

# Design and Simulation of an Efficient Real-Time Traffic Scheduler with Jitter and Delay Guarantees

Fu-Ming Tsou, Hong-Bin Chiou, and Zsehong Tsai, *Member, IEEE*

**Abstract**—In this paper, we propose a framework for real-time multimedia transmission in asynchronous transfer mode (ATM) networks using an efficient traffic scheduling scheme called multilayer gated frame queueing (MGFQ). MGFQ employs only one set of FIFO queues to provide a wide range of QoS for real-time applications. We also propose special cell formats for real-time multimedia transport and a hybrid design to allow MGFQ to combine its scheduling scheme with Age Priority Packet Discarding scheme. For this hybrid design, the cell level performance as well as the packet level QoS can be improved at the same time. Simulation results show that this hybrid design will be useful for packetized voice and progressive layer-compressed video transmission across the backbone networks. With the presented framework and the MGFQ algorithm, real-time multimedia traffic streams can be much better supported in terms of cell/packet delay and jitter.

**Index Terms**—Jitter, real-time, scheduler.

## I. INTRODUCTION

AS IS well known, asynchronous transfer mode (ATM) is designed to provide integrated services to all traffic types including voice, video, data, etc., within one transport architecture. Therefore, ATM networks are required to provide harmonization of divergent services with different levels of quality of service (QoS), such as cell transfer delay (CTD), cell delay variation (CDV), and cell loss ratio (CLR) [1], as is demanded by streams with a wide range of bandwidth requirement and burst characteristics. In the past, studies on ATM network performance or designs of ATM switches and schedulers have been done with the focuses primarily on the quality of service at the cell level. However, what end users concern most may not just be the lower layer performance. They may concern more about the IP layer QoS and, even, the application layer performance. Take real-time MPEG I/II video stream to be transferred on UDP datagram over ATM network for instance. Even though the cell loss rate is kept below a guaranteed level negotiated by traffic contracts, a user may not be able to accept the QoS offered by the underlying ATM service. The reason is that frequent CTD or CDV violation of end-of-message cells will lead to serious error events at MPEG level. Hence, further studies are

required to address the problems of designing an efficient ATM scheduler, which may provide buffer management mechanism for real-time packet streams, to not only maintain QoS on the cell level, but also to improve QoS on the packet level for the upper layer applications.

Many scheduling algorithms, such as weighted fair queueing (WFQ) [2], weighted round-robin (WRR) [3], etc., have been proposed for general data communications. However, these algorithms simply deal with the reduction of implementation complexity and the improvement of packet delay bound and fairness. In other words, they are not designed to meet the requirements of real-time traffic streams. For example, one may not need to reduce the cell delay bound as small as possible when real-time streams are conveyed. Instead, one can choose to increase the statistical multiplexing gain as large as possible and meet the delay/jitter constraints at the packet level at the same time. Currently, nearly all data communication scheduling algorithms adopt work-conserving disciplines. As a result, they can only limit the CDV to trivial bounds. As is known, scheduling algorithms, such as WFQ and its extensions, inherently face the problem of trading off between jitter bound and statistical multiplexing gain. In other words, the duration over which the statistical multiplexing gain is performed must be restricted if a tight jitter bound is desired. Conversely, if the multiplexing gain is to be maximized, then the jitter bound must be relaxed and this may lead to the need for transmission overhead for source clock recovery.

A typical representative of the scheduling algorithm supporting both flexible delay and jitter guarantees is the jitter-earliest-due-date (JEDD) [4] proposed by Verma *et al.* After each packet is served and prepared to transmit to its downstream node, the *due-date*, which is the difference between its local transmission deadline and actual transmission time, is inserted into a field of packet header. A regulator at the ingress of the next node holds the packet for a period before it is made eligible to be scheduled without violating the jitter bound. The node then transmits those eligible packets in an increasing order of their due-dates. However, the complexity of “winner selection” or “queue insertion” operation in the JEDD algorithm makes it difficult to be realized by a cost-effective hardware implementation. It is proved that the advanced buffer management mechanisms, such as the PPD [5] and the APPD [6], [7] schemes, can avoid the waste of network resources and improve the quality of service in the TCP or application layer at the receiving node. However, the implementation cost for such buffer management to be applied in combination with the JEDD algorithm could be very high because of the winner selection operations of the JEDD in the output buffer.

Manuscript received December 14, 1999; revised October 9, 2000. This work was supported by the National Science Council, R.O.C., under Grant NSC89-2213-E-002-078 and by the Ministry of Education, R.O.C., under Grant 89E-FA06-2-4-7. The associate editor coordinating the review of this paper and approving it for publication was Prof. Tak-shing Peter Yum.

F.-M. Tsou and Z. Tsai are with the Graduate Institute of Communication Engineering, National Taiwan University, Taipei 10617, Taiwan, R.O.C. (e-mail: d6942003@ee.ntu.edu.tw; ztsai@cc.ee.ntu.edu.tw).

H.-B. Chiou is with Northern Taiwan Business Group, Chunghwa Telecommunication Co., Ltd., Taipei, Taiwan, R.O.C. (e-mail: hbchiou@chttl.com.tw).

Publisher Item Identifier S 1520-9210(00)11053-3.

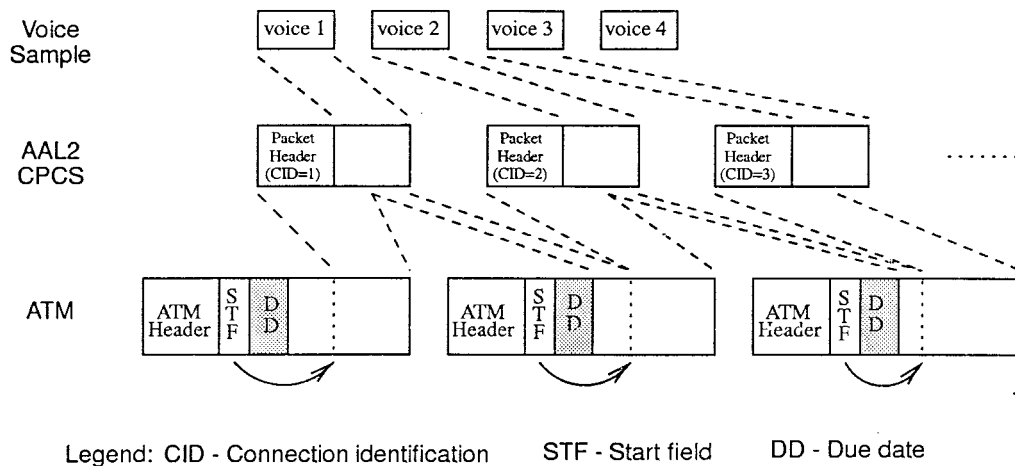


Fig. 1. Cell format for voice traffic in the MGFQ algorithm.

In order to reduce implementation complexity, Pocher *et al.* introduces the delayed frame queueing (DFQ) service discipline [8] and adopts the concept of rotating-priority-queue (RPQ) [9] proposed by Liebeherr *et al.* to achieve the delay and jitter guarantees. The service queue of each link is organized as a sequential row of several FIFO buffers. Service priority is given to the cells buffered in the FIFO queues with the smallest index value. Via employing RM cells, DFQ discipline can support a variety of delay and jitter bound combinations without sacrificing the fair distribution of QoS violations among the traffic streams. However, a trade-off exists between the scheduling performance and transmission overhead when DFQ scheme is employed. In addition, the number of the FIFO-queue sets in DFQ scheme must be the same as the number of the supported jitter levels. This implementation cost leads to the limitation on the scalability and granularity of jitter level for DFQ.

Hence, in this paper we propose a framework for multimedia transmission using a novel traffic scheduling scheme, called multilayer gated frame queueing (MGFQ) for real-time traffic. The rationale of the MGFQ algorithm is to accommodate an arriving cell into the proper FIFO queues according to its due-date in the current node, where the due-date is calculated based on the previous due-date passed over from its upstream node. The goal of the MGFQ algorithm is to provide efficient real-time traffic scheduling, with a minimum level of processing, and yet satisfies different QoS including jitter and delay of various scales and granularities. The framework is supplemented with special cell formats for real-time voice and video streams. In addition, we also propose a hybrid design, called MGFQ with APPD [6], [7], to combine scheduling scheme and packet discarding scheme. With this hybrid design, cell discarding does not follow the so called “tail-drop” policy and the loss ratio is improved directly at the packet level.

The organization of this paper is as follows. In Section II, the special cell formats for real-time transport and the proposed traffic scheduler combined with selective packet discarding schemes is presented. The due-date calculation procedure is described in Section III. And the implementation complexity is discussed in Section IV. Simulation results of voice and video

traffic are shown in Section V and Section VI, respectively. And finally, in Section VII, we draw our conclusions.

## II. MULTILAYER GATED FRAME QUEUEING (MGFQ) DISCIPLINE FOR REAL-TIME TRAFFIC

In this section, we first describe the cell format for MGFQ scheme, which carries necessary due-date information without incurring significant protocol overhead. And, we illustrates how due-date is carried in voice and video streams. Then, the MGFQ operations in an ATM switch are described.

### A. Cell Format for Real-Time Transport

We adopt AAL2 defined in ITU-T recommendation I.363.2 [12] as our ATM adaptation layer protocol for voice and make some modifications to AAL2 in order to carry necessary due-date information. The resulting protocol stack of voice over ATM and cell format is shown in Fig. 1. In order to improve network utilization, we assume one or multiple voice calls can be carried in one ATM virtual channel (VC). A new field to support MGFQ, i.e., the due-date field, is assigned behind the *Start field* (STF) and is allocated 2 bytes for each ATM cell. In the *due-date* field, if we denote 12 bits as  $x$  and other 4 bits as  $y$ , then this *due-date* field represents  $x \cdot 2^y$  time slots. The definitions and formats of other fields follow the definitions in [12].

Next, we introduce the protocol stack for transport video over ATM with MGFQ. We recommend the use of AAL5 and also adopt RTP [13] and the principle of Application Level Framing [14] to minimize the impact on the receiver’s frame-level QoS degradation due to cell losses. With the protocol stack of RTP/UDP/IP/AAL5, the header information of all layers can be accommodated in the first two cells of a video frame. We use an additional *DD* (*due-date*) field in AAL5 overhead to carry necessary information for MGFQ as shown in Fig. 2. An optional RTP header extension is also suggested. The first field, *FT*, denotes the type of the video frame, such as *I*-frames, *P*-frame and *B*-frames. The field is useful to enforce selective discarding to further improve video frame playback performance. The *frame sequence number* and *frame size* (in cells) should be helpful to additional buffer

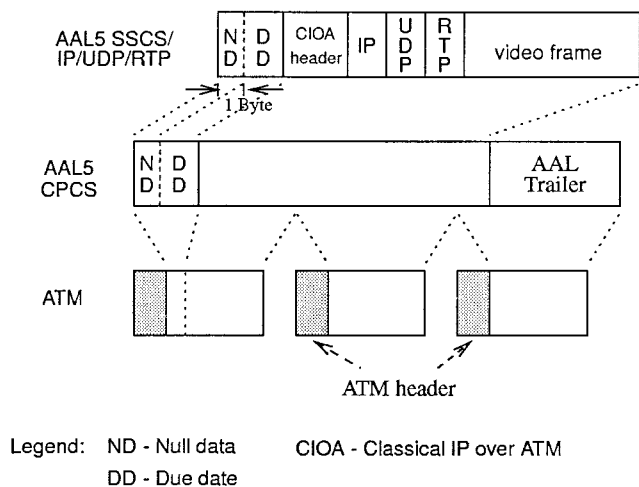


Fig. 2. Protocol stack and cell format for video traffic.

management and error control. The *timing field* carries timing information needed in the RTP and applications. If this RTP header extension is adopted and is passed over to the AAL layer, the timing field of RTP header extension can be extracted and transformed into *DD* field in AAL5. Otherwise, the due-date can be calculated with the timing information that is passed over directly from application programming interface, in addition to application PDU. Overhead of classical IP over ATM [15] is then included. In the service specific convergence sublayer (SSCS) of AAL5, we assign a 1-byte dummy data and a 2-byte due-date field before the regular video data.<sup>1</sup> After an SSSC PDU pass through common part convergence sublayer (CPCS) and is segmented by SAR of ATM, the due-date field of the first cell of a video frame should be at the same position as the voice cell. Therefore, the operations of the ATM switch are similar for both voice and video cells. As long as the switch can detect whether an incoming cell is a beginning of message (BOM) cell, the switch always extracts the correct due-date information of a video frame packet.

**B. Operations of the MGFQ Algorithm**

The queueing model of MGFQ is shown in Fig. 3. Each virtual path (VP) is assigned a dedicated FIFO queue. We assume each virtual path is dedicated to a class of services with a set of pre-determined cell-level QoS parameters, including delay, jitter, and cell loss ratio, etc. Thus, the cells in the same VP queue can be served by FCFS discipline. In addition, VPs of the physical link are organized as several groups according to VPs' jitter bounds. The jitter bounds of all VPs in Group *i* are within  $[(i - 1)T + 1, iT]$  slot times. Thus, *T* decides the granularity of the jitter bounds. *T* also denotes the length of the period that parameters in the scheduling operations are updated. This period is called *refreshing-period*, and is explained in further details later. Without loss of generality, we assume the assignment of the group identifier is in the increasing order of the jitter bounds. In other words, Group *i* is assigned the tighter

<sup>1</sup>It is noted that if the switch is able to perform different processing on voice and video cells according to their VPIs and VCIs respectively, then the *null data* field in the video cell can be eliminated.

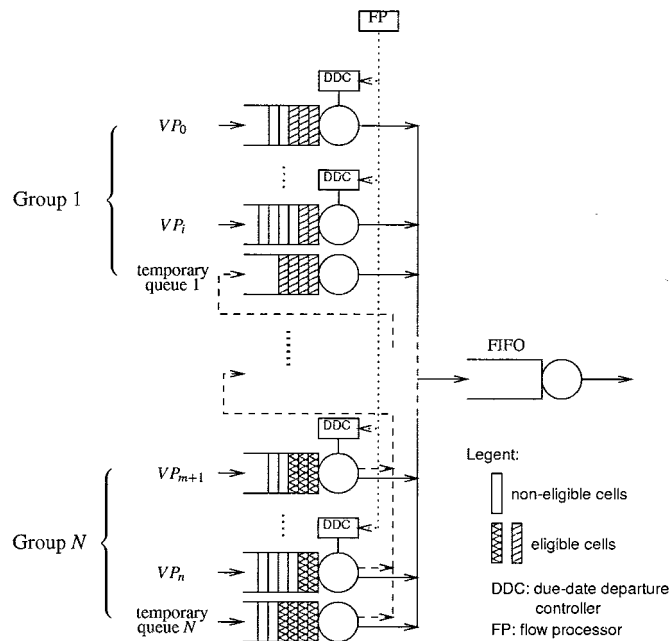


Fig. 3. Queueing model of the MGFQ algorithm for real-time traffic.

jitter bound than Group *i + 1*. Some FIFO queues called *temporary-queues* in this scheme are also dedicated for each group. The function of *temporary-queue* of Group *i* is to buffer the cells which were eligible in Group *i + 1* during the last refreshing-period. Here, a cell is called *eligible* if it does not violate the nodal delay bound and nodal jitter bound. In other words, the *temporary-queue i* buffers the cells whose due-dates were within the interval  $[iT + 1, (i + 1)T]$  in the last refreshing-period. Next, the flow processor (FP) informs the due-date departure-controllers (DDCs) to open the “gate” with period *T*. When DDCs of Group *i* open their gates, eligible cells belonging to Group *i* are moved to the temporary queue *i - 1*. In order to reduce the implementation complexity, the jitter bound of each VP has to be ceiled as the integer multiple of *T*. In order to improve QoS regarding packet loss ratio, the output buffer can employ a FIFO queue combined with two packet discarding schemes: Partial Packet Discarding scheme (PPD) [5] and Aged Priority Packet Discarding scheme (APPD) [6], [7].

The operations of the MGFQ algorithm are described as follows. Suppose the nodal jitter bounds of all VPs in this node is within the interval  $[0, NT]$ , then *N* temporary queues are dedicated to buffer eligible cells. Each time when a cell arrives, the initial nodal due-date and the eligible time of the cell are calculated. If the cell is for voice, the initial nodal due-date is calculated directly based on its own *DD* field. For video cells, such calculation is based on the *DD* field of the BOM cell. If this arriving cell is not eligible right now, it is attached to its own VP queue. Otherwise, it is put into the corresponding temporary-queue according to its due-date. When the flow processor informs all DDCs to open the “gate,” the eligible cells in the temporary queue *i* are moved to the temporary-queue *i - 1*. Next, eligible cells originally belonging to Group *i* are also moved to the temporary-queue *i - 1*. And, the cells whose due-dates are within  $[(i - 1)T + 1, iT]$  are marked eligible. After above operations, the eligible cells in Group 1 are moved to the output buffer during the period. And eligible cells in Groups

2, 3, 4, and so on, can be sent to output buffer during this period, if the high priority Groups are empty. We called this procedure as the “refreshing procedure.” At the same time, APPD and PPD mechanisms can be also applied to arrange eligible cells into the output buffer to avoid unnecessary waste of transmitting the cells that can not be re-assembled into a useful data unit in the receiver due to cell overdue. The detailed operations of APPD and PPD mechanisms are available in [6] and [7], respectively. In order to reduce the implementation complexity, we perform the refreshing procedure only at the starting epoch of refreshing-periods. If the output buffer is empty during the refreshing-period, then the eligible cells are served in the increasing order of group number. If the output buffer is not empty at the starting epoch of the refreshing-period, the remaining cells in the output buffer are discarded because of their violations of delay bounds. In order to more precisely describe the operation of the MGFQ algorithm, in Fig. 4 we present the pseudo code for an implementation of MGFQ.

### III. DUE-DATE CALCULATION PROCEDURE

Before describing the calculation procedure of the due-date of every arriving cell, we introduce the following notations. Without loss of generality, our due-date calculation procedure is focused on the virtual channel  $j$  ( $VC_j$ ) of virtual path  $i$  ( $VP_i$ ). Hence, in the following discussion, we omit the subscripts  $i$  and  $j$  which represents  $VC_j$  and  $VP_i$ . Three delay and jitter parameters are essential:

- $ND^h$ : nodal cell delay bound assigned to  $VP_i$  at node  $h$ ,  $h = 1, 2, \dots, H$ ;
- $J^h$ : nodal cell jitter bound assigned to  $VP_i$  at node  $h$ ,  $h = 1, 2, \dots, H$ ;
- $LD^{h-1, h}$ : propagation delay of the link between node  $h-1$  and node  $h$ ,  $h = 2, \dots, H$ .

Here, we simply assume node 1 and node  $H$  are the ingress node and the egress node of the network, respectively. Although the assigned nodal delay bound  $ND^h$  and nodal jitter bound  $J^h$  are  $ND^h$  and  $J^h$  could be arbitrary, one should note that the actual nodal delay bound and jitter bound provided by the MGFQ scheduler are  $\lceil ND^h/T \rceil T$  and  $\lceil J^h/T \rceil T$ , respectively.

Because the approaches of carrying due-dates for voice traffic and video traffic are different, the notations and the procedure of updating due-date for voice cells and video cells must be presented separately. Notice that the calculations are based on the modified cell format introduced and the operation algorithm mentioned in Section II.

#### A. Voice Traffic Streams

Additional definitions of notations to calculate the due-date information for voice traffic streams are as follows.

- $P_k^A$ :  $k$ th voice cell of  $VP_i$ , where the superscript “A” stands for “audio;”
- $AT_k^{A, h}$ : the arrival time of voice cell  $P_k^A$  at node  $h$ ;
- $LTT_k^{A, h}$ : latest transmission time of voice cell  $P_k^A$  at node  $h$ ;
- $ET_k^{A, h}$ : eligible time of voice cell  $P_k^A$  at node  $h$ ;
- $DT_k^{A, h}$ : departure time of voice cell  $P_k^A$  at node  $h$ ;
- $J_k^{A, h}$ : cell delay jitter of voice cell  $P_k^A$  at node  $h$ ;

```

main()
{
  while (1){
    t = system time;
    if (cell_arrival_event_occurs == TRUE){
      p = arrived cell;
      calculate initial nodal due-date and eligible time of p;
      enqueue p into its VP queue;
    }
    if (refreshing_event_occurs == TRUE){
      discard all cells in the output buffer;
      refreshing_procedure();
      schedule next refreshing_event at time t + T;
      alpha = 2;
    }
    if (cell_output_event_occurs == TRUE){
      if (output_buffer != EMPTY)
        p = HOL cell of the output buffer;
      else{
        do{
          p = HOL eligible cell in Group alpha;
          if (p == NULL)
            alpha = alpha + 1;
        }while((p == NULL) and (alpha <= N))
      }
      if (p != NULL){
        calculate the due-date of p;
        transmit p;
      }
      schedule next cell_output_event at time t + 1;
    }
  }
}

```

```

refreshing_procedure()
{
  for (i = N; i >= 2; i --){
    move eligible cells in Group i to temporary queue i - 1;
    mark cells in VP queues of Group i whose due-dates are
      within the range [(i - 1)T + 1, iT] as eligible cells;
  }
  mark cells in VP queues of Group 1 whose due-dates are
    within the range [0, T] as eligible cells;
  move eligible cells in Group 1 to the output buffer
  with enforcing APPD and PPD procedure;
}

```

Fig. 4. Pseudo code of MGFQ algorithm applied to the real-time traffic streams.

- $IND_k^{A, h}$ : initial nodal due-date of voice cell  $P_k^A$  at node  $h$ ;
- $DD_k^{A, h}$ : due-date of voice cell  $P_k^A$  when it leaves node  $h$ .

In the above notations, the latest transmission time at a node means the latest time epoch (or so-called deadline) at which a cell transmission still does not violate its nodal delay bound. Therefore,  $LTT_k^{A, h}$  can be obtained via

$$\begin{aligned}
 LTT_k^{A, 1} &= AT_k^{A, 1} + ND^1, \\
 LTT_k^{A, h} &= AT_k^{A, 1} + \sum_{j=1}^h ND^j \\
 &= LTT_k^{A, h-1} + ND^h, \quad h > 1. \quad (1)
 \end{aligned}$$

In other words,  $LTT_k^{A, h}$  can be calculated recursively.

If the voice cell  $P_k^A$  arrives at node  $h$  at time  $AT_k^{A,h}$ , then the initial nodal due-date,  $IND_k^{A,h}$ , is designated as

$$IND_k^{A,h} = LTT_k^{A,h} - AT_k^{A,h}. \quad (2)$$

We also know that the due-date of a cell at its departure time is the difference between the latest transmission time and the departure time. Therefore, when the cell  $P_k^A$  departs for node  $h+1$ , its due-date is calculated via

$$DD_k^{A,h} = LTT_k^{A,h} - DT_k^{A,h}. \quad (3)$$

Combining (1), (2) and adopting the property  $AT_k^{A,h} = DT_k^{A,h-1} + LD^{h-1,h}$ , we can derive

$$IND_k^{A,h} = ND^h + LTT_k^{A,h-1} - DT_k^{A,h-1} - LD^{h-1,h}. \quad (4)$$

Hence, according to (3) we derive the recursive formulas for the initial nodal due-date as

$$IND_k^{A,1} = ND^1, \quad (5)$$

$$IND_k^{A,h} = DD_k^{A,h-1} + ND^h - LD^{h-1,h}, \quad h > 1. \quad (6)$$

Following to the mathematical definition of delay jitter in [16], the cell delay jitter  $J_k^{A,h}$  is given by

$$J_k^{A,h} = \left| \left( DT_k^{A,h} - AT_k^{A,1} \right) - \left( DT_{k-1}^{A,h} - AT_{k-1}^{A,1} \right) \right|. \quad (7)$$

Now, it is easy to derive from (1), (3), and (7) that  $J_k^{A,h}$  can also be calculated using

$$J_k^{A,h} = \left| DD_k^{A,h} - DD_{k-1}^{A,h} \right|. \quad (8)$$

In other words, the notion of jitter described in Section II-A is actually consistent with the definition in

As we know, the eligible time of a cell is the time when that cell can be transmitted immediately without jitter bound violation. In the following, we show that  $J_k^{A,h}$  is bounded by

$$-J^h \leq J_k^{A,h} \leq J^h \quad (9)$$

if we set the eligible time,  $ET_k^{A,h}$ , as

$$\begin{aligned} ET_k^{A,h} &= LTT_k^{A,h} - J^h \\ &= AT_k^{A,h} + IND_k^{A,h} - J^h, \quad h \geq 1. \end{aligned} \quad (10)$$

By combining (3) and (8), one can write

$$J_k^{A,h} = \left| LTT_k^{A,h} - LTT_{k-1}^{A,h} - \left( DT_k^{A,h} - DT_{k-1}^{A,h} \right) \right|. \quad (11)$$

Because  $DT_k^{A,h}$  has to satisfy the relationship  $ET_k^{A,h} \leq DT_k^{A,h} \leq LTT_k^{A,h}$ , we can argue that the difference between  $DD_k^{A,h}$  and  $DD_{k-1}^{A,h}$  satisfies

$$ET_{k-1}^{A,h} - LTT_{k-1}^{A,h} \leq DD_k^{A,h} - DD_{k-1}^{A,h} \leq LTT_k^{A,h} - ET_k^{A,h}, \quad (12)$$

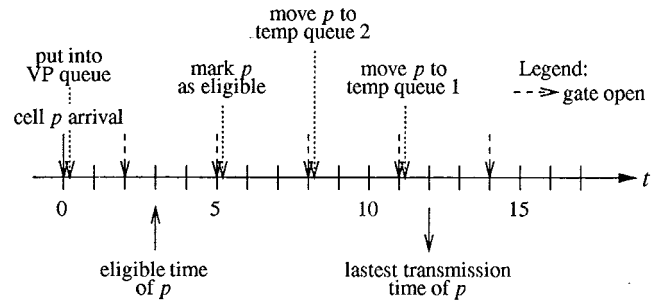


Fig. 5. Example of the operations of MGFQ algorithm.

which is equivalent to (9).

Therefore, when cell  $P_k^A$  is allowed to depart for the node  $h+1$ , it violates neither the delay bound nor the jitter bound of node  $h$  in the most strict sense. When  $P_k^A$  departs from node  $h$ , its due-date  $DD_k^{A,h}$  updated via (3) also carries the necessary due-date information.

By summarizing the whole due-date calculation procedure, the computation involves only five additions per cell. Since the due-date of a cell does not have to be updated slot-by-slot and only has to be updated when that cell leaves a node, such computation complexity should not be difficult to handle.

In the following, we use an example as shown in Fig. 5 to illustrate the operations of MGFQ algorithm when a voice cell is processed. We consider a VP whose assigned nodal delay bound is 12 slots and jitter bound is 9 slots. The length of the refreshing-period,  $T$ , is assumed as three time slots. Then, this VP is assigned to Group 3. We assume voice cell  $p$  arrives at time  $t=0$ , and then its latest transmission time is set to be 12 based on (1) and its eligible time is 3 according to (10). Suppose the "gate" opens at time  $t=2, 5, 8, 11,$  and  $14$  as shown in Fig. 5. When the "gate" opens at time  $t=5$ ,  $p$  is marked as eligible cell because its due-date is within  $[7, 9]$ . In the same way,  $p$  is moved to temporary-queue 2 and 1 at time  $t=8$  and  $11$ , respectively. If cell  $p$  is still not transmitted until time 14, then  $p$  will be discarded because of its overdue. On the other hand, if  $p$  can be transmitted within the time interval  $[5, 14]$ , the jitter of  $p$  will conform to (9).

## B. Video Traffic Streams

Additional notations to calculate the due-date for video traffic streams are as follows.

- $X^V$ : peak cell rate of the considered VC,  $VC_j$  of  $VP_i$ , of the video traffic, where the superscript "V" stands for "video"
- $P_{k,l}^V$ :  $l$ th cell of  $k$ th video frame of the considered VC,  $VC_j$  of  $VP_i$ ;
- $AT_{k,l}^V$ : arrival time of video cell  $P_{k,l}^V$  at node  $h$ ;
- $LTT_{k,l}^V$ : latest transmission time of video cell  $P_{k,l}^V$  at node  $h$ ;
- $ET_{k,l}^V$ : eligible time of video cell  $P_{k,l}^V$  at node  $h$ ;
- $DT_{k,l}^V$ : departure time of video cell  $P_{k,l}^V$  at node  $h$ ;
- $IND_{k,l}^V$ : initial nodal due-date of video cell  $P_{k,l}^V$  at node  $h$ ;
- $DD_{k,l}^V$ : due-date of video cell  $P_{k,l}^V$  when it departs from node  $h$ .

Suppose the video cell  $P_{k,l}^V$  arrives at node  $h$  at time  $AT_{k,l}^{V,h}$ . Therefore, the initial nodal due-date calculation formulas for the video cells are as

$$IND_{k,1}^{V,1} = ND^1, \quad (13)$$

$$IND_{k,l}^{V,h} = DD_{k,l}^{V,h-1} + ND^{V,h} - LD^{h-1,h}, \quad h > 1. \quad (14)$$

For BOM cells (i.e.,  $l = 1$ ), their due-dates are extracted from the DD field. Otherwise, for the cells other than BOM cells, their initial nodal due-dates are derived via the following procedure:

We know  $DD_{k,l}^{V,h-1}$  ( $l > 1$ ) can be obtained from

$$\begin{aligned} DD_{k,l}^{V,h-1} &= LTT_{k,l}^{V,h-1} - DT_{k,l}^{V,h-1}, \\ &= LTT_{k,l}^{V,h-1} - \left( AT_{k,l}^{V,h} - LD^{h-1,h} \right). \end{aligned} \quad (15)$$

Because the delay and jitter distributions for cells in a virtual path are all the same, the relationship

$$LTT_{k,l}^{V,h-1} - LTT_{k,1}^{V,h-1} = AT_{k,l}^{V,h-1} - AT_{k,1}^{V,h-1} = \left\lfloor \frac{l-1}{X^V} \right\rfloor \quad (16)$$

should hold. Then, from (14) and (15), one can derive

$$\begin{aligned} IND_{k,1}^{V,h} &= DD_{k,1}^{V,h-1} + ND^{V,h} - LD^{h-1,h}, \\ &= LTT_{k,1}^{V,h-1} - \left( DT_{k,1}^{V,h-1} + LD^{h-1,h} \right) + ND^{V,h}, \\ &= LTT_{k,1}^{V,h-1} - AT_{k,1}^{V,h} + ND^{V,h}. \end{aligned} \quad (17)$$

Combining (14)–(17), we obtain  $IND_{k,l}^{V,h}$  as

$$IND_{k,l}^{V,h} = IND_{k,1}^{V,h} + \left( AT_{k,1}^{V,h} + \left\lfloor \frac{l-1}{X^V} \right\rfloor - AT_{k,l}^{V,h} \right). \quad (18)$$

Hence, we obtain a recursive formula for computing the initial nodal due-dates of video cells:

$$IND_{k,1}^{V,h} = DD_{k,1}^{V,h-1} + ND^{V,h} - LD^{h-1,h}, \quad h > 1, \quad (19)$$

$$IND_{k,l}^{V,h} = IND_{k,1}^{V,h} + \left( AT_{k,1}^{V,h} + \left\lfloor \frac{l-1}{X^V} \right\rfloor - AT_{k,l}^{V,h} \right), \quad l > 1, h > 1 \quad (20)$$

with initial condition listed in (13). With (20), the corresponding  $DD_{k,l}^{V,h-1}$  in (14) is simply  $IND_{k,l}^{V,h} - ND^{V,h}$ .

Similar to the voice connections, the eligible time  $ET_{k,l}^{V,h}$  for video cells is calculated via

$$ET_{k,l}^{V,h} = AT_{k,l}^{V,h} + IND_{k,l}^{V,h} - J^h, \quad l \geq 1, h \geq 1. \quad (21)$$

Different from voice traffic, we only have to update the due-date of the BOM cell of every video frame. Suppose the first cell of a video frame  $k$  departs from node  $h$  at time  $DT_{k,1}^{V,h}$ , then its due-date  $DD_{k,1}^{V,h}$  is updated via

$$DD_{k,1}^{V,h} = IND_{k,1}^{V,h} - \left( DT_{k,1}^{V,h} - AT_{k,1}^{V,h} \right) \quad h \geq 1. \quad (22)$$

The total complexity of the DD calculation involves at most seven additions and one multiplication per cell for video.

### C. Design Issues

Because the propagation delay of a link between two nodes is fixed, the effect of the propagation delay on the due-date calcu-

lation procedures can be combined with the effect on the nodal delay. Therefore, in the following discussions, we neglect the propagation delay temporarily.

According to the operations of the MGFQ algorithm, we know that the jitter bound of a VP is constrained by the egress node of the network. Suppose the local jitter bound assigned to  $VP_i$  at the egress node  $H$  is  $J^H$ . Then, the end-to-end transmission delay (CTD) of  $VP_i$  is

$$\left( \sum_{h=1}^H ND^h \right) - \left\lceil \frac{J^H}{T} \right\rceil T \leq \text{CTD} \leq \left( \sum_{h=1}^H ND^h \right) + T. \quad (23)$$

Therefore, the end-to-end jitter bound [or called Cell Delay Variation (CDV)] is  $\lceil J^H/T \rceil T + T$ .

In addition, we do not expect cell losses to have any impact on scheduling performance. For voice traffic, each cell carries the *due-date* information. Therefore, the *due-date* calculations are mutually independent. For video traffic, the *due-date* calculations depend on the BOM cell. If the BOM cells can not be distinguished, the *due-date* calculations will be in error until the next distinguished BOM cell. However, because the PPD mechanism is applied, these cells, whose *due-date* are in error, will all be discarded by the ATM switch. Therefore, scheduling performance should not be affected by the cell loss for video traffic. Hence, MGFQ is robust against cell loss events if PPD is used.

## IV. DISCUSSIONS ON IMPLEMENTATION COMPLEXITY FOR MGFQ

Since maintaining a sorted priority queue often introduces significant processing overhead, much emphasis on QoS scheduler design is put on methods to simplify the task of maintaining a sorted priority queue. However, the implementation complexity will be an important metric to evaluate the worth of the schedulers. Hence, we investigate the implementation complexity for MGFQ and make a comparison with other scheduling algorithms, such as JEDD [4], DFQ [8], RPQ [9], RPQ<sup>+</sup> [17], etc., in this section.

As is well known, the algorithmic complexity for maintaining a sorted priority queue with  $N$  arbitrary entries is  $O(\log N)$  in the worst case. The cost to maintaining the sorted queue usually is due to the queue insertion operation upon each cell arrival. Alternatively, a “winner selection” procedure to select the cell with the shortest *due-date* in an unsorted queue can be applied. Via either the “winner selection” or “queue insertion” operation to select the cell with minimum *due-date* in JEDD, the implementation cost is still not easy to reduce, especially in the large-scale switches. Meanwhile, schedulers designed to operate with lower complexity have proposed. For example, Liebeherr and Wrege have proposed an approach that attempts to approximate a sorted priority queue at an output-buffered switch with significant complexity reduction [17]. In the following, we provide a comparison of the implementation complexity on those schemes employing the technique of approximating a sorted priority queue, such as MGFQ, DFQ, RPQ and RPQ<sup>+</sup> [17].(7).

Given the *due-date* supported by RPQ is in the range  $[0, NT]$ , RPQ employs extra  $N$  FIFO queues with one pointer-operation

TABLE I  
COMPARISON OF IMPLEMENTATION COMPLEXITY AMONG VARIOUS SCHEDULING ALGORITHMS WHERE  $J$  IS THE NUMBER OF DELAY JITTER LEVELS PROVIDED BY THE SCHEDULER

| Scheduler                 | DFQ  | RPQ | RPQ <sup>+</sup> | MGFQ     |
|---------------------------|------|-----|------------------|----------|
| No. of FIFO Queues        | $NJ$ | $N$ | $2N$             | $N$      |
| No. of Pointer Operations | $J$  | 1   | $N + 1$          | $\leq N$ |

to achieve the sorting operation. Nevertheless, RPQ will cause the problem of *rotation anomaly* [17]. To solve the rotation anomaly, RPQ<sup>+</sup> employs extra  $N$  FIFO queues and increase the number of FIFO queues to  $2N$ . In addition, extra  $N$  pointer-operations are needed to concatenate the cells in FIFO  $i^+$  into FIFO  $i$  [17]. An alternative approach, called DFQ [8], also adopts the concept of RPQ to achieve low-complexity traffic scheduler with delay/jitter guarantees. However, as mentioned in Section I, the number of the FIFO-queues set is increased linearly proportional to the supported jitter levels. In other words, there is a trade-off between the supported jitter levels and implementation complexity for the DFQ scheme.

In contrast, MGFQ needs  $N$  FIFO queues (called *temporary queues* in this paper) to accommodate the delay bound in the range  $[0, NT]$ . For each refreshing-period, MGFQ needs  $N$  pointer-operations to move cells from temporary queue  $i$  to temporary queue  $i - 1$ . However, these pointer-operations also can be achieved via rotating the FIFO queues. Therefore, the implementation complexity of MGFQ is higher than RPQ while it is lower than RPQ<sup>+</sup>. Afterward, we will show MGFQ can be combined easily with advanced buffer management schemes, such as APPD and PPD via simulations. Hence, the packet level QoS in terms of packet loss ratio can be improved at the same time via employing MGFQ algorithm. In Table I, we summarize the implementation complexity in terms of the number of FIFO queues employed in different schedulers and the scheduling complexity in terms of the number of pointer operations.<sup>2</sup>

V. SIMULATION OF VOICE TRAFFIC STREAMS

Next, we evaluate the performance of MGFQ scheme for voice traffic streams. The examined QoS parameters include cell delay, cell jitter distribution and cell discarding ratio. Here, the cell discarding ratio only accounts for those cells discarded due to delay or jitter violations. We employ FCFS and JEDD as baseline comparisons. The assumed queuing model of JEDD is shown in Fig. 6. The arriving cells of each virtual path are buffered in the corresponding regulators until they do not violate their jitter bounds, i.e., when they become eligible. Then, eligible cells are moved to the output buffer. The scheduling algorithm of the output buffer for JEDD algorithm is Earliest Due-Date (EDD). Here, we also assume the instantaneous movements of cells from the input regulators to the output buffer. Therefore, the JEDD algorithm adopted in this paper is considered as ideal cases for illustrating baseline performance.

<sup>2</sup>Note that the number of respective VP queues and the number of pointer-operations between VP queues and scheduler FIFO queues are not included in the table.

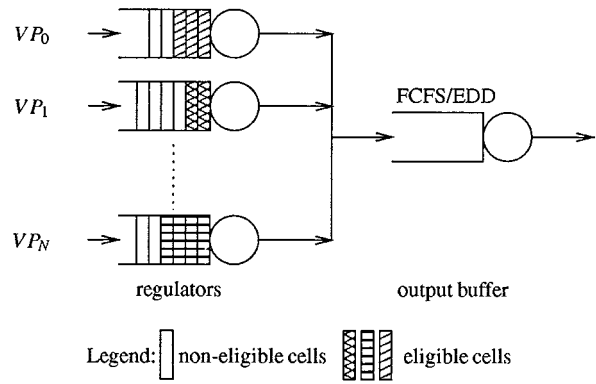


Fig. 6. Queuing model of JEDD algorithm in the simulation experiments.

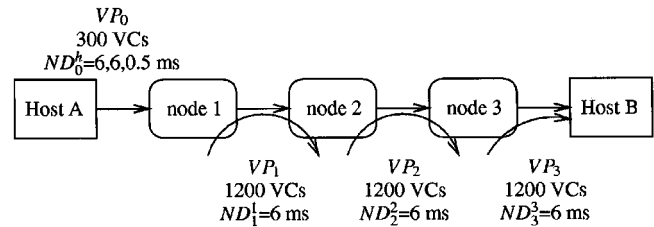
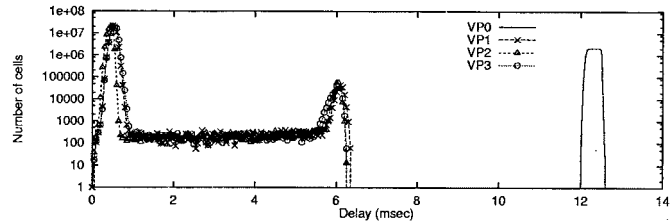
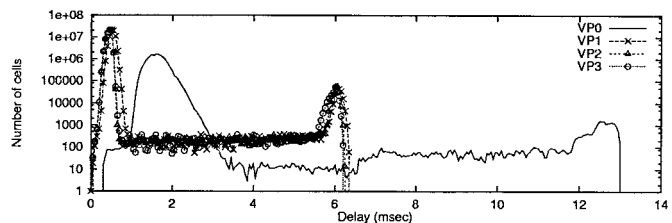


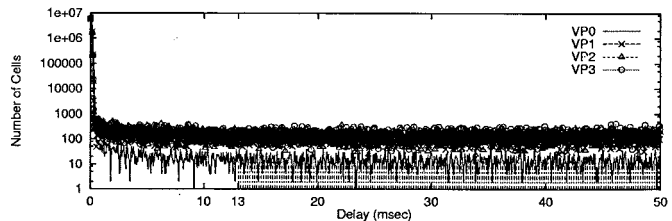
Fig. 7. Simulation model of MGFQ network for voice traffic.



(a) Tight Jitter Control ( $VP_0$ )



(b) Loose Jitter Control ( $VP_0$ )



(c) FCFS without any control mechanism

Fig. 8. Cell delay distributions for voice traffic with two classes of guaranteed jitter bounds. For tight jitter control, jitter bound of  $VP_0$  is 1 ms; while for loose jitter control, jitter bound of  $VP_0$  is 13 ms. The shadow part of (c) represents the discarded cells of  $VP_0$  due to violations of delay constraints.

TABLE II  
CELL DELAY OF METRICS OF MULTIPLE CTD/CDV BOUNDS FOR VARIOUS SCHEDULING DISCIPLINES

|                               | Virtual Path | Scheduling Algorithm | Max. Delay (msec) | Min. Delay (msec) | Avg. Delay (msec) | Jitter (msec) |
|-------------------------------|--------------|----------------------|-------------------|-------------------|-------------------|---------------|
| No Control                    | $VP_0$       | FCFS                 | 1091.26           | 0.03              | 11.47             | 12.92         |
|                               | $VP_1$       |                      | 499.07            | 0.03              | 4.91              | 6.40          |
|                               | $VP_2$       |                      | 350.69            | 0.03              | 3.33              | 6.44          |
|                               | $VP_3$       |                      | 290.29            | 0.03              | 2.90              | 6.49          |
| Tight Jitter Bound ( $VP_0$ ) | $VP_0$       | JEDD                 | 13.00             | 12.00             | 12.01             | 1.00          |
|                               |              | MGFQ                 | 13.00             | 12.00             | 12.29             | 0.97          |
|                               | $VP_1$       | JEDD                 | 6.50              | 0.03              | 0.05              | 6.47          |
|                               |              | MGFQ                 | 6.46              | 0.03              | 0.51              | 6.41          |
|                               | $VP_2$       | JEDD                 | 6.50              | 0.03              | 0.04              | 6.47          |
|                               |              | MGFQ                 | 6.50              | 0.03              | 0.43              | 6.44          |
|                               | $VP_3$       | JEDD                 | 6.50              | 0.03              | 0.06              | 6.47          |
|                               |              | MGFQ                 | 6.48              | 0.03              | 0.53              | 6.45          |
| Loose Jitter Bound ( $VP_0$ ) | $VP_0$       | JEDD                 | 13.00             | 12.00             | 0.26              | 12.87         |
|                               |              | MGFQ                 | 13.00             | 12.00             | 1.61              | 12.98         |
|                               | $VP_1$       | JEDD                 | 6.50              | 0.03              | 0.05              | 6.47          |
|                               |              | MGFQ                 | 6.46              | 0.03              | 0.51              | 6.41          |
|                               | $VP_2$       | JEDD                 | 6.50              | 0.03              | 0.04              | 6.47          |
|                               |              | MGFQ                 | 6.50              | 0.03              | 0.43              | 6.44          |
|                               | $VP_3$       | JEDD                 | 6.50              | 0.03              | 0.04              | 6.47          |
|                               |              | MGFQ                 | 6.48              | 0.03              | 0.43              | 6.45          |

The simulation model shown in Fig. 7 consists of a three-node network and 1500 voice streams in each node. All voice streams are assumed to follow ITU-T G.764 voice packetization recommendation [18] and the silence suppression mechanism is implemented. Therefore, the voice stream can be modeled as an ON-OFF traffic source.

Suppose the link bandwidth is 45 Mbps and  $VP_0$  consists of 300 virtual connections (VCs). These VCs are assigned nodal delays of 6.0 ms, 6.0 ms, and 0.5 ms at nodes one, two, and three, respectively.  $VP_1$ ,  $VP_2$  and  $VP_3$  serve as competing cross traffic and each of them contains 1200 VCs. The nodal delay assigned to cross traffic are all 6 ms. The ON/OFF duration of all these voice connections are with exponential distribution with mean 1.5 and 2.25 s, respectively. While ON, a voice source transmits one cell every 703 slot times, which is sufficient to support a 64 Kbps stream with silence suppression and necessary compression. Therefore, the average bottleneck link utilization is about 0.85. In order to avoid man-made simultaneous arrivals of cell bursts at the multiplexer, the starting epoch of each voice source is uniformly distributed over the 3.75 s interval. The refreshing-period is set to be 0.5 ms and all simulations last for  $10^8$  time slots.

#### A. Cell Delay Distribution

According to (23), it can be estimated that the guaranteed upper bounds on queueing delay are 13 ms for  $VP_0$ , and 6.5 ms for  $VP_1$  to  $VP_3$ . Two classes of guaranteed jitter bound are simulated for  $VP_0$ : a tight bound of 1 ms and a loose bound of 13 ms. We assume only a loose jitter bound of 6.5 ms is guaranteed for the cross traffic. Note that our MGFQ algorithm requires only one set of FIFO queues in node 3 to provide two classes of jitter bounds, while two sets of FIFO queues are needed in DFQ [8]. Fig. 8(a) and (b) show the

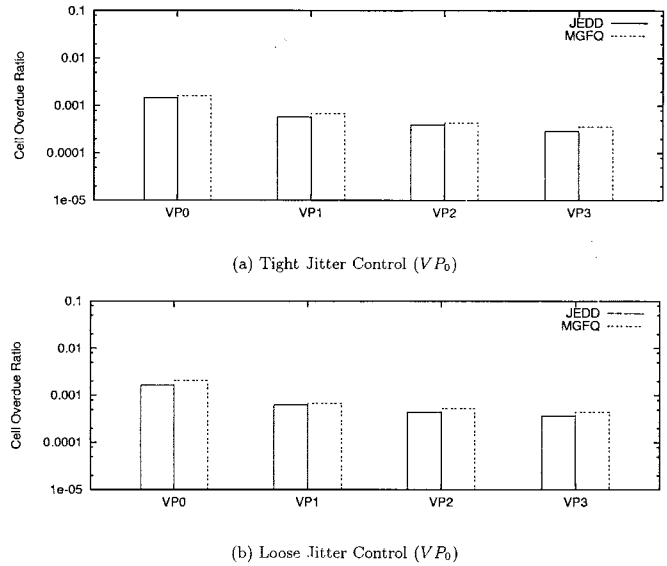


Fig. 9. Cell overdue ratios of multiple CTD/CDV bounds for JEDD and MGFQ.

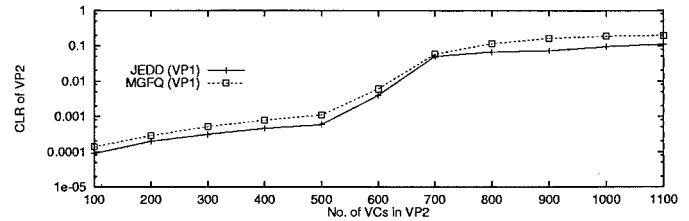


Fig. 10. Cell loss ratios of  $VP_1$  for JEDD and MGFQ under various traffic loads.

delay distributions of various VPs under the MGFQ algorithm for different jitter constraints. We can find that the delay distributions of all VPs conform to the delay constraints. The cell delay distribution under FCFS discipline, i.e., without any control mechanisms and regulators, is shown in Fig. 8(c) for the baseline comparison. All switch nodes in this baseline system perform nothing except forwarding the cells. The cells with delay bound violations are discarded only by the receiver. The cell delay distributions of all VPs spread over a wide range, and a large shadow part in Fig. 8(c) represents the cells of  $VP_0$  with delay beyond 13 ms. We can observe that if no control mechanism is adopted, the cell delay distributions of all VPs are beyond control and a large portion of cells violate their delay constraints. Hence, in the following simulation studies for voice traffic, statistics of the FCFS case are not included.

#### B. Cell Delay Metrics

Subsequently, in Table II we illustrate the cell delay performances of different scheduling algorithms. First, we observe that the mean queueing delay for the cross traffic is less than 0.5 ms, and is also less than a frame period in the DFQ algorithm. Thus, the transmission of multiple RM cells during a single frame period is required to improve the performance if DFQ is employed [8]. But note that this operation will increase the overhead of network. Secondly, the mean delay of the cross traffic in the MGFQ algorithm is larger than that in the



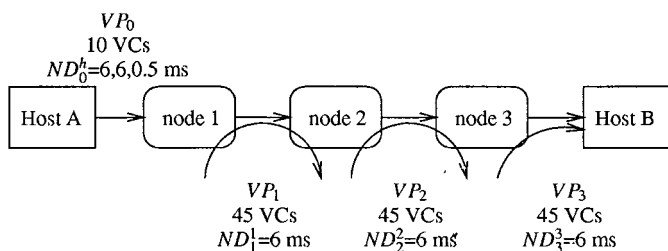
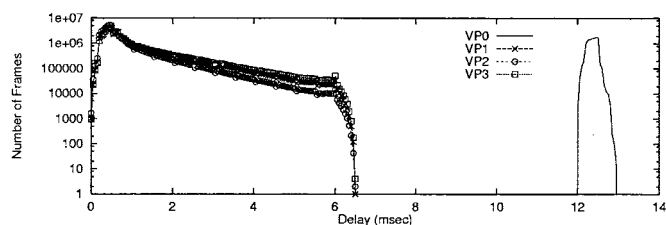


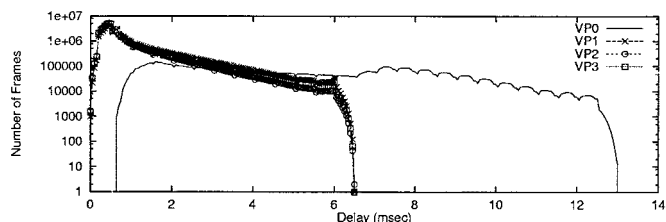
Fig. 11. Simulation model of the MGFQ network for video traffic.

TABLE III  
GENERAL INFORMATION OF THE MPEG VIDEO TRACE IN SIMULATIONS

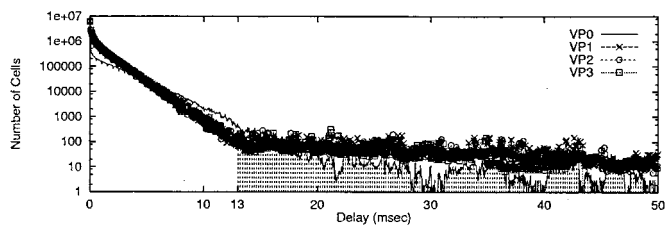
|   | I-Frame      | P-Frame  | B-Frame |
|---|--------------|----------|---------|
| Mean frame length (cells)                     | 217.401      | 108.376  | 27.868  |
| Variance of frame length (cell <sup>2</sup> ) | 4609.706     | 2757.451 | 119.330 |
| Maximum frame length (cells)                  | 637          | 543      | 145     |
| Overall mean rate                             | 587.881 Kbps |          |         |



(a) Tight Jitter Control ( $VP_0$ )



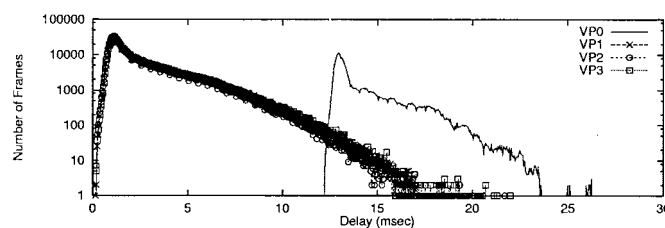
(b) Loose Jitter Control ( $VP_0$ )



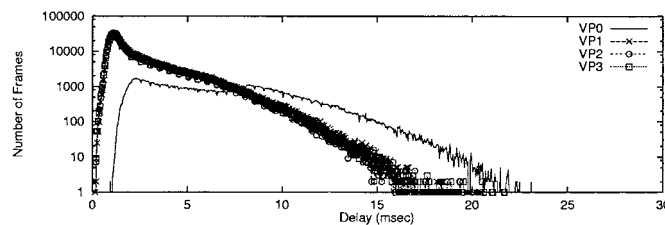
(c) FCFS without any control mechanism

Fig. 12. Cell delay distributions of multiple CTD/CDV bounds for video traffic simulation.

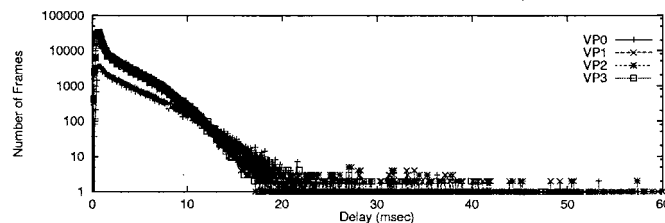
JEDD scheme. This is because the MGFQ algorithm is a sub-optimal scheduling algorithm. In the MGFQ algorithm, the eligible cells belonging to the same temporary queue have the same priority, regardless of the order of their eligible times and departure times, while the JEDD algorithm schedules the eligible cells according to their departure times. Hence, the coarse granularity in the service order of the MGFQ algorithm increases mean delay



(a) Tight Jitter Control ( $VP_0$ )



(b) Loose Jitter Control ( $VP_0$ )



(c) FCFS without any control mechanism

Fig. 13. Frame delay distributions of multiple CTD/CDV bounds for video traffic simulation.

slightly. However, the maximum delay and maximum jitter experienced by all VPs still conforms the delay and jitter bounds.

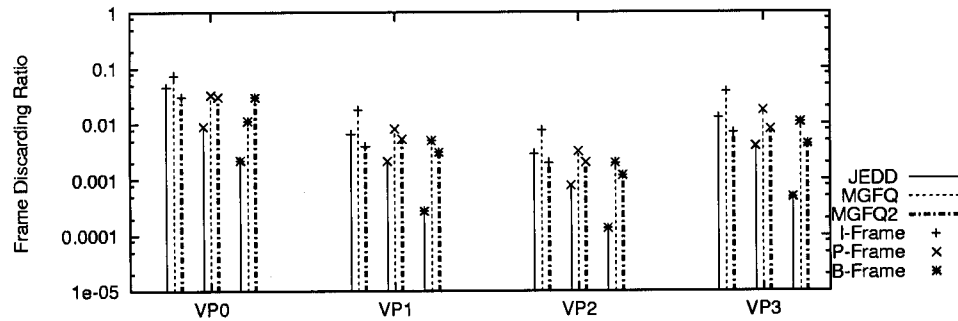
### C. Cell Overdue Ratio

Fig. 9 shows the cell overdue ratios of JEDD and MGFQ algorithms under two different jitter constraints. It can be observed that the cell overdue ratio under our MGFQ algorithm for  $VP_0$  can achieve a performance level close to that of the JEDD algorithm. For handling cross traffic, though the cell overdue ratio of the MGFQ algorithm is slightly higher than that of the JEDD algorithm, we have to note that the implementation complexity of the MGFQ algorithm is much lower than JEDD.

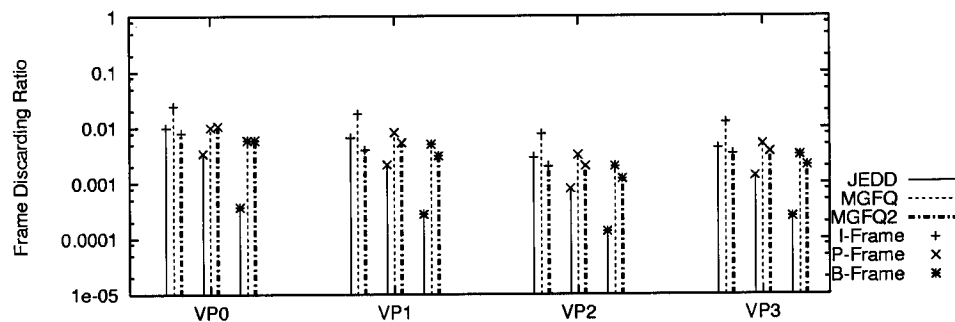
### D. Impact of Congestion

This simulation scenario illustrates the impact on the cell loss ratio among all connections during congestion periods. The simulation configuration consists of only one switching node and two VPs,  $VP_1$  and  $VP_2$ .  $VP_1$  contains 1100 VCs while the number of VCs in  $VP_2$  is increased from 100 to 1100. Hence, the total traffic load is increased from 0.683 to 1.252 when the number of VCs in  $VP_2$  is increased. We assume the delay bounds of two VPs are all equal to 6.5 ms. The jitter bound of  $VP_1$  is set to 1 ms while the jitter bound of  $VP_2$  is 6.5 ms.

From the simulation results shown in Fig. 10, we find that the performances of JEDD and MGFQ are very close under all levels of traffic loads. In other words, the CLR achieved



(a) Tight jitter bound.



(b) Loose jitter bound.

Fig. 14. Frame discarding ratios of various scheduling algorithms, where MGFQ2 represents the MGFQ algorithm combined with APPD and PPD schemes.

by MGFQ is very similar to JEDD, but with much lower complexity.

## VI. SIMULATION OF VIDEO TRAFFIC STREAMS

In this simulation scenario, video traces are applied to investigate the performance of MGFQ algorithm supporting real-time MPEG video over ATM. In this simulation, not only the cell level performance is shown, but also the frame level QoS, such as frame delay distribution and frame discarding ratio are presented. Notice that any cell is discarded while it violates the delay constraint and that a video frame is discarded if any cell of the frame is discarded.

The video traffic simulation model, which is similar to the model illustrated in Fig. 7, is shown in Fig. 11. A 45 Mbps link bandwidth is still assumed. The target virtual path,  $VP_0$ , consists of 10 VCs. These VCs are also assigned nodal delays of 6.0 ms, 6.0 ms, and 0.5 ms at nodes  $i$  (where  $i = 1 \sim 3$ ) respectively.  $VP_1$  to  $VP_3$  serve as competing cross traffic and each of them contains 45 VCs. The nodal delays assigned to cross traffic are all 6 ms. Each VC carries a video stream and each video stream is a replay of “James Bond: Gold finger” MPEG-1 video trace obtained from University Wuerzburg [19], with equally separated starting points within the 39996 frame positions. Since the frame rate is 24 frames/s, each stream is equivalent to a video of the length 1666.5 s. Again, in order to

avoid simultaneous arrivals of cell bursts at the multiplexer at the beginning, the starting epoch of each cell stream is uniformly distributed over the 1 s interval. All simulations last for  $10^8$  cell slot periods. Related statistical information of the video trace is listed in Table III. When a VC has a video frame to send, it uses peak rate to transmit the cell burst of the video frame. In this simulation experiment, we assume the peak rate of each VC is 15 Mbps. The average bottleneck link utilization is 0.72 in this simulation scenario.

### A. Cell Delay Distribution and Frame Delay Distribution

Figs. 12 and 13 show the cell delay distribution and frame delay distribution of the MGFQ algorithm for video traffic under two different jitter constraints, respectively. The cell delay distribution and frame delay distribution under FCFS discipline, i.e., without any control mechanism and regulators, are also shown in Fig. 12(c) and 13(c) for the baseline comparison. Again, all switch nodes in this baseline system perform nothing except forwarding the cells. The receiver of host B is responsible for discarding the cells with delay constraint violations. Again the cell distributions of all VPs spread over a wide range. The ratio of cells of  $VP_0$  with delay beyond 13 ms, indicated by the shadow in Fig. 12, is significant. In a precise fashion, the frame delay is defined as the time interval between the time when the first cell of the frame is transmitted and the time when the last cell is received by the receiver. According

to this definition, it is not trivial to control the delay jitter at the frame level. Nevertheless, we find that if the cell delay jitter is under control, then the frame delay jitter becomes small, as illustrated by the simulation results. Therefore, it is possible for the receiver to allocate smaller buffer to compensate the frame delay jitter if MGFQ traffic scheduler is implemented in the network. In turn, the receiver must allocate very large buffer to compensate the disturbed cell arrivals under FCFS.

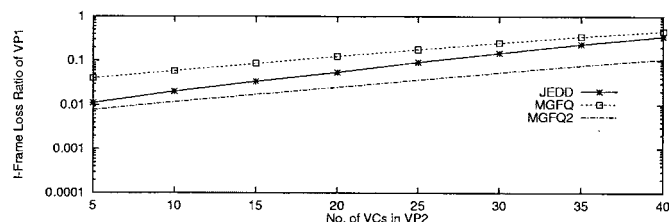
### B. Frame Discarding Ratio

Fig. 14 shows the frame discarding ratios of each frame type of four VPs under JEDD, MGFQ, and MGFQ combined with APPD and PPD, which is denoted as MGFQ2 in the figures. Although JEDD algorithm has the smallest frame discarding ratio for  $VP_0$ , the implementation complexity is the major cost. The reason that frame discarding ratio of  $I$ -frame is higher than  $B$ -frame or  $P$ -frame under JEDD and pure MGFQ algorithm, as observed from the simulation results, is due to large cell bursts of  $I$ -frames. Meanwhile, if the MGFQ algorithm is combined with APPD and PPD scheme, there are significant improvements in terms of fairness among different types of frames. Although the frame discarding ratios of  $P$ -frame and  $B$ -frame are higher than other schemes, when APPD and PPD schemes are adopted, the frame playback performance is not expected to degrade seriously since the layering codec technique is used. Therefore, we believe the MGFQ algorithm should be an excellent candidate to be used with advanced buffer management schemes. In contrast, this feature has not been well investigated in other scheduling algorithms.

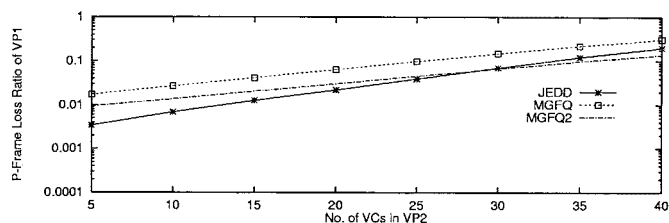
### C. Impact of Congestion

This simulation scenario describes the impact on the frame loss ratios under heavy loaded conditions. Similar to the simulation experiment of voice traffic, only one switching node and two VPs,  $VP_1$  and  $VP_2$ , are included in the simulation configuration.  $VP_1$  contains 40 video VCs while the number of VCs in  $VP_2$  is increased from five to 40. Each VC carries a video stream as mentioned in Section IV. Hence, the total traffic load ranges from 0.588 to 1.045 when the number of VCs in  $VP_2$  is increased. We assume the delay bounds of two VPs are all equal to 6.5 ms. The jitter bound of  $VP_1$  is set to 3.5 ms while the jitter bound of  $VP_2$  is 6.5 ms.

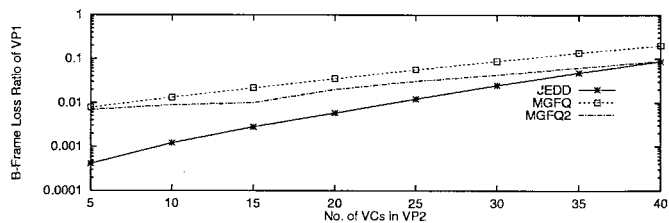
The frame discarding ratios of each frame type for  $VP_1$  under JEDD, MGFQ, and MGFQ combined with APPD and PPD, which is denoted as MGFQ2 in the figures, are shown in Fig. 15. When the traffic load increases (see Fig. 15), the  $I$ -frame discarding ratio of  $VP_1$  under MGFQ2 is better than JEDD at the expense of increasing the frame discarding ratios of  $P$ -frame and  $B$ -frame. Since the error in the  $I$ -frame could propagate and influences the quality for a sequence of frames,  $I$ -frame is considered more important. Meanwhile, the loss of  $B$ -frame is expected to have only limited impact. Hence, we believe it does not have much impact on the perceived visual quality. Last but not least, the MGFQ algorithm, which is a suboptimal scheduling discipline compared to JEDD, can still accommodate a good frame level performance even under heavy loaded conditions if it combines with advanced buffer management schemes such as APPD and PPD.



(a) I-frame discarding ratio.



(b) P-frame discarding ratio.



(c) B-frame discarding ratio.

Fig. 15. Frame discarding ratios of  $VP_1$  versus traffic loads under different scheduling disciplines.

## VII. CONCLUSIONS

In this paper, we first designed two ATM cell formats for carrying timing information in the upstream node to the downstream node along the transmission path for voice and video traffic, respectively. Based on these special cell formats, we presented a framework which includes an efficient scheduling algorithm called MGFQ for transporting real-time traffic over ATM networks with minimum processing and protocol overhead. Unlike previous studies [8], MGFQ employs only one set of FIFO queues to provide a wide range of QoS for real-time applications. Thus, it not only reduces the hardware implementation complexity significantly but also achieves high multiplexing gain. In addition, we had shown it can be combined easily with advanced buffer management schemes, such as APPD and PPD. Hence, both the cell level performance and the packet level QoS can be improved.

From the simulation results, we found that MGFQ can provide much better control of delay and jitter, and yet improve cell/packet discarding ratio. Because MGFQ allows the target traffic stream be granted higher priority than an interfering traffic stream, it may even accomplish better performance than JEDD by slightly degrading the performance of cross traffic. With the help of APPD and PPD, MGFQ can improve packet level QoS significantly in term of frame discarding ratio for video traffic. Although many people believe it is difficult to efficiently control packet level QoS, such as frame delay jitter,

using pure cell level QoS mechanisms, we showed that effective QoS control mechanisms at the cell level (especially jitter) should be able to achieve a commensurate packet level QoS. With MGFQ, a receiver can allocate less resources, such as buffers, to compensate the disturbed cell arrivals. Nevertheless, how to precisely map QoS parameters from the frame level or the packet level to the cell level needs further investigation.

To summarize, the presented framework, with MGFQ, provides a novel approach to implement real-time multimedia transport and the efficient traffic scheduling with flexible jitter and delay guarantees. Various kinds of customer requirements, such as flexible end-to-end jitter constraints, transmission via AAL1/2/5, and adaptive playout, etc., can be achieved by employing MGFQ-enabled switches. We believe that the MGFQ scheme should be able to support a wide range of other jitter and delay sensitive applications for multimedia communications.

## REFERENCES

- [1] The ATM Forum, *ATM User-Network Interface Specification Version 3.0*. Englewood Cliffs, NJ: Prentice-Hall, 1993.
- [2] A. Demers, S. Keshav, and S. Shenker, "Analysis and simulation of a fair queueing algorithm," in *Proc. ACM SIGCOMM '89*, 1989, pp. 1–12.
- [3] M. Katevenis, S. Sidiropoulos, and C. Courcoubetis, "Weighted round-robin cell multiplexing in a general-purpose ATM switch chip," *IEEE J. Select. Areas Commun.*, vol. 9, pp. 1265–1279, Oct. 1991.
- [4] D. Verma, H. Zhang, and D. Ferrari, "Guaranteeing delay jitter bounds in packet switching networks," in *Proc. Tricomm '91*, Chapel Hill, NC, Apr. 1991, pp. 35–46.
- [5] A. E. Kamal, "A performance study of selective cell discarding using the end-of-packet indicator in AAL type 5," in *Proc. IEEE INFOCOM '95*, pp. 1264–1272.
- [6] H.-B. Chiou and Z. Tsai, "An age priority packet discarding scheme for ATM switches on internet backbone," *IEICE Trans. Commun.*, vol. E81-B, no. 2, pp. 380–391, Feb. 1998.
- [7] —, "Performance of ATM switches with age priority packet discarding under the ON-OFF source model," in *Proc. IEEE INFOCOM '98*, San Francisco, CA, Mar. 31–Apr. 2 1998.
- [8] H. L. Pocher, V. C. M. Leung, and D. W. Gillies, "An efficient ATM voice service with flexible jitter and delay guarantees," *IEEE J. Select. Areas Commun.*, vol. 17, pp. 51–62, Jan. 1999.
- [9] J. Liebeherr, D. E. Wrege, and D. Ferrari, "Exact admission control for networks with a bounded delay service," *IEEE/ACM Trans. Networking*, vol. 4, pp. 885–901, Dec. 1996.
- [10] D. J. Wright, "Voice over ATM: An evaluation of implementation alternatives," *IEEE Commun. Mag.*, vol. 34, pp. 72–80, May 1996.
- [11] Z. Tsai, W. D. Wang, C. H. Chiou, J. F. Chang, and L. S. Liang, "Performance analysis of two echo control designs in ATM networks," *IEEE/ACM Trans. Networking*, vol. 2, pp. 30–39, Feb. 1994.
- [12] *B-ISDN ATM Adaption Layer Specification: Type 2 AAL*, *ITU-T Recommendation I.363.2*, Sept. 1997.
- [13] H. Schulzrinne, S. Casner, R. Frederick, V. Jacobson, and Internet Engineering Task Force, Audio-Video Transport Working Group, "RTP: A Transport Protocol for Real-Time Applications," RFC 1889, Jan. 1996.
- [14] A. Tanenbaum, *Computer Networks*, 3rd ed. Englewood Cliffs, NJ: Prentice-Hall, 1996.
- [15] J. Heinanen and Internet Engineering Task Force, Network Working Group, "Multiprotocol Encapsulation over ATM Adaption Layer 5," RFC 1483, July 1993.
- [16] H. Zhang, "Service disciplines for guaranteed performance service in packet-switching networks," *Proc. IEEE*, vol. 83, pp. 1374–1396, Oct. 1995.
- [17] J. Liebeherr and D. E. Wrege, "Priority queue schedulers with approximate sorting in output buffered switches," *IEEE J. Select. Areas Commun.*, vol. 17, pp. 1127–1144, June 1999.
- [18] *General Aspects of Digital Transmission Systems—Packetization Guide*, *ITU-T Recommendation G.764*, Nov. 1995.
- [19] O. Rose. (1995) *MPEG-1 Video Trace, James Bond: Goldfinger*. Inst. Comput. Sci. III, Univ. Würzburg, Würzburg, Germany. [Online] <ftp://ftp-info3.informatik.uni-wuerzburg.de/pub/MPEG/traces>



**Fu-Ming Tsou** received the B.S. and M.S. degrees in electrical engineering from National Taiwan University (NTU), Taipei, Taiwan, R.O.C., in 1995 and 1997, respectively. He is currently pursuing the Ph.D. degree in the Graduate Institute of Communication Engineering at NTU. His current research interests include high performance LAN protocols, traffic shaper design and analysis, real-time transport protocols of high speed networks, deterministic performance evaluation in ATM networks.



**Hong-Bin Chiou** received the B.S. degree from National Taiwan University (NTU), Taipei, Taiwan, R.O.C., the M.S. degree from National Tsing Hua University, Taiwan, and the Ph.D. degree in electrical engineering from NTU, all in electrical engineering, in 1986, 1989, and 1997, respectively.

From 1989 to 1991, he was a Project Researcher in the R&D Division of Acer, Inc. In 1991, he joined Telecommunication Laboratories, Chunghwa Telecommunication Co., Ltd., Taiwan, and was involved in the research and development of multimedia communication systems. He now works as the Vice President's special assistant in the Northern Taiwan Business Group, Chunghwa Telecom. Co., Ltd. Since 1998, he has been a part-time Associate Professor at National Taipei University of Technology, National Taiwan Normal University, and Taipei Medical University, respectively. His current research interests include QoS control over Internet, buffer management schemes and traffic scheduling for high-speed networks.

Dr. Chiou was a recipient of the Chinese Institute of Engineers (CIE) technical paper award in 1997. He is a member of Phi Tau Phi.



**Zehong Tsai** (M'88) received the B.S. degree in electrical engineering from National Taiwan University (NTU), Taipei, Taiwan, R.O.C., in 1983, and the M.S. and Ph.D. degrees from the University of California, Los Angeles, in 1985 and 1988, respectively.

During 1988–1990, he was Member of Technical Staff at AT&T Bell Laboratories, where he investigated performance aspects of network management systems. Since 1990, he has been with the Department of Electrical Engineering, NTU, where he is currently a Professor. He is also with the Graduate Institute of Communication Engineering and the Graduate Institute of Industrial Engineering at NTU. He was responsible for the technical support of the ATM testbed deployment and many ATM interoperability trials at NTU. His research interests include high speed networking, network management, network planning, spread spectrum, and broadband Internet.

Dr. Tsai was a recipient of the Chinese Institute Engineers (CIE) Technical Paper Award in 1997. He is a member of the ACM.