast congestion control method can be applied to both wired and wireless scenarios.

6. REFERENCES

[1] S. McCanne, V. Jacobson, and M. Vetterli, "Receiver-driven layered multicast," Proc. SIGCOMM'96, ACM, Stanford, CA, Aug. 1996, pp. 117–130.
[2] A. Legout and E.W. Biersack, “PLM: fast convergence for cumu-lative layered multicast transmission schems,” Proc. of ACM SIGMETRICS, June 2000, 13–22.
[3] Q. Liu and J. N. Hwang, “A New Congestion Control Algorithm for Layered Multicast in Heterogeneous Multimedia Dissemination,” ICME 2003, Baltimore, MD, USA, July 2003.
[4] J. Lin, W. Hwang, S. Pei, and J. Hwang, “A Hybrid Coarse/fine Layered Multicast Scheme Based on Hierarchical Bandwidth Inference Congestion Control,” IEEE Trans. on Circuits and Systems for Video Technology, vol 18, pp. 1776-1780, Dec 2008.
[5] ITU-T Rec. H.264 and ISO/IEC 14496-10 (MPEG-4 AVC), ITU-T and ISO/IEC JTC 1, Advanced Video Coding for Generic Audiovisual Services, Version 8 (including SVC extension): November 2007.
[6] Y. Wang and Q.-F. Zhu, "Error control and concealment for video communication: A review," Proc. IEEE, vol.86, no.5, pp.974–997, May 1998.
[7] Y.-K. Wang, M. M. Hannuksela, S. Pateux, and A. Eleftheriadis, “System and transport interface of SVC,” IEEE Trans. Circuits Syst. Video Technol., vol. 17, no. 9, pp. 1149–1163, Sep. 2007.
[8] The Network Simulator Ns-2. Available [online]. http://www.isi.edu/nsnam/ns/
[9] Joint Video Team of ITU-T VCEG and ISO/IEC MPEG ,“Joint Scalable Video Model JSVM-9” Jun. 2007.