Design of a Generalized Priority Queue Manager for ATM Switches

H. Jonathan Chao, *Senior Member, IEEE*,
Hsiling Cheng, Yau-Ren Jenq, and Daein Jeong

The authors are with the Department of Electric Engineering, Polytechnic University, 6 Metrotech Center, Brooklyn, NY, U.S.A.
(email:{chao, hscheng, yjeng, djeong}@antioch.poly.edu)

Manuscript received May 1, 1996.

Manuscript revised December 11, 1996.

This research was supported by NSF grant NCR-9216287 and Center of Advanced Technology for Telecommunications in New York State.

This paper was presented in part at GLOBECOM '94, San Francisco, CA, U.S.A., November 28 - December 2, 1994 and at IEEE ATM '96 WORKSHOP, San Francisco, CA, U.S.A., August 26-28, 1996.

Abstract

Meeting QoS requirements for various services in ATM networks has been very challenging to network designers. Various control techniques at either the call or cell level have been proposed. In this paper, we deal with cell transmission scheduling and discarding at the output buffers of an ATM switch. We propose a *Generalized Priority Queue Manager* (GPQM) that uses per virtual connection queueing to support multiple QoS requirements and achieve fairness in both cell transmission and discarding. It achieves the ultimate goal of guaranteeing the QoS requirement for each connection. The GPQM adopts the *Earliest-Due-Date* (EDD) and *Self-Clocked Fair Queueing* (SCFQ) schemes for scheduling cell transmission and a new *Self-Calibrating Pushout* (SCP) scheme for discarding cells. The GPQM's performance in cell loss rate and delay is presented. An implementation architecture for the GPQM is also proposed, which is facilitated by a new VLSI chip called the *Priority Content Addressable Memory* (PCAM) chip. The chip is being designed using CMOS 0.8-μm technology.

keywords: Asynchronous transfer mode, Quality of service, Queue manager, Scheduling, Discarding, Fairness, Content addressable memory

1 Introduction

Due to the large bandwidth-delay products involved, the heterogeneous traffic types that need to be supported, and the various quality of service (QoS) requirements imposed on ATM networks, traffic control and management is difficult and challenging for network designers. For example, real-time traffic requires small end-to-end delay and jitter, while data transfer requires low cell loss rates to reduce the number of retransmissions so as to prevent the network from becoming more congested. Thus, it is necessary to control users' traffic so that network resources can be shared effectively and the QoS can be met for all users.

A queue manager of an output-buffered ATM switch, as shown in Fig. 1, receives multiple cells and transmits one cell in each time slot. Cells waiting for transmission are stored in a buffer in the queue manager, which will schedule ATM cells' departing and discarding sequences based on both of their required QoS and the system status. During call setup, each connection can be assigned a service class with a delay and a loss priority, as shown in Fig. 2. Time-constrained services are usually assigned higher delay priorities in order to have a lower cell delay, and loss-sensitive services are usually assigned higher loss priorities in order to have a smaller cell loss rate. This classification enables network nodes to support various QoS requirements.

Several algorithms dealing with cell scheduling based on different priority service disciplines have been proposed [1, 2, 3, 4, 5]. For instance, the Earliest Due Date (EDD) scheduling scheme achieves each service class's delay requirement by minimizing the *tail probability*, the probability that the cell's queueing delay exceeds its delay requirement [3]. Meanwhile, one of the main objectives of scheduling is to achieve fairness in the amount of services provided to competing connections, in which the share of service is proportional to some specified weighting factor. In addition, cells from different connections tend to be interleaved with each other as they are multiplexed onto a transmission link. Consequently, user traffic's burstiness is reduced, which in turn reduces the network congestion probability and increases the network throughput. Several cell scheduling algorithms that deal with fairness have been proposed [6, 7, 8, 9], specifically dealing with the amount of service share provided to each connection. Processor sharing algorithms provide a lower limit of assigned bandwidth and are appropriate for traffic requiring deterministic QoS [10]. However, the assignment of service share (i.e., the *weighting factor* in the processor sharing scheme) for each connection or grouped connections depends on the required QoS as well as traffic characteristics. Due to the difficulty in determining the weighting factor of a connection associated with its required QoS requirement, we will adopt the EDD scheme to select a class that is to be served and the fair queueing algorithm to achieve fairness based on the allocated bandwidth that could be determined by the call admission control procedure.

Several cell discarding policies have also been proposed. They are complete buffer sharing with *pure pushout* [11], partial buffer sharing with nested thresholds [12] for different loss priorities, and the so-called expelling policy [13]. In the pure pushout policy, an arriving high-priority cell can overwrite a low-priority cell if the buffer is full. In the nested threshold policy, a threshold is assigned to each loss priority. Cells of a particular class are accepted to enter the buffer only if the current queue length is smaller than the associated threshold. In the expelling policy, when the number of high priority cells exceeds a predetermined threshold, all low-priority cells in front of the queue will be expelled until a high-priority cell is found and transmitted. This policy has been proven to outperform the other two policies in the case of two loss priority levels. The pure pushout scheme does not provide guaranteed QoS to low priority traffic, while the other two schemes have the difficulty of determining proper thresholds for different priorities under time-variant parameters, such as traffic burstiness and offered load. Since each service class has its own required cell loss rate, a devised discarding policy must guarantee each class's cell loss rate requirement. Recently, we have proposed a new cell discarding scheme, *Self-Calibrating Pushout* (SCP) [14], which maintains a constant cell loss rate ratio between two loss priority classes while the buffer is completely shared by all service classes.

In this paper, we propose a Generalized Priority Queue Manager (GPQM) that is capable of handling the delay and loss priority jointly while satisfying each service class's QoS requirement and achieving fairness for connections in the same service class. The GPQM adopts the EDD scheme to meet various delay requirements and the *Self-Clocked Fair Queueing* (SCFQ) [9] algorithm to achieve fairness. In other words, it handles multiple delay priorities in class level and fair scheduling in connection level. The GPQM also adopts the SCP scheme for selective cell discarding. Moreover, by taking advantage of the fair queueing, the GPQM achieves almost identical cell loss rates for all connections in the same loss priority without increasing implementationrm complexity. It thus meets each individual connection's cell loss rate.

Section 2 gives a detailed description of the adopted cell scheduling schemes: the EDD and the SCFQ. Section 3 describes the SCP scheme and the fairness in cell discarding. Section 4 shows the performance of the GPQM through computer simulations. Section 5 presents an implementation architecture for the GPQM, which is facilitated by a new VLSI chip called a *Priority Content Addressable Memory* (PCAM) chip. Section 6 describes the PCAM chip. Section 7 gives final conclusions.

2 Cell Service Scheduling

Several cell scheduling algorithms have been proposed to achieve fairness, specifically dealing with the amount of service share provided to each connection. Among them, the Packet-by-

Packet Generalized Processor Sharing (PGPS) algorithm [8] is implemented using the concept of *virtual time*, *v(t)*, which was introduced in [7]. The computational complexity associated with the evaluation of *v(t)*, however, is not feasible in a broadband ATM network. Golestani [9] proposed the *Self-Clocked Fair Queueing* (SCFQ) scheme, which is an implementable version of fair queueing. He defined the *virtual time* as the indication of the progress of work in a real queueing system instead of a hypothetical fluid-flow reference system model. Golestani showed that this approach approximates the optimal fairness found in fluid flow models. The SCFQ scheme, as applied to ATM, is described below.

1. The k^{th} cell, p_i^k, of connection i arriving at time a_i^k is stamped with a virtual finishing time F_i^k, which is determined as below, and then placed in the queue.

$$F_i^k = \frac{1}{\rho_i} + max\{F_i^{k-1},\ \hat{F}(a_i^k)\}\ , \tag{1}$$

$$with\ \ F_i^0 = 0, \tag{2}$$

   in which ρ_i is the offered load of connection i. The cells in the queue are chosen for service in increasing order of the values of the associated virtual finishing times.

2. $\hat{F}(t)$, regarded as the system's virtual time at time t, is equal to the virtual finishing time of the cell receiving service at time t.

Although we can achieve fairness among competing connections by scheduling their cells based on their allocated bandwidth using the SCFQ scheme, we cannot meet their various delay requirements. Thus, we will also adopt the EDD scheme to handle various delay requirements between classes.

The EDD scheme works as follows. A cell from connection i, arriving at time a_i^k with local tolerable delay d_i, is assigned a time stamp value $\{a_i^k + d_i\}$, called *due time* (T). When a cell is to be transmitted, the server chooses the one that has the minimum *due time* value. Here, we show how the SCFQ and the EDD schemes work together. Each arriving cell is time stamped with two values, virtual finishing time (F) and due time (T). Within each class, cells are sorted based on their F value and are served according to that order, thus achieving fairness among the connections in the same class. The HOL cell in each service class is the one that has the smallest F value in that class. When the server is to choose a cell from the HOL cells to transmit, it will choose the one that has the smallest T value, thus implementing the EDD scheme.

There is one more interesting feature of sorting cells in each service class based on their virtual finishing time (F). It can distribute the cell loss evenly among the connections in the same class, thus achieving loss fairness (described in detail in section 3.2).

3 Buffer Management

3.1 Self-Calibrating Pushout (SCP)

The main objective of the SCP scheme is to balance the cell loss rates by measuring, in real time, the numbers of discarded low and high loss priority (LP) cells such that each priority will meet its cell loss rate requirement (e.g., 10^{-6} for low LP and 10^{-9} for high LP). Since the SCP automatically balances cell loss rates among different classes, it may simplify call admission control rules when different cell loss rates are to be guaranteed.

The primary advantage of the SCP scheme over the other proposed cell discarding schemes is its capability of automatically calibrating in real time each service class's cell loss rate so as to meet each class's QoS requirement, i.e., it is "self-calibrating". Many proposed cell discarding schemes rely on finding different discarding thresholds off-line based on traffic characteristics and QoS requirements. However, when cell discarding is considered jointly with cell scheduling, finding the thresholds may become very difficult (if not impossible). On the other hand, the SCP scheme can easily be combined with any cell scheduling scheme because cell discarding is determined on-line by the measured data from a system, not by statistical prediction like other schemes. We briefly describe the SCP cell discarding algorithms in the following.

Let us define the target cell loss rates of low and high LP classes to be P_L and P_H, respectively, and the measured cell loss rates of low and high LP classes to be $\hat{P}_L$ and $\hat{P}_H$, respectively. Our goal here is to selectively discard low or high LP cells such that the ratio $\hat{P}_L/\hat{P}_H$ approaches P_L/P_H. For example, if P_L is 10^{-6} and P_H is 10^{-9}, we will try to keep the ratio $\hat{P}_L/\hat{P}_H$ around 1,000 regardless of the traffic characteristics.

A control parameter used in this algorithm, called **loss_weight**, is analogous to the bandwidth-weight in weighted round robin service, where the bandwidth received by each virtual connection (or number of cells transmitted) is proportional to the bandwidth-weights. Similarly, in the SCP scheme, the number of discarded cells is proportional to the **loss_weight** of different loss priorities. Now let us consider two loss priorities, low and high. Let ρ_L and ρ_H be the offered load of low and high LP traffic, respectively. The total offered load ρ is equal to $\rho_L + \rho_H$. The value of **loss_weight** is defined as the following:

$$\mathbf{loss\_weight} = \frac{\mathbf{P_L}}{\mathbf{P_H}} \times \frac{\boldsymbol{\rho}_\mathbf{L}}{\boldsymbol{\rho}_\mathbf{H}} \tag{3}$$

For instance, if $P_L = 10^{-6}$, $P_H = 10^{-9}$, and $\rho_L = \rho_H$, then the **loss_weight** is equal to 1,000. In other words, when 1,000 low LP cells have been discarded, a high LP cell will be the next candidate to be discarded as the buffer is filled. A variable called **CNT** in this algorithm is used to keep track of the discardings. Initially, **CNT** is set to zero. It is incremented by 1 when a low LP cell is discarded, and decremented by **loss_weight** when a high LP cell is discarded.

Another control parameter is the threshold (**TH**). **TH** is compared with the number of high priority cells currently in the buffer. If the number of high priority cells currently in the buffer is less than **TH**, cells of both classes will be admitted into the buffer. If it is larger than **TH** and the **CNT** value is less than the **loss_weight**, we will discard low LP cells. From our performance study, it is shown that the target ratio $\hat{P}_L/\hat{P}_H$ can be kept constant for a wide range of **TH** values. That means that choosing a **TH** value to ensure the SCP scheme works properly is quite relaxed, which eliminates the necessity of adjusting **TH** for different traffic characteristics. This is because the **TH** control parameter only provides a coarse calibrating guideline to bring $\hat{P}_L$ and $\hat{P}_H$ to the range of interest, while the **CNT** of discarded cells is the control parameter that fine-tunes the cell loss of low and high LPs to ultimately keep $\hat{P}_L/\hat{P}_H$ at a desired ratio. The algorithm is shown in Fig. 3.

3.2 Loss Fairness

As described above, the SCP scheme balances the ratio of the cell loss rates of low and high loss priorities. However, our ultimate goal is to ensure that each individual connection's cell loss rate is met, and not just for each service class. Now that the SCP scheme determines a priority level from which a cell is to be discarded, the remaining problem is to select a connection in that priority to discard its cell.

Since there may be up to a few thousand virtual connections in each transmission link, it will be very difficult to control their cell loss rates on an individual basis. Here, we propose a simple method to achieve cell level fair loss rate within same loss priority class. By taking advantage of the fair-queueing scheduling scheme, which provides fair service among connections in the same class, we can also achieve fair loss with *bounded* variation. As a result, we can guarantee each individual connection's cell loss rate requirement under the SCP discarding policy with a properly designed call admission control policy.

In the following, we explain how fair loss with bounded variation can be automatically achieved by adopting the fair-queueing scheduling policy. Suppose ℓ ($\leq N$) cells arrive at the buffer when the buffer is filled, resulting in ℓ cells to be discarded from the buffer. For fairness concern, we expect the portion of loss from connection i (ℓ_i) to be

$$\ell_i = \frac{\rho_i}{\sum_{j \in B} \rho_j} \ell \;, \quad {}^{\forall} i \in B \tag{4}$$

in which B is the set of connections that has at least one backlogged cell at the moment of buffer overflow, and ρ_i is the average arrival rate of connection i. Note that Eqs. (4) and (1) have the same feature in that the amount of discarded (or served) information is proportional to ρ_i, which suggests us to select the cell in the position of next transmission as the one to be pushed out. We can imagine the execution of discarding cells as a fictitious momentary transmission (with an infinite link rate). Then, the total served data from each connection, the sum of the actually served and the fictitiously transmitted (i.e., discarded), conforms to the proportional property. As a result, the queue manager will discard ℓ cells close to the front of the sorted queue when ℓ cells arrive and see the buffer filled. In Appendix A, we show that this discarding scheme can achieve fair loss among connections of the same loss priority with bounded variation. Note that discarding cells from the head of the queue also yields smaller average queueing delay [15].

4 Performance of the Generalized Priority Queue Manager (GPQM)

4.1 Simulation Set-up and Traffic Model

Four different service classes, I, II, III, and IV, are considered here and arranged in a priority matrix as shown in Fig. 4(a). Each service class is associated with a logical queue. Cells in the same logical queue are FCFS. The SCFQ is not considered here because we just need to investigate the performance of each service class, while each virtual connection's performance will be met as the SCFQ is applied. The total occupancy of these four logical queues is always less than or equal to the buffer size. The time division multiplexer (TDM) distributes cells from an ATM switch to a particular logical queue according to their service classes. Cells are stored or discarded in the corresponding logical queue according to the rules described in the SCP algorithm. Namely, high LP cells are always stored in the buffer when the buffer is not filled. However, low LP cells may be pushed out from the buffer (even if the buffer is not filled) when the number of high LP backlogged cells exceeds **TH** and the **CNT** is less than the **loss_weight**. The server serves the HOL cell from one of the four logical queues that has the smallest due time (T). The tie of the due time can be broken by random choice.

In this performance study, we assume the source traffic is bursty and alternates between the active and idle period, as shown in Fig. 4(b). More specifically, cells arrive in consecutive slots in the active period, and no cells arrive in the idle period. Traffic sources are assumed to be independent. Both the active and idle periods are assumed to be geometrically distributed with mean burst length L_{active} and mean idle time L_{idle}. Thus, the offered load ρ is given by

$$\rho = \frac{L_{active}}{L_{active} + L_{idle}} \tag{5}$$

4.2 Cell Loss Performance

In our simulation study, we assume that the buffer size $B = 512$, the total offered load $\rho = \rho_H + \rho_L = 0.8$, where $\rho_H = \rho_I + \rho_{III}$ and $\rho_L = \rho_{II} + \rho_{IV}$, and the average burst length

$L_{active} = 15$ *cells*. We also assume that $P_H = 10^{-6}$ and $P_L = 10^{-3}$, instead of $P_H = 10^{-9}$ and $P_L = 10^{-6}$; this is because the latter will take a very long time for simulations. The traffic mix ratio, ρ_H/ρ_L, varies from 0.1 to 10.

Fig. 5 shows the loss probabilities, $\hat{P}_L$ and $\hat{P}_H$, versus the traffic mix ratio ranging from 1 to 10 with different **TH** values. With **TH** set to 64, 128, 192, the ratio can be maintained around 1,000 (P_L/P_H), shown by the two curves that are equally spaced in each case. This means when a cell of low or high LP is to be discarded, most of the time its associated logical queue is not empty. As a result, the measured loss probability ratio, $\hat{P}_H/\hat{P}_L$, can be kept to the target value. $\hat{P}_H$ is in the order of 10^{-6} and $\hat{P}_L$ is in the order of 10^{-3}. Note that in Fig. 5 both $\hat{P}_H$ and $\hat{P}_L$ increase when ρ_H/ρ_L is larger than 3. This is because, when ρ_H/ρ_L becomes larger, high LP cells will dominate. It would be quite likely that cells in the buffer are all high LP cells when the buffer is filled. Thus, when a high LP cell arrives and finds the buffer filled, it will be discarded because there are no low LP cells in the buffer, resulting in $\hat{P}_H$ increasing. This will make the **CNT** more negative because the **CNT** is decremented by **loss_weight** when a high LP cell is discarded. On the other hand, since most of the time the number of high LP backlogged cells exceeds **TH**, low LP cells are blocked more often. Thus, the $\hat{P}_L$ is also increased. When ρ_H/ρ_L is larger than 4, $\hat{P}_H$ and $\hat{P}_L$ become more stable. Thus, we can conclude that because of **TH**, the ratio of loss probabilities, $\hat{P}_H/\hat{P}_L$, can be maintained constant.

Fig. 6 shows the ratio $\hat{P}_L$ and $\hat{P}_H$ versus the traffic mix ratio varying from 0.1 to 1 with **TH** set to 64 and 128. As expected, the ratio still remains around 1,000. The reason that the case of **TH** equal to 192 is not shown in Fig. 6 is that the $\hat{P}_H$ value is too small to be obtained from computer simulations in reasonable time. Also note that the smaller the traffic mix ratio ρ_H/ρ_L is, the less likely the condition "$\mathbf{Q_H} \geq \mathbf{TH}$" will occur. For ρ_H/ρ_L from 0.1 to 0.3, $\hat{P}_H$ and $\hat{P}_L$ fluctuate. The reason is the same as before only now the dominating cells in the buffer are of low LP class. Note that both $\hat{P}_H$ and $\hat{P}_L$ in this case are smaller than those in the case where the traffic mix ratio varies from 1 to 10. This is because low LP cells prevail in the buffer and are always available when discarding is needed. Therefore, high LP cells will not be mistakenly discarded due to the lack of low LP cells in the buffer. This then decreases the number of discarded cells of high LP and makes $\hat{P}_H$ smaller. Meanwhile, since low LP cells dominate in the buffer, the number of high LP backlogged cells will usually be under **TH** and low LP cells will not be blocked from entering the buffer. As a result, $\hat{P}_L$ is smaller than that in the case of the traffic mix ratio ranging from 1 to 10.

From Fig. 5 and Fig. 6, we notice that, for a traffic mix ratio ρ_H/ρ_L from 0.1 to 10, the target ratio $\hat{P}_L/\hat{P}_H$ can be kept constant for a large range of **TH** values. That means that choosing a **TH** value to ensure the SCP scheme works properly is quite relaxed, eliminating the necessity of adjusting the **TH** for different traffic distributions. This is because the **TH** is the control parameter that only provides a coarse calibrating guideline to bring $\hat{P}_L$ and $\hat{P}_H$ to the range of interest, while the **CNT** is the control parameter that fine-tunes the cell loss of low and high LPs to eventually keep $\hat{P}_L/\hat{P}_H$ to a desired ratio.

Fig. 7 shows the cell loss probabilities for each service class versus the traffic mix ratio ranging from 1 to 10 and from 0.1 to 1. The **TH** is set to 64. In both cases, we can see that Class II and Class IV (low LP classes) almost have the same loss rates, while Class I and Class III (high LP classes) are very close to each other. The discrepancy between Class I and Class III is due to the difference of their delay priorities. Since Class I cells have higher delay priority, they have a higher chance of being served before Class III cells. As a result, Class III cells will stay in the buffer longer than Class I cells and have a higher probability of being pushed out upon a turn to discard a high LP cell. This is why Class III has a larger cell loss rate than Class I. One way to eliminate the discrepancy between the service classes that have the same loss priority is to provide a separate counter for each class. By keeping track of each individual's counter value, we will be able to maintain the same cell loss rate for the service classes with the same LP. However, this will increase implementation complexity. Since the discrepancy is small, we would rather combine the cell loss from Classes I and III, and that from Class II and IV, to simplify the implementation.

4.3 Cell Delay Performance

Let us assume the 99 percentile delay requirement for high DP cells is 150 μsec and for low DP cells it is 5 $msec$ [16]. For an OC-3 (155.52 Mb/sec) transmission line, each cell time is about 2.83 μsec. Then, the delay requirements for high and low DP are 53 and 1767 cell slots, respectively. Note that cells from the same class are served in a first-come-first-served manner because their due times are their arrival times plus the same tolerable delay.

We use the *delay probability mass function (pmf)* for each class to show its delay distributions and see the percentage of cells violating their delay requirements. The pmf usually gives us more useful information in practice than the average delay. Fig. 8 shows the delay distributions of Classes I, II, III, IV, where the traffic mix ratio ρ_H/ρ_L is equal to 1. For Classes I and II with a higher DP, 2 percent of cells violate the delay requirement (150 μsec, or 53 cell slots). For Classes III and IV with a lower DP, less than one percent of the cells violate the delay requirement (5 $msec$ or 1767 cell slots). Note that the delay distributions for the traffic mix ratio ρ_H/ρ_L varying from 0.1 to 10 are similar to Fig. 8.

5 GPQM Implementation Architecture

One of the main functions of a queue manager is to schedule cells' transmission based on their bandwidth share or delay requirement by using various algorithms, such as the SCFQ and the EDD described in section 2. One way to schedule cells' transmission is to time-stamp arriving cells, store them in the buffer, and transmit them according to their time stamp values. This time stamp can be calculated based on any scheduling algorithm. For instance, if the SCFQ algorithm is adopted, the time stamp will be the virtual finishing time (F). We have designed an ASIC, called the *Sequencer* chip [17], to sort cells' time stamp values in such a way that cells with a smaller time stamp value are scheduled to depart earlier. However, each Sequencer chip can only handle up to 256 virtual connections (VCs) or 256 cells. For a practical application where there are thousands of VCs, the number of required Sequencer chips would be too large to be cost-effective. Here, we present a new ASIC, called the *Priority Content Addressable Memory* (PCAM) chip, which can accommodate any number of VCs and any size buffer. The chip also resolves the overflow problem of the time stamp.

The main difference between these two chips is that the Sequencer performs the sorting function and the PCAM chip performs the searching function. The Sequencer sorts the time stamp values such that the smallest one appears at the rightmost side of the chip. Since the sorting is done globally inside the Sequencer, it can be done in one clock cycle. The PCAM chip arranges the time stamp values in fixed locations and starts searching the first one that has a valid bit. As long as the searching speed is not a bottleneck, this approach is not limited by the VC number nor by the buffer size.

5.1 Searching by the PCAM Chip

The PCAM chip consists of many entries that are addressed by the F values (assuming the SCFQ is adopted). Each entry contains a *zone bit* (Z) and a *validity bit* (V), denoted as a (Z, V) pair, as shown in Fig. 9. The Z bit is used to resolve the overflow problem of the F (details will be explained later). The V bit indicates whether there is any cell assigned to the F.

The (Z, V) pairs in the PCAM chip are addressed by different F values and are arranged in such a way that pairs with smaller F values are on top of those with the larger F values. For those cells with identical F values, a logical queue is formed to link all of them and is called the timing queue. Since cells are arranged by the F values, the PCAM chip facilitates the required search function by identifying the first (Z, V) pair that has the V bit set to "1". Cells that are associated with the identified (Z, V) pair will be transmitted to the network. For instance, in Fig. 9, when the (Z, V) at location 2 is found, cells "a", "b" and "c" of the timing queue at $F=2$ will be transmitted in sequence. Once they are all transmitted, its V bit is reset to zero, indicating that no more backlogged cells are assigned with $F=2$. Thus, during the next round of searching, cell "d" at location t will be chosen and transmitted.

On some occasions, the calculated F may overflow, i.e., exceeding the maximum number that the hardware can handle, say 2^{14}-1 for 14 bits of the F values. This is because as time passes, the F increases monotonically and eventually exceeds its maximum number. To overcome the overflow problem, we have previously proposed using two sorting devices to store non-overflow and overflow time stamps separately [18]. Here we store them in the same device to save hardware by using the Z bit to indicate whether the calculated time stamp is overflow or not. The definition of overflow here is different from the traditional one. We use a CZ (*current zone*) bit to indicate the zone of the cells that are currently being served. Whenever the 15th bit of calculated F (i.e., the overflow bit) has the same value as the CZ, the F is defined as non-overflow. However, if its 15th bit has a different value from the CZ bit, the F is defined as overflow. Thus, when searching the V bits in the PCAM chip, the CZ facilitates the PCAM chip to choose the first V bit from an appropriate zone. When all the V bits in the current zone are zero and there is at least one non-zero V bit in the other zone, the CZ will be toggled after sending a cell from the other zone, indicating that the service zone is flipped. For example, when all cells "a", "b", "c" and "d" are transmitted and the V bit at the other zone (Z=1) is found at (t+3), the CZ bit is toggled from zero to one. From then on, cells in zone "1" will be scheduled before those in zone "0". In conclusion, the searching zone in the PCAM chip is alternating between CZ=0 and 1. As long as the 15th bit of the calculated F does not change more than once when serving in the current zone, no cell out-of-sequence problem will occur.

5.2 Head-of-Line Time Stamp

In order to ultimately solve the time-stamp overflow problem, we will only assign the HOL cell from each virtual connection (VC) the virtual finishing time (F) and *due time* (T). Upon the departure of the current HOL cell, the next cell from the same VC that becomes the HOL cell will be time stamped. If cells are time stamped as they arrive at the GPQM, those cells that are from extreme low bit-rate sources and arrive in a burst can easily cause the virtual finishing time value overflow. This is because the term $1/\rho_i$ in Eq. (1) is very large due to the low bit rate. Although the zone bit (Z) in the PCAM chip is designed to handle the time stamp overflow problem, when there are more than one time overflow from each VC at any moment, the single zone bit fails to resolve the overflow problem. Thus, by assigning the virtual finishing time only to the HOL cell, it is guaranteed that there is only one time overflow from each VC at any given moment. As a result, Eqs. (1) and (2) can be simplified as

$$F_i^k = \frac{1}{\rho_i} + \hat{F}(h_i^k)\ , \tag{6}$$

$$with\ \ F_i^0 = 0, \tag{7}$$

where h_i^k is the moment at which the k^{th} cell of VC_i becomes the HOL cell. If we assume the minimum rate of any VC is not less than (*line rate*/2^m), where m is the number of bits used to represent the F value, there should be no more than one time stamp overflow from each VC at any moment. This assumption ensures that we can maintain a correct order of the time stamp values by using a single zone bit in the PCAM chip. It has been proven by us that the system that assigns the F value only to the HOL cell of each VC operates identically to the one that assign the F to all arrived cells (i.e., the SCFQ scheme).

5.3 Architecture of the GPQM

The implementation architecture for the Generalized Priority Queue Manager (GPQM) shown in Fig. 10 adopts the SCFQ and EDD schemes for cell scheduling, and the SCP scheme for cell discarding. It consists of a Cell Time-Division Multiplexer (CTDM), a Cell Memory, a Queue Manager Controller, four PCAM chips (one for each service class), two Microprocessors (i.e., one for T calculation and the other for F calculation), an Idle-Address FIFO, and a group of virtual-channel queues (VC queues) and timing queues. In addition, Selectors (SEL's $0 \sim 5$) are used to select the data path properly. The CTDM multiplexes multiple cells from a switch fabric and store them in the Cell Memory. The Queue Manager Controller generates the necessary signals to control all functional blocks. In this architecture, four service classes, I, II, III, and IV are considered.

Newly arriving cells that belong to the same virtual connection (VC) are linked together in a logical queue, called the VC queue. The content of each VC queue is cell's address in the Cell Memory and the cell's due time (T). Only the HOL cell in each VC queue joins the timing queue, where HOL cells that have the same service class and virtual finishing time value (F) are linked together. The content of the timing queue is cell's VCI and the due time (T). Each VC queue and timing queue is confined by two pointers, the Head Pointer (HP) and the Tail Pointer (TP). The former points to the first cell of the logical queue while the latter points to the last cell of the logical queue. Timing queues in each service class are handled by the associated PCAM chip in a way that timing queues with smaller F values are served sooner. Detailed operations of the timing queues have been described in section 5.1.

A counter, increased by one in every cell clock tick, is used to record cell's arrival time and to calculate cell's *due time* (T). Both the due time (T) and virtual finishing time (F) increase monotonically as time goes on and are reset to zero when the buffer is empty. Here, we use a large number (e.g., 2^{32}) to accommodate the T values. For a cell slot of $2.83\mu s$ at $OC-3c$ line, a 32-bit T value can sustain the busy period of the GPQM up to $2.83\mu s \times 2^{32}$, or more than 3 hours. A 48-bit T value can sustain the busy period for more than twenty years. Since the T values are stored in the memory (VC queues and timing queues), we can afford to use a large number of bits to represent them.

In the following sections, we will explain how cells are written to, read out or discarded from the Cell Memory.

5.3.1 Write-in Procedure

When a cell arrives at the GPQM, it is first stored in the Cell Memory at a location provided by the Idle-Address FIFO, which stores the address of current vacant cell locations. Its VCI (Virtual Channel Identifier) in the cell header is extracted by the T-Microprocessor to calculate the arriving cell's T value. The cell's address and T value are then written to the corresponding VC queue. If the arriving cell is the HOL cell, its VCI is also extracted by the F-Microprocessor to calculate its F value. The F value is then used as an address to store the (Z, V) pair in the associated PCAM chip according to its service class and to link the arriving cell's VCI value and T value to the corresponding timing queue.

5.3.2 Read-out Procedure

According to the scheduling policy described in Section 2, when a cell is to be transmitted, the smallest F value in each class will be identified from each PCAM chip. These four F values are used to access the HOL cell's T value from the selected timing queue in each class. They are then compared and the VC that has the smallest T is chosen to transmit. Its VCI is then used to access the cell's address from the VC queue and read out the cell from the Cell Memory. In the mean time, its address is stored in the Idle-Address FIFO for next arrivals. As soon as a cell is transmitted, the F value is returned to the F-Microprocessor to calculate the F value of the next HOL cell in the same VC queue. The new F value is used as the address to store the (Z, V) pair in the associated PCAM and to store the T and VCI to the timing queue. The Queue Manager Controller will check whether the timing queue, from which a cell just departed, is empty or not. If the timing queue becomes empty, its associated V bit in the PCAM chip will be reset to zero. The Queue Manager Controller also keeps track of the CZ (current zone) bit. Whenever there are no more cells that can be found in the current zone and there is at least one cell in the other zone, the Queue Manager Controller will toggle the CZ bit after sending a cell from the other zone, meaning from now on the search will start from the other zone.

5.3.3 Pushout Procedure

The procedure of discarding cells is similar to that of transmitting cells. The Queue Manager Controller uses the SCP scheme to identify the service class from which a cell is to be discarded. The HOL cell of the class, i.e., the one with the smallest F, will be identified from the associated PCAM chip and discarded, thus achieving loss fairness described in

Section 3.2. Once the cell is discarded, its address becomes available to the next arrivals and is stored in the Idle-Address FIFO.

6 Priority Content Addressable Memory (PCAM) Chip

6.1 Block Diagram

Fig. 11 shows the PCAM chip's block diagram. It consists of a CAM Array, a few Selectors, one I/O Interface, two Inhibit circuits, two 7_To_128 Decoders, two 128_To_7 Encoders, and the Chip Controller.

The CAM Array has 32K bits arranged into 16K modules, each module with 2 bits. The input data (2 bits) are written to the Array through the input bus IN[0:1] with an address provided on the input address bus, X[0:13]. When a pattern in the Array is to be searched, the searching pattern is fed through the input bus IN[0:1]. If a pattern that is closest to the top of the list is found ("HIT" signal is asserted), its corresponding address will appear at the output address bus, Y[0:13]. Signals such as MASK[0:1] and INH[0:1] are used when part of the data pattern in the Array is to be searched or updated. The data pattern can also be directly accessed using the address bus X[0:13]. The "INIT" signal initializes the chip by resetting all data patterns in the CAM Array to zero. The "CS" and "HIT" signals are used for connecting multiple PCAM chips when necessary. The "READ" and "WRITE" signals are to read or write the data pattern using the X[0:13] address. The "SEARCH" signal is asserted during a search operation. These input/output (I/O) signals are summarized in Table 1.

When the CAM Array is accessed for either a write or read operation, the first 7 bits of the address are the column address, and the rest are the row address. The row decoder (7_To_128 Decoder) decodes the row address to enable one of the 128 row select lines, and the column decoder (7_To_128 Decoder) decodes the column address to enable one of 128 column select lines. At the intersection of the row select and column select lines, a module is identified. The column select line enables the bit-line pair, which routes the addressed data through the I/O data bus.

When the PCAM chip is operated at the search mode, each module in the CAM Array will compare its internally stored data with the broadcast pattern through the IN[0:1] bus. If matched, the associated row match line and column match line will be asserted. If multiple matches occur, the Row Inhibit circuit will select the one that is closer to the top of the list (RM0 $\sim$ RM127) and forward the match enable lines (ME0 $\sim$ ME127), among them only one being enabled, to the CAM Array so that only those modules on the enabled line will be able to participate the column search. Again, all 128 column match lines (CM0 $\sim$ CM127) are sent to the Column Inhibit circuit, which will then choose only one that is closer to the left of the list (CM0 $\sim$ CM127). Finally, two 128_To_7 Encoders simultaneously encode the results from both Row and Column Inhibit Circuit to the address output Y[7:13] and Y[0:6]. The PCAM chip uses parallel hardware in the Inhibit Circuit to identify the top most or the left most bit, resulting in only a few gates delay in the Inhibit Circuit.

Note that the PCAM chip is different from the traditional CAM chip. First, the PCAM chip is deep and narrow (16K$\times$2), while the traditional CAM chip is shallow and wide (e.g., 1K$\times$64). Second, the PCAM chip gives out the address of the pattern that is found, while the traditional CAM gives out the data field associated with the pattern. Third, the PCAM chip can write the pattern to a specific location associated with the time stamp, while the traditional CAM chip cannot. Fourth, the PCAM reduces wiring complexity by searching and encoding the first matched pattern through vertical and horizontal dimensions, thus only processing $2\sqrt{M}$ match lines instead of M match lines (M is the size of the PCAM chip).

6.2 Chip Operations

There are four different operations for the PCAM Chip: *initialization*, *write-in*, *search*, and *read-out*. As shown in Fig. 12, the *initialization* is started by asserting the "INIT" and setting input bus (Z, V) to (0, 0). Following the *initialization*, three pairs of (Z, V) are written to the locations W, U, and R. Let us assume $|W| > |R| > |U|$ ($|\bullet|$ means the value of $\bullet$) and R, W belong to the overflow zone, meaning their Z value is different from the CZ value. Following the three *write-in* operations is a *read-out* operation. The (Z, V) at location R is accessed by asserting the "READ" signal and setting the input address value to R. After the *read-out* operation, there are two *search* operations. First, the search patterns (IN[0:1]) are set to (1, 1). The chip will search the first valid cell in zone one (i.e., R). Among the two matches, since the address value R is less than W, it is chosen and its value R appears at the output address bus Y[0:13]. The second search patterns (IN[0:1]) are set to (0, 1). Since the (Z, V) at the location U is the only match, its value U appears at the output address bus Y[0:13].

6.3 Connecting Multiple PCAM Chips

Multiple PCAM chips can be connected to accommodate a larger F value. The connection is achieved by using the "CS" and "HIT" signals. Fig. 13 shows a system with four PCAM chips connected in parallel, which can accommodate a maximum F value up to 64K (2^{16} or 4 $\times$ 2^{14}) with each chip handling 16K (2^{14}) F entries. The most significant bits, X[14:15], are decoded to determine the page to which the timing queue belongs. X[0:13] is connected to all chips. During the *read-out/write-in* operations, only the PCAM chip that has "CS" signal asserted through the Page Decoder will be enabled. However, during the *search* operation, four PCAM chips are enabled to search for a pattern simultaneously. The output "HIT" signal from each chip indicates if there is any pattern matched in the chip. These four "HIT" signals are encoded by a Priority Encoder. The output of the Priority Encoder, Y[14:15], selects a proper F value, Y[0:13], through the 4_To_1 Selector. They are then combined to form a final result Y[0:15].

7 Conclusion

QoS guarantee is very important and challenging for ATM network designers. In this paper, we proposed a *Generalized Priority Queue Manager* (GPQM) of an output-buffered ATM switch, which can handle cell scheduling and selective discarding so as to meet each connection's QoS requirement. The GPQM adopts the *Earliest-Due-Date* (EDD) and the *Self-Clocked Fair Queueing* (SCFQ) schemes for delay control and fair queueing. It also adopts the *Self-Calibrating Pushout* (SCP) to determine a service class to discard cells. The SCP has the capability of maintaining the cell loss rate ratio between loss priority classes while fully utilizing the memory and eliminating the difficulty of setting thresholds that are normally required by other discarding schemes. Both delay control (by EDD) and cell loss rate control (by SCP) are applied on a class basis. By taking advantage of the fair queueing, each connection's service amount and number of discarded cells are proportional to its allocated bandwidth. This effectively achieves the ultimate goal of meeting each individual connection's QoS requirement. The performance of the GPQM was studied through computer simulations and shown to be satisfactory.

We presented an implementable architecture for the GPQM, facilitated by a new VLSI chip (PCAM). The proposed architecture can implement any scheduling algorithm that schedules cell transmission based on time stamp values. The PCAM chip has several advantages over the traditional CAM chip. The PCAM chip is being designed using CMOS 0.8-μm technology.

APPENDIX A

Lemma 1 *Assume that cells are sorted according to their virtual finishing time (F) in an increasing order. Let us consider the first ℓ cells from the head-of-line. Then, among these ℓ cells, the number of backlogged cells from connection i is proportional to ρ_i with bounded variation, which is independent of ℓ.*

Proof. Let us define F_1 and F_ℓ as the F values of the head-of-line cell and the ℓ_{th} cell of the sorted queue, respectively. Let us denote the set of connections that have at least one backlogged cell at the moment of discarding as B. Assume that there are n connections in B. By referring to Eq. (1), the backlogged cells from connection i form a sequence of F values, incremented by a fixed amount of $1/\rho_i$. Let us denote a *non-negative integer* $\hat{N}_i$ as the maximum possible number of cells from connection i within the range of F values, $F_\ell - F_1$, starting from F_1. Then,

$$\hat{N}_i = \lfloor \rho_i R \rfloor + 1 \quad {}^{\forall} i \in B, \tag{8}$$

$$where \;\; R = F_\ell - F_1 \;\;, \tag{9}$$

in which $\lfloor x \rfloor$ represents the closest integer value not greater than x. With a fixed range of F values of R in Eq. (9), $\hat{N}_i$ is the maximum possible number of cells from connection i whose F values fall into that range. Now, let N_i be the *actual* integer number of cells from connection i among ℓ cells. Expression of N_i in terms of $\hat{N}_i$ can be obtained by considering an example in Fig. 14, where F_1 is 10 and F_ℓ is 20. Assume that $1/\rho_i$ of connection i is 10. Then, according to Eq. (8), $\hat{N}_i = 2$. Since F values of connection i can have many different sequences, N_i can vary from $\hat{N}_i$ in different ways. But the variation is bounded. For instance, if the F values sequence is $\{10, 20, \cdots\}$, N_i is 2. If the F values sequence is $\{13, 23, \cdots\}$, $N_i = 1$ as shown in Fig. 14 (a). In addition, cells from different connections may have identical F value and result in a smaller N_i. For example, Fig. 14 (b) shows 3 cells with F value of 10, and 2 cells with F value of 20. Assume that the buffer is filled after 2 cells with F value of 10 have departed. Since the cell of connection i with F value of 20 is outside the discarding region (the first ℓ cells), N_i is zero. From the above observations, N_i can be expressed by $\hat{N}_i$ with a variation of at most two cells, i.e.,

$$\max\{0, \hat{N}_i - 2\} \leq N_i \leq \hat{N}_i \quad {}^{\forall} i \in B. \tag{10}$$

If we define an *integer* value $\hat{\ell}$ as the summation of all $\hat{N}_i$, then from Eq. (8),

$$\hat{\ell} = \sum_{i \in B} \hat{N}_i = \sum_{i=1}^{n} \lfloor \rho_i R \rfloor + n \;. \tag{11}$$

Here, by the fact that

$$\lfloor \rho_i R \rfloor \leq \rho_i R \;\;,$$

and from Eq. (11), we obtain

$$R \geq \frac{\hat{\ell} - n}{\sum_{i=1}^{n} \rho_i} \ . \tag{12}$$

Then, from Eq. (12), Eq. (8) becomes

$$\hat{N}_i \geq \lfloor \frac{\rho_i}{\sum_{j=1}^{n} \rho_j} (\hat{\ell} - n) \rfloor + 1 \quad . \tag{13}$$

By assigning $\hat{N}_i$ the minimum value in Eq. (13), Eq. (10) becomes

$$\max\{0, \lfloor \frac{\rho_i}{\sum_{j=1}^{n} \rho_j} (\hat{\ell} - n) \rfloor - 1\} \leq N_i \leq \lfloor \frac{\rho_i}{\sum_{j=1}^{n} \rho_j} (\hat{\ell} - n) \rfloor + 1, \quad {}^{\forall} i \in B. \tag{14}$$

Eq. (14) implies that, for every connection i, N_i is proportional to ρ_i with respect to a specified amount $\hat{\ell} - n$ with variation bounded by 1. Notice that this variation is independent of ℓ value. Comparing N_i in Eq. (14) to ℓ_i in Eq. (4), ℓ in Eq. (4) is replaced with $(\hat{\ell} - n)$ in Eq. (14) due to the nature of a discrete system. □

References

[1] Y. Takagi, S. Hino, and T. Takahashi, "Priority assignment control of ATM Line Buffers with Multiple QoS Classes," *IEEE JSAC*, vol. 9, no. 7, Sep. 1991.

[2] R. Chipalkatti, J. F. Kurose, and D. Towsley, "Scheduling policies for real-time and non-real-time Traffic in a Statistical Multiplexer," *Proc. IEEE INFOCOM 1989*, pp. 774-783.

[3] Y. Lim and J. E. Kobza, "Analysis of a delay-dependent priority discipline in an integrated multiclass traffic fast packet," *IEEE Transactions on Communications*, vol. 38, no. 5, May 1990.

[4] J. R. Jackson, "Queues with dynamic priority Discipline", *Management Science*, vol. 8, no. 1, 1961.

[5] H. M. Goldberg, "Analysis of the Earliest Due Date scheduling rule in Queueing Systems", *Math. Oper. Res.*, vol. 2, no. 2 1977.

[6] A. Demers, S. Keshav, and S. Shenker, "Analysis and simulation of a fair queueing algorithm," in *Proc. ACM SIGCOMM'89,* Austin, TX, pp. 1-12, Sept. 1989.

[7] L. Zhang, "Virtual Clock: A new traffic control algorithm for packet switching networks," in *Proc. ACM SIGCOMM'90,* Philadelphia, PA, pp. 19-29, Sep. 1990.

[8] A. K. Parekh, "A generalized processor sharing approach to flow control in integrated services network," PhD thesis, MIT, Feb. 1992.

[9] S. J. Golestani, "A self-clocked fair queueing scheme for broadband applications" in *Proc. IEEE INFOCOM'94*, Toronto, Canada, June 1994.

[10] G. de Veciana and G. Kesidis, "Bandwidth allocation for multiple qualities of service using generalized processor sharing," *IEEE Transaction on Information Theory*, vol. 42, no. 1, pp. 268-272 , Jan. 1996.

[11] A. Y. Lin and J. A. Silvester, "Priority queueing strategies and buffer allocation protocols for traffic control at an ATM integrated broadband switching system," *IEEE J. Select. Areas Commun.*, vol. 9, no. 9, pp. 1524-1536, Dec. 1991.

[12] D. W. Petr and V. S. Frost, "Nested threshold cell discarding for ATM overload control: optimization under cell loss constraints," in *Proc. IEEE INFOCOM'91*, Bal Harbour, FL, April 1991.

[13] L. Tassiulas, Y. Hung, and S. S. Panwar, "Optimal buffer control during congestion in an ATM network node," in *Proc. IEEE INFOCOM'93*, San Francisco, CA, 1993.

[14] H. J. Chao and H. Cheng, "A new QoS-guaranteed cell discarding scheme: self-calibrating pushout," in *GLOBECOM'94* San Francisco, CA, Dec. 1994.

[15] N. Yin and M. G. Hluchyj, "Implication of dropping packets from the front of a queue," *IEEE Trans. Commun.*, vol. 41, no. 6, pp. 846-851, June 1993.

[16] "Broadband ISDN switching system generic requirements", Bellcore, TA-NWT-001110, Issue 1, Aug. 1992.

[17] H. J. Chao and N. Uzun, "A VLSI Sequencer chip for ATM traffic shaper and queue manager," *IEEE J. Solid-State Circuits*, vol. 27, no. 11, pp. 1634-1643, Nov. 1992.

[18] H. J. Chao, "A novel architecture for queue management in the ATM network," *IEEE J. Select. Areas Commun.*, vol. 9, no. 7, pp. 1110-1118, Sep. 1991.

List of Tables

List of Figures

H. Jonathan Chao (S'82-M'85-SM'95) received the B.S.E.E. and M.S.E.E. degrees from National Chiao Tung University, Taiwan, in 1977 and 1980, respectively, and the Ph.D. degree in electrical engineering from the Ohio State University, Columbus, in 1985.
He is Associate Professor of Electrical Engineering at Polytechnic University, New York, which he joined in January 1992. His areas of research include large-scale multicast ATM switches, photonic ATM switches, multimedia service over IP/ATM networks, and congestion/flow control in ATM networks. He holds 14 patents and has published over 60 journal and conference papers in the above areas. From 1985 to 1991, he was a Member of Technical Staff at Bellcore, NJ, where he has implemented several high-speed VLSI chips for SONET/ATM-based broadband networks. He has also proposed various architectures, along with their performance studies, for B-ISDN applications. These architectures include an optical customer premises network, a large-scale multicast ATM switch, traffic shaper, and priority queue manager for ATM network congestion control. He played a key role in designing a HIPPI-to-SONET/ATM network interface for the Nectar Gigabit Testbed, one of the five gigabit network testbeds in the US. From 1977 to 1981, he worked at Taiwan Telecommunication Laboratories, where he was engaged in the development of a digital switching system. He is serving as Editor for IEEE/ACM Transactions on Networking.

Hsiling Cheng received the B.S. degree in electrical engineering from the Feng-Chia University, Taichung, Taiwan, ROC, in 1984, and the M.S. degree from the Polytechnic University, Brooklyn, New York, in 1991. He is currently pursuing the Ph.D degree in electrical engineering at the Polytechnic University.

He joined the GTE Taiwan Branch as an engineer from 1986 to 1988, where he was involved in M13 multiplexers and fiber optics communication systems. His research interests include congestion control and connection admission control in ATM networks.

Yau-Ren Jenq received the B.S. degree in electrical engineering from the Feng-Chia University, Taichung, Taiwan, ROC, in 1986, and the M.S. degree from the Polytechnic University, Brooklyn, New York, in 1991. He is currently pursuing the Ph.D degree in electrical engineering at the Polytechnic University.

From 1988 to 1989, he was an engineer at Microelectronics Technology INC, Taiwan, where he was involved in the RF panel design including high power amplifier, up converter and down converter for INMARSAT-A satellite telecommunication systems.

He is currently pursuing the Ph.D. degree in the Department of Electrical Engineering at Polytechnic University, Brooklyn, N.Y.. He is a Research Assistant in the Center for Advanced Technology in Telecommunications (CATT). His main research interests include architecture designs and VLSI implementations for ATM networks.

Daein Jeong received B.A. and M.S. degrees from Seoul National University, Seoul, Korea, both in electrical engineering, in 1984 and 1986, respectively.

From 1987 to 1991, he was a Researcher at Korea Telecom, Korea, where he is involved in the development of TDX digital switching system. He is currently pursuing the Ph.D. degree at Polytechnic University, Brooklyn, N.Y.. He is a Research Assistant in the Center for Advanced Technology in Telecommunications (CATT). His current research interests are in the area of quality-of-service and congestion control in the ATM networks.

Table 1: I/O pin signals of the PCAM chip

Pin	I/O	Description
X[0:13]	IN	Address input (e.g., F value)
IN[0:1]	IN	Data input or searching pattern input
MASK[0:1]	IN	Mask bits for searching pattern
INH[0:1]	IN	Inhibit bits for writing data
INIT	IN	Initialization signal
WRITE	IN	Write enable
SEARCH	IN	Search enable
READ	IN	Read enable
CS	IN	Chip select
CK	IN	Clock signal
HIT	OUT	Indicate pattern found
OUT[0:1]	OUT	Data output for read operation
Y[0:13]	OUT	Address output (e.g., F value)

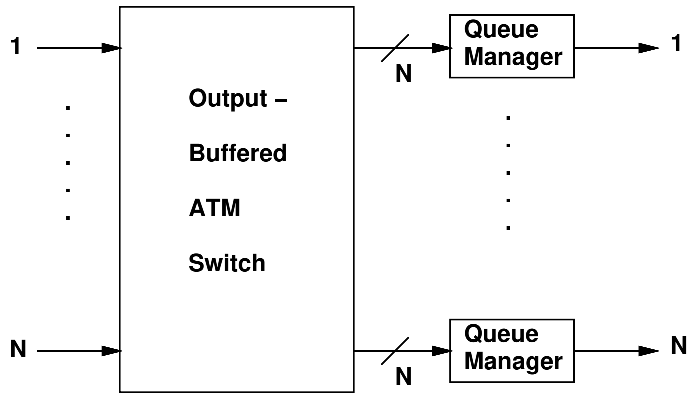


Figure 1: Queue manager of an output buffered ATM switch

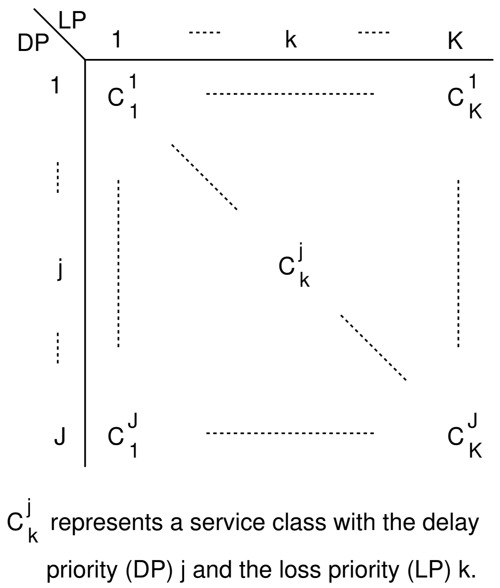
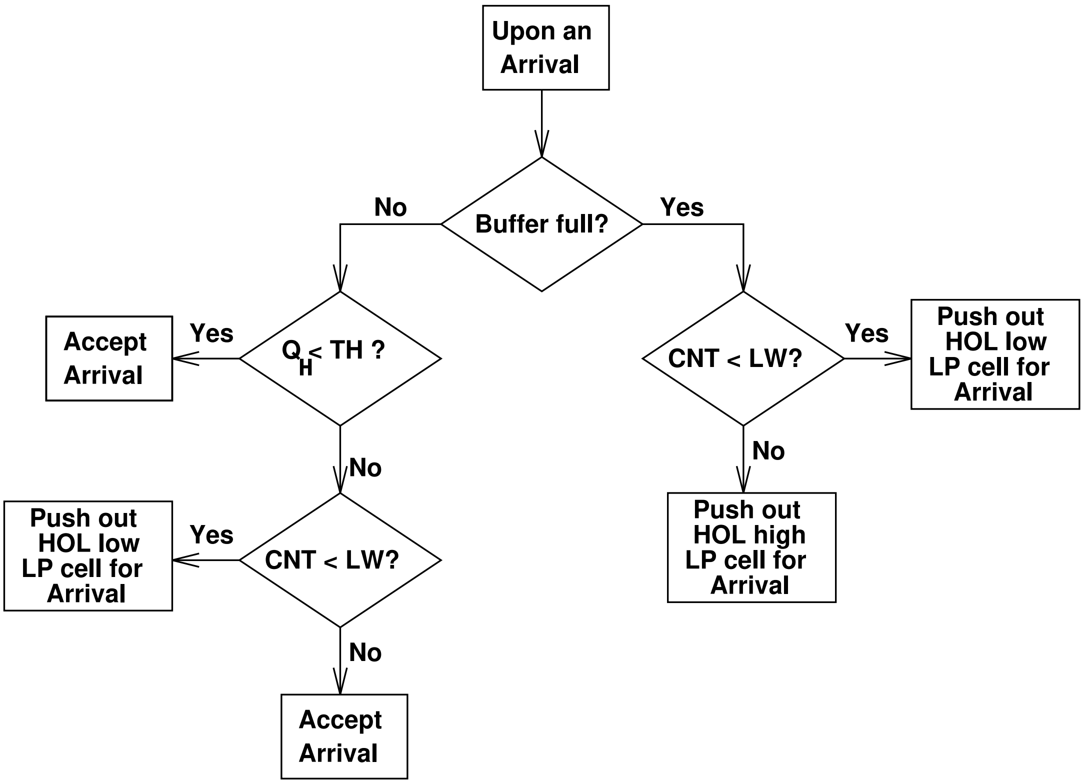


Figure 2: A traffic priority matrix



Q_H : number of high priority cells in the buffer

CNT : counter value

LW : loss_weight

TH : Threshold

HOL : Head of Line

Figure 3: Self-Calibrating Pushout (SCP) algorithm

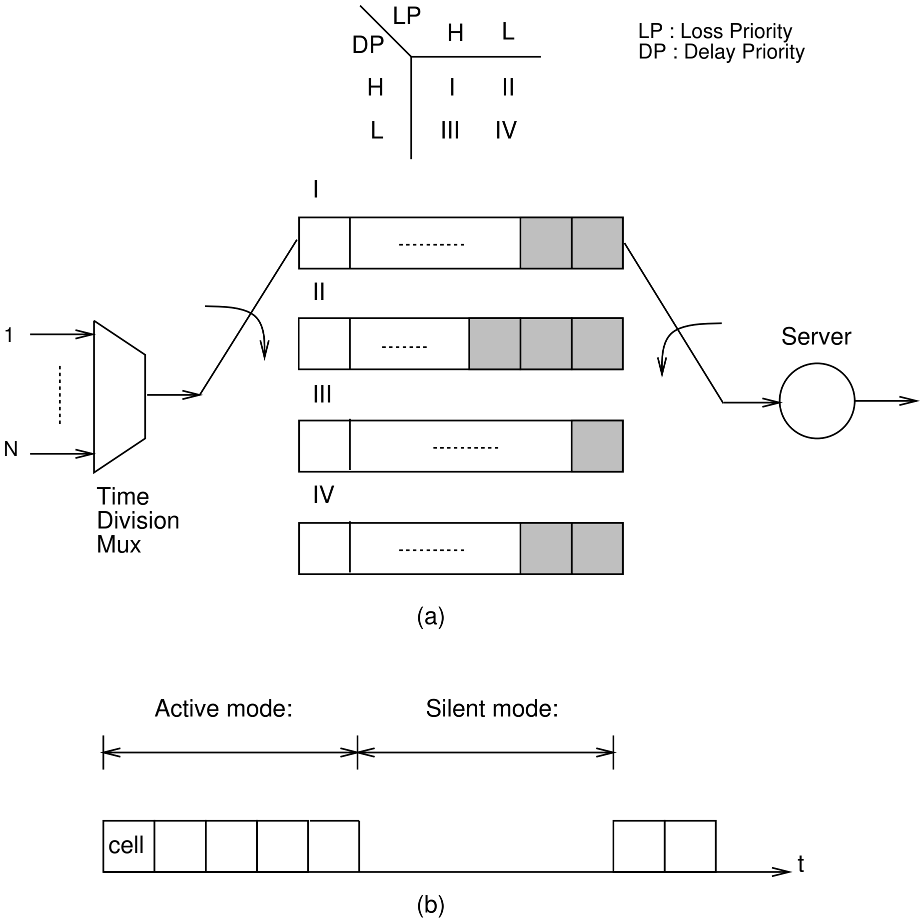
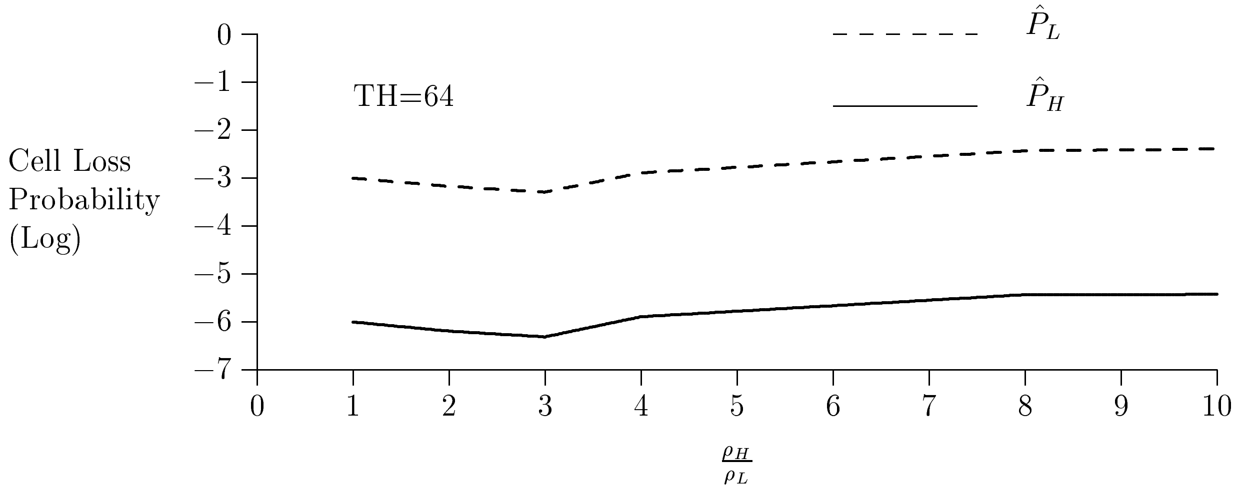
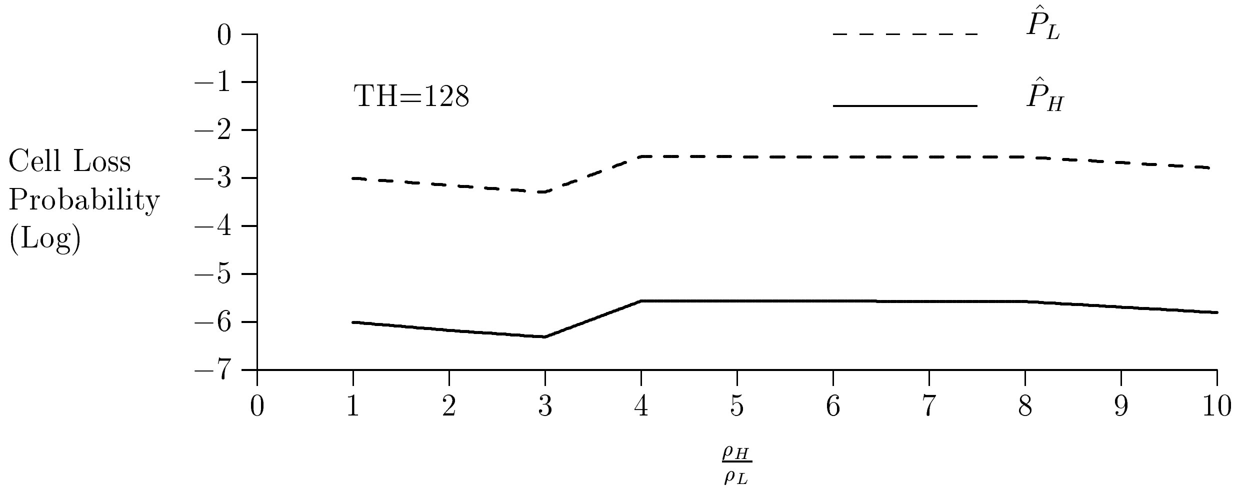


Figure 4: (a) Simulated GPQM model, (b) Cell arrival model





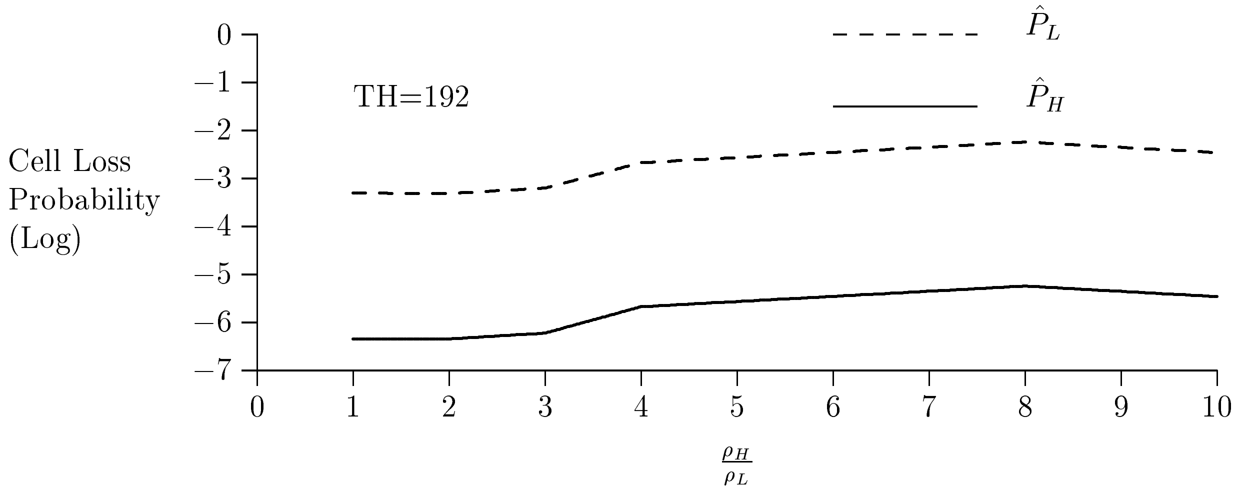


Figure 5: Cell loss probabilities versus trafic mix ratio, ρ_H/ρ_L, from 1 to 10, with $\rho_H + \rho_L = 0.8$, buffer size=512, and $L_{active} = 15$

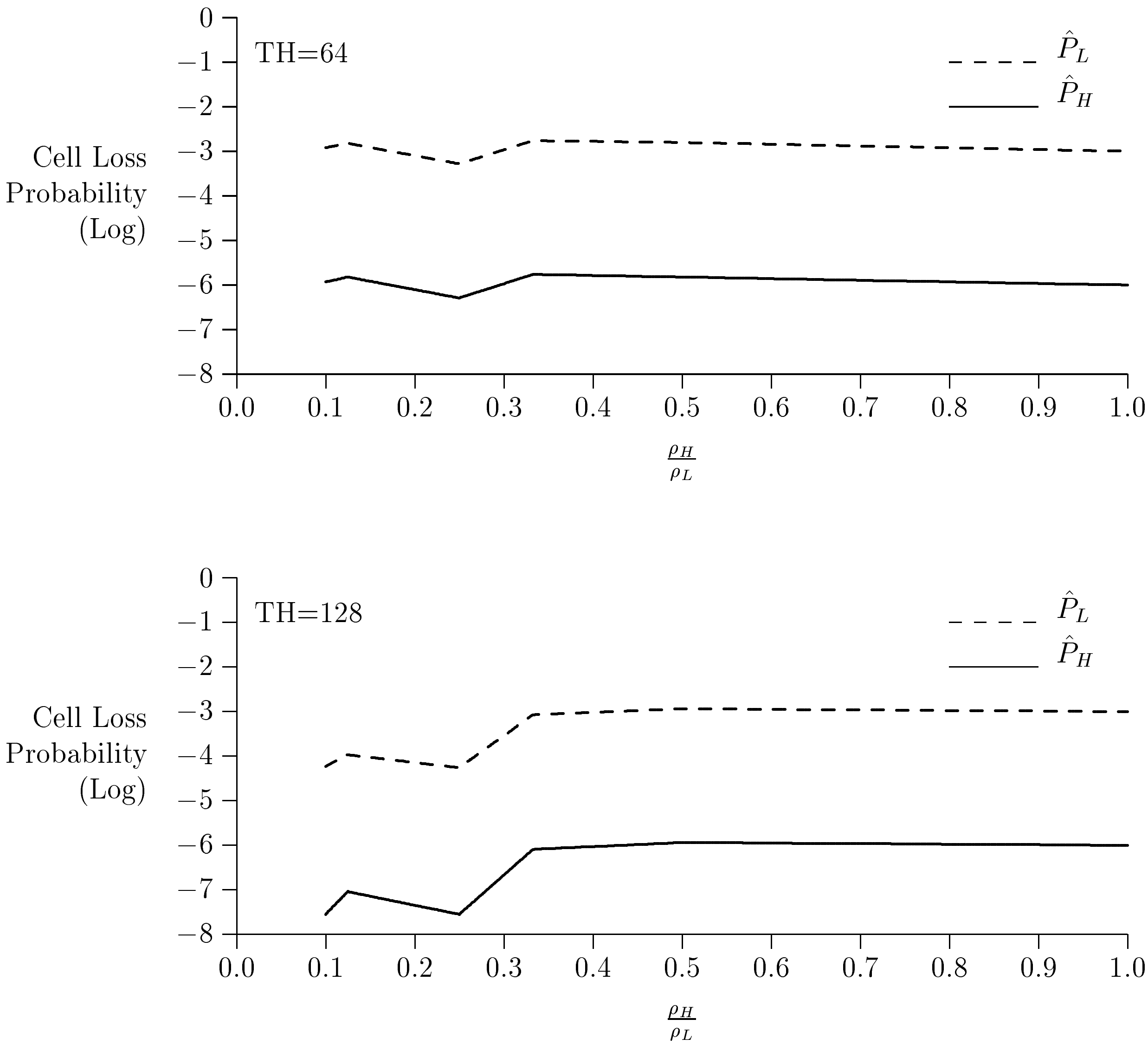


Figure 6: Cell loss probabilities versus trafic mix ratio ρ_H/ρ_L, from 0.1 to 1, with buffer size=512, $\rho_H + \rho_L = 0.8$, and $L_{active} = 15$

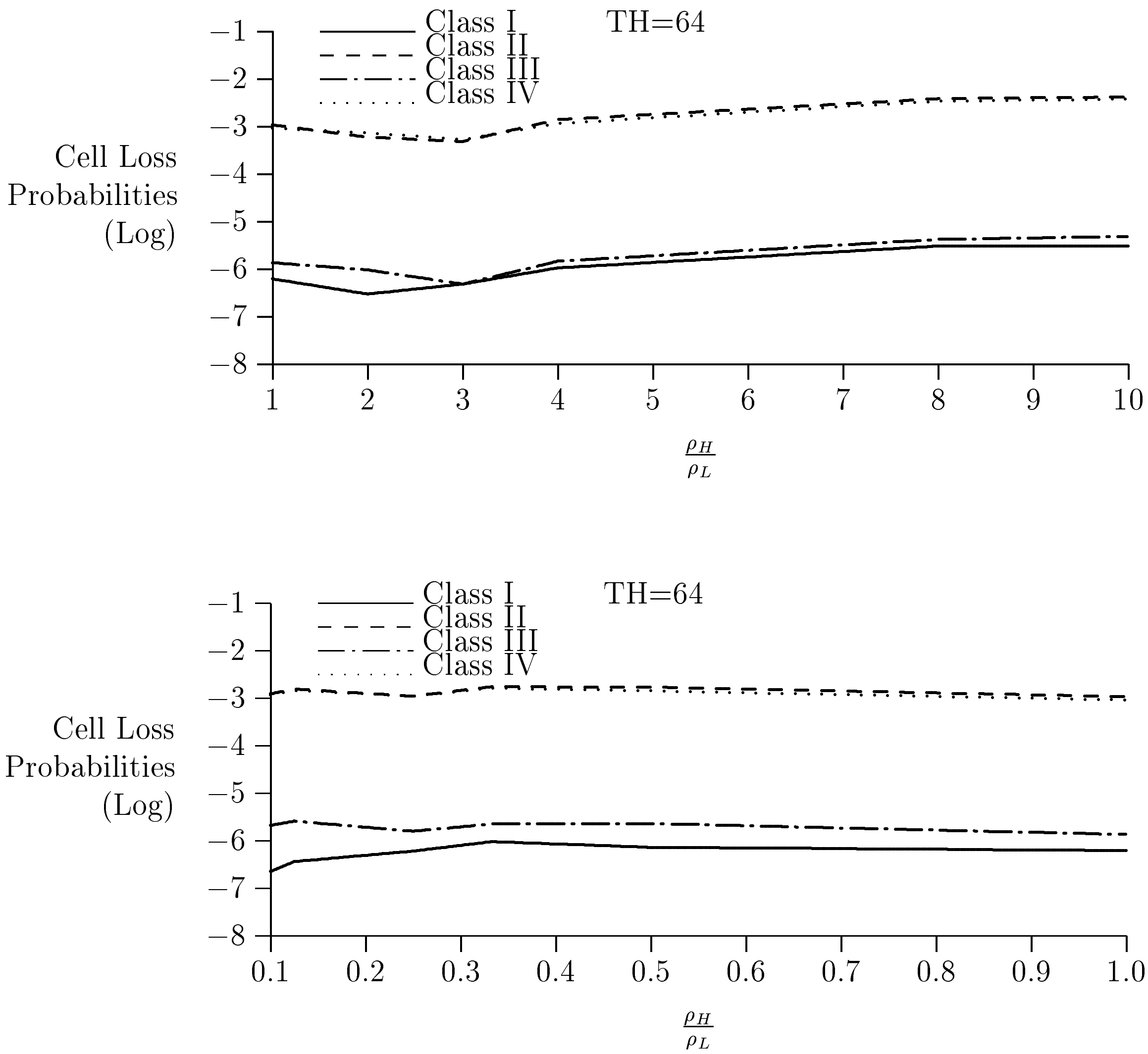


Figure 7: Cell loss probabilities for each class versus trafic mix ratio , ρ_H/ρ_L, from 0.1 to 10, with buffer size=512, $\rho_H + \rho_L = 0.8$, and $L_{active} = 15$

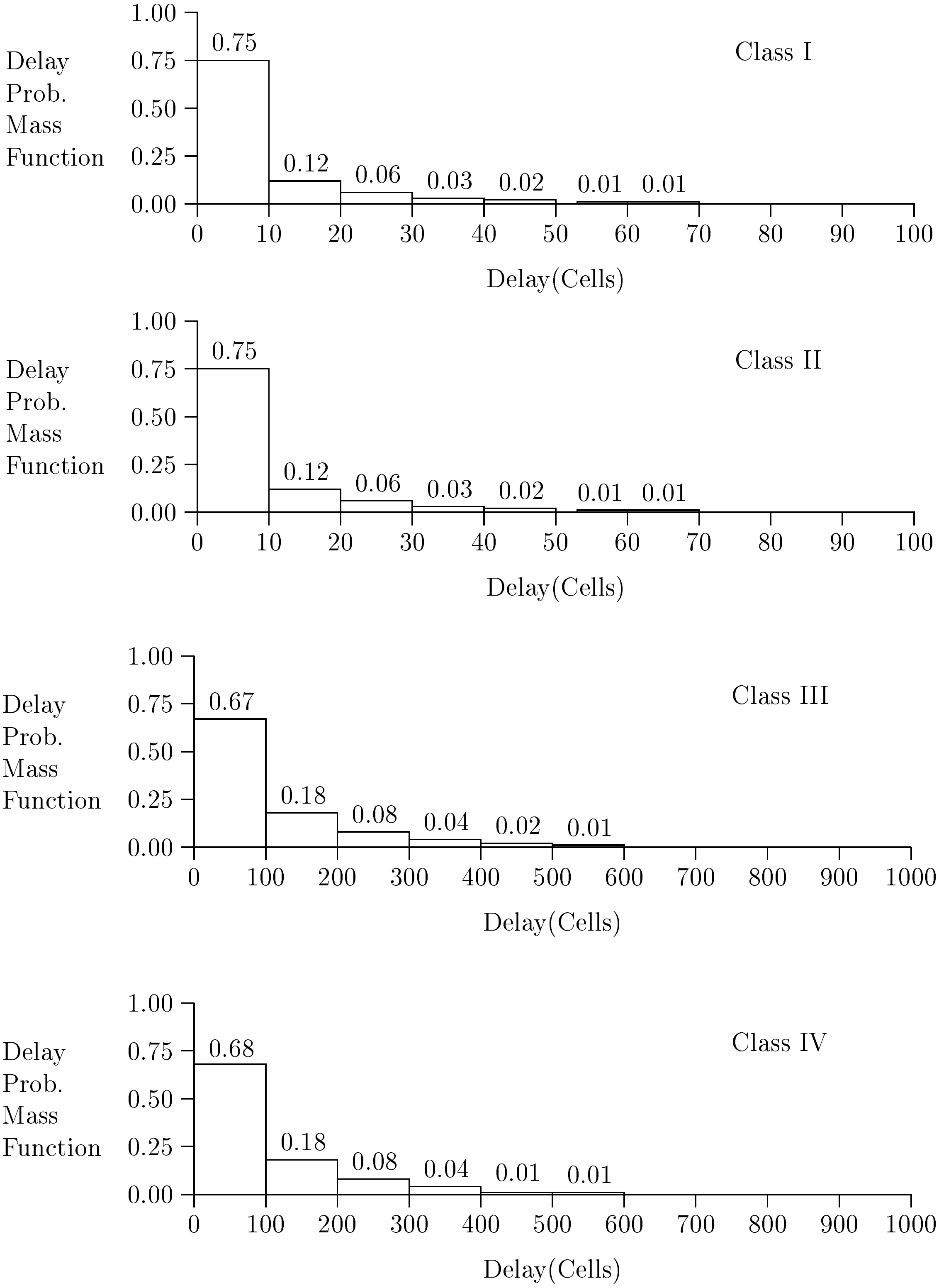


Figure 8: Delay distributions for Classes I, II, III, and IV, with buffer size=512, $\rho_H + \rho_L = 0.8$, $\rho_H/\rho_L = 1$, $L_{active} = 15$, and **TH**=192

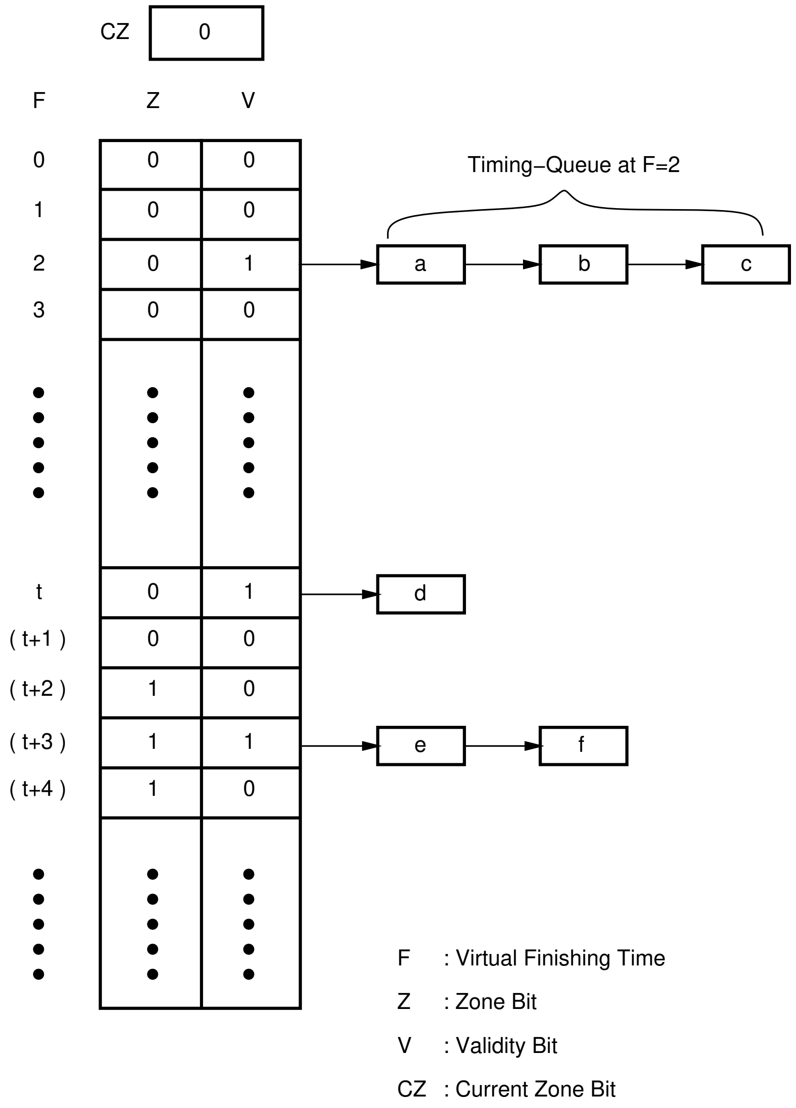
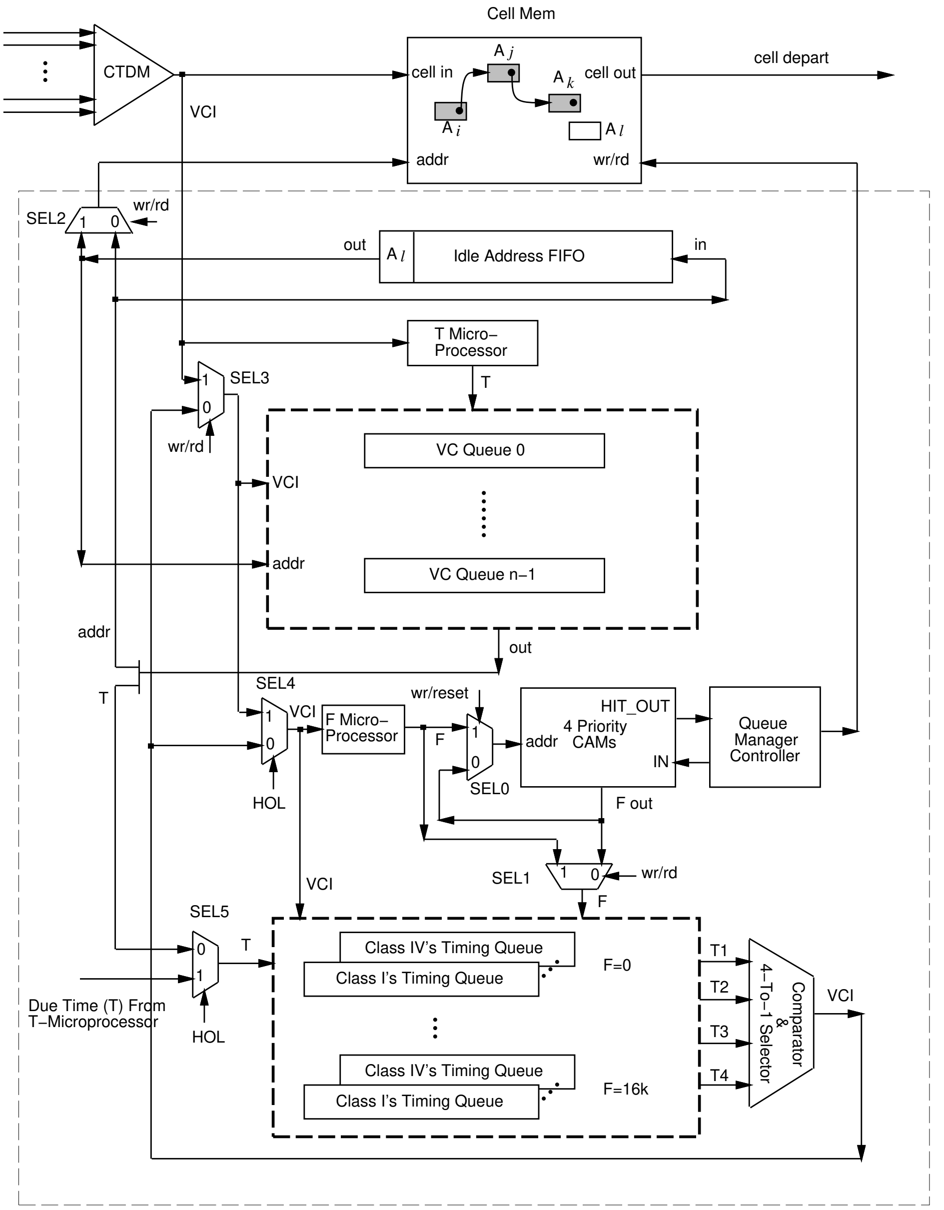


Figure 9: (Z,V) pairs in the PCAM chip



CTDM : Cell Time Division Multiplexer

F : Virtual Finishing time

T : Due time

Figure 10: Implementation architecture for the GPQM

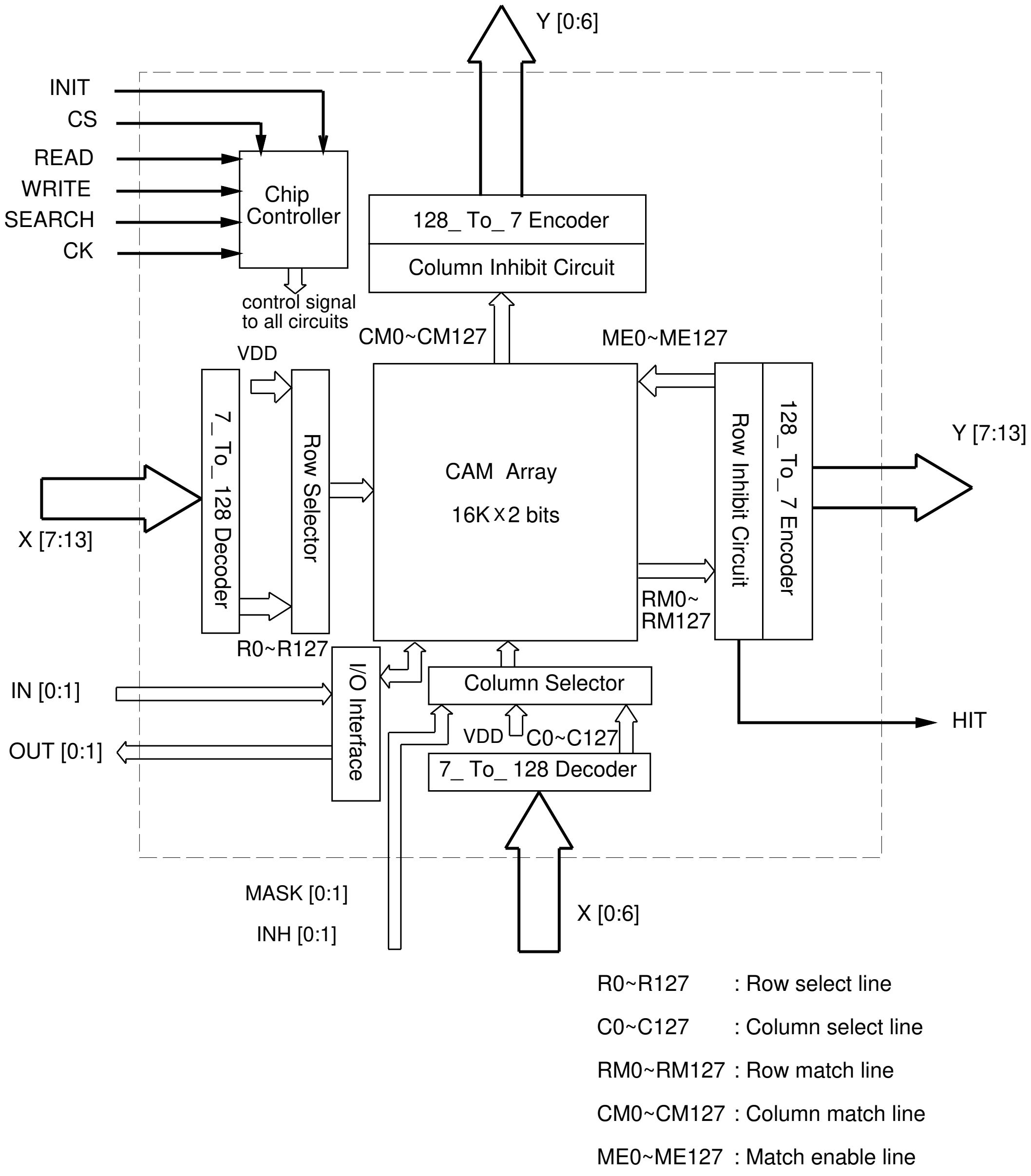


Figure 11: Block diagram of the PCAM chip

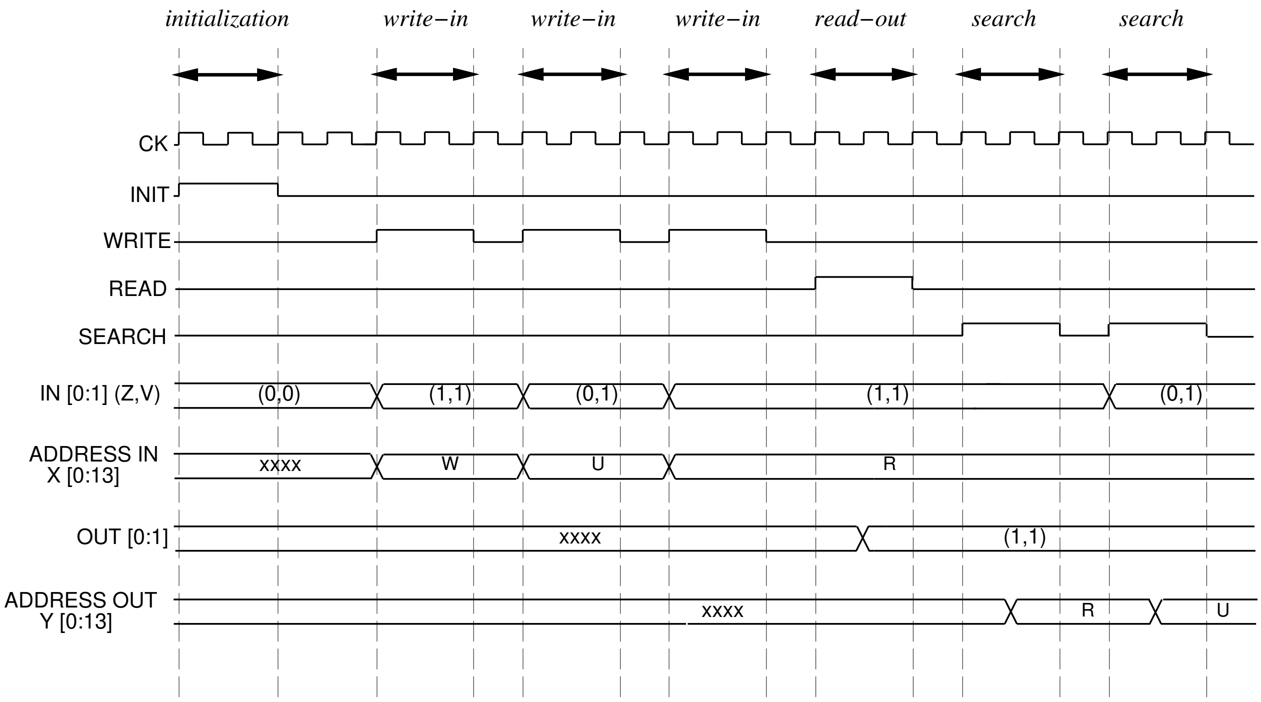


Figure 12: Timing diagram of PCAM chip operations

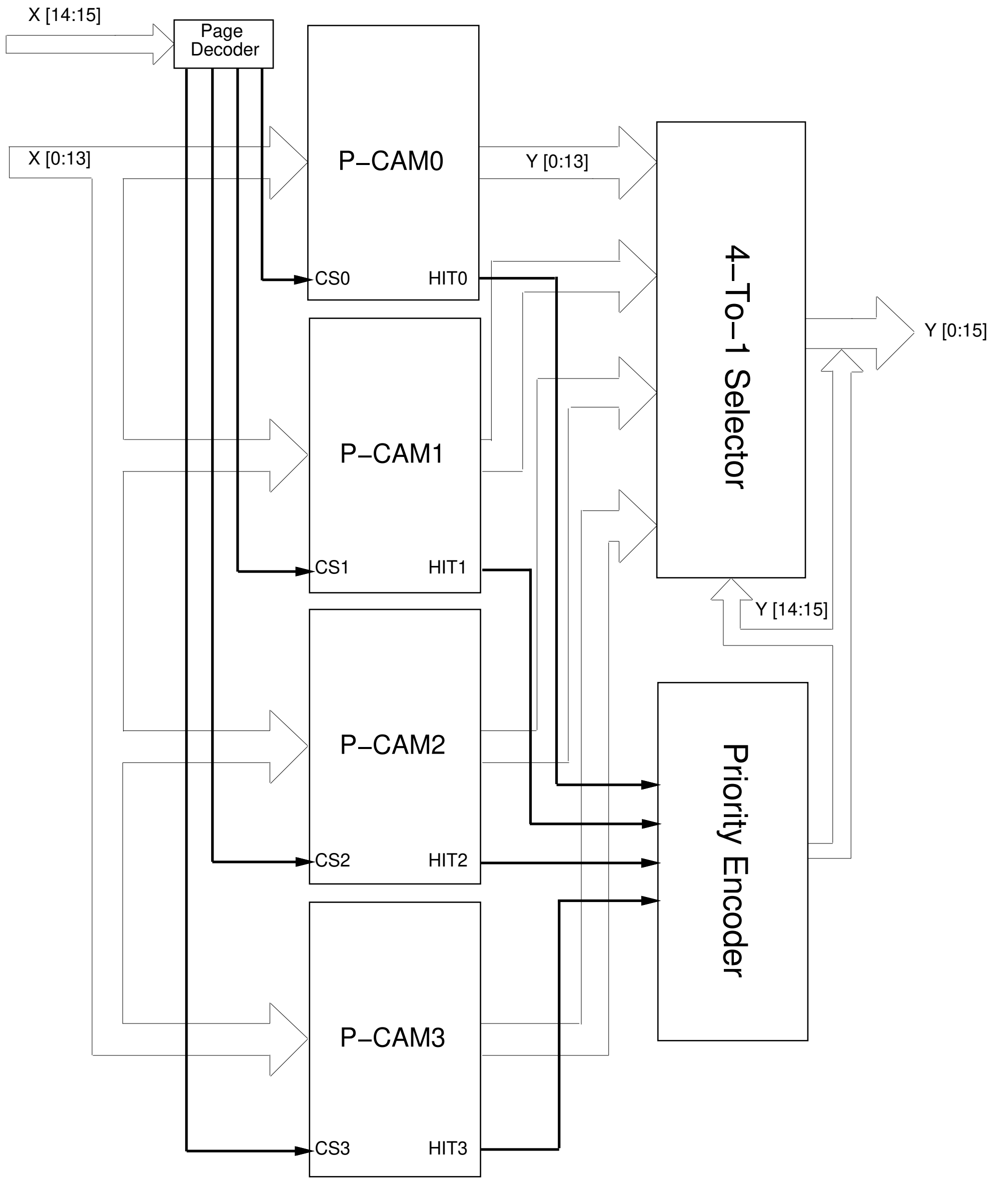


Figure 13: Parallel connection of four PCAM chips

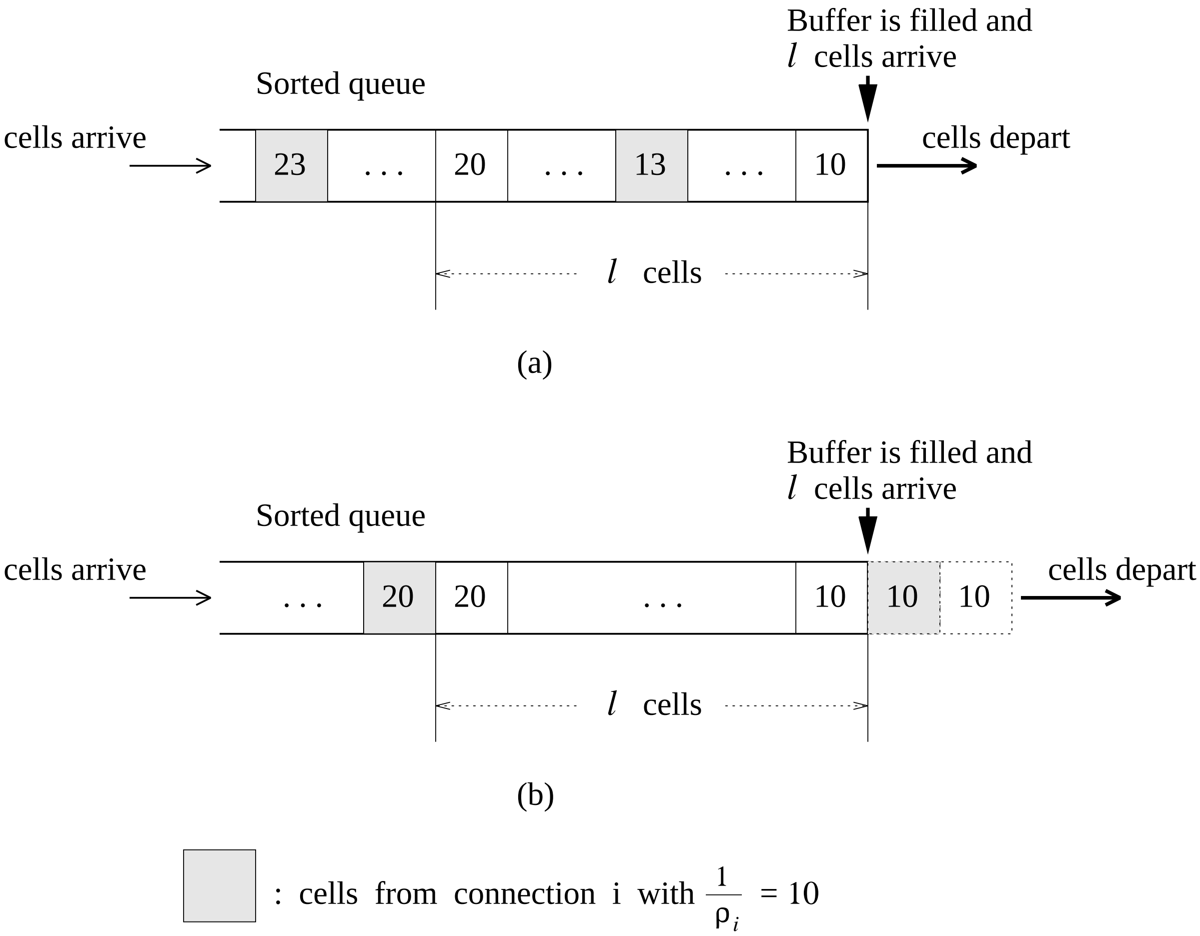


Figure 14: Examples of different N_i values with $\hat{N}_i = 2$: (a) Case of $N_i = 1$, (b) Case of $N_i = 0$.