A Network Channel Buffer Scheduling Algorithm for an Interactive VOD Server

Sungyoung Lee[J], Hyungill Kim[J], Jongwon Lee[JJ]

Department of Computer Engineering[J]
Kyunghee University, Seoul, Korea
Software Research Lab.[JJ]
Korea Telecom, Seoul, Korea

Abstract

A VOD server handles various types of media such as continuous media (CM) and non-continuous media (NM). Conventional VOD server buffer scheduling algorithms like Round Robin (RR) and Earliest Deadline First (EDF) have tradeoffs between guaranteeing the deadline of CM and getting a fast response time of NM. The former performs well in a NM environment while the latter is suitable for CM. In this paper, we present a network channel buffer scheduling algorithm to support VOD server based on the dynamic-Critical Task Indicating algorithm [8] which was developed by the authors. The goal of the proposed algorithm is to get fast response time for NM while guaranteeing the deadlines of CM. Our simulation study show that the proposed algorithm demonstrates a similar blocking rate for CM, however considerable improvement of the average response time for NM compared to EDF scheduling.

1. Introduction

Over the last few decades, the development of technology for the high speed computer networks has enabled us to receive various types of services such as multimedia applications [1][12][14]. We however, have recognized that since multimedia traffic is characterized by a wide dynamic range of burstiness and requires a variety of Quality of Service (QOS) parameters to be satisfied, the current network technology can not fully support the requirements of multimedia applications. For example, since a round robin scheduling algorithm is mainly to focus on the fairness of resources allocation, it can not guarantee the continuity of the CM (video and audio) which have to meet stringent timing requirement. On the other hand, an EDF algorithm suffers from low average response time for NM (text and image). We therefore, try to find a reasonably good scheduling method to handle mixture of the different types of media in an Asynchronous Transfer Mode (ATM) network. Since the ATM does not provide a fixed bandwidth for synchronous traffic, it overcomes the limitations, which are mainly caused by the bursty of multimedia traffic, of currently popular networks including FDDI and DQDB.

Considerable research have been done in the field of multimedia network channel scheduling [2][3]. Gibbson [5] proposed LAP (limited a priori) scheduling in an FDDI network. The scheduler assumes that it knows enough about the system a priori to schedule the next period of, or limited portion of the presentation. By dividing a multimedia presentation into periods, the scheduler adapts to changes in the system such as dynamic user input or network load changes. Ferrari [4] is working on the design and development of real-time protocols and network support for CM applications in an ATM environment. The real-time protocols developed by Ferrari include real-time multicast channel, reservation of real-time channels, dynamic channel management, and a channel group abstraction. Krishnamurthy [6] proposed the use of the ATM to support multimedia traffic in a Local Area Network (LAN) environment. An adaptive RR queuing technique was considered which was shown to provide better performance at the cost of a high buffering requirement. Zheng [15] suggested an EDF preemptive scheduling technique in a point-to-point network on which the periods, deadlines of data, and amount of the transmission traffic were known in advance.

Our approach however, is to reserve bandwidth capacity for each CM and NM a priori to meet QOS based on the user's requirements especially, in an interactive VOD application with an ATM network environment. The basic idea of this scheme is to group the same type of media for different users into a transmission period of a certain media (i.e., one video frame should be transmitted for every 1/30 second) which can be regarded as a scheduling period of a periodic task. To meet stringent timing constraint and guarantee data continuity, the CM should be scheduled periodically. To increase utilization and get a fast response time, the NM can be scheduled aperiodically. From a scheduler point of view, consequently, the former data type can be regarded as periodic tasks while latter as aperiodic tasks. Such a circumstance seems to be analogous to the principle of the aperiodic task scheduling in real-time systems. We therefore, have adapted an aperiodic task scheduling, called the dynamic-Critical Task Indicating (CTI) algorithm which is an extension of the fixed-CTI algorithm developed by the authors, to meet stringent timing constraints and guarantee data continuity for CM, and to get fast response time for NM.

The fixed-CTI algorithm is a preemptive joint scheduling of hard deadline periodic, and both soft and hard deadline aperiodic tasks in real-time systems. The tasks are scheduled in a way of mixed scheduling of a fixed [10][11] and dynamic priority algorithm [11]. The goals of the algorithm are not only to guarantee the deadline of all periodic tasks, but also to provide an efficient acceptance test mechanism for hard aperiodic tasks and to get the response time for the soft aperiodic tasks as small as possible. The simulation study shows that the CTI algorithm for soft aperiodic tasks has a considerable performance improvement over the other soft aperiodic task schedulings [9][13], especially under a heavy transition overload. The time complexity of an acceptance test for the hard aperiodic tasks in the CTI algorithm is O(n) [7] since it uses a one or two step slack discriminant formula on the deadlinewise preassignment table, called CTI table, while that of the other hard aperiodic task schedulings, such as the slack stealing algorithm, is $O(n^2)$ [13]. The merit of the CTI algorithm lies in the fact of its simplicity and scheduling predictability.

The remainder of the paper is organized as the following: Section 2 describes the background of the algorithm including queueing model. Section 3 explains the proposed network channel buffer scheduling algorithm. Section 4 shows the simulation study, and Section 5 concludes the paper.

2. Background of the Buffer Scheduling

2.1. Background

Since multimedia traffic is characterized by a wide dynamic range and burstiness and requires a variety of QOS parameters to be satisfied, it is important that the network be able to provide statistical multiplexing to handle the changing intensity of traffic. On the VOD applications in an ATM network, the scheduling of multimedia packets to the network channel is expected to be more significant than VOD server processor processing. Because the resource capacity of the storage and network channel in VOD system are limited, the server should be carefully designed to provide services to users fairly and effectively.

Multimedia data generally occurs at random and the number of occurrences of that media is also unexpectable. Thus, a direct application of the static real-time schedulings to the network channel buffer management may be undesirable. On the other hand, pure dynamic real-time schedulings (e.g., an EDF) suffer from poor response time of NM even though it guarantees the deadlines of CM. In order to resolve these problems as described in Section 1, we apply the dynamic-CTI algorithm to handle the CM as well as NM, mainly in an interactive VOD environment. The dynamic-CTI algorithm [8], an extension of the fixed-CTI scheduling, is a new type of joint scheduling that performs in such a way that dynamic priority assignment strategy for the given periodic tasks and the information on a deadlinewise preassignment table based on dynamic priority policy, called dynamic CTI table built off-line, are mixed dynamically according to the aperiodic task's arrival. The algorithm consequently, can satisfy the real-time requirement of CM and improve the response time of NM.

Meanwhile, since the scheduling overhead in a network channel buffer scheduler may cause a serious problem, it is important to design a buffer manager with a low scheduling overhead and a high utilization. For this, it is necessary to synchronize the deadlines of the same types of CM. For example, a video image frame must be transmitted for every 1/30 second. This condition will be applied to every user so that all the video images which occurs within 1/30 sec must have the same deadlines. In a case of audio track, the delay permission of 200msecs (1/5 sec) can be regarded as its deadline. This simplified method of assigning the synchronized deadlines (make same deadline) to each type of media can dramatically reduce the network channel buffer scheduling overhead.

2.2. Queuing Model

We assume that there are four different types of media, i.e., video, audio, fixed-length, and text. All of them except fixed-length media have variable bandwidths. As stated before, text media will be treated as aperiodic tasks, while the other three types of media as periodic tasks with their own deadlines.

In our queuing model, a single queue will be provided for each type of media so that the network channel buffer for a VOD server consists of four different queues (refer to Fig.1). The output link manager (i.e., channel buffer scheduler) will serve one packet in each queue before moving on to the next one based on the dynamic-CTI scheduling policy described in the next section.

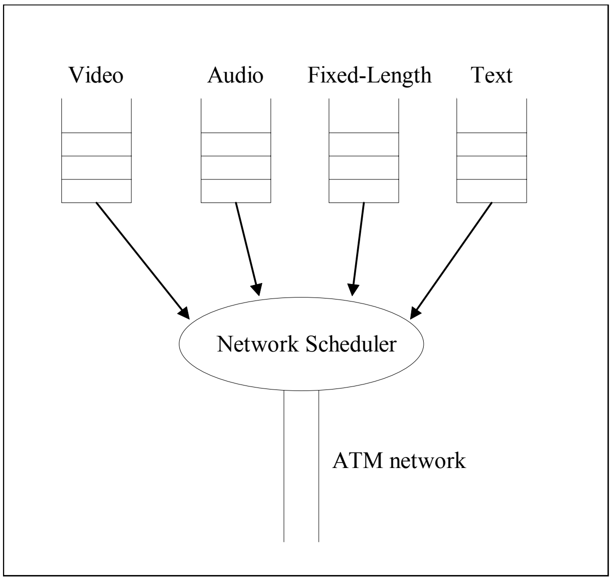

Fig. 1. Queuing Model for VOD Buffer

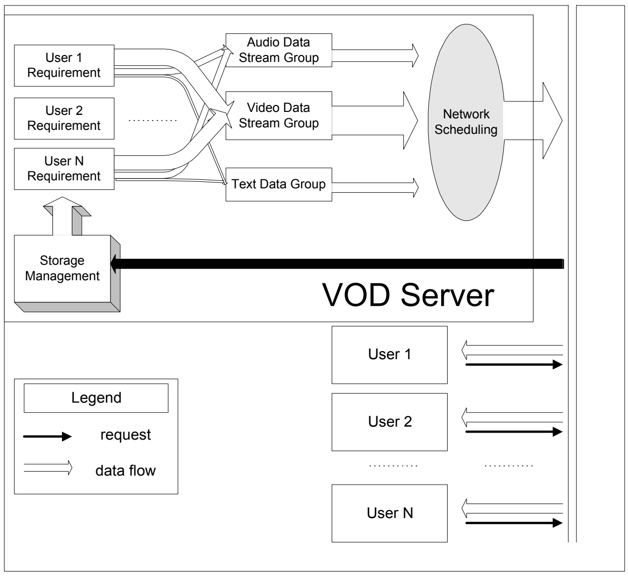

Fig. 2. An Example of Media Grouping

3. Algorithm Description

One of the most important factors to be considered in the network channel buffer scheduling is to decide the network bandwidth for each type of media in advance. Since the traffic requirements of the CM are known a priori, we can group same type of media into one scheduling unit (i.e., a CM group can be divided into 30 video and 5 audio samples within one second). Consequently, once we know the channel capacity and the number of different types of frames for a CM, it is easy to determine the size of channel bandwidth in advance. If NM should be scheduled with CM, we have to decide the bandwidth capacity for each media after considering requirements of both media. Notice that, as far as CM is concerned, guaranteeing the deadlines is more important than the fast processing.

The grouping and allocation strategy of the CM and NM before scheduling them into the multimedia network channel is as follows:

S1) Group the same types of CM from different users (refer to Fig. 2) and synchronize the deadlines of those requests in a transmission period.
S2) Allocate available maximum network bandwidth to each group of CM in advance.
S3) Allocate the available minimum network bandwidth to NM a priori. This minimum allocation bandwidth may be dependent on the QOS of CM service.

Pseudocode of the proposed algorithm is as follows:

Step 1. Allocate a portion of network bandwidth for CM in advance of which execution time is deferred at its maximum toward the deadline. Allocate a portion of network bandwidth for NM after considering the requirement of CM.
Step 2. When a request of NM occurs, the scheduler allocates it within an available bandwidth of the preassigned NM.
Step 3. If a request of NM does not occurs, the CM ready will be scheduled based on the on-line EDF scheduling.
Step 4. If some slacks are available within a bandwidth of CM, after deferring the execution time of CM toward the deadline, the NM will be scheduled at that slack.

Meanwhile, the characteristics of the proposed algorithm are as follows:

C1) The CM which arrived after no more bandwidth capacity is available, will be eliminated form the scheduling domain.
C2) The minimum bandwidth of NM should be guaranteed in advance and the slacks (unused bandwidth) in the CM bandwidth during run time, will be assigned for the NM.

Media	Distribution	Max. bitrate	Ave. bitrate	Min. bitrate	Period
Video	exponential	8.0 Mbps	2 Mbps	0.5 Mbps	33 ms
Audio	exponential	0.441 Mbps	0.221 Mbps	0.110 Mbps	200 ms
Fixed-length	fixed bitrate	0.16 Mbps	0.16 Mbps	0.16 Mbps	500 ms
Text Data	exponential	1 Mbps	0.5 Mbps	0.1 Mbps	infinite
SUM	N/A	9.601 Mbps	2.881 Mbps	0.870 Mbps	N/A

Table 1. The Characteristics of Media

4. Simulation

For the evaluation purpose, we use a simple model of an interactive VOD in 622Mbps ATM network. The server supports MPEG-II level of video and audio qualities, and NM such as a still picture, file, and text. Two types of scheduling schemes (NM First (NMF) and EDF) are simulated and compared to the dynamic-CTI scheme. We assume that the same types of media groups discussed in the previous section are synchronized. The NM is chosen at random from the range of 0.1Mbps to 1Mbps. Other simulation environments are depicted in Table 1.

We find that each user has a bandwidth range from 0.870 Mbps to 9.601 Mbps at maximum. In 622Mbps ATM Network, the server provides VOD services to 32 users at its minimum and 115 users at its maximum.

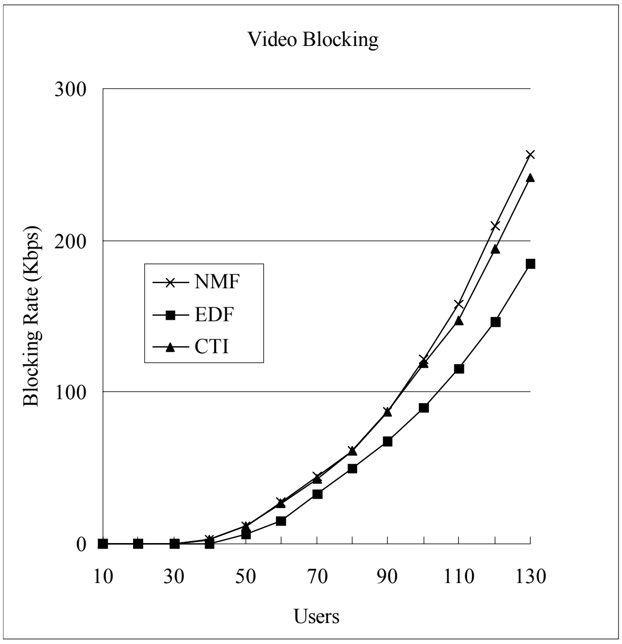

Fig. 3. The Rate of Video Blocking

In the case of video blocking, an EDF scheduling shows the lowest blocking rate (refer to Fig. 3). The proposed dynamic-CTI algorithm shows a little bit higher blocking rate than that of an EDF. However, it still reasonably performs almost similar to the EDF in most cases. In case of audio blocking, when the number of users is more than 100, the blocking rate of the proposed algorithm has been increased since the preallocated bandwidth of the algorithm is about to saturate at that point (refer to Fig. 4).

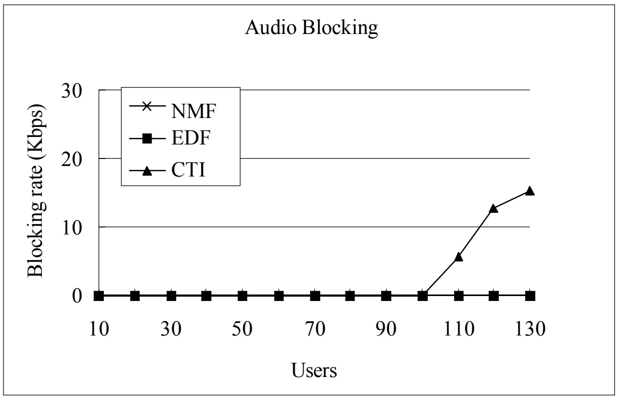

Fig. 4. The Rate of Audio Blocking

However, the proposed algorithm which has adopted a bandwidth preallocation mechanism for each type of media in advance, in this case, 10% of the bandwidth is assigned to the NM, dramatically improves the delay time of NM compared to the EDF scheduling which has no room for allocation of NM since it never considered its bandwidth in advance (refer to Fig. 5).

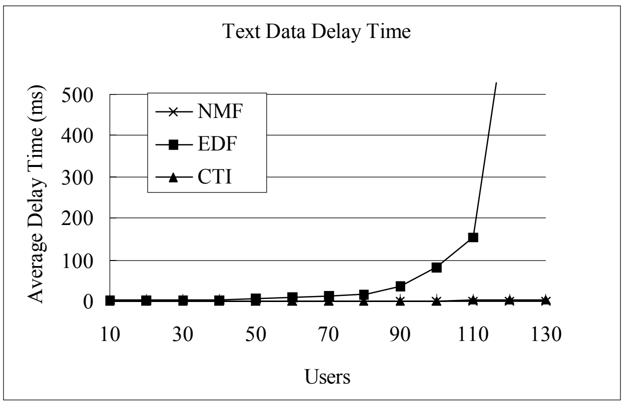

Fig.5. Delay Time of NM

5. Summary

In this paper we present a network channel buffer scheduling algorithm to support a VOD server. In order to satisfy the continuity and rigorous timing constraint for CM, we adopted a dynamic real-time scheduling policy, dynamic-CTI algorithm developed by the authors. Moreover, to reduce the scheduling overhead, we make the same types of media as a group and synchronize their deadlines to be same within a scheduling unit. By using this scheme, we have considerably improved the delay time of NM and still maintained a reasonably good blocking rate for CM. In the future, we will investigate an effective bandwidth decision mechanism for each type of media and continue more study on how to reduce the blocking rate for CM.

References

[1] D. P Anderson, S.Tzou, R. Wahbe, R. Govindan and M. Andrews, "Support for Continuous Media in the DASH System," Proceedings of the 10th International Conference on Distributed Computing Systems, Paris, May 1990.

[2] J-T Amenyo, A. A. Lazer, and G. Pacifici, "Proactive Cooperative Scheduling and Buffer Management for Multimedia Networks", Multimedia Systems, Vol. 1, No. 1, 1993, pp. 37-49.

[3] A. Dan, D. Siraram, and P. Shahabuddin, "Scheduling Policies for and On-Demand Video Server with Batching," Proceedings of the ACM Multimedia, pp. 15-24, October 1994.

[4] D. Ferrari, "A New Admission Control Method for Real-Time Communication in an Internetwork," Advances in Real-Time Systems, ed. S. Son, Prentice-Hall, Englewood Cliffs, NJ, 1995, Chapter 5, pp. 105-116.

[5] J. F. Gibbon, "Real-Time Scheduling for Multimedia Services Using Network Delay Estimation", Technical Report 05-20-1994, Boston University, 1994.

[6] A. Krishnamurthy, An ATM LAN for Multimedia Traffic, MS Thesis, Boston University, 1992.

[7] J.W. Lee, S.Y. Lee, and H.I. Kim, "Scheduling of Hard-Aperiodic Tasks in Hybrid Static/Dynamic Priority Systems," ACM SIGPLAN '95 Real-Time Workshop on Programming Language Design and Implementation, June 1995, pp.7-19.

[8] S.Y.Lee, J.W.Lee, and H.I.Kim, "A Soft Aperiodic Task Scheduling Algorithm in Dynamic-Priority Systems," submitted for publicatoin.

[9] J.P. Lehoczky and S. Ramous-Thuel, "An Optimal Algorithm for Scheduling Soft-Aperiodic Tasks in Fixed-Priority Preemptive Systems," Proceedings of the IEEE Real-Time System Symposium, December 1992, pp. 110-123.

[10] J.Y.-T. Leung and J. Whitehead, "On the Complexity of Fixed-Priority Scheduling of Periodic Real-Time Tasks," Performance Evaluation, 2:253-250, 1982.

[11] C.L. Liu and J.W. Layland, "Scheduling Algorithms for Multi-Programming in a Hard Real-Time Environments," Journal of the Association for Computing Machinery 20(1):46-61, January 1973.

[12] C.W. Mercer, S. Savage and H. Tokuda, "Process Capacity Reserves: Operating System Support for Multimedia Applications," Proceedings of the IEEE International Conference on Multimedia Computing Systems, Boston, pp. 90-99, May, 1994.

[13] S.R. Thuel and J.P. Lehoczky, "Algorithms for Scheduling Hard-Aperiodic Tasks in Fixed-Priority Systems Using Slack Stealing," Proceedings of the IEEE Real-Time System Symposium, December 1994, pp.22-33.

[14] H. Tokuda, and T.Kitayama, "Dynamic QOS Control Based on Real-Time Threads," In Proceedings of the 4th International Workshop on Network and Operating System Support for Digital Audio and Video, 1993.

[15] Zheng. Q. Real-time Fault-tolerant Communication in Computer Networks. Ph.D. thesis, University of Michigan, 1993.