

M-10 An Adaptive Scheduling Algorithm for On-Demand Video Delivery System

*, **Chih-Chang HSU **Youichi ISHIBASHI ***Takuyo KOGURE

****Fumio HASEGAWA *Terumasa AOKI *Hiroshi YASUDA

1. INTRODUCTION

Although there is the widespread deployment of broadband technology, large-scale on-demand video delivery systems still have not become commercially available. The heavy traffic and long-lived nature of digital video streaming is the main challenge to the designs of the on-demand video delivery system which provides users with the probability of watching the digital videos at any time from any locale via the Internet.

Currently, there are three issues being addressed in the majority of Video-on-Demand (VoD) research. These concern video server bandwidth requirement, network delivery cost and startup delay for video playback. Existing schemes, such as video patching, broadcasting protocols [1] and merging algorithms [2], have mainly focused either on the designs of the *golden factor* (used for dividing the video into segments and allocating the channel bandwidth) or the constructions of the *merge tree*. They have shown the performance of the VoD system, in terms of less server bandwidth requirement and guaranteed startup delay, can be greatly improved. However, most of them assumed the existence of the multicast-enabled routers set in all the transmission paths. They, obviously, are not easily deployable in the current state of IP multicast networks. To address this issue, we propose a new mechanism called AVD, an Active Networks approach, that includes (i) the video delivery tree construction [3], and (ii) the video merging algorithm which will be discussed in this paper.

2. ACTIVE VIDEO DELIVERY (AVD)

The main goal of the AVD system is to achieve Multicast-like delivery service by utilizing traditional Unicast forwarding mechanism combined with the video merging techniques. The Multicast-like delivery mechanism does not require all the routers along the delivery paths to implement the AVD specific protocol but some routers located in the strategic points of the networks. The delivery tree of the AVD is constructed and maintained mainly by a center-based server controller. The Active Routers set in the networks only need to perform simple tasks, such as packet replicating/redistributing, after analyzing the lightweight codes contained in the active packets generated from the video server controller. The AVD tree is dynamically changed according to the utilization of the video server bandwidth, the behaviors of users (i.e., Join/Leave), and the conditions of network connectivity. More technical information related to the AVD tree is described in further detail in ref. [3].

3. VIDEO MERGING

As depicted in Fig. 1, in this section, we discuss the basic algorithm of the video merging of the AVD, that is, how we combine the pure Unicast (Patching) and the Multicast-like (AVD sharing) stream for achieving multicast-like AVD service.

* The University of Tokyo ** TAO Yamagata Video Archive Research Center
*** Matsushita Electric Industrial Co., Ltd.
**** Tohoku University of Art and Design

In addition, we present an offline (video stream initiations are known ahead of time) for both the video server bandwidth requirement (i.e., the maximum number of concurrent out-going connections) and the total network cost of the AVD mechanism.

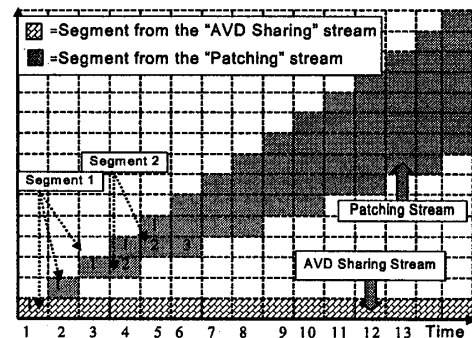


Fig. 1. The Basic Algorithm of the AVD Video Merging

Throughout the paper, we assumed that (i) the video length could be suitably partitioned into L units; (ii) the user access pattern is "fully loaded arrivals" (i.e., every time slot contains at least one request arrival); (iii) the time required for transmitting one segment is one unit of time (i.e., the total time T for delivering the whole video segments is equal to L); (iv) the network cost for delivering a segment via one channel is 1; (v) A new stream, either in the form of AVD_Sharing or Patching, always begins at segment 1 of the video.

For instance, the 1st user, who arrives in time=1, receives only the AVD sharing stream initiated from the video server directly (* these segments will be playback immediately). The 2nd user, who arrives in time=2, would require to receive both (i) "segment 2~the last segment" of the video, which are duplicated from the AVD sharing stream destined to user 1 originally and then generated by the Active Routers. (* these segments will be stored in local disk for latter playback temporally); and (ii) "segment 1" of the video from the patching stream generated independently by the video server (*this segment will be playback immediately). Similarly, the 3rd user, who arrives in time=3, would also require to receive both (1) the AVD sharing stream for "segment 3 to the end segment", and (2) the patching stream for "segments 1 and 2".

Given the above rules for client reception of streams, the total number of active concurrent logical channels generated by the video server at time t and the total network cost during the period T can be expressed as:

$$Active_Srv_Channel(t) = AVD_Sharing(t) + \sum_i Patch_i(t) \quad (1)$$

$$where \quad AVD_Sharing(t) = 1 \quad t = [0, T] \quad (1)$$

$$Patch_i(t) = \begin{cases} 1 & t = (i, 2^i + 1) \\ 0 & \text{else} \end{cases} \quad (2)$$

From Equation (2), we can obtain the following time series

$$\left\{ \sum_i Patch_i(t) \mid_{t=0 \sim T} \right\} = \{0, 1, 1, 2, 2, 3, 3, \dots, A\} \quad (3)$$

$$where \quad A = \begin{cases} \frac{T-1}{2} & \text{if } T_{\text{odd}} \\ \frac{T}{2} & \text{if } T_{\text{even}} \end{cases}$$

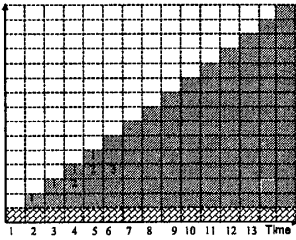


Fig. 2-A: Pure Unicast

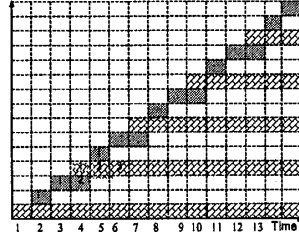


Fig. 2-B: Mul_AVD_Patch(Pth=2)

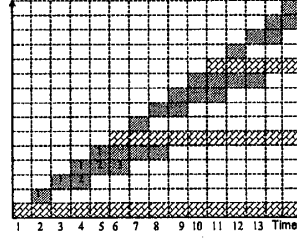


Fig. 2-C: Mul_AVD_Patch(Pth=3)

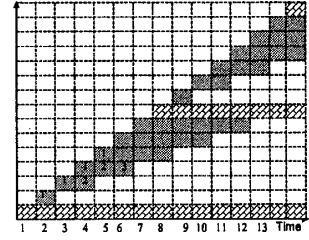


Fig. 2-D: Mul_AVD_Patch(Pth=4)

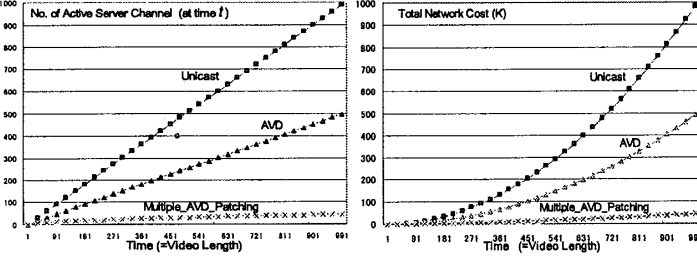


Fig. 3: No. of the active server channel

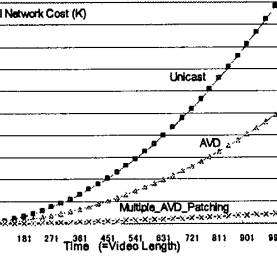


Fig. 4: Total Network Cost

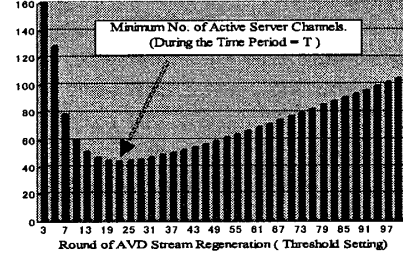


Fig. 5: Threshold Setting v.s. No. of Active Channels

For simplicity, we choose $T \in \text{odd}$ for the following analysis. Therefore, we can obtain the number of the active server channels at any time t and the total network cost for delivering all video segments:

$$Active_Srv_Channel(t) = \frac{t-1}{2} + 1 = \frac{t+1}{2} \quad 0 < t \leq T$$

where "1" = the AVD sharing stream channel.

$$Total_Network_Cost = \sum_{i=0}^T AVD_Sharing(t) + \sum_{i=1}^T Length_Patch_i(t)$$

$$= T + [1 + 2 + 3 + \dots + (T-1)] = \frac{T(T+1)}{2}$$

The simplest mechanism that can be used in the AVD is shown in Fig. 1. In this case, the video server generates only one main Unicast stream to perform multicast-like sharing, and several independent Unicast streams for performing patching. However, similar to the pure Unicast approach (Fig 2-A), this mechanism clearly would not perform well because of the high demand on the video server bandwidth and the heavy cost on the networks.

To solve this problem, we can consider other mechanisms shown in Figs 2-B~D, called the *Multiple_AVD_Patching* mechanism. That is, if the number of the fan-out *Patching* streams issued by the video server reach the threshold ($P_{threshold}$) which is set in advance for the first time (t_{1st}),

$$i.e., \sum_i Patch_i(t) \geq P_{threshold}$$

the video server re-generates a new full-length main stream to perform a new AVD sharing source for those users who arrive later. Note that, we let this form of the AVD sharing stream regeneration repeat after a fixed slot time continuously. As a result, the traffic produced by the *Patching* streams can be greatly reduced.

From Equation (3), we know the time, t_{1st} , for performing the first AVD stream regeneration is

$$\frac{t_{1st}-1}{2} = P_{threshold} \rightarrow t_{1st} = 2 * P_{threshold} + 1$$

and the total round of AVD stream regeneration during the time period T is $Round_{avd} = L / t_{1st} = L / (2 * P_{threshold} + 1)$ (4)

therefore, the $Active_Srv_Channel(t)$ can be obtained:

$$= No. of the AVD Sharing Streams + \sum_i Patch_i(t)$$

$$= Round_{avd} + P_{threshold} \quad (5)$$

and the total network cost for this case can be expressed as:

$$\sum_{j=1}^{Round} Length_AVD_j + \sum_{j=1}^{Round} [\sum_i Length_Patch_i(t)]$$

$$= L(Round_{avd} + P_{threshold})$$

For simplicity, we use R and P to denote $Round_{avd}$ and $P_{threshold}$ respectively. Since $L = R * (2P + 1)$, and from Equation (5), we have $Active_Srv_Channel(t) = (2R^2 - R + L) / 2R \rightarrow F(R)$. If $F'(R) = 0$, then we can obtain the minimum total number of the active server channels and the total network delivery cost, when

$$R = \sqrt{\frac{L}{2}} \quad (6)$$

4. ANALYTIC RESULT AND CONCLUSION

In Figs 3 and 4, we compared the performance metrics, in terms of the number of the active channels (i.e., the maximum server bandwidth requirement) and the total network cost, for three different delivery mechanisms (* $L=1000$). Obviously, the *Multiple_AVD_Patching* mechanism (*set $P_{threshold}$ at 22) obtains considerably less network delivery cost and at an overall lower video server bandwidth requirement than that of the other two.

In Figure 5, we showed the relation between the settings of the threshold and the total number of the active server channels. We observed that the min. total number of the concurrent out-going channels, generated by the video server during the time period T , could be obtained if the threshold is set to 19 ~ 25 (* $L=1000$). Equation (6) provides a simple way to find this optimal threshold setting. That is, from Equations (6) and (4), we can easily obtain the following:

$$R = \sqrt{\frac{1000}{2}} = 22.5 \Rightarrow P_{threshold} \cong 22$$

There are several issues for future works, such as a simulation for evaluating the above analytic result and an analysis on different user arrival patterns and multiple videos delivery.

REFERENCE
 [1] A. Hu, et al, "Video-on-Demand Broadcasting Protocols: A Comprehensive Study", Infocom2001
 [2] A. Bar-Noy, et al, "Efficient Algorithms for Optimal Stream Merging for Media-on-Demand", 2000.05
 [3] Chih-Chang Hsu, et al, "An Active Networks-based Video Delivery Tree", 2002 IEICE Society Conference 2002.09