Harmonic Data Placement: File System Support for Scalable Streaming

Youjip Won, *Member*, IEEE, Seungheon Yang, and Sooyong Kang

Abstract — *Scalable encoding scheme enables the player or streaming server to adaptively change the playback rate of multimedia content. However, in scalable streaming of layer encoded content, sequential playback of content does not necessarily coincide with the sequential scan of a file. This property introduces another dimension of complexity in the scheduling of data block retrieval. In this work, we develop a novel file organization strategy, Harmonic Interleaving, which can effectively handle the dynamically changing playback rate of multimedia data retrieval. The proposed scheme not only eliminates the retrieval of unnecessary blocks but also minimizes disk head movement. Via experiment, Harmonic Interleaving exhibits superior disk utilization on a moderately loaded network.*

Index Terms — Scalable Streaming, Multimedia File System, Multimedia Data Transmission

I. Introduction

Due to the sequential access nature of real-time multimedia playback, file system puts great emphasis on placing data in *seek* optimized fashion. Hence, file systems support for real-time multimedia playback has been the subject of intense research recently. Some of these studies strive to find the relationship between real-time requirements of individual playbacks and file system level data retrieval scheduling[1, 4, 8]. Others exploit the variability in linear bit density of the hard disk track in placing the multimedia blocks[10, 12, 6]. More recently, a number of studies propose to support not only real-time multimedia playback but also legacy text based file in single file system partition and propose various scheduling algorithms[13, 11, 14, 7].

However, those works do not consider the disk access pattern under scalable streaming service. When the file is organized as a sequence of logical units, e.g. frames, playback of a single layer multimedia file yields a simple sequential scan. However, when content is created with a layered encoding scheme, playback no longer yields a sequential scan. This is because only a subset of individual frame information may be selected for transmission. The subset selected for transmission dynamically changes as a function of network bandwidth availability. Therefore, a layered encoding scheme introduces another dimension of complexity from the file system's point of view. A legacy file system model and a disk scheduling strategy do not incorporate the characteristic of layered encoding and leaves much to be desired.

In this work, we examine the effect of a layered encoding scheme on file system efficiency and propose a new file organization technique for layer encoded multimedia content. Multimedia content consists of a collection of logical data units. Each data unit consists of layers. Application dynamically selects the proper subset of layers for transmission for each logical data unit. There can be two extreme ways of organizing a file: (i) As in legacy file organization, the file is physically organized as a sequence of logical units (frames) and each logical unit is organized as a collection of a layers, or (ii) a file is physically organized as a sequence of layers and each layer consists of the data blocks of the respective layer for each logical unit. Neither of these organizations yields satisfactory efficiency under dynamically changing network conditions. Fig. 1 illustrates the underlying server organization.

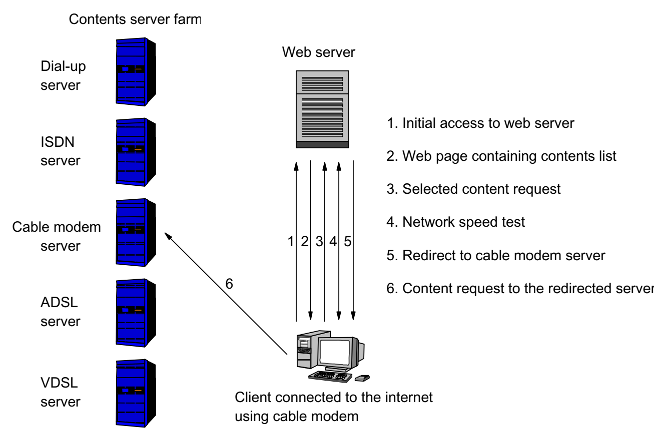


Fig. 1 Underlying Service Model

User can access the content via wide variety of different network medium whose bandwidth capability varies widely, e.g. cable modem, xDSL, T1, 56 Kbps modem and etc. To effectively cope with the variety in connection medium, content providing system consists of a number of servers and each server is dedicated to harbor the contents for a given bandwidth connection. Once the connection speed of the incoming request is determined, the incoming request is directed to the appropriate server and is serviced from the respective server. In this work, we propose a novel file organization technique, *Harmonic Interleaving* which effectively exploits the access characteristics of layer encoded

Youjip Won is with the Department of Electronics and Computer Science, Hanyang University, Seoul, Korea (e-mail: yjwon@ece.hanyang.ac.kr).
Seungheon Yang is with the LG Electronics Inc. (e-mail: shyang@lge.com).
Sooyong Kang (Corresponding author) is with the Department of Computer Science Education, Hanyang University, Seoul, Korea (e-mail: sykang@hanyang.ac.kr).

Contributed Paper
Manuscript received May 15, 2006

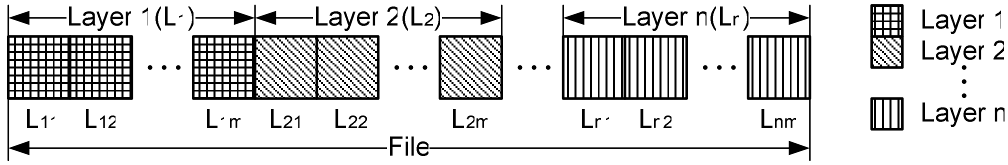


Fig. 2 Progressive placement scheme

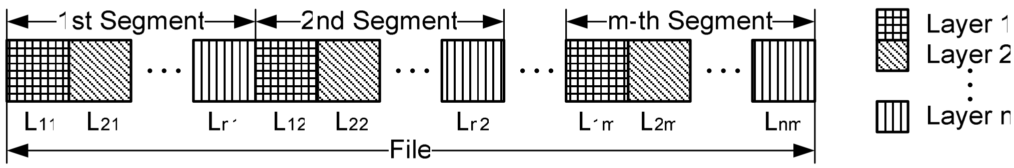


Fig. 3 Interleaved placement scheme

multimedia content. *Harmonic Interleaving* combines advantages of the above mentioned two file organization types.

The rest of the paper is organized as follows. In section 2, we present three different file organization techniques. In section 3, we develop the performance model for each file organization scheme. In section 4, we examine the efficient of individual file organizations via simulation as well as physical experiments. In section 5, we conclude the paper.

II. Layer Allocation Strategy

In this section, we present three file organization strategies for scalable streaming. We view a media file as a collection of logical storage units, called *segments*. A segment can be a frame or group of pictures. Table 1 illustrates the notations used in this paper.

TABLE I
Notations

Parameters	Description
n	Number of layers
m	Number of segments in a media file
B_i	Data rate of *i*-th layer in an object
B_{movie}	Total data rate of an object
α_i	Transmit ratio of layer *i*
L_{ij}	Data unit for *j*-th segment of *i*-th layer
L_i	Set of data units belonging to *i*-th layer
T	Playback length of a movie
T_{seg}	Playback time of a segment (eg. 1 sec.)
l	Number of lower layers
C_{movie}	Number of cylinders that movie occupy
T_{total}	Time of reading data from disk
$T_{overhead}$	Disk overhead
T_{data}	Time of reading data purely of T_{total}

A. Progressive Placement

Progressive placement strategy clusters the data blocks in a layer together. This allocation strategy manifests itself when network bandwidth availability is very limited and when the streaming server can transport the lowest layers most of the time. The *Progressive Placement* scheme entails significant disk head movement overhead when the server transports larger numbers of layers. Fig. 2 illustrates the *Progressive Placement* strategy.

B. Interleaved Placement

In *Interleaved Placement* scheme, segments are placed in temporal order. This is plain sequential placement. When the streaming server retrieves data blocks in all layers, disk access yields sequential access pattern not only from a logical aspect but also from a physical aspect. When the server transports only the proper subset of layers, either file access entails undesirable seek operation or streaming server needs to discard the retrieved data blocks. Interleaved placement scheme manifests itself when the server transports most of the layers. Fig. 3 illustrates the *Interleaved Placement* scheme for scalable coded data.

C. Harmonic Interleaving

The *Progressive* placement and *Interleaved* placement strategies are two extremes in a wide spectrum of file organization techniques. In this section, we propose a novel file organization strategy, *Harmonic Interleaving*. Fig. 4 illustrates the file organization under the *Harmonic Interleaving* placement strategy.

In Harmonic Interleaving, data blocks are partitioned into two groups based upon the layers which they belong to. The layers are categorized into two groups: a set of lower layers, L_{lower} and a set of upper layers, L_{upper}. For example, with five layers, the layers can be partitioned as follows: $L_{lower} = \{L_1, L_2, L_3\}$ and $L_{upper} = \{L_4, L_5\}$. In this case, we assume that up to layers 3 are frequently requested and layer 4 and layer 5 are hardly requested. *Harmonic Placement* adopts

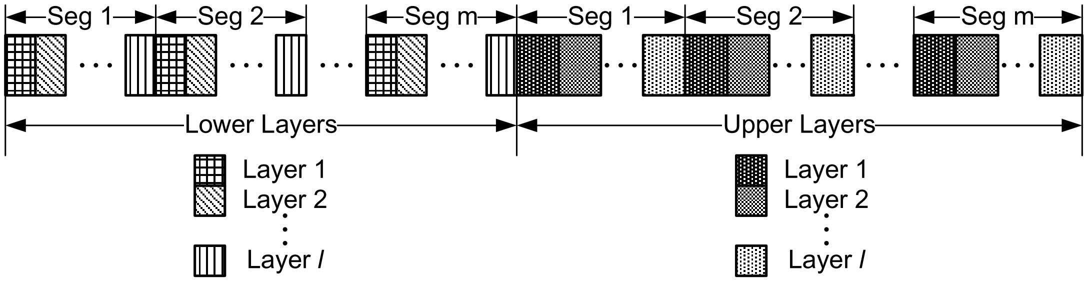


Fig. 4 Harmonic Interleaving placement scheme

Progressive Placement for *inter-group* placement and Interleaving for *intra-group* placement. Using this scheme, we can reduce disk seek time by increasing the physical continuity of data blocks belonging to frequently serviced layers. Hence, when only lower layers of information is streamed in most of the time and information in the upper layers is rarely used, this scheme outperforms other schemes. However, if the upper layers are accessed frequently, the required disk seek overhead can result in lower performance. Therefore, the boundary of the layer groups needs to be carefully determined by considering both network bandwidth and the client device.

The effectiveness of Harmonic Interleaving is subject to the layer partitioning policy and the variability in network bandwidth availability. We will explore this issue more formally in the following section.

III. MODELING OF DISK RETRIEVAL OPERATION

We use disk utilization, ρ, to quantify the effectiveness of the placement strategy. Disk utilization, ρ, is a ratio between total elapsed time to read data and the time spent on reading the actual information from the disk excluding the disk overhead(Eq. 1). Total elapsed time, T_{total}, consists of the time to read the data, T_{data}, and overhead, $T_{overhead}$. $T_{overhead}$ consists of seek, rotational latency, head switch, command processing, etc. T_{data} depends upon the amount of data to read. $T_{overhead}$ is governed by the data placement strategy.

$$\rho = \frac{T_{data}}{T_{total}} = \frac{T_{data}}{T_{data} + T_{overhead}} \qquad (1)$$

Let $\alpha_i (0 \le \alpha_i \le 1)$ be the fraction of playback duration during which $L_1, ..., L_i$ are presented. Fig. 5 illustrates the layers presented during playback. During T_1, only the first layer is presented. In T_2, the first and the second layers are presented. Let D_i and B be the amount of data blocks in L_i (Byte) and maximum disk transfer rate(Bytes/sec), respectively. Then, the amount of time spent on reading data blocks, excluding disk overhead, during the entire service time can be calculated as in Eq. 2.

$$T_{data} = \sum_{i=1}^{n} \left(\sum_{j=1}^{n} \alpha_j \right) \frac{D_i}{B} \qquad (2)$$

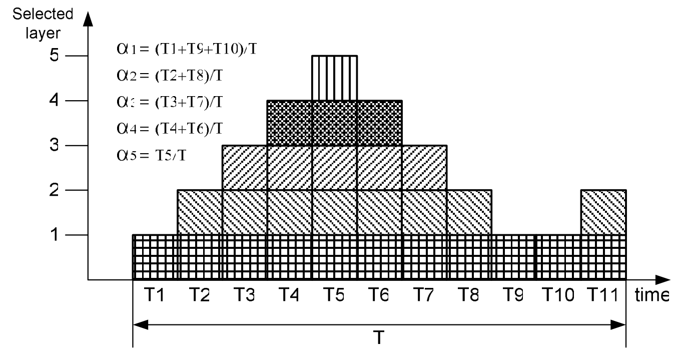


Fig. 5 Layers presented during the playback.

Modern disk drives consist of one or more platters and heads. Each platter consists of concentric circles. Circles on a platter are partitioned into a number of *zones.* A zone is a group of consecutive cylinders with the same number of sectors. Developing a file organization technique mandates an accurate disk model which enables us to precisely predict the performance of a given placement strategy.

Let N_P and T_R be the number of tracks per cylinder and time to read single track, respectively. A number of disk heads are attached to the disk arm assembly. Usually, only one of the disk heads is active at any time. The process of reading cylindrical data consists of the time to read individual tracks and the time to switch the active disk head, T_s. Eq.3 formulates the time to read a single cylinder(T_{cyl}). When all tracks are read, the first disk head becomes re-activated, and it is repositioned. We define this cylinder switch time as T_c. T_c is subject to the position of the next data blocks to be read. When a file is placed in consecutive fashion, T_c consists of the head switch and a *very* short seek. Very short seek is equivalent to *resettle* in modern disk drive due to the high *tracks/inch*(TPI) factor.

$$T_{cyl} = (N_p - 1) \times T_s + T_c + N_p \times T_r \qquad (3)$$

A. Modeling Progressive Placement Scheme

Progressive placement clusters the data blocks in the same layer together. Accessing data blocks across multiple layers can incur excessive disk *seek*. Let C_i be the number of cylinders occupied by L_i. Then, $C_i = \left\lceil \frac{B_i}{B_{movie}} \times C_{movie} \right\rceil$.

Let $T_{seek}(i)$ be the seek time for i cylindrical distance. Let us assume that the disk retrieves data for the lower i layers. The overhead incurred in retrieving a single data unit is the sum of seek times between adjacent layers and the seek time returning to the lowest layer from layer i, i.e. $\sum_{j=1}^{i-1} T_{seek}(C_j) + T_{seek}(\sum_{j=1}^{i-1} C_j)$. The total overhead, $T_{overhead}$ in playback can be formulated as in Eq. 4.

$$T_{overhead} = m\sum_{i=2}^{n} \alpha_i \left(\sum_{j=1}^{i-1} T_{seek}(C_j) + T_{seek}(\sum_{j=1}^{i-1} C_j) \right) \quad (4)$$

As we can see from the equation, *Progressive Placement* scheme results in large disk overhead when α_1 is small and $\alpha_i, (2 \le i \le n)$ is large. On the other hand, if α_1 is very large, e.g. due to poor network bandwidth, then this file organization scheme yields efficient disk utilization.

B. Modeling Interleaved Placement Scheme

In *Interleaving Placement* scheme, segments are placed with respect to their temporal order. For a segment, the data blocks are placed with respect to their layers. If the application needs to retrieve all layers for a segment, it will simply yield a sequential scan. A problem may occur when the application retrieves data blocks in a proper subset of all layers. When we do not need data blocks in all layers, there can be two retrieval strategies: (i) we can either read all layers and discard unnecessary data blocks, or (ii) we can read only the data blocks in the selected layers.

In second strategy, which is called *selected retrieval*, the file system retrieves data blocks of selected layers and the disk head *jumps* to the beginning of the next data block. It is possible that dependent upon the cylindrical distance between the current disk head position and the beginning of the next block, the file system can adaptively switch between *blind scan* and *selective retrieval*. In this paper, we assume *blind scan* strategy. Using *blind scan*, the total time to read data blocks can be modeled as in Eq. 5, where T_t represents the seek time between adjacent cylinders.

$$T_{total} = C_{movie} \times T_{cyl} + (C_{movie} - 1) \times T_t \quad (5)$$

C. Modeling Harmonic Interleaving Placement Scheme

Both *Progressive* and *Interleaved* file organization have their own advantages and disadvantages. In *Progressive Placement*, *inter* layer data block accesses can cause excessive disk seek. On the other hand, Interleaved placement can cause retrieval of unnecessary data blocks. We propose a novel placement scheme, Harmonic Interleaving, which effectively addresses these issues. Let C_{lower} and C_{upper} be the number of cylinders occupied by the layers in L_{lower} and L_{upper}, respectively. Fig.~\ref{fig:harmodel3} shows the general data retrieval sequence in Harmonic Interleaving scheme. In this figure, $L_{lower} = \{L_1, L_2\}$ and $L_{upper} = \{L_3, L_4\}$. We are to read segment 3. Based upon the current network bandwidth availability, data blocks in L_1, L_2, L_3 and L_4 are selected for transmission. The total time for data read here consists of (i) time to read lower layer blocks T_1, (ii) seek time from the end of segment 3 in a lower layer to the start of segment 3 in an upper layer T_i, (iii) time to read upper layer blocks T_2 and (iv) seek time from the end of segment 3 in the upper layer to the start of segment 4 in the lower layer T_o. i and o is used to represent *in-sweep* and *out-sweep* of the disk head. If only lower layers are requested, T_i, T_2 and T_o are not necessary. Since the disk head skips $m-1$ segments during T_i and m segments during T_o, T_i and T_o are approximately the same. Let C_o and C_i be the cylindrical distance in T_o and T_i. Cylindrical distance is linearly proportional to the seek distance. Let segment size in L_{lower} and L_{upper} be s_{lower} and s_{upper}. In our case, C_o and C_i corresponds to $\kappa(s_{lower} * (m-3) + s_{upper} * 3)$ and $\kappa(s_{lower} * (m-3) + s_{upper} * 2)$, respectively. κ is a constant coefficient which scales the linear byte distance into cylindrical distance. Therefore, when the application is to read k^{th} segment with up to L_4, C_o and C_i can be formulated as in Eq. 6 and Eq. 7.

$$C_o = \kappa(s_{lower} * (m-k) + s_{upper} * k) \quad (6)$$

$$C_i = \kappa(s_{lower} * (m-k) + s_{upper} * (k-1)) \quad (7)$$

C_o and C_i are subject to the segment size and the segment offset as well. The expected value of C_o and C_i is $(C_{lower} + C_{upper})/2$. Subsequently, T_i and T_o can be formulated as in Eq. 8.

$$T_i \approx T_o = T_{seek}\left(\frac{C_{lower} + C_{upper}}{2} \right) \quad (8)$$

Assuming that up to the i^{th}-layer belonging to L_{upper} are selected for transmission, we can calculate T_2 as in Eq. 9.

$$T_2 = \frac{\left(\sum_{j=l+1}^{i} D_j \right) / m}{B} = \frac{\sum_{j=l+1}^{i} D_j}{mB} \quad (9)$$

Throughout the playback, data blocks belonging to layers in L_{lower} are retrieved once and only once. Hence, the aggregated time of T_1 can be calculated as $C_{lower} \times T_{cyl}$. Therefore, the total time to read a file in the *Harmonic Interleaving* placement scheme is computed as:

$$
\begin{aligned}
T_{total} &= m\left(T_1 + \sum_{j=l+1}^{n} \alpha_j (T_i + T_2 + T_o) \right) \\
&= C_{lower} \times T_{cyl} + m \sum_{j=l+1}^{n} \alpha_j \left(T_2 + 2 \cdot T_{seek} \left(\frac{C_{lower} + C_{upper}}{2}\right) \right) \\
&= C_{lower} \times T_{cyl} + m \sum_{j=l+1}^{n} \alpha_j \left(\frac{\sum_{k=l+1}^{i} D_k}{mB} + 2 \cdot T_{seek} \left(\frac{C_{lower} + C_{upper}}{2}\right) \right) \\
&= C_{lower} \times T_{cyl} + \sum_{j=l+1}^{n} \alpha_j \left(\frac{\sum_{k=l+1}^{i} D_k}{B} + 2m \cdot T_{seek} \left(\frac{C_{lower} + C_{upper}}{2}\right) \right)
\end{aligned}
$$

As we can see from the above formula, when layers in L_{upper} are frequently selected for transmission the Harmonic Interleaving placement scheme is expected to show large overhead, while only layers in L_{lower} are selected mostly it is expected to show little overhead.

IV. Performance Experiment

A. Experimental Setup

We examine the efficiency of each file organization technique. We generate synthetic trace for network bandwidth variability and transform the trace into a sequence of layers. This equence of *selected* layers are fed to the disk. We examine the utilization of the disk under different file organization techniques. It is very important that the analytical model for each file organization technique properly reflect the behavior of the physical disk. We examine the disk utilization obtained from the analytical models(Eq. 4, Eq. 5 and Eq. 10) as well as the physical experiment. The disk model used in this experiment is the IBM ultrastar 36LP DPSS-309170 (Table II).

TABLE II
Disk Specification : IBM DPSS-309170

Interface	Ultra 160 SCSI
Number of zones	11
Number of heads	3
Number of disks	2
Rotation speed	7200RPM
Track to track seek time	0.6ms

We obtain the seek time of the disk[11]. Fig. 6 illustrates the result.

The objective of the proposed file organization technique is to effectively support scalable streaming service from the file system's point of view.

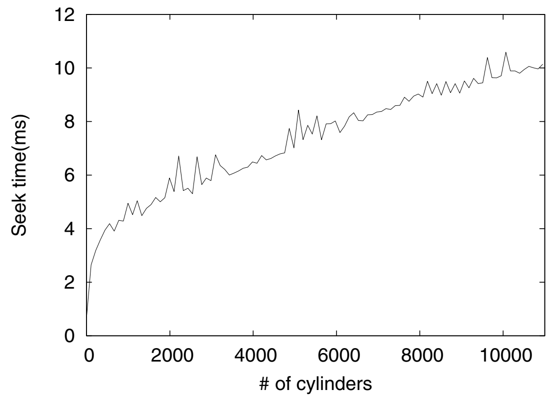


Fig. 6 Seek time characteristics of DPSS-309170

Thus, it is prerequisite to obtain the network bandwidth availability trace. In this experiment, we obtain the bandwidth trace from the simulation. Fig. 7 illustrates the network topology in our network simulation. We obtain bandwidth availability trace using *ns*[12].

There are four sources and four sinks. Two sources generate TCP flow. The two network pairs {6,8} and {7,9} are ftp server and ftp client pairs. The other two sources generate real-time video traffic. The two sources which generate the multimedia streaming traffic adaptively change the number of layers to cope with the congestion status of the subnet. We assume that these nodes control the transmission rate using RAP protocol. Table III shows the network simulator parameters.

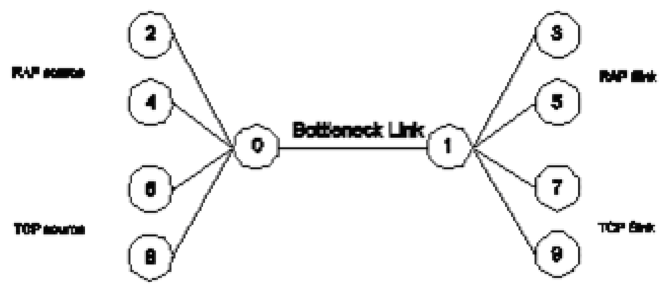


Fig. 7 Network topology for simulation

TABLE III
Parameter Settings in Simulation

Parameter	Value
Flows	4(TCP × 2, RAP × 2)
Fine Grain Adaptation	True
Bottleneck Bandwidth	6,000 Kbps, 2400 Kbps, 1440 Kbps
Bottleneck Delay	50ms
Bottleneck Queue Type	Drop Tail
TCP Source Type	TCP/Sack1
TCP Delay Acks	False
TCP Timer Granularity	50ms
Data Packet Size	1024bytes
Acknowledge Packet Size	40bytes
Simulation Length	2400secs

To examine the network bandwidth trace under various different network conditions, we use three different bottleneck bandwidths: 6,000 Kbps, 2400 Kbps and 1440 Kbps. TCP Sack1 protocol is used. Drop tail management is used in the bottleneck queue. Fig. 8 illustrates one of the bandwidth traces used in our simulation. It shows that RAP flow exhibits relatively smoother bandwidth fluctuation than TCP flow. The network throughput for the RAP nodes fluctuates between 180 KBytes/sec and 250 KBytes/sec.

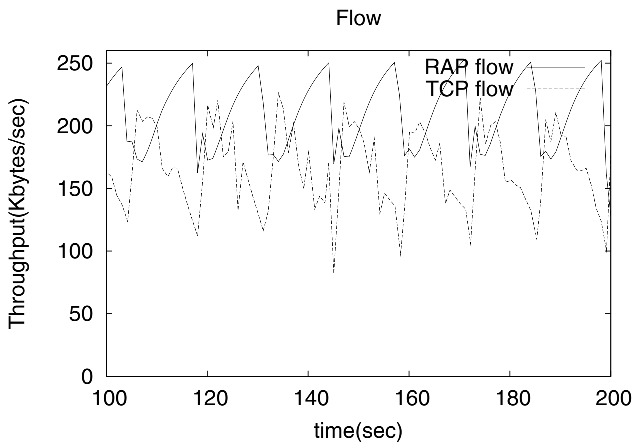


Fig. 8 Network throughput of a RAP(Rate Adaptive Protocol) session and a TCP session (Bottleneck bandwidth:6,000 Kbps)

The file used in our experiment is 40 min long and partitioned into 5 layers. The bandwidth requirement for each layer should be carefully chosen so that the file can support a wide variety of connection bandwidth. Table IV shows the bandwidth requirement for each layer used in our experiment. This partition strategy is based upon Helix Producer user's guide[13].

TABLE IV
Partition of Layers

Layer	Bandwidth (Kbps)	Cumulative Bandwidth(Kbps)	Network Medium
1	34	34	56K Dial-up
2	46	80	128K Dual ISDN
3	270	350	384K DSL or Cable Modem
4	350	700	768K DSL or Cable Modem
5	800	1500	DVD Quality

We compare the disk utilization of individual placement strategies under different network settings. We use three different bottleneck bandwidths between node 0 and node 1 in Fig. 7 : 1440 Kbps, 2400 Kbps and 6,000 Kbps.

B. Statistics on Layer Selection

In *Harmonic Interleaving*, layers are partitioned into two groups. The disk utilization and the efficiency of *Harmonic Interleaving* is largely dependent on the layer partitioning. To examine the performance of *Harmonic Interleaving*, we examine the performance under three different layer grouping policies: H_2, H_3 and H_4.

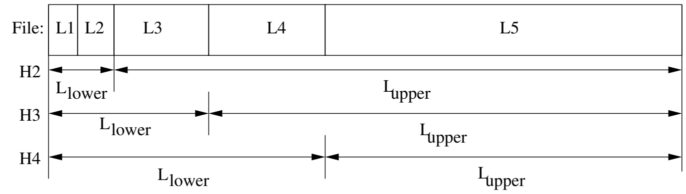


Fig. 9 Grouping of Layers in Harmonic Placement Strategy

In Fig. 9, H_2, H_3, H_4 illustrate the different partitioning schemes. H_2 partitions the layers into two groups $\{L_1, L_2\}$ and $\{L_3, L_4, L_5\}$. H_2 is intended to be used for 120K Dual ISDN connections. H_3 partitions the layers into two groups $\{L_1, L_2, L_3\}$ and $\{L_4, L_5\}$. H_3 is used for 384 Kbps DSL connections. H_4 partitions the layers into two groups $\{L_1, L_2, L_3, L_4\}$ and $\{L_5\}$. H_4 is used for 768 Kbps DSL connections.

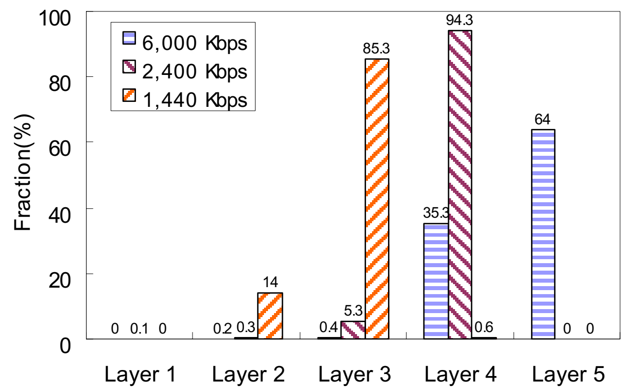
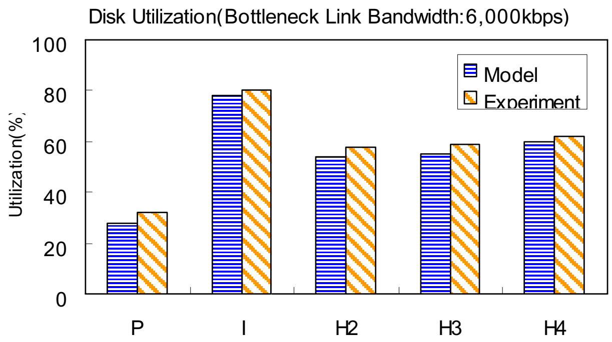


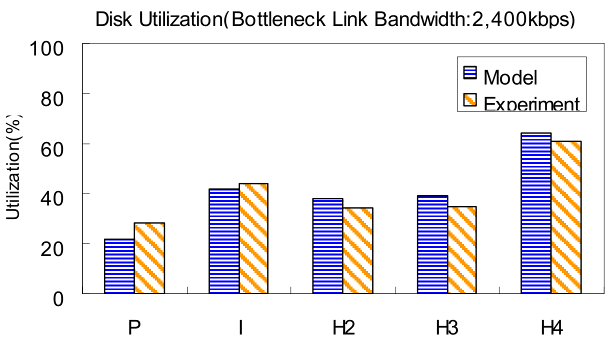
Fig. 10 Fraction of selected layers under different bottleneck links

Fig. 10 illustrates the percentage of layers transmitted during playback. The Y-axis denotes the transmitted layers and the X-axis denotes the percentage. Let us consider the case when bottleneck link capacity is 1,440 Kbps. When four sessions evenly share the available bandwidth, the data can be sent on the average at 360 Kbps. This intuition is well aligned with the test result. As can be seen in the figure, during 85% of the playback duration, three layers(L_1, L_2 and L_3) are transmitted when the bottleneck bandwidth is 1,440 Kbps. Data blocks in L_4 are rarely transmitted(0.6% of the playback time) and L_5 is not transmitted at all. Therefore, in this situation, placement strategy H_3 outperforms others as shown in Fig. 11. When network bandwidth bottleneck is 2,400 Kbps, four layers { L_1, L_2, L_3 and L_4 } are transmitted in most of the playback duration(94.3% of the playback duration). In this case, placement strategy H_4 yields the best disk utilization. Let us consider the case when bottleneck link is 6,000 Kbps. If four sessions uniformly share the bandwidth capacity, individual session can have 1.5 Mbps transmission bandwidth on the average. All five layers are transmitted during 65% of the time and four layers { L_1, L_2, L_3 and L_4 } are transmitted during 34% of the time. In this case, Interleaving placement outperforms other placement strategies, and we can think of the Interleaved placement scheme as one extreme case of Harmonic Interleaving scheme where all layers are included in the lower layer group.

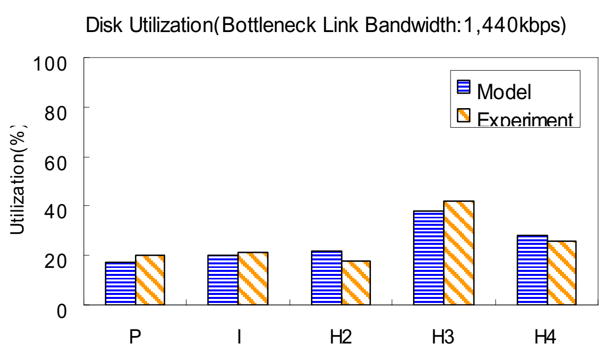
C. Disk Utilization



(a) bottleneck bandwidth:6,000 Kbps



(b) bottleneck bandwidth:2,400 Kbps



(c) bottleneck bandwidth:1,440 Kbps

Fig. 11 Disk Utilization

Fig. 11 illustrates the relationship between the placement strategy and disk utilization under different bottleneck bandwidths. When network capacity is 6,000 Kbps, Interleaving exhibits the best disk utilization. This is because there is sufficient network bandwidth availability and thus most layers are transported. When the bandwidth of a bottleneck link is 2,400 Kbps, each of the four sessions is allocated on average 800 Kbps data rate and hence up to layer 4 is transferred most of the time. In this case, placement strategy H_4 yields the best performance. When the bandwidth of a bottleneck link is 1,440 Kbps, placement strategy H_3 yields the best performance.

V. CONCLUSION

Temporal variation of network bandwidth availability gives rise to layered encoding and scalable streaming technology. However, in rate adaptive transmission of layer encoded multimedia content, sequential playback of data does not necessarily imply the sequential retrieval of data blocks in a file. Scalable coding and rate adaptive transmission brings another dimension of complexity from file system's point of view. Legacy file organization of multimedia contents places the data blocks in a temporal order. This placement strategy may not yield optimal performance in a scalable streaming environment. We find that organizing the file data blocks solely in a temporal order or via layer major order does not properly exploit the disk performance. In this work, we propose a novel file organization technique, *Harmonic Interleaving*. Harmonic Interleaving elaborately combines the temporal order based file organization and layer order based organization. It partitions the data blocks into two groups based upon its layer. We use a *progressive* placement strategy for *inter*-group placement and an *interleaving* strategy for *intra*-group strategy. We developed an elaborate model to capture the behavior of the disk. In our experiment, we examine the disk utilization via a physical experiment and verify that our disk model closely captures the physical behavior. Via simulation and physical experiment, we show that harmonic placement scheme yields superior disk utilization on a moderately loaded network. This placement technique effectively addresses the issues aroused by layered encoding scheme and TCP-friendly rate adaptive protocol. The results of this study serve as a guide on how to store scalable coded files. They can also be used by file system designers or multimedia content publishers.

REFERENCES

[1] M.-S. Chen, D. D. Kandlur, and P. S. Yu, "Optimization of the grouped sweeping scheduling (GSS) with heterogeneous multimedia streams", in *Proceedings of the first ACM international conference on Multimedia.* ACM Press, 1993, pp. 235-242.

[2] Renu Tewari and Richard King and Dilip Kandlur and Daniel M. Dias, "Placement of Multimedia Blocks on Zoned Disks", in *Proceedings of DPIE West '96*, 1996.

[3] R. V. Meter, "Observing the effects of multi-zoned disks", in *Proc. of Usenix Technical Conference*, San Jose, CA, USA, 1997.

[4] R. Wijayaratne and A. N. Reddy, "System support for providing integrated services from networked multimedia storage servers", in *Proceed-ings of ACM Multimedia Conference*, Ottawa, Canada, November 2001, pp. 270-279.

[5] P. Shenoy, P. Goyal, and H. Vin, "Architectural considerations for next generation file systems", in Proceedings of ACM Multimedia Conference, Orlando, FL, USA, Nov. 1999, pp. pp. 457 – 467.

[6] G. Nerjes, P. Muth, and G. Weikum, "Stochastic service guarantees for continuous data on multi0zone disks", in *Proceedings of the 16th Symposium on Principles of Database Systems*, Tucson, Arizona, 1997, pp. 154-160.

[7] M. Chen, D. Kandlur, and P. S. Yu, "Storage and retrieval methods to support fully interactive playout in a disk-array-based video server", *Multimedia System*, vol. 3, no. 3, pp. 126-135, July 1995.

[8] R. N. Inc., "Helix producer user's guide", June 2002.

[9] W. Li, "Overview of Fine Granularity Scalability in MPEG-4 Video Standard", *IEEE Trans. On Circuits and Systems for Video Technology*, vol. 11, no. 3, March 2001.

[10] R. Rejaie, M. Handely, and D. Estrin, "Rap : An end-to-end rate-based congestion control mechanism for realtime stream in the internet", in *Proceedings of IEEE Infocom*, New York, NY, USA, March 1999, pp. 1337-1345.

[11] Z. Dimitrijevic and R. Rangaswami and E. Chang and D. Watson and A. Acharya, "User-Level SCSI Disk Feature Extraction", UCSB, July 2001.

[12] S. McCanne and S. Floyd, "Network Simulator", www-mash.cs.berkeley.edu/ns, 2002.

[13] Real Networks. Inc., "Helix Producer USER'S GUIDE", June 2002.

Youjip Won received his BS and MS in Computer Science from Seoul National University in 1990 and 1992, respectively. He received Ph. D in Computer Science from University of Minnesota, USA in 1997. After his Ph.D, he worked for Intel as Server Performance Analyst. From 1999, He has been teaching in Dept. of Electrical and Computer Engineering, Hanyang University, where he is now Associate Professor. His research interests include Operating System, Storage System, High Speed Internet, Wireless Communication, Internet Model and etc.

Seunghun Yang received his BS and MS degrees in Electrical and Computer Engineering from Hanyang University in 2001 and 2003, respectively. He is currently a software engineer at LG Electronics. His research interests are on File System, Mobile Computing, Operating System and Wireless Network.

Sooyong Kang received his B.S. degree in mathematics and the M.S. and Ph.D degrees in computer science, all from Seoul National University (SNU), Seoul, Korea, in 1996, 1998 and 2002, respectively. He was then a Postdoctoral Researcher in the School of Computer Science and Engineering, SNU. He is now with the Department of Computer Science Education, Hanyang University, Seoul. His research interests include multimedia systems, especially multimedia data transmission system and multimedia storage system. He is also interested in the security area including secure communication on the Internet, intrusion detection systems and attack detection systems